The Pedagogy of Computer Programming Using Cognitive Development Through an E-learning Object

A thesis submitted to Middlesex University in partial fulfilment of the requirements for the degree of PhD

PANG Ka Man (M00596861)

School of Health and Education
Middlesex University London

Submission Date: May 2020

Abstract

Motivated by the needs of a pedagogy focusing on minimizing the learning difficulties in program semantics knowledge and logical reasoning, this research project develops a cognitive development-based pedagogy for introductory programming to support students in organizing and constructing knowledge to learn computer programming. A pedagogy is described as a practice and learning theory that defines the teaching and learning. Regarding the practice of this pedagogy, it uses a cognitive learning tool, called e-learning object, to support the scaffolding. With regard to the theory, this pedagogy is developed based on Vygotsky's Zone of Proximal Development and Piaget's theory for cognitive development.

In particular the scaffolding of this pedagogy includes three major learning processes. The first two learning processes focus on supporting students constructing knowledge on program semantics and conceptually map this knowledge to the coding process. The last learning process extends the learning to self-practice by demanding students to complete a set of exercises independently. All of these learning processes are supported by using the e-learning object, which is the major cognitive learning tool used in this pedagogy to support cognitive development. It is called e-learning object as it is designed by organizing a group of learning objects, in which each of them is to deliver the concepts of a specific unit topic of program control. Together with the course materials, these learning objects are accessed through the college's 'Blackboard System'.

In addition to the major objective of improving students' learning performance, this cognitive development-based pedagogy also extends from this objective to find out whether the positive learning outcome connects to cognitive development, and also whether this pedagogy can be embraced by teachers for use in their teaching processes. With these objectives, six research questions are defined in two stages of study. Research questions Q1 and Q2 are used to study students' learning outcomes in year 1 and 2, and research questions Q3 to Q5 are used to find out whether students' learning outcomes are connected to cognitive development. Research question Q6 focuses on whether this pedagogy matches teachers' knowledge of using it, based on their knowledge of applying technology-based pedagogy.

The research methodology of this project is the triangulation design where quantitative data are enriched by the collection of qualitative data. This mixture of quantitative and qualitative data collection in different research questions enables this study to interpret the values of this cognitive development-based pedagogy with different views from students and teachers. The research methods mainly include the quasi-experimental method, survey method and the rating scale anchoring method. With these methods, data are collected by using pre-test and post-test papers, questionnaires, and a checklist of rating scale anchoring mental specifications. They are analysed by two-tailed t-test, descriptive method with mean analysis and the one-way repeated measure ANOVA. These research and data analysis methods have been proven effective and used widely, in educational research projects.

This research project makes four major contributions: (i) the e-learning object used in this pedagogy can be used to improve students' learning performance in computer programming; (ii) evidence that a pedagogy focusing on cognitive development can be used to improve students' learning performance without being limited by programming languages; (iii) development of a cognitive development-based pedagogy for wide use in introductory programming without being limited by teachers' knowledge and programming languages; and (iv) learning with this cognitive development-based pedagogy builds up students' problem-solving skills and applies them to different subject areas.

With these achieved goals, this project therefore provides a conceptual and operational model for a pedagogical approach to Computer Science teachers design and use in their teaching process.

Acknowledgements

I would like to express my deep gratitude to my supervisors, Professor Paul Gibbs, School of Health and Education, Middlesex University, and Dr George Dafoulas, Associate Professor, School of Science and Technology, Middlesex University, for their patient guidance, enthusiastic encouragement and useful critiques of this research work. I also specifically express my deep and sincere gratitude to Professor Paul Gibbs for his advice on research methodologies, and assistance in keeping my progress on schedule. My grateful thanks are also extended to the IT teaching and technical support teams in Hong Kong Institute of Education, School of Professional and Continuing Education, for their support in conducting the research and their technical support in hosting the e-learning object. I also extend grateful thanks to my friends and colleagues for their support in investigating the e-learning object, making effective use of it in this project. Finally, I wish to thank my wife and my family for their support and encouragement throughout my study.

Contents

List of Tables

List of Figures

Chapter 1 Overview

1.1 Introduction

Introductory programming is a core module in a typical Computer Science programme. It aims to provide basic programming knowledge for year 1 students at undergraduate and tertiary levels (ACM Curriculum Committee on Computer Science, 2001). To learn this module, students are required to familiarize themselves with the programming terminologies, and establish the capacity for mental operations, logical reasoning and problem-solving to program. In this sense, there is always a strong demand for pedagogy to learn how to program while it is highly focused on these learning aspects. Motivated by this demand, this research project developed a cognitive development-based pedagogy to facilitate students' learning by using an online learning tool to help students organize and construct their knowledge of the learning contents. This learning tool is termed as e-learning object because it was developed by grouping 31 learning objects, and was developed as an online learning model, used together with the course materials on introductory programming through an e-learning platform. Every learning object is used to deliver a particular topic of program controls including the decision, iterative, array and functional controls. Figure 1.1 shows one of the learning objects used for delivering the topic of 'if-else' controls. The detail of the design of the e-learning object is discussed in Chapter 3.

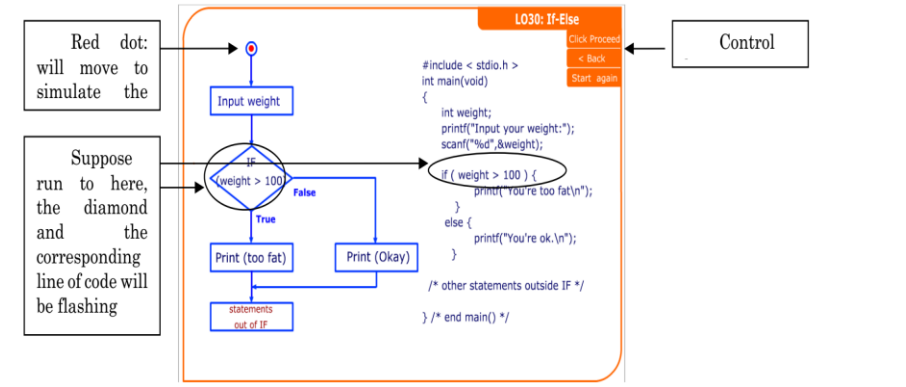


Figure 1.1 The typical layout of a learning object grouped in the e-learning object. It is focused on the learning contents of 'if-else' control.

This cognitive development-based pedagogy was evaluated with two major stages in this project. Stage 1 focuses on students' performance. It was defined based on some classic models relating to the attainment of cognitive domain levels, cognitive processing steps and instructional strategy for learning computer programming. These models provide crucial information for identifying how the e-learning object can be used in the scaffolding of this pedagogy to facilitate the cognitive development. This framework is illustrated in Table 2.7.

The stage 2 study focused on teachers' capacity, concerning whether this new pedagogy meets with teachers' knowledge of using technology for teaching and learning. The study was designed by referencing a pedagogically and operationally focused framework namely the 'technological, pedagogical and content knowledge (TPACK) framework'. The detail of how it was selected from diverse available frameworks and models, such as the Triple-E, SAMR, RAT and ADDIE, is discussed in Chapter 2.

For these two stages of study, there were six research questions defined. Research questions Q1 to Q5 belong to the stage 1 study, while only Q6 is included in the stage 2 study. Research questions Q1 and Q2 focus on students' learning outcomes on their year 1 and 2 studies, while research questions Q3, Q4 and Q5 focus on students' learning attitude, satisfaction with using the e-learning object and mental engagement with the pedagogical environment, respectively. They are used to evaluate whether the outcome of research questions Q1 and Q2 are connected to the improvement of students' cognitive presence on the learning context. Research question Q6 focuses on teachers' knowledge of using technological pedagogy based on the theoretical issues provided by the TPACK framework.

The research design is an approach where quantitative data are enriched by the collection of qualitative data. The purpose of methodological triangulation is to use more than one methodological approach, and the mixture of two or more methods to collect data in a single study enables this study to interpret different views from different stakeholders for a holistic picture of the use of this new pedagogy with the e-learning object in Hong Kong. The research methods for the different research questions include the quasi-experimental method, survey method and rating scale anchoring method. Data analysis includes the methods of two-tailed paired and independent t-tests, survey with descriptive mean analysis, and one-way repeated measure ANVOA. These methods have been widely used for educational research and have proven effective for qualitative and quantitative data analysis. The detail of how these methods relate to different research questions is summarized in Table 1.1.

Table 1.1 Summary of instrumentation, research and data analysis methods for the research questions

Stage	Research question	Research method	Data collection method	Instrumentation	Data analysis method
1	Q1	Quasi-experimental	Quantitative	Pre-test and post-test (C/C++ programming)	Two-tailed paired t-test & independent t-tests
	Q2	Quasi-experimental	Quantitative	Post-test (Java programming)	Two-tailed paired t-test & independent t-tests
	Q3	Survey	Quantitative	Student questionnaire	Descriptive method with mean analysis Two-tailed independent t-test
	Q4	Survey	Quantitative	Student questionnaire	Descriptive method and mean analysis Two-tailed independent t-test
	Q5	Rating scale and anchoring method	Quantitative	7 scales rating bipolar anchoring specifications	One-way repeated measure ANOVA (GLM) & Two-tailed independent t-test
2	Q6	Survey	Qualitative Quantitative	Questionnaire open-ended question	Descriptive method with mean analysis

1.2 Problem Statement

The problems of learning computer programming have been evidenced in many studies. For example, a study by Lahtinen, Ala-Mutka and Järvinen (2005) investigated 550 students and 34 teachers from many universities, focusing on (i) students' understanding of using program controls of decision, iteration and array to program; and (ii) the pedagogical approaches of teaching these program controls by teachers. The study found that most students could not correctly use loop control variables to establish the correct looping number, and that they were unable to effectively use program controls for program design because of poor logical reasoning skills and lack of understanding of using diverse program controls in an appropriate way. For teachers' study, it indicates that there is a demand on an alternative pedagogical approach that supports students in learning program semantics.

Another study from Gomes and Mendes (2007) pointed out that the reasons for difficulty in learning program semantics are closely connected to the taxing mental effort it takes to visualize the concept as an understandable model in the learners' mind. This mental problem is likely due to the poor design of pedagogical approaches. Moreover, a study from Piteira and Costa (2013) provided a similar result, namely that the high drop-out rates in year 1 Computer Studies (CS1) are due to students' mental difficulty and their paucity of abstract thinking in learning computer programming. It aligns with another related finding by Mhashi and Alakeel (2013) that also indicated the high drop-out rate in introductory programming because of the lack of support of an effective pedagogical approach.

In respect of using technology for learning computer programming, studies also present diverse problems and limitations. These studies concluded that a lot of technological, web-based models were designed primarily focusing on providing a drill and practice environment to increase students' skills of speeding up the programming processes, while fewer of them were set up to improve their cognitive skills to learn (e.g. Ala-Mutka, 2005; Belland, French, & Ertmer, 2008; Keengwe, Onchwari, & Wachira, 2008; Nguyen, 2008; Welsh, Harmes, & Winkelman, 2011). Although studies showed that some of these models were able to engage students in improving cognitive skills to learn, and they are likely to be designed comprising animations and simulations on specific programming languages. However, most of these models are still very limited in facilitating learners to develop their abilities of program design (Brent, Laura, & Donna, 2014; Keith, Aidan, & Susan, 2015; Pappas, 2017).

1.3 Motivations

The problem statement in Section 1.2 indicates diverse novices' difficulties in learning computer programming. These can be summarized in connection to students and teachers. For students, most difficulties relate to poor cognitive skills and low engagement with the taxing mental efforts required for learning the semantics of using a program language for program design. For teachers, the studies in Section 1.2 indicate a demand on a pedagogical approach which is not driven by learning programming language. These problems are basically consistent with diverse related studies (e.g. Apiola & Tedre, 2012; Lau & Yuen, 2009; Queirós, 2014; Teague, 2011), and the literature discussions in Chapter 2.2. They point out a gap between the existing pedagogical approaches and teachers' expectations of focusing primarily on learning program semantics and minimizing students' difficulties with mentally taxing efforts.

Motivated by this concern, this project developed a cognitive development-based pedagogy that focuses on improving students' semantics knowledge of computer program controls, and intends to provide an alternative way for teachers that is highly focused on improving students' cognitive presences in learning computer programming. The concept of developing this pedagogy is based on the cognitive benefits (Akyol & Garrison, 2011) of learning computer programming. As discussed in Chapter 2.5 and many reviewed studies (e.g. Mason & Cooper, 2013; Mayer & Fay, 2013; Scherer, Siddiq, & Sánchez Viveros, 2018), they conclude that the skills of cognitive development are a crucial attribute to help students to establish a mental model to interpret the abstract meanings of program semantics and logical reasoning, particularly with regard to novices learners, and the learning process is deeply connected with learners' cognitive presences on computer programming.

This research is focused on students taking sub-degree programmes, from diverse fields of studies, including 'Software Development', 'Computer Network Administration', and 'Computer Studies'. Most of them are local to Hong Kong and mainland China. Upon finishing the study, they would like to pursue a top-up degree in computer programming. This cognitive development-based pedagogy was examined in one of my teaching modules, 'Programming with C/C++ language', which was offered in the School of Continuing and Professional Education, the then Hong Kong Institute of Education, (SCPE, HKIED[1]). This module has two classes

[1] Renamed the Education University of Hong Kong (EdUHK).

with headcounts more than 50 students. As an teacher with more than 20 years' teaching experience in diverse computer programming languages, including the C/C++, Java and HTML/CSS, across many institutes and colleges, including the Institute of Vocational Education, ST Campus, SCPE, HKIED and UOWCC City University of Hong Kong, I also was motivated to get involved in this research project because it could facilitate of further developing my teaching competence and knowledge of pedagogical approaches and design in computer programming, while these are linking to my teacher professionalism.

The self-developments from this project is important for me as a teacher of sub-degree programmes in Hong Kong. We are strongly encouraged to improve our teaching performance, specifically for introductory programming in year 1 Computer Studies programmes, as it is a major subject and its knowledge crucially can be used across diverse subjects areas, including software and computer network developments. Besides, this project focuses of introductory programming as it essentially focuses on helping students to establish a systematic approach. It could be referenced to design an alternative approach of teaching computer programming and promotes to all teachers in Computer Studies including my working department. For example, the positive outcome in this research project can be used to encourage them considering of facilitating students' cognitive skills to learn how to program, instead of focusing on the developments of codes line by line. Also, the findings and recommendations in this research project can be used to provide pertinent further information for those who are interested in studying pedagogical approaches for teaching and learning computer programming.

1.4 Use of the E-learning Object

A distinguishing characteristic of this cognitive development-based pedagogy is the use of a cognitive learning tool to facilitate students' cognitive presences in understanding the program semantics. It is called the 'e-learning object' in this research project, of which it was designed by grouping 31 learning objects by segregating the topic of program controls into a unit of learning content. These learning objects were delivered together with the course materials through the HKIED's e-learning platform (Sooriamurthi, 2009; Tuparov, Tuparova, &

Tsarnakova, 2012). The e-learning object is claimed as cognitive tool as it was used to help students organize and construct knowledge of the learning contents (Apiola & Tedre, 2012; Fullan & Langworthy, 2014). The major advantages of using the e-learning object in the pedagogy of this project, as summarized as follows.

- Rather than use static materials, this pedagogy was designed with animated flowchart and program walk-through techniques to help teachers demonstrate the dynamic features of the program logics and semantics. These cognitive artefacts have proven to be effectively facilitating learners in linking up the abstract concept as a mentally concrete model in their mind, while is able to absorb the details of instructions (Lau & Yuen, 2009; Schulte & Bennedsen, 2006).
- By using the e-learning object, this pedagogy can be used as an alternative conceptual and operational approach that focuses on Computer Science teachers for use in their taught programming modules with a strong connection of facilitating cognitive processing.
- The design features of this pedagogy meet teachers' expectations as discussed in Section 1.2, where they expected to have a pedagogical approach which focuses primarily on learning program semantics rather than emphasizing the memorizing of keywords reserved for programming language and struggling with how to program with loose concepts (Gomes & Mendes, 2007; Mhashi & Alakeel, 2013).

1.5 Overview of Pedagogical Design

This newly developed cognitive development-based pedagogy is scaffolded with theories related to cognitive developments including Vygotsky's Zone of Proximal Development (Shabani et al., 2010) and Piaget's Theory of Cognitive Development (PsychoHawks, 2010), and the theories of subsequent works discussed in Chapter 2. At the pedagogical level, the learning processes were designed with three scaffolding functions, which are conceptual scaffolding, strategic scaffolding, and metacognitive scaffolding. In view of these scaffoldings, there are three major learning processes to: (i) facilitate the cognitive development of learning program semantics; (ii) help students conceptually map their program semantic knowledge to the coding process; and (iii) provide exercises to enhance students' learnt knowledge. The process of relating the e-learning to the scaffoldings, and the learning processes are illustrated in Figure 1.2.

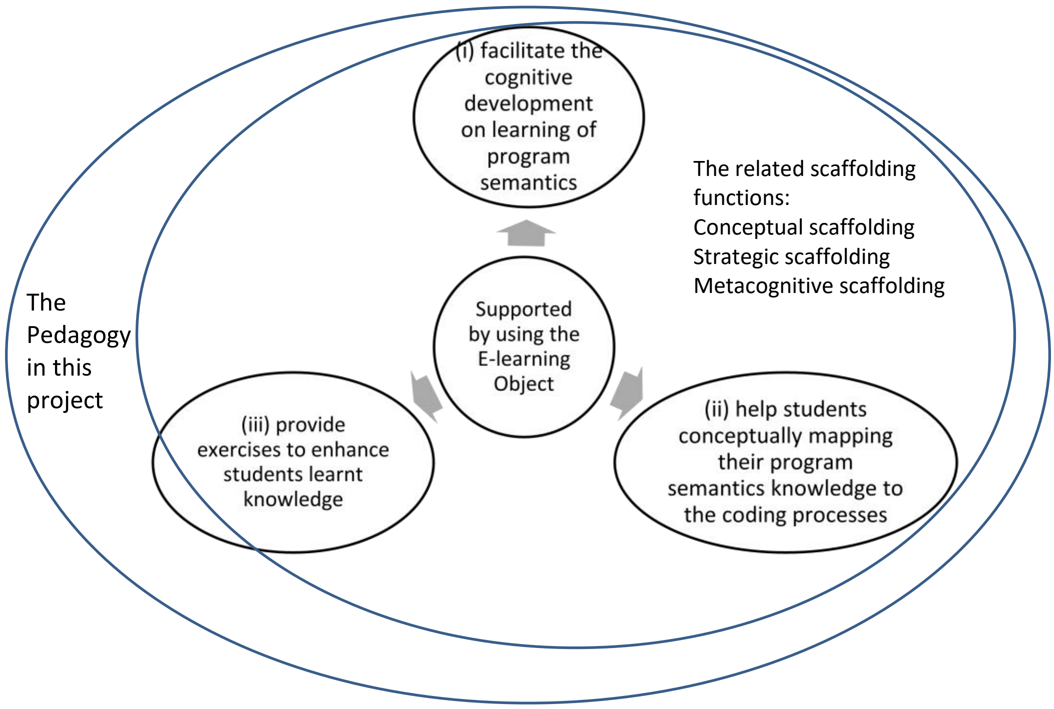


Figure 1.2 Scaffolding for the pedagogy developed in this project

In alignment with Vygotsky's Zone of Proximal Development (Shabani et al., 2010), the learning processes of (i) and (ii) are collaboratively interact between students and instructors, and they are crucially focusing on helping students construct knowledge based on their individual view of the learning contexts. For the learning process (iii), it provides students with a series of practices and requires students to complete them by themselves. By removing support from instructors, this learning process can be used to evaluate whether students' cognitive presence on the learning contexts is improved through using this pedagogy. The detail of the scaffolding and how it relates to these learning processes is discussed in Chapter 2, while the strategy of using the e-learning object to support them is discussed in Chapter 3.

1.6 Research Design

This research project includes a ‘design stage’ and two ‘study stages’. The ‘design stage’ used to validate the e-learning object could be used in this research project. For the two ‘study stages’, stage 1 focuses on students’ learning outcomes of the modules ‘Programming with C/C++ Language’ and ‘Introduction to Java Programming’ in year 1 and 2 respectively. This study relates to two consecutive years and without changed the organization of research group and control group as it intends to provide evidence that program semantics knowledge could give benefits for different program languages. For this reason, the research group only learnt with the new pedagogy in year 1, while learnt with normal approach in year 2. Therefore, if the research group still have a significantly better performance than the control group, this may indicate that the students’ built-up knowledge on program semantics through this pedagogy in year 1 can also provide advantages while use in different program languages. The detail of this research design is elaborated in Chapter 4.

For the stage 2, it used to evaluate whether the design of this cognitive development-based pedagogy matches teachers’ capacity for using technology for teaching and learning, based on a framework of ‘Technological, Pedagogical and Content Knowledge (TPACK)’. The detail of how this framework used to define the study also is discussed in Chapter 4.

Research Subjects

There were 51 students studying in year 1 and 2. They were randomly assigned to class A and class B and this was not changed in either year. This project randomly selected class A to be the research group, while class B was the control group. The stage 2 study focused on 52 experienced teachers. They were all specialists in the subject areas of Computer Science and Computer Studies. This is summarized in Table 1.2 below.

Table 1.2 Research subjects in this project

* Note: the difference in the numbers of students in years 1 and 2 is due to attrition.

Attribute	**Number of research subjects**	
Year 1 Research (n=51)	Research group (Class A)	Control group (Class B)
Programming with C/C++ language	25 students	26 students
Mean age of years	19.6 (SD=1.45)	20.1(SD=2.20)
Year 2 Research (n=*47)	Research group	Control group
Introduction to Java programming	23 students	24 students
Mean age in years	19.9(SD=2.13)	19.8 (SD=1.72)
Teacher study	52 IT/Computing teachers	

Research Questions

With the objectives of understanding whether this cognitive development-based pedagogy helps improve students' learning in their year 1 module 'Programming with C/C++ Language', as well as successfully transfer the learnt knowledge of program semantics to their year 2 module 'Introduction to Java Programming', six research questions were defined in this research project. They are listed as follows:

- Q1. Does the cognitive development-based pedagogy help students learn better in their first-year module 'Programming with C/C++ Language'?
- Q2. Do students who learnt with the cognitive development-based pedagogy in their first year still have better learning outcomes in their second-year module 'Introduction to Java Programming'?
- Q3. Is the exposure to learning with the cognitive development-based pedagogy associated with better attitudes towards learning computer programming?
- Q4. Do students feel that learning with the e-learning object provides them with a better approach to their learning?
- Q5. Is the exposure to learning with the e-learning object associated with better mental engagement in the pedagogical environment?
- Q6. Does the design of the cognitive development-based pedagogy match teachers' knowledge of how to use it, as suggested by the TPACK framework?

Research questions Q1 to Q5 were evaluated in stage 1, while Q6 was evaluated in stage 2. Research questions Q1 and Q2 were mainly used to evaluate whether students in the research group performed significantly better than the control group, while research questions Q3, Q4 and Q4 were defined to find out whether the learning outcomes in the research group are connected to the improvement of the cognitive presences in the learning contents by focusing on students' attitude, satisfaction and engagement with the learning environment by using the e-learning object.

Research question Q6 focuses on the stage 2 study. It is defined based on a framework, namely the technological, pedagogical and content knowledge (TPACK) framework (Pahlevi, 2017) to evaluate teachers' capacity for using technology for teaching and learning. The reason for applying TPACK to this study is that it is operationally focused, and a framework free of subject-matter evaluation (Snape & Fox-Turnbull, 2011). This is elaborated in Chapter 4.

Methodology

The research design of this project is an approach where quantitative data are enriched by the collection of qualitative data. This approach is embraced with the concept of triangulation design methodology. It mixes the quantitative and qualitative to collect data in different research questions, enabling this study to interpret the values of this cognitive development-based pedagogy with different views from students and teachers. The use of quantitative methods can provide a wide range of standardized data systematically to compare and explain the reality. This method also supports the precise definition of the goals by focusing on the summative outcome of the research variables.

The research methods applied to different research questions mainly include the quasi-experimental method, survey method and rating scale anchoring method. The quasi-experimental research was used for research questions Q1 and Q2, as it can provide an effective means to statistical analysis of the empirical study for learning with or without using the new pedagogy (Creswell, 2009). The quantitative data collected by the pre-test and post-test were analysed with two-tailed paired t-test and independent t-test. These are effective data analysis methods for measuring students' learning progress, and have been widely used in educational research for comparing differences between two independent study groups (Laerd.com, 2013a).

Research questions Q3, Q4 and Q6 used the Likert scale five-point response type survey (Robert et al., 2009) to obtain a wide range of views on the related studies. Students' responses on research question Q3 were analysed with two-tailed paired and independent t-test. It has been seen to be effective to measure the mean of responses on the research variables one by one. Research questions Q4 and Q6 used the descriptive method with mean analysis to analyse the attributes defined in the questionnaire individually.

Specifically, for research question Q6, there were two open-ended questions defined in the questionnaire to acquire a greater flexibility that does not limit respondents, focusing on the predefined research variables in the questionnaire (Robert et al., 2009), and it provides a flexible means for teachers to comment on the knowledge domains relating to the TPACK framework. These qualitative comments mean the result can offer more nuanced, authentic explanations of complex realities to validate the outcomes by measuring the consistency between the qualitative and quantitative data (Day, Sammons, & Gu, 2013; Lee & Smith, 2012), and can be used for comparing with the quantitative data.

Research question Q5 is focused on the variances of a set of pre-identified variables throughout a given period, therefore data analysis needs to be processed at multiple points in this period. To minimize the disturbance to students and to shorten students' response time, it applied the rating scale and anchoring method survey (Flick, 2009, pp. 214, 216, 306) focusing on a set of mental specifications (Robert et al., 2009, pp. 248–254). Students' responses were analysed with the one-way repeated measure ANOVA for data analysis. This method is effective for studying the variances of a research variable over different times in a predefined period (Laerd.com, 2013b).

The methodology of this research project is summarized in Table 1.1 and the detail is discussed in Chapter 4, while the outcomes of data analysis are presented in Chapter 5, and the findings drawn from the analysis results are discussed in Chapter 6.

1.7 The Contributions

There are four contributions made by this research project. They are briefly described in the following, while details of the accomplishment of these contributions is elaborated in Chapter 7.

The e-learning object used in this pedagogy can be used to improve students' learning performance in computer programming

In this research project, the e-learning object plays an important role in this cognitive development-based pedagogy. This finding of the research project also indicates that this pedagogy can significantly improve students' performance. Therefore, it could be concluded that the e-learning object itself can be used to improve students' learning performance. This contribution is important for showing a highly positive view on using a cognitive tool for facilitating the learning of computer programming.

Evidence that a pedagogy focusing on cognitive development can be used to improve students' learning performance without being limited by programming languages

The pedagogy developed in this research project is primarily focused on improving students' cognitive presences on program semantics. Besides, the findings of this research project indicate that it is not limited to use in the module 'Programming with C/C++ Language', but also gives positive effects on learning the module 'Introduction to Java Programming'. In this sense, this research project can provide evidence that if a pedagogy is developed focusing on cognitive development, it is not limited to only improving students' learning outcome in a specific programming language but also significantly provides positive effects in the use of different programming languages.

Development of a cognitive development-based pedagogy for wide use in introductory programming without being limited by teachers' knowledge and programming languages

The findings of this research project indicate this cognitive development-based pedagogy is not limited to a particular computer programming language, as shown in the previous contribution, but also the design matches teachers' knowledge of using it. With this concept, it can be concluded that this cognitive development-based pedagogy can be used in different introductory programming modules, which are focused on different programming languages and are conducted by different teachers. This contribution importantly provides evidence that this pedagogy possibly could be generalized to wider use in the field of introductory programming.

Learning with this cognitive development-based pedagogy builds up students' problem-solving skills and their application to different subject areas

Evidence from various research projects has shown cognitive development comes to be a meaningful way of establishing problem-solving skills (Bouzid, 2015; Spector, Lockee, Smaldino, & Herring, 2013; Wang & Chiew, 2018). Since this new pedagogy is focused on facilitating students' cognitive skills to learn how to program, theoretically it is not limited to only being effective for learning computer programming, but will also help students to develop their problem-solving skills, and apply these skills to different subject areas.

1.8 Reliability

The major concern of this research project is whether this cognitive development-based pedagogy can improve students' learning performance. This research project uses empirical study as it is engaged in evaluating students' learning outcomes by directly comparing their learning processes with using or not using this pedagogy. With this outcome, this project is also concerned with students' cognitive presences in the learning contexts delivered by using it. Therefore, the study also focuses on three critical factors which are specific to indicating cognitive presences: (i) if the students' learning attitude is changed by learning with this pedagogy; (ii) if students are satisfied with using the e-learning object to learn; and (iii) if students are positively engaged with the learning environment mentally. These have been widely used for diverse research projects that focus of attitudinal studies.

The research methodology for all research questions is suitable for and comparable to other similar research. Research questions Q1, Q2 and Q3 use two-tailed paired t-test to evaluate progression in the research and control groups, and use an independent t-test to compare the difference between them. This data analysis method is seen to be effective for quasi-experimental research, and also has been widely used in educational research.

With regard to research question Q4, students' satisfaction with using the e-learning object, the questionnaire for this research question was defined with four key focuses: 'Design', 'Effectiveness', 'Helpfulness', and 'Motivation'. They were designed by referencing the critical features of cognitive tool development and technology integration. These factors have been adopted in a wide range of educational applications for cognitive tool evaluation (Kaya & Akdemir, 2016; Robertson et al., 2014; Schatsky & Schwartz, 2015).

Research question Q5 focuses on knowing whether students can make use of the e-learning object in their learning processes rather than being restricted by it. As mental engagement needs to go through the whole learning process, this research question used the survey method, focusing on four mental specifications in four major points in the program modules. Their responses to these specifications were analysed with the one-way repeated ANOVA analysis. This is an effective method for comparing the variations of students' responses from time to time (Field, 2012; Laerd.com, 2013). Therefore, both the research and data analysis methods are secure for use in this research question.

All focused students were assigned to the control and research groups based on the class they belong to. The forming of these two classes was at the time they registered to the programme prior to the commencement of semester A and were not changed throughout the two years of study. These students were recruited from the pool of post-secondary colleges in Hong Kong, and there is no pre-condition to assigning students to classes. Therefore, the forming of the research and control groups can be reasonably assumed to be randomly selected.

There were five statements defined in research question Q3's pre-module survey to study students' background in learning computer programming. It indicates that student subjects have a very similar background and prior knowledge in computer programming. Both of the modules related to this research project were developed under the curriculum of introductory programming recommended by the ACM Curriculum Committee on Computing and Computer Science (ACM Curriculum Committee on Computer Science, 2001), and went through the accreditation exercise via the Hong Kong Council for Accreditation of Academic and Vocational Qualifications (HKCAAVQ). The accreditation report showed that the standard of the curriculum, its contents and process, are comparable to similar programmes offered in other institutes and colleges in Hong Kong. It indicates that the programming modules focused on in this research project are generally similar to other computer programming modules offered in Hong Kong's colleges and institutes. Moreover, I was the only lecturer for the classes of the research and control groups, so delivery discrepancy between these classes is avoided.

For the stage 2 study, all teachers invited to attend the research were well-qualified with adequate experience in teaching computer programming. With respect to using the TPACK framework for designing the questionnaire of research question Q6, the TPACK is context-free, operationally, and pedagogically focused, and has been proven in diverse literatures to be effective for the purpose pursued in this research question.

All the pre-test and post-test papers used, and the questionnaires, had been commented on by teachers. The responses likely indicate that they were designed at appropriate levels and that the statements in the questionnaire were precisely described. Moreover, the responses of these questionnaires were analysed with Cronbach's alpha reliability coefficient. All figures indicated the responses were reliable.

The e-learning object is a crucial cognitive learning tool for this cognitive development-based pedagogy. To find out whether it was capable of being used, it had been evaluated by teachers before being used in the stage 1 study. The feedback from this evaluation was positively indicated. Therefore, the e-learning object is secure for use in this research project.

1.9 Research Ethics

There are ethical issues in this project related to the students' learning in the classroom, the research process and procedure, and data privacy issues that need to be considered. The first is about students' learning activities. This project focuses on two different classes defined as the research group and control group. While only the research group learnt with the newly developed cognitive development-based pedagogy, therefore, there are ethical issues due to this discrepancy between both the research and control groups. For example, learning in the classroom generally relates to an unpredictable and multiple-faceted environment (Konza, 2012). To cope with these ethical issues, an adjustment to the teaching and learning process had been considered before beginning the module. Students were told about this research and that they may be involved in either the research or control groups. However, the teaching and learning processes in both groups were standardized including the course contents, strategy of materials delivery, instructional approaches and the learning environment used.

As a result, both the research and control groups used the same curriculum in turn, taking a session per week with the identical learning topic in all weeks, and the laboratory, lecture contact hours and the self-learning hours were the same as indicated in Table 3.4. Also, the teaching and learning approach for every session was conducted with a consistent teaching plan where there was a clear identification of the procedure and time of the lecture and tutorial within which the e-learning object would be used, while at the same time the context in the control group needed to be similar. This detail is provided in Table 3.5. The environment of materials delivery was not only the research group using the institute’s Blackboard system, which is an e-learning platform for supporting the teaching and learning process, but also the control group which used this platform to access course materials and resources, as illustrated in Figures 3.4 (a) and (b). This schedule was used in all learning sessions in both the control and research groups to maintain the consistency of the studying variables (Creswell, 2013; Cohen et al, 2011) and to ensure the discrepancies in the teaching and learning processes between the groups could be minimized. Besides, I was the only teacher in this study module, therefore the teaching and learning processes as above-mentioned in both groups could be controlled by me and consistently performed so as to minimize any teaching bias.

However, there is another concern that students’ overall grade on this programme would be affected by only the research group having used the e-learning object in the pedagogical approach. What a student learnt in this module is only part of their taking the programme in the institute, such that this research only partially measures their performance with the indicators concerned in the research. This additional benefit to the research group may provide positive effects not only in the research content but also beneficial to the final outcome of the programme (Creswell, 2013; Konza, 2012). This comes to be an ethical issue regarding whether the control group experiences an unfair process because of this research.

With regard to this concern, while my position as an inside researcher allowed me to access greater insight into complex data in the research (Costley, Elliott, & Gibbs, 2010, pp. 6–7), I also requested to be the programme leader of the study module. This position also allowed me to monitor the progress of the research and control groups not only as to the outcome of this module, but also their performance over the whole programme. Concerning this, I proposed the moderation of the final grade for the research group and control group should be separated to minimize the positive effects bias on the research group. This proposal was approved by the Programme Committee of School of Continuing and Professional Education, Hong Kong Institute of Education (SCPE HKIED).

As for the ethical issue on the process and procedure of the research, this research project has been approved by the Education, Research Ethics Committee of Middlesex University, London. All the related research was conducted by me with a precise and understandable protocol, and there were no major concerns raised about the procedures and processes. Before the research, all students and teachers received an invitation, with a particular information sheet (PIS) attached, together with a consent form attached to the questionnaire. It was used to explain in detail about purpose, potential risk, confidentiality, and right and process to withdraw from the research. In addition, there is a consent form with the questionnaire sent to students. Therefore, students can find out about the questionnaires before they decide on consenting to being involved.

There were two research questions using empirical study, as it involved the use of this newly developed cognitive development-based pedagogy. The approval from the Head of Department and the Programme Management Committee had been acquired subject to the concerned. The protocol of conducting the pre-test and post-test complied with guidelines from the Quality Assurance Committee of the then Hong Kong Institute of Education, School of Continuing and Professional Education (SCPE HKIED) to minimize the effect on the students' grading results in the overall programme. Besides, students received detailed explanations about their roles and involvement while they were required to attend the study.

As for data privacy, the consent form specified that all the collected information would be kept securely, and no personal identity could be figured out from the information. Besides, the topic of this research did not include diagnosis from specialist instruments or observations of inappropriate behaviour by participants. Therefore, the disclosure of these types of private information is unlikely to occur. The collection of data was only subject to the need for analysis of the attainment of this research project, and they were focused in a way that would not cause substantial disturbance or distress to any individual, and not further processed in any manner incompatible with the original purpose. All analysis sheets and drafts of responses were physically retained in a locked drawer in my college office. They will be disposed of a year after the completion of this research project.

1.10 Limitations of this Research Project

The first limitation is specific to the use of the e-learning object in this study, considering that this model was only commented on by 12 teachers based on their experience and professional knowledge, rather than using an empirical study. The ultimate goal of using it in the introductory programming may not be appropriately addressed. Bearing this concern in mind, therefore, teachers are only seen as being consulted for a professional view on whether improvements are required upon using it in the programming module. To know the significance of using the e-learning object, there was a research question defined to directly focus on the capacity of the e-learning object regarding the key areas of cognitive tool design. These factors are ‘Pedagogical Issues’, ‘Content Design’, ‘Delivery Strategy’ and ‘Layout Design’. The outcome of this research question therefore can be linked to whether the e-learning object has significant effects on the learning processes.

Another limitation relating to the objective of generalizing this cognitive development-based pedagogy for wider use in introductory programming is that the study of teachers’ capacity was based on using the TPACK framework. In this regard, it is not an empirical study, and some features of teachers’ adoptability may not adequately reflect in the study: for example, the TPACK framework does not focus on the curriculum approach, teachers’ culture and their teaching behaviours.

The last limitation relates to the findings of this project are not effective in ameliorating two major sources of students' difficulty in learning computer programming: their boredom and their attitude to learning computer programming. As this pedagogy intends to improve students' learning outcome, and was evaluated based on using a questionnaire and a checklist focusing on students' feelings to determine their attitude, and some essential students' attributes, such as their motivation and the learning culture, such that it may not be examined sufficiently. Therefore, though these findings indicate that this pedagogy cannot significantly improve students' boredom or attitude to the learning process, however, based on this insufficiency related to the study, it cannot determine whether the outcomes link to this newly developed pedagogy or has other reasons.

Focuses on these limitations, there are 12 recommendations addressed in Chapter 7, providing strategy of using this cognitive development-based pedagogy.

1.11 Outline of Chapters

Chapter 1: Overview

This chapter provides an overview of this research. It summarizes its context, along with the problem statement, motivation, and the overview of scaffolding. It follows with the methodology of this research and data analysis methods to be used in this project. As the first chapter, it also discusses reliability, ethics, and limitations. The main objective of this chapter is to give readers an overview of the motivation, processes, and contributions of this project.

Chapter 2: Literature Review

This chapter provides a review and discussion of the literature that importantly relates to this research project. The review processes surround many major areas, which include the difficult of introductory programming, cognitive development, diverse instructional methods, and pedagogical and scaffolding design. It also provides discussion on the developments of learning technologies including the e-learning and learning objects, and some existing models and frameworks used to evaluate the process and quality of using technologies for teaching and learning. This chapter also includes some discussion on information processing models and is followed with design of the e-learning object in later chapters. The last part of this chapter establishes two theoretical frameworks for defining the research processes of this project. These researches focus on whether this cognitive development-based pedagogy delivers better learning outcomes for students, and whether it is designed in a way that matches teachers' capacity for using it in teaching and learning.

Chapter 3: The E-learning Object

This chapter discusses design of the e-learning object and how it can be used to support the three-steps scaffolding of this cognitive development-based pedagogy. The main focuses of this chapter include the shape and size of the e-learning object, the delivery strategy, and the layout design of an individual learning object. At the end of this chapter, there is a discussion of the process and outcome of the pre-project evaluation. It is used to identify whether the e-learning object meets the needs for being used in this research project.

Chapter 4: Research Design and Methodology

This chapter focuses of the research design and methodology, providing an overview of the process, research, instrumentation, and data analysis methods of this research project. The discussions include the research paradigm, its advantages and strategy for use in this project. This chapter also provides the rationale for defining the research questions in respect of the requirements defined in the project objectives.

Chapter 5: Data Collection and Analysis

This chapter provides the spectrum of data collection and the analysis outcomes in the two-stages of study. The discussions are organized by research questions with the data analysis, and they are presented according to the presentation style consistent with using the SPSS statistical software package. The goal of this chapter is to provide evidences for the findings from the data analysis outcomes, while these findings are discussed in following chapter.

Chapter 6: Discussions

This short chapter focuses on summarizing the findings of this project, of which there is evidence from the data analysis discussed in the previous chapter. These findings are discussed and presented individually with the related research questions.

Chapter 7: Conclusion and Recommendations

This chapter starts by recapping the background and rationale of this project. It also primarily provides conclusions on those findings discussed in the previous chapter, and based on them develops four major contributions from this project. Twelve recommendations are provided on implementing this cognitive development-based pedagogy for introductory programming. As the final chapter of this thesis, it also discusses some major features that are valuable for future study.

1.12 Summary

Chapter 1 provides an overview of this research project that includes the problem statements, motivations, definition of the research questions, and overview of the scaffolding of the cognitive development-based pedagogy developed in this project. It also includes an introduction to the target subjects and methodology, and some major attributes relating to the research such as ethics, reliability, and limitations. At the end of this chapter, there is a summary of all remaining chapters. Instead of detailed discussion, this chapter aims to give readers an overview of the what, why and how of this research project.

Chapter 2 Literature Review

2.1 Introduction

Difficulties with learning computer programming are multidimensionally specific to poor pedagogical approaches that focus on improving students' cognitive presence on understanding the logical reasoning and minimizing the taxing mental difficulties. Motivated by these identified difficulties, this chapter conducts a literature review of the critical theories of teaching and learning, instructional approaches, and functions of scaffolding relating to design a cognitive development-based pedagogy to facilitate students learning in introductory programming.

Learning is a series of comprehensive studies and processes describing how people learn. It helps people to understand the complexity of diverse educational objectives and outcomes of a learning context (Driscoll, 2002). The approaches of learning can be described by three major paradigms: behaviourism, constructivism and cognitivism (Emerging Technology, 2000; Hobbs, 2002). The perspective of cognitivism and constructivism was the major concern of this research project. Their development is based on Jean Piaget's (1896–1980) and Lev Vygotsky's (1896–1934) cognitive development theories (PsychoHawks, 2010; Psychology Notes HQ, 2018). These two paradigms quickly came to be fundamental for scaffolding design, and afterwards were widely supported by the use of diverse technologies (e.g. H. Brown, 2012; Kaya & Akdemir, 2016; Peach-Squibb, 2013; Schatsky & Schwartz, 2015). With an insight of these theoretical issues, this chapter discusses diverse approaches to, and strategies for using technology for teaching and learning, specifically in computer programming. These discussions begin with some basic classic models, and to which discussions of some modern technological models. These include the revolutionary e-learning and learning objects.

This research project mainly focuses on the development of a cognitive tool defined as e-learning object, which is used to facilitate the cognitive processes in the scaffolding of a cognitive development-based pedagogy. 'Cognitive processes' is defined as a series of processes that synthesize the cognitive development from different perspectives and contexts of cognition to transfer existing knowledge to new knowledge (Mnguni, 2014), and this cognitive tool is called as ‘e-learning object’ as it was developed by grouping 31 learning objects, each of them focused on delivering a learning topic of program controls, and was used together with the course materials. In this sense, the use of an e-learning object in this cognitive development-based pedagogy is a major characteristic distinct from other pedagogical approaches. This aspect is discussed in Chapter 3, the e-learning object design.

Scaffolding was described by Wood, Burner and Ross (1976) as the activities between learners and tutors that assist the learner in accomplishing a difficult task with independence. This cognitive development-based pedagogy is related to three major learning processes and was defined with three major functions of scaffolding. These learning processes are to: (i) facilitate the cognitive development on learning program semantics; (ii) help students in conceptually mapping their program semantic knowledge to the coding process; and (iii) provide exercises to enhance students’ learnt knowledge. They were defined to achieve the functions of conceptual scaffolding, strategic scaffolding, and metacognitive scaffolding. They are the major contexts to be reviewed in this chapter.

Along with scaffolding, this chapter also establishes a cognitive processing framework for learning computer programming that defines a full spectrum of the cognitive processing steps and the attainment of cognitive domains to identify how the three learning processes are supported by the e-learning object. The workings of the framework are based on a classic model of cognitive processing steps defined by Gerald Grow in 1996 and the following related works (e.g. Kaya & Akdemir, 2016; Keogh & Pearson, 2011; Pappas, 2014; Schatsky & Schwartz, 2015). Therefore, this cognitive processing framework can be achieved by mapping the cognitive processing steps to the instructional strategy. For identifying the attainments of cognitive domain levels, this framework applies the Bloom’s Taxonomy cognitive domain model (Bevier, 2010). This detail is presented in Table 2.7.

While teachers are major implementors of a pedagogy, it needs to be understood whether they would adopt the pedagogy of this project in their teaching processes. Therefore, it needs to show whether this pedagogy was designed in line with teachers’ capacity for using it in their teaching and learning. As a result, this chapter conducts a comprehensive review of diverse evaluation models that could be used for this purpose. These include: The Technological, Pedagogical, Content Knowledge (TPACK) framework; Substitution Augmentation Modification Redefinition (SAMR); Triple-E; Technology Integration Matrix (TIM); Analyse, Design, Develop, Implement and Evaluate (ADDIE); and Replacement, Authentication & Transformation (RAT). The study reveals that the TPACK framework is suitable for used, as it is context-free and focused on pedagogical operations (Koehler & Mishra, 2009), unlike many others, which are mere guidelines or standards to measure the extent of integrating information technology into teaching and learning (Annaweisspol, 2016{Learn NC article, 2015 #1152; Tuğba, 2016).

In summary, the reviewed contexts in this chapter involve a wide range of theoretical issues including cognitive development, functions of scaffolding specific to learning computer programming, technology revolutions from classic models to modern approaches including e-learning and learning objects. Along with these focuses, Table 2.1 provides a summary of how these contexts relate to the research questions of this project.

Table 2.1 Context of reviewed literatures relating to the research question of this project

Context of the reviewed literatures	**Rationale**	**Related research questions**
Instructional approach and difficulties of learning computer programming	A field of study of learning computer programming which is the main focus of this research project	Motivation of this project
Theories of pedagogy, scaffolding and instructional methods theories	Provide research evidence that the pedagogy in this project can be used to improve students' learning outcomes	Q1 and Q2
Cognitive development and modelling, visualization techniques, the theory of information process	Provide research evidence for the design of the cognitive strategy applied to the cognitive development-based pedagogy of this research project	Q1, Q2, Q3 and Q4
Pedagogy for e-learning and the use of learning objects.	Review the literature related to the application of the e-learning and learning object in relation to this cognitive development-based pedagogy in cognitive developments	Q4 and Q5
Theoretical framework of this research project including the pedagogical issues and the TPACK framework	Provide theoretical foundation that defines this research project	Q6

2.2 Difficulties of Introductory Programming

The introductory programming module was endorsed by the Association for Computer Machinery Curriculum Committee on Computer Science (2001) in the 1960s. It was designed for novice learners of computer programming to provide basic knowledge and skills for program development (Bills & Bile, 2005). However, learning computer programming is widely recognized to be difficult because of hard mental problems, such as low cognitive presences, poor understanding of the semantics and logical reasoning, and lack of problem-solving skills (I. T. Chan, 2008; Sun-Ongerth, 2012). For novices, besides these mental problems, there are also some that are specific to introductory programming. These problems relate to orientation, consistency, proneness to error, and role expressiveness (Piteira & Costa, 2012, 2013).

Environmental Orientation

This problem relates to students' lack of familiarity with the endemic jargon and terminologies in Computer Science, and the low adaptivity of the learning context (Pears et al., 2007). Although these terms and contexts are defined in learning objectives, if they are unfamiliar with these terminologies, students find it difficult to match them to the context. Besides, environmental orientation also includes the skill to cope with the integrated development programming (IDE) tool. This complex and professionally oriented software tool often blurs students' focus on the chief purpose of introductory programming (Cetin & Andrews-Larson, 2016).

Consistency

The absence of consistent thought between different program languages gives students great problems in learning logical concepts. For example, it is common to see assignment statements 'a=2' and afterwards 'b=a'; novices may think the variable 'a' no longer has the value of '2', and instead it is now 'b'. Also, they always find it difficult to understand the meaning of 'k=k+1', since k is never equal to 'k+1'. The reason is that novices are likely to use mathematical concepts to interpret the logical meaning of a variable. Research indicates most program bugs stem from consistency problems (Chan, 2008), as most of the context of consistency problems are beyond the learners' domain knowledge. Regarding these issues, this project uses a syntax-free approach to minimize students' difficulties and to encourage them to concentrate on the semantics and logic of programming (Kedar & Thakare, 2009, pp. 1–8).

Proneness to Error

This problem mostly arises from learners' cognitive deficiency. It may be due to their poor understanding of program semantics (Wagner, 2008) and lack of a systematic strategy for program design. As a result, novice learners tend to work with loose, fragmental and basic knowledge (Chan, 2008). Proneness to error may be minimized by introducing effective pedagogy.

Role Expressiveness

Role expressiveness is a concept that refers to the extent to which information about the program elements can be determined from their structure. For example, the function names println() and readline() explicitly identify their purpose, so they are regarded as having 'high role expressiveness'. Good use of the programming language with high expressiveness helps students to improve their cognition of the learning aspects. In contrast, poor role expressiveness degrades students' understanding of the program. This problem results in students' low interest in learning computer programming. The use of a syntax-free approach to learning computer programming is an effective way to increasing role expressiveness (Tuparov et al., 2012).

Syntax vs Semantics

A program is constructed by both syntax and semantics. Syntax defines the grammatical issues of a program, while semantics presents the meaning of that program, and needs to deal with multiple layers of abstraction of a program's logic (Bruce, 2005). Evidenced by studies, understanding program semantics is mentally taxing for novice learners. For example, a survey by Lahtinen, Ala-Mutka and Järvinen (2005) evaluated 550 students and 34 teachers from several universities' introductory programming modules to see if students could remember syntax but still have a poor understanding of the semantics. This study discovered a large number of students could write a control block correctly, but could not construct it in a meaningful way: for example, only very few students could define the loop control variables in a suitable way to increase and decrease the looping appropriately. Besides, many studies also indicated that the difficulties of learning semantics were connected to poor pedagogical design, for which they lack a strategy to visualize the complex concepts of program semantics and generalize the learning processes to focus on it (Gomes & Mendes, 2007; Nguyen, 2008; Oswalt, 2010). The other study from Chen (2010, pp. 389–394) revealed a consistent result, finding that students were often unable to map a corrected concept to a specific problem because of poor cognitive skills to mentally create a learning model. All of these studies concluded that the use of visualization techniques, such as the concept map strategy, well-structured animations, and simulations, can minimize these mental difficulties.

2.3 Problem-solving Model for Computer Programming

The concept of problem-solving skills in learning computer programming is observed to be an inverse relationship in earlier studies. Much evidence concluded that problem-solving skills are directly involved in learning computer programming (Apiola & Tedre, 2012; Fidge & Teague, 2009). However, some others described learning computer programming as strategically based on tackling problems by using coding skills (Bouzid, 2015; Fidge & Teague, 2009; Havenga & Maria, 2009). Both of these concepts linking up learning computer programming can be modelled by problem-solving steps (Spector et al., 2013). As indicated in the early 1980s, the problem-solving model was developed by mathematical concepts. The most profound model is George Pólya's (1887–1985) problem-solving model (Alfeld, 2011). It identifies four major steps in problem-solving: 'recognize the problem'; 'design a plan'; 'implement the plan'; and 'generalize the solution'. This linear processing model provided a concept for subsequent related works (Morrosty, 2017; Pedroni, 2003, p. 19) and was extended for use in the instructional strategy of learning computer programming. In early 1996, computer programming strategy was identified as two sequenced major phases: 'problem-solving' and 'program implementation'. Problem-solving includes the steps of 'analysis and specification', ‘general solution (algorithm)’ and ‘verify’, while program implementation includes ‘create a solution (program)’, ‘test’ and ‘maintenance’ (Ismail, Ngah, & Umar, 2010).

Until learning computer programming was materialized with curriculum and textbooks, instructional strategy still followed the linear model using a stepwise, incremental style with problem identification and design before coding and testing (Abid, Farooq, Farooq, Abid, & Shafiq, 2015; Adalbert, Jignesh, & Richard, 2018; Kannan, A. Muthu, & Jennifer, 2015). The processing models of learning computer programming developed later were stimulated by this incremental style. As indicated in early work from David Perkins in 1985, Chris Hancock and Rebecca Simmons in 1986, etc. (Michael, 2008, p. 9; Perkins, Hancock, Hobbs, Martin, & Simmons, 1986; Perkins, Schwartz, & Simmons, 1985), the instructional strategy of computer programming was designed including five major steps: ‘Problem Identification’, ‘Program Design’, ‘Program Implementation’, ‘Testing and Debugging’ and ‘Program Evaluation’. This stepwise model is adopted for this project for scaffolding design as it had been widely adopted in the academic curriculum for a long period. Table 2.2 below illustrates how this linear style of learning computer programming relates to the problem-solving model.

Table 2.2 The relationship of problem-solving model and computer programming instructional strategy

Stage/ step	**Problem-solving phases**	**Linear problem-solving model applied to computer programming strategy**
1	Problem Understanding	Problem Identification (e.g. determine what the input/output is, what problem the program intends to solve)
2	Design a Plan	Program Design (e.g. elaborate the program’s logics and flow of control. Using appropriate program controls to construct the logical blocks)
3	Implement the Plan	Program Implementation (e.g. code the program with specific program language)
4	Review and Verify the Solution	Testing and Debugging (e.g. find and correct programming bugs, including the syntactical and semantics errors)
		Program Evaluation (e.g. validate and verify the program. In some cases, redesign of the program may be required)

The instructional strategy of computer programming presented in Table 2.2 provides two major advantages for introductory programming. They are discussed as follows:

- This strategy suggests clearly defining a borderline breaking down of the complexity of programming to manageable steps (Michael, 2008, pp. 10–11). It makes it possible for all specific deliverables to be reviewed without overlapping with the teaching process, and thus can be precisely materialized based on the course curriculum (Mayer, 2013, pp. 153–158).
- This strategy facilitates logical thinking and is more specific to novice working style (Prieto, 2016). With this benefit, it is possible to design instructions which clearly focus on the two major learning phases of computer programming: ‘Program Design’ and ‘Program Implementation’ (Mohorovičić & Strčić, 2011). These phases are particularly focused on by the pedagogy proposed in this project as they are highly connected to students’ understanding of the semantics of computer programming (Eid & Millham, 2012).

2.4 Instructional Approaches

There are diverse instructional approaches for learning computer programming designed for different paradigms of programming language. However, a large number of these approaches still rely heavily on memorizing the learning contents by familiarization with terminologies, keywords and syntactical detail to construct the program (Caspersen & Bennedsen, 2007; Mayer & Fay, 2013, p. 67). These instructional approaches are not effective in helping students to develop a mental model of the programming process. Some other instructional approaches are more focused on facilitating cognitive development and problem-solving skills improvements. This section reviews some of these major approaches to instructions and uses them to identify whether they can be used in this project.

Instruction-based Approach

This approach uses well-structured instructions to explain the learning contents directly (Mohorovičić & Strčić, 2011). It mainly supports incremental learning, and is an ideal approach to exploring problem-solving skills in a stepwise learning style (Sooriamurthi, 2009). This instructional approach possibly can be supported by using learning objects, as its learning contents are chunked by design into manageable learning units. In this sense, specific contents can be supported by introducing cognitive elements (Pears et al., 2007). Therefore, the instructional approach can be closely connected to cognitive-based instructions and move students to further accomplishment (Linn & Dalbey, 2010).

Structured Programming Approach

This approach is specifically suitable for structure programming as it delivers the programming contents from basic program controls, and then extends to an understanding of the entire programming process. This approach highlights the structural features in computer programming, including selection, decision, iterative and functional controls (Lau & Yuen, 2009). It adopts a top-down manner that disaggregates large problems into a manageable size. Therefore, this approach is suitable for novice learners and is particularly used for introductory programming.

Walk-through Approach

The walk-through approach is self-exploration pedagogy. It is designed to employ the advantages of collaborative learning while emphasizing learners' work instead of the instructor's presentation (Castelli, Bergman, Lau, & Oblinger, 2010). This approach builds up learners' programming ability through reading, tracing and understanding of a given algorithm, by the need to explore the semantics in an algorithm (Sooriamurthi, 2009), and facilitates concept mapping (Jyotsana & Ajay, 2016).

Syntax-free Approach

The syntax-free approach focuses on teaching program semantics through a strategy of not using a programming language, but instead using some 'mind realization' techniques (Fidge & Teague, 2009). It is promoted by following the concept of semantics as independent of the syntactical, and aims to reduce students' effort to understand the syntax of a programming language (Wagner, 2008). The early design of the syntax-free approach relied on using static flowcharts and pseudocode (Garner, 2003; Holman, Ong, Domeshek, & Mohammed, 2002). However, in later approaches, it has been changed to employ mindtools and cognitive tools, such as

Mindstorms robots (Mason & Cooper, 2013), program logic simulators (Tuparov et al., 2012) and animated flowcharts (Dol Aher, 2015), to support learning.

Problem-based Approach

The problem-based approach provides learners with opportunities to accumulate problem-solving experience and explore new knowledge. This approach generally relies upon ‘mindtools’ to help students create a problem-solving model for programming (Apiola & Tedre, 2012). This approach sometimes is independent of program language, therefore it seems closer to psychological issues rather than learning how to program (Sternberg & Zhang, 2009). Therefore, this approach is contested for novices and requires instructors to provide clear and meaningful clues to the problem, as well as to effectively monitor progress (Bouzid, 2015; Havenga & Maria, 2009, pp. 97–99).

Theory-based Approach

This approach is designed following the theory of cognitive domains, and avoids cognitive development being held back by a poor pedagogical approach (Lau & Yuen, 2009). For this project, as the pedagogy is driven by cognitive development, the scaffolding for this approach is subject to cognitive and information processing theories.

Inductive Reasoning Approach

Inductive reasoning takes a logical approach to designing algorithms. The learning process is based on defining rules and predicates for a series of reasoning activities to analyse problems and forecast the algorithm execution outcomes (Changing Mind, 2016; Flener & Schmid, 2010). As this approach involves construction of programs from the standpoint of exemplary behaviour, and exchanges theory and techniques to progress the inductive reasoning, it presents most academic students with difficulties in learning with this approach (Rafae, Lars, & Mehdi, 2006).

The Approach of Improving Learning Environment

This approach covers improving the environment to enhance learning. Its implementation includes apprenticeship learning (Caspersen & Bennedsen, 2007) and uses technological tools to contextualize learners' cognition (Dehnadi, 2009, pp. 24–30). This approach is contested, as it takes a huge effort to improve the environment and develop an effective strategy to mentally engage students in the environment. It needs the learning concepts, facts and objectives that are consistent with the environment to be explicitly defined (Keith et al., 2015).

2.5 Cognitive Development for Learning Computer Programming

The theory of cognitive development is dominated by Jean Piaget's (1896–1980) cognitive development stages and Lev Vygotsky's (1896–1934) social cultural cognitive theory. Piaget claimed that cognitive development is at the centre of the human organism, a strategy of addressing learners' interpretations of perceptions, skills of knowledge to understand a new and complex concept in a specific learning context (PsychoHawks, 2010), through the processes of absorption and application with four major stages, which are sensorimotor stage, preoperational stage, concrete operational stage, and formal operational stage (Dede, 2014; Scherer & Siddiq, 2016). On this other side, Lev Vygotsky's theory focuses on how culture and social interaction lead cognitive development. He believed that social interaction plays an important role in the way we develop cognitively (McLeod, 2014). In this sense, cognitive development is seen as a 'progressive reorganization of mental process resulting from biological maturation and environmental experience', and knowledge is built from the learning experiences between the existing knowledge and the learning processes (Breivik, 2016).

While the cognitive theories form Jean Piaget and Lev Vygotsky earlier works received the greatest attention, the outcome of cognitive development is described as cognitive presences. Cognitive presence was inspired by the works of John Dewey (1933) on reflective teaching, and described as 'the exploration, resolution and confirmation of understanding a learning context'. The degrees of cognitive presence can be described as 'attainment of cognitive domain levels' (Carlston, 2013; Garrison, 2006).

Cognitive development is crucial for learning computer programming specifically for the understanding of program semantics and logical reasoning. They are the major learning difficulties for novice learners (Akyol & Garrison, 2011), as indicated in a meta-analysis on 105 quasi- and experimental studies on educators, policymakers and computer scientist examiners. This study found that the performance of learning computer programming is closely connected with learners' cognitive presences on the learning contexts (Scherer et al., 2018). Besides, this study also pointed out that there was a large number of studies indicating a high degree of attainment of cognitive presences crucially facilitating students to be problem-solvers, creative thinkers, and logical reasoners.

Another study from White (2005) consolidated diverse prior researches that formulated a theory of students' ability in cognitive development and the cognitive characteristics possessed by programmers, such as their ability in information processing. The study concluded that if students do not possess the cognitive characteristics of a programmer, they may be bored by the abstractive concepts of programming logics. However, in contrast, if students met such cognitive characteristics, they were able to face challenges and optimize their success. A further study by Kodat (2007, #853), which assessed 302 students at the sixth grade of college level with the Langeot Test, also indicated students with a higher degree of cognitive presence on logical reasoning generally achieve a better performance in learning computer programming.

In concluding of the studies above-mentioned, cognitive development comes to be a crucial attribute for learning computer programming. Therefore, it is the major feature to be concerned in this project, and rationale the design of this cognitive development-based pedagogy.

2.6 Information Processing

Information processing is a concept of encoding the learning information to the human memory, and processing it to be working memory by reconstructing the information as learners' manageable forms of presentation, and eventually converts as long-term memory (McLeod, 2018; Pappas, 2014). In this sense, a quality information processing strategy is a major priority for supporting cognitive development in a pedagogy. Regarding the use of technologies, this strategy relates to two major techniques: the concept map and visualization techniques (Schatsky & Schwartz, 2015).

The concept map is rooted in cognitive psychology theory. It seeks to produce a graphical representation of the concept that is important to a given domain and how they are related. It encourages students to use a new way to think about the meanings and logics of that new concept in a learning context (Lu & Dosher, 2007). In respect to implementation, the concept map normally acts as a graphic organizer to enrich learners' understanding of a complex, new concept, and connects them to the learner's existing knowledge (Brent et al., 2014; Jyotsana & Ajay, 2016).

The visualization technique relates to processing information by making use of different types of cognitive artefacts to reconstruct the information by visualizing the underlying concepts of the information. It enables learners to develop their cognitive skills by using four major categories of properties: visual imagery, major technologies, style and process, and delivery structure (Brant, 2013; Mnguni, 2014; Olubunmi & Adesope, 2007; Ware, 2013). They are discussed as follows:

- Visual imagery is defined as 'image and imagination' that includes the use of dynamic graphics, simulations and animations to transform information into sensible forms (Chen, 2010; Heersmink, 2013). It is a major technique for reinforcing humans' cognitive presence on a new concept (D. Jonassen, Howland, Marra, & Crismond, 2010; J. Lee, 2007).

- Style refers to learners' interaction with the model. It is a major step of information reconstruction and also relates to strategically supporting model navigation. The style of a cognitive tool needs to provide a degree of autonomy in line with students' learning style (Carlston, 2013, p. 149).

- Major technology refers to using technology-based external stimuli to construct a pictorial image in learners' working memory. Stimuli can be any type of cognitive artefacts, such as video, audio or 2D and 3D animations, and modern technologies such as virtual reality (VR) and augmented reality (AR) (Heersmink, 2013), that strategically provide interactions and controls for instructors and learners (Morrosty, 2017, pp. 109–111).

- The delivery structure concerns how the cognitive model can be used with the existing learning materials. It focuses on enabling the environment to offer an information space that can be traversed in different ways by individual learners (Bodemer & Faust, 2006). Because of technology, a learning environment needs to be used for optimizing students' stylistic strengths and complements their weaknesses (Soto, 2013).

The application of information processing concepts to design the e-learning objects is further discussed in Chapter 4.

2.7 Cognitive Processing Steps

Cognitive processes refer to a series of processes that synthesize the cognitive elements from different perspectives and contexts of cognition in a developing field to transfer existing knowledge to new knowledge (Mnguni, 2014). It is a mental action of acquiring knowledge through experience and senses. A model of cognitive processes normally comprises a group of cognitive artefacts for effective information processing (Garrison, 2017).

An early model of cognitive processing steps was developed by Gerald Grow in 1996, known as the 'cognitive processing steps' (Grow, 1996). It is a five-step model including 'Problem Comprehension'; 'Information Reconstruction'; 'Learning'; and 'Information Recalling and Reuse'. In later works, diverse cognitive processing models were developed based on the similar objective of Grow's cognitive processing steps, of which a model revises the cognitive processing steps with more pertinent to use technologies based on the concept of information processing called as information processing model, as discussed in Section 2.6, and it has been widely used for technology-based learning tool design (Mario, 2017 ; Nadir, Yavuz, & Mehmet, 2016). This model includes six major cognitive processing steps relating to information processing: 'Attention', 'Perception', 'Repetition', 'Coding', 'Storing' and 'Retrieve' (Halford & Andrews, 2010). They are described as follows:

- Attention: the ability to focus on a certain stimulus, it constitutes the focal point of consciousness to respond to stimuli. This stage determines what information can pass to short-term memory and how great is an individual's capacity for directing their cognitive strengths towards certain resources of information in a learning environment.

- Perception: the process of describing the stimuli received through sensory organs or the process of turning sensory signals into meaningful experiences.
- Repetition: information is stored through repetition to stay in short-term memory longer, while stimulus or stimuli should head towards reaction.
- Coding: most of the information coming from outside is stored temporarily without coding. It is the process of transferring the short-term memory, through information reconstructions, to long-term memory.
- Storing: information is stored in long-term memory. However, during the process of storing, information is stored in the appropriate area of episodic, semantic, and procedural memories.
- Retrieving: looking for finding and activating the information stored in the long-term memory.

In comparison with Grow's cognitive steps, this information processing model was designed to be more closely connected to the use of diverse technologies, and with visualization techniques and concept mapping for helping humans' thinking and mind mastering. This characteristic is more concerned with the strategic design of this cognitive development-based pedagogy to achieve the goals pursued in cognitive processing. In contrast, Grow's cognitive processing steps are developed based on problem-solving, but they did not highlight support with technologies. In this sense, the information processing model is more specific with reference to designing the scaffolding of this pedagogy which is supported by the e-learning object.

However, to identify the cognitive processing steps in this information processing model supported by the e-learning object, it needs to also concern of, identifying the cognitive processing steps to instructional strategy of learning computer programming, and whether the cognitive attainment of using the e-learning object is beneficial to novice learner of computer programming. It stimulates the need of development of a cognitive processing framework for this project that precisely maps the information processing model to instructional strategy of learning computer programming, and to the levels cognitive attainments, so as based on this work to identify the scope of using the e-learning object. This cognitive processing framework is elaborated in Section 2.16, and how it defines the use of the e-learning object is elaborated in Chapter 3, while the searching of a suitable cognitive attainment model used in the cognitive processing framework is pursued in Section 2.8.

2.8 Attainments of Cognitive Processes

An early model of cognitive domain levels known as Bloom's Taxonomy was developed by Benjamin Bloom with collaborators Max Englehart, Edward Furst, Walter Hill and David Krathwohl in early 1956. It was widely used as a model to categorize educational goals by focusing on the cognitive processing outcome corresponding to a series of learning activities. Therefore, it was also described as a Taxonomy of Educational Objectives. Bloom's Taxonomy consists of six domains of attainment in cognitive processing defined as Knowledge, Comprehension, Application, Analysis, Synthesis and Evaluation (Armstrong, 2010). It quickly became a useful descriptor for identifying the achievement of a cognitive tool.

However, upon the emergence of Bloom's Taxonomy, an enhanced model described as the Revised Bloom's Taxonomy (RBT) redefined the original model into six levels with new descriptors: 'Remember', 'Understand', 'Apply', 'Analyse', 'Evaluate' and 'Create' (Kelly, 2017). This revised model identifies knowledge in four domains which are factual (terminology and discrete facts); conceptual (categories, theories, principles, models); procedural (knowledge of technique, process or methodology); and metacognitive (including self-assessment ability and knowledge of various learning skills and techniques) (Bevier, 2010; Iwuchukwu, 2016). The major differences between Bloom's Taxonomy and the Revised Bloom's Taxonomy (RBT) are summarized in Table 2.3, while the discussion of which one is more suitable for use in this project is discussed in what follows.

Table 2.3 Comparison of Bloom's Taxonomy and the RBT

Bloom's Taxonomy	**Revised Bloom's Taxonomy (RBT)**
Knowledge: at this level, learners can recall prior knowledge and recognize the information. They may develop a learning strategy mentally	Remember: retrieve relevant knowledge from long-term memory
Comprehension: at this level, learners can translate, comprehend, and interpret information into new knowledge	Understand: construct meaning from the instructional message, including oral, written graphic communication
Application: at this level, learners can select, transfer, and use the information to undertake tasks with minimal instruction	Apply: carry out or use procedure in a given situation
Analysis: at this level, learners can distinguish and classify information and relate it to assumptions and hypotheses	Analysis: carry out or use a procedure in a given situation
Synthesis: at this level, learners can originate, integrate, and combine ideas into production, planning and proposals that are new to them	Evaluation: at this level, learners can judge and criticize the learnt content
Evaluation: at this level, learners can judge and criticize the learnt content	Create: put elements together to form a coherent whole; recognize in a new pattern or structure

Table 2.3 summarizes the major differences between Bloom's Taxonomy and the Revised Bloom's Taxonomy (RTB). The RTB can be seen as a modern version of the original Bloom's Taxonomy, while it is more specific in evaluating a wider range of applications of different types of modern technologies. For example, at the last level, the RTB is concerned with patterns or structures of production. Those are concerned with the types of technologies, such as virtual reality and augmented reality technologies, used to achieve cognitive attainment.

In contrast, Bloom's Taxonomy is more coherent on the cognitive processing outcome of pedagogy and scaffolding design (Penney, 2014; Tutkun, Guzel, Koroğlu, & Ilhan, 2012). It is a holistic form of using educational aspects to precisely define the objectives and outcomes of cognitive development. For example, it emphasizes the stepwise cognitive processing from initial level 'Knowledge' through advanced levels of 'Comprehension' to the ultimate level of 'Evaluation'. These levels concern teachers' perspective of teaching, and students' attainment of learning. In this sense, Bloom's Taxonomy is a two-dimensional model relating to teachers' understanding of the scaffolding of a standards-based curriculum, as well as students' cognitive presences through the learning process. Therefore, Bloom's Taxonomy is still a superb model for educators to understand the achievements of a series of cognitive processing steps pursued by cognitive tools in a scaffolding (Forehand, 2005; Kelly, 2017).

For this cognitive development-based pedagogy, it needs a model that is able to be used to analyse the complexity of the steps in cognitive processing to know the cognitive outcomes needed for this pedagogy, so as to determine the aspects that need to be achieved by using the e-learning object (Halford & Andrews, 2010). Therefore, Bloom's Taxonomy is more specific to the needs of defining the cognitive processing steps in this pedagogy. The picture of how Bloom's Taxonomy maps with different steps of the cognitive processing steps is illustrated in Table 2.4.

Table 2.4 The mapping of the cognitive processing steps to Bloom's taxonomy cognitive domain levels

Cognitive Processing Steps	**Possible learning activities needed to achieve the cognitive levels**	**Cognitive processing outcome indicated with Bloom's Taxonomy Cognitive Domain Levels**
Attention & Perception	Understand the meaning, translation, interpolation and interpretation of instructions and problems	Knowledge Accretion
Repetition	Able to state a problem in a meaningful way by reconstructing the information with manageable style	
Repetition & Coding	The information is converted to be one's own knowledge	Knowledge Understanding
Coding & Storing	Use the knowledge to solve a domain-based problem and deepen the information to be solid knowledge	Comprehension & Application
Storing & Retrieving	Reuse the knowledge on internal/external context Finding and modifying simple errors. Finding errors Redesign and redevelop	Analysis & Synthesis & Evaluation

Table 2.4 precisely defines the attainments of cognitive processing steps at different levels of cognitive domains defined in Bloom's Taxonomy. It breaks down the level of 'Knowledge' to two finer levels of 'Knowledge Accretion' and 'Knowledge Understanding', as this level needs more precise cognitive processes to achieve. For example, to achieve the domain 'Knowledge Accretion', learners need to go through the cognitive processing steps of 'Attention & Perception' and

‘Perception’. It means that they have to ‘understand the meaning, translation, interpolation, and interpretation of instructions and problems’, and be ‘able to state a problem in a meaningful way by reconstructing the information with their understandable style’. Similarly, to achieve the levels of ‘Analysis & Synthesis & Evaluation’, learners have to support the cognitive processing steps of ‘Storing and Retrieving’. It means they have to be able to ‘reuse the knowledge on internal/external context’ and ‘hold the ability to find and modify simple errors, as well as redesign the works’.

Table 2.4 is specifically used to identify the aspects that need to be supported by the e-learning object in the scaffolding of this cognitive development-based pedagogy. For example, the cognitive processing step ‘Perception’ is closely connected to students’ cognitive skills, therefore the related learning process is one of the major parts suggested for use of the e-learning object. On the other hand, the cognitive processing steps ‘Storing & Retrieving’ are related to ‘Analysis & Synthesis & Evaluation’. In considering the curriculum of introductory programming, this cognitive domain level needs to be achieved by providing students with applications and case studies. This attainment may not suit the purposes of using the e-learning object in this pedagogy. The details of defining the aspect of using the e-learning object in the pedagogy are discussed in Chapter 3, upon completing work on defining a cognitive processing framework for learning computer programming in Section 2.16.

2.9 Pedagogy

Pedagogy is defined by theories and practices related to different learning behaviours and psychological approaches (Shapiro et al., 2017). On the theory side, the pedagogical design is based on three major paradigms: behaviourism, cognitivism and constructivism. On the practical side, it refers to the scaffolding designed in a pedagogy. This will be discussed in Section 2.10.

Behaviourism was specified early in Skinner’s *About Behaviorism* (Skinner, 1974), which stated that learning is a process with positive and negative reinforcements, where learners’ behaviours are determined by responses to the stimuli. Until the mid-1990s, a study from Edward Tolman (1948) delivered a concept that knowledge needs to be built by construction, and that it is impacted by cognitive ability. Jean Piaget, a Swiss psychologist, reinforced this concept by relating cognitive development to cognitivism with four stages known as sensorimotor, preoperational, concrete operational and formal operational stages in a fixed order (Kodat, 2007). The movement from one stage to the next occurred, it is suggested, when learning reaches an appropriate level of maturation (Huitt & HUmmel, 2006). These stages differ not only in the quantity of information acquired but also in the quality of how this information is reconstructed and understood by learners.

Unlike Piaget, Lev Vygotsky stressed that cognitive development had a fundamental role in social interaction and strongly believed that the community plays a central role in 'making meaning to learn'. He argued that learning is a universal aspect of the process for developing culturally organized, specifically human psychological function (Vygotsky, 1978). The key point from Vygotsky is that the concept of cognitive development for learning can happen at diverse levels from guided learning within the zone of proximal development (McLeod, 2014). While Vygotsky suggests cognitive learning can stem from interactions, such as guidance from instructors, Piaget maintains that cognitive development stems largely from independent explorations in which learners construct knowledge of their own. In this regard, Vygotsky's zone of proximal development focuses on the scaffolding of a pedagogy, while Piaget delivered the concept of cognitivism.

Another famous psychologist Jerome Bruner (1979) provided a concept consistent with both Vygotsky's zone of proximal development and Piaget's cognitivism. He said learning needs to be interactive with teachers by identifying teaching as an intellectual trial whereby learners construct new ideas or concepts based upon their works on the cognitive structure, e.g. scheme and mental models, providing meaning to allow individuals to go beyond the given information. This early view from Bruner declared that educational content needs to be retransmitted to enhance individuals' cognitive capacity (Takaya, 2008).

Theories from Vygotsky, Piaget and Bruner provide meaning to constructive learning design for cognitive development, where there needs to be a wider perspective of crucial attributes in the scaffolding, in which knowledge is acquired constructively from learners' manageable information by reconstructions (H. Brown, 2012; Kaya & Akdemir, 2016; Thramboulidis, 2003). These, in turn, can promote learners' positive attitudes, confidence, motivation and mental engagement (CUREE, 2012, pp. 4–7). These features are concerned with the design of the pedagogy of this project and they are explored and discussed in Section 2.15.

2.10 Scaffolding

Scaffolding was described by Wood, Burner and Ross (1976) in their early works as the activities between learners and tutors when assisting the learner in accomplishing a difficult task with independence. This term has become widely used in educational aspects describing the wide variety of practices of pedagogy (Israel & Duffy, 2011, pp. 472–473). Influenced by Bruner's and Vygotsky's later works, scaffolding connects to how teachers temporarily aid students in completing tasks and developing new understandings, enabling them to work on similar tasks alone (Israel & Duffy, 2011, p. 29). Bruner also emphasizes that learners must recognize the goals before they can produce their scaffolding, while Vygotsky offers key principles on scaffolding by describing learning as taking place through interaction with appropriate support by others.

Scaffolding defines the practice of pedagogy. In relating to instructional design, it needs to allow students, organizing their learning process independently and also supported by instructor interactions, to effectively absorb new concepts and skills in a learning context. Based on these aims, instructional scaffolding is at variance with its functional features (Alber, 2014). There are three scaffolding functions concerned in the pedagogy of this project: conceptual scaffolding, strategic scaffolding and metacognitive scaffolding (Alber, 2014; Belland, Kim, & Hannafin, 2013). They can be summed up as follows:

- Conceptual scaffolding guides students to find an effective way of solving a learning problem. To make use of technology, it needs to make sense of the received information by reconstructing the information to be students' manageable forms, helping them to map the received concept to knowledge context (Belland et al., 2013). Conceptual scaffolding needs a more structured and organized interaction with the learners and instructors to lead students to pursue a greater achievement (Burns & Joyce, 2005; Sun-Ongerth, 2012).

- Strategic scaffolding concerns the development of a strategy to achieve the objective defined in the conceptual scaffolding (Alber, 2014). The concept of strategic scaffolding is left open for use in different learning scenarios. For example, some strategic scaffoldings may be supported by learning tools, while others are supported through instructional methods, or both of them. The fundamental of strategic scaffolding is based on defining an effective strategy for a pedagogy to achieve the learning objectives (Belland et al., 2013; Winnips, 2001).
- Metacognitive scaffolding helps students to evaluate their thinking through monitoring, regulating and reflection (Quintana, Zhang, & Krajcik, 2010). The major concept of metacognitive scaffolding is to perform backwards the conceptual scaffolding and strategic scaffolding to significantly worse learning-outcome students. Therefore, this scaffolding is scheduled and defined by focusing on students' outcomes delivered by the conceptual and strategic scaffoldings (Zhang & Quintana, 2012).

The advantages of using technological scaffoldings have been indicated in diverse studies. For example, a meta-analysis focused on 144 experimental studies found that students who learn with technology-based scaffoldings perform better than those who do not receive technology-based scaffoldings on a variety of assessment types (Belland, Walker, Kim, & Lefler, 2017). A literature review from Gayton (2008) also found that diverse technology-based scaffoldings functionally can help students to learn while they use them with appropriate instructions. Besides, this review also reveals that, with the diversified functions in technology-based scaffoldings, low-achieving students can independently recognize when they need to learn by utilizing an available metacognitive scaffolding. Therefore, as a recommendation, Gayton (2008) advised teachers need to directly teach students, and prepare students with alternative scaffoldings based on the different needs of students in a pedagogy.

In this project, the three functions of scaffolding above-mentioned form the major approach for the scaffolding design of the cognitive development-based pedagogy. This is discussed in Section 2.15.

2.11 E-learning and Learning Object

Learning technology is developed to help learners master the abstractions of principles and skills (Settle, Vihavainen, & Miller, 2014), and grasp the complexity of the learning context by diversified presentations (Dede, 2014; Jonassen et al., 2010). Its developments can result from the early approach of using courseware, online materials (Pang, 2003a), such as Taylor R Three-T model (Boulos, Maramba, & Wheeler, 2006; Shank, 2005), up to modern technologies such as multimedia, mobile technology, virtual reality (VR) and augmented reality (AR) (Lynch, 2016), to focus on cognitive learning (Keengwe et al., 2008). There are two major streams of using technology for educational purposes known as the e-learning and learning objects (Pang, 2007) focused on in this project.

E-learning

E-learning was developed with the emergence of high-speed Internet technologies. In an early definition, it was described as ‘a networked learning platform, which makes instructions and information capable of instant updating, storage/retrieval, distribution, sharing within a learning community’ (Rosenberg, 2001, p. 28). By using advanced technology such as cloud computing and mobile technologies, e-learning diversifies to apply to different learning strategies, such as online learning, blended learning and deep learning (Brown, Krasteva, & Ranieri, 2016; Epignosis LLC, 2014, pp. 61–69; Gannon-Leary & Fontainha, 2007; Pang, 2003b; Tastle, White, & Shackleton, 2005). Conceptually, the development of e-learning has gone through four generations.

- The first-generation was the pioneer used to support distance learning with limited face-to-face mode with two major goals (Rosenberg, 2001, pp. 13–15; Tastle et al., 2005). They were first used with text-based information with moderate graphics and audios to provide cost-effective training platforms to learners (Newton & Ellis, 2006; Rosenberg, 2001, pp. 13–15), and secondly to maximize the access time and release of location from delivering the contents online (Varlamis & Apostolakis, 2006).

- The second-generation added a lot of course and learning management features on top of the first-generation platforms (Epignosis LLC, 2014, p. 53; Klett, 2007; Pang & Au, 2002; Rosenberg, 2001), and made use of more affordable multimedia and hypermedia to support teaching and learning (Boyle, 2006; Conole, 2017; Mullier, Hobbs, & Moore, 2002).

- The third-generation emerged with the availability of high-speed optical-based intranet to schools. With these technologies, e-learning becomes more efficient in delivering multimedia-based learning content (Epignosis LLC, 2014, pp. 29–39; Liu & Cheng, 2008; Pang, 2007). This generation of e-learning makes it possible for constructive learning and cognitive learning to be designed and implemented online (Klett, 2007). With these developments, there are many terminologies used to define the e-learning systems. They include Knowledge Management Systems (KMS) (Bouzeghoub, Defude, Duitama, & Lecocq, 2006), Learning Content Management Systems (LCMS) and Learning Management Systems (LMS) (Rosenberg, 2001, pp. 65 & 161; Varlamis & Apostolakis, 2006).

- The fourth-generation was developed and accomplished with the concepts of the ‘intelligent agent’, ‘learning agent’, ‘mobile-learning’ and ‘cloud-based learning’ (Neji & Ammar, 2007; Pappas, 2017). These systems provide learners with fruitful features on using reusable and flexible learning resources (Jonassen et al., 2010; Jonassen & Churchill, 2004) to cater for diversified learning situations (Contreras, Galindo, & Caballero, 2006).

Learning Object

There are many definitions of a learning object. In the beginning, the term ‘learning object’ was defined as a ‘technology solution’ by the vendor Netg Inc. and was called ‘NETg learning objects’. With the others, learning objects were also referred to as a ‘learning objective’ in some off-the-shelf courseware such as the ‘ToolBook II learning objects’. Until Merrill, Li and Jones used the terms ‘knowledge objects’ and ‘learning objects’ to refer to a component of instructions, the concept of learning objects became closer to e-learning developments. The definition of learning object in the scope of e-learning was also reinforced by Merlot (2000) and Escot (2000), who described online learning materials as learning objects.

In 2006, the Institute of Electrical and Electronics Engineers (IEEE) quality assurance and standards body defined Learning Objects as ‘any entity, digital or non-digital, which can be used, re-used or referenced during technology-supported learning’ (IMS, 2006). This definition was followed by a large number of studies and evaluations of learning objects (Cakiroglu, Baki, & Akkan, 2012; Churchill, 2007, 2012). This definition is followed in this project.

Despite the subject matter, granularity is the first major factor for consideration in learning object design. This is the process of systematically disaggregating the learning content into instructionally grounded units of learning information and packaging them to be standard non-divisible learning objects (Dahn, 2006; Williams, 2004). The granular needs to be rigorously broken down to a finer level (Churchill, 2007; Jonassen et al., 2010). It could be a piece of reusable textual information, graphics, animation, concepts, knowledge, experience, principle, process and procedure (Collis & Strijker, 2003; Mogharreban & Guggenheim, 2008; Pappas, 2016). The identity of granular for a learning object is linked to two major design concepts: decontextualization and contextualization.

Decontextualization is the process of extracting specific learning context from a subject area to a new learning domain (Wiley et al., 2004). This can exist internally or externally. Internal context is a primitive asset for a specific learning topic of the subject area in a course. This content directly relates to the context itself. On the other hand, external context normally is not only visible internally yet can be reused externally (Ljuboevic, Cook, & Boyle, 2004; Pappas, 2016).

In this project, the e-learning object is designed by decontextualizing the topics of program control from the curriculum of introductory programming while contextualizing this context as a new learning domain that focuses on delivery of students’ concept of the semantics of program controls (Apiola & Tedre, 2012). Therefore, the context of all learning objects integrated into the e-learning object are self-contained spatial or temporal elements for storing learning information (Dahn, 2006). It can exist externally in other programming languages as its focus on the semantics of program control does not limit it to a particular programming language.

2.12 Pedagogy for E-learning

Pedagogy for e-learning encompasses similar elements as generic designed pedagogy by introducing more specific scaffolding features of using an Internet-based learning environment (Branun, 2008). For this project, there are three attributes concerned in using the e-learning object: teaching presence, social presence and cognitive presence (Akyol & Garrison, 2011), and these are explained as follows.

Teaching Presence

Teaching presence takes advantage of behaviourism, stimulated by B. F. Skinner and John B. Watson, focusing on learners' behaviours in the design of online pedagogy (Watson, 2013). It focuses on the need to provide immediate feedback for individual learners, and also recommends that a pedagogy needs to be scaffolded by combining online activities and classroom interactions to increase students' learning experience (Clifford, 2016). This cognitive development-based pedagogy applies the function of conceptual scaffolding to maintain the teacher presence in the pedagogical environment. This is further discussed in Section 2.15.

Social Presence

Social presence relates to Vygotsky's Sociocultural Theory, the zone of proximal development (ZPD) that focuses on structure activities to wider student contexts by learning with the communities including the students and instructor (Psychology Notes HQ, 2018). This attribute is scaffolded by focusing on the differences between what learners can do, with assistance from teachers, and how they move to complete a task on their own (Shabani et al., 2010). To bridge this gap, the scaffolding needs to emphasize communication, collaboration and understanding of the factors underpinning the communities (Akyol & Garrison, 2011), and potentially supports diverse types of online-based communications (H. Brown, 2012). With regard to the pedagogy of this project, social presence concerns the application of the functions of strategic scaffolding. This detail is discussed in Section 2.15.

Cognitive Presence

Cognitive presence on pedagogical design for e-learning inherits diverse properties from cognitivism with implementation in an online-based environment, such as it needs to help cognitive development by designated cognitive processes strategically scaffolded through exploration, integration, and resolution (Garrison & Arbaugh, 2007). Studies of cognitive presence on e-learning often emerge from inquiry into mental engagements with the pedagogical environment and the social community (Tsai, Shen, & Fan, 2013). Vygotsky's theory on constructivism also summarizes social interactions, e.g. zone of proximal development (ZPD), as critical for knowledge co-construction between teachers and schoolmates (Verenikina, 2010). For this cognitive development-based pedagogy, cognitive presence concerns the use of the e-learning object to improve students' mental engagement with the pedagogical environment and their cognitive processes in the scaffolding. This detail is discussed in Section 2.15.

2.13 Evaluation of Technology Integration

There are diverse frameworks and models that can be used to evaluate the efficiency and performance of using technology for educational purposes. This section provides a review of these models and frameworks to search for one which is suitable for evaluating whether the design of this cognitive development-based pedagogy can match teachers' capacity for using technology for their teaching process. These models include the TPACK, SAMR, the Triple-E, TIM, ADDIE, and RAT.

TPACK (Technological Pedagogical Content Knowledge)

This is a theoretical framework used to evaluate teachers' knowledge using technology by the dynamics and transactional relationship between the combination of content, technology and pedagogy (Jamieson-Proctor et al., 2013; Koehler & Mishra, 2009; Pahlevi, 2017). These relationships are identified as primary forms and domain knowledge. The primary forms include the 'Technology Knowledge (TK)', 'Pedagogical Knowledge (PK)' and 'Content Knowledge (CK)', while domain knowledge consists of primary forms defined as 'Pedagogical Content

Knowledge (PCK)’, ‘Technological Content Knowledge (TCK)’ and ‘Technological Pedagogical Knowledge (PTK)’. They are described as follows:

- Technological Knowledge (TK): it concerns teachers’ knowledge of ‘thinking about’ and ‘working with’ the technologies that directly engages with the use of technological resources, and the adaptability of new technologies (Pahlevi, 2017). This primary form is fundamental to teachers’ understanding of using technology to achieve learning goals.
- Pedagogical Knowledge (PK): it concerns teachers’ knowledge of the methods and practices for teaching and learning. It relates to teachers’ understanding of applying knowledge to students, and the ability to establish a learning environment with essential components such as lesson plans and assessments. This primary form encompasses the overall educational purpose, values and aims of a pedagogy (Koehler & Mishra, 2009).
- Content Knowledge (CK): it concerns teachers’ knowledge of the subject matter, identified as learning contexts which it crucially needs to address through the learning process.
- Technological Pedagogical Knowledge (TPK): it is the teachers’ understanding of technologies specific to their pedagogy. Teachers are required to process knowledge of pedagogical affordance and constraints in the range of usable technologies. It is closely linked to teachers’ knowledge of disciplines, specialism and pedagogical approaches (Jamieson-Proctor et al., 2013).
- Technological Content Knowledge (TCK): it is the interaction of technology and the learning content (Jamieson-Proctor et al., 2013; Sahin & Kelesoglu, 2011). It relates to the understanding of using technology and how the contents influence and constrain each other. Under the rationale of this domain, teachers are required to master more than the subject matter and to be capable of developing a deep understanding of using technology. It critically relates to identifying the types of technology that support presenting and delivering the subject effectively.
- Pedagogical Content Knowledge (PCK): it is the notion of transforming the subject matter through teaching. This transformation occurs when teachers interpret the subject matter and tailor the instructional materials to alternative

conceptions of students' prior knowledge. This domain specifically engages the core of teaching, learning, curriculum, assessment and reporting (Koehler & Mishra, 2009; Scherer, Tondeurb, & Siddiqc, 2017).

- TPACK: with the interplay between the primary forms and domain knowledge, the TPACK framework provides a multidimensional approach to evaluating teachers' technological and pedagogical capacity, carrying out meaningful and skilled teaching. It addresses the crucial ways in which technology helps to redress the problems that students may encounter, with knowledge created from prior knowledge, and how technologies can be used from existing knowledge to develop a new, strengthened learning context. Therefore, this framework directly relates to a teacher's knowledge of technology integration and their capacity to apply a technology-based pedagogy to cope with diversified teaching needs.

SAMR (Substitution Augmentation Modification Redefinition)

The SAMR model offers a method for evaluating how computer technology impacts on teaching and learning. It enables teachers to design, develop and infuse digital learning experiences that utilize technology (Romrell, Kidder, & Wood, 2014). The goal of SAMR is to transform learning experiences to push students' achievement to a higher cognitive level. The SAMR model consists of four classifications of technology use for learning activities (Puentedura, 2013), illustrated in Table 2.5.

Table 2.5 Use of technology for learning activities in the SAMR model

Level	Definition	Example	Functional change
Substitution	Computer technology is used to perform the same task as before the use of computers.	Students print out worksheet, finish it, hand it in.	No functional change in teaching and learning. When an appropriate level of work can be conducted without a computer, there is a real gain to be had from computer technology
Augmentation	Computer technology offers an effective tool to perform common tasks.	Students take a quiz using a Google Form instead of using pencil and paper.	There is some functional benefit here in that paper is being saved, students and teacher can receive almost immediate feedback on students' level of understanding of the material.

Modification	This is the first step over the line between enhancing the traditional classroom activities and transforming them. Common tasks are accomplished using computer technology.	Students are asked to write an essay; an audio recording of the essay is made along with an original musical soundtrack, where the recording will be played in front of an authentic audience such as parents, or college admission counsellors.	There is significant functional change in the classroom. While all students are learning similar writing skills, the reality of an authentic audience gives each student a personal stake in the quality of the work. Computer technology is necessary for this classroom to function, allowing peer and teacher feedback, easy rewriting and audio recording. Questions about writing skills increasingly come from students themselves.
Redefinition	Computer technology allows for new tasks that were previously inconceivable.	A classroom is asked to create a documentary video answering an essential question related to important concepts. Teams are expected to contact outside sources for information.	At this level, common classroom tasks and computer technology exist not as ends but as supports for student-centred learning. Students learn content and skills in support of important concepts as they pursue the challenge of creating professional quality. Collaboration becomes necessary and technology allows such communications to occur.

As a characteristic of the SAMR model, the first two levels ‘Substitution’ and ‘Augmentation’ represent enhancements of existing ways of working. While digital technology is not necessary in these stages, technology may provide a digital medium in which learning takes place. Subsequent levels, ‘Modification’ and ‘Redefinition’, represent the transformational stages where technology is actively used to transform students’ learning within a technological environment (Fastiggi, 2014).

TIM (Technology Integration Matrix)

The TIM framework was developed by the Florida Center for Instructional Technology (FCIT) (TIM Development Team University of South Florida, 2015) to provide a framework of descriptors for using technology to enhance learning. It incorporates five interdependent characteristics of meaningful learning environments: ‘Active’; ‘Collaborative’; ‘Constructive’; ‘Authentic’; and ‘Goal-Directed’, while each is associated with five levels of technology integration: ‘Entry’; ‘Adoption’; ‘Adaptation’; ‘Infusion’; and ‘Transformation’. By integrating the interdependent characteristics, TIM creates a matrix of 25 cells to guide the evaluation of using technology. This structure means TIM is often seen as a rubric to guide teachers to determine if the diffusion of technology is at a suitable level of transformation. Therefore, TIM is like a reference model to assist schools using technology in the classroom, instead of an evaluation model to measure the achievement of using technology (Ruman & Prakasha, 2017). The descriptors of TIM are illustrated in Table 2.6.

Table 2.6 Descriptors of TIM Model

	Descriptor
ACTIVE	
Entry	Teachers may be the only persons actively using presentation software to support delivery of a lecture for drill and practice activities.
Adoption	Teachers control the type of technology and how it is used. Make sure students can complete each step in the same sequence.
Adaptation	Teachers choose which technology tools and how to use them. They do not need to guide students, instead act as facilitators.
Infusion	Teachers guide, inform and contextualize student choices of technology with flexible and open ideas.
Transformation	Teachers serve as a guide, mentor and model in use of technology. They encourage students' engagement with technology resources and facilitate students' engagement in higher-order learning.
COLLABORATIVE	
Entry	Teachers direct students to work alone on tasks involving technology.
Adoption	Teachers direct students in conventional use of technology tools for working with others.
Adaptation	Teachers provide opportunities for students to use technology to work with others and encourage students to begin exploring the use of these tools.
Infusion	Teachers encourage students to use technology tools collaboratively.
Transformation	Teachers seek partnerships outside the setting to allow students to access experts and peers.
CONSTRUCTIVE	
Entry	Teachers use technology to deliver students' information.
Adoption	Teachers provide students with opportunities to use technology in conventional ways.
Adaptation	Teachers can design lessons in which students' use of technology is integral to building an understanding of a concept.
Infusion	Teachers consistently allow students to select technology tools and provide a context in which technology is seamlessly integrated into lessons.
Transformation	Teachers facilitate higher-order learning in which students regularly engage in activities that may be impossible to achieve without the use of technology.

AUTHENTIC	
Entry	Teachers assign work based on a predetermined curriculum beyond the instructional setting.
Adoption	Teachers direct students in the conventional use of technology for learning activities.
Adaptation	Teachers create instruction that integrates technology and provides access to information in community.
Infusion	Teachers encourage students to use technology to make connections outside the instructional setting.
Transformation	Teachers encourage innovative use of technology.
GOAL-DIRECTED	
Entry	Teachers use technology to give students directions and monitor step-by-step completion of tasks.
Adoption	Teachers can direct students step by step in the conventional use of technology tools to either plan, monitor, or evaluate an activity.
Adaptation	Teachers select technology and clearly integrate it into the lesson and facilitate students' independent use of technology tools.
Infusion	Teachers create learning context in which students regularly use technology for planning, monitoring, and evaluating learning activities.
Transformation	Teachers create a rich learning environment in which students regularly engage in higher-order planning activities.

The Triple-E Framework

The Triple-E is a practical framework to measure whether the degree of using technology meets the learning goal (Triple-E framework, 2013). It draws on educational research on effective and ineffective practices with technology tools, specially developed by K-12 teachers and administrators, for use in lesson plan development and the potential effectiveness of educational apps in learning. The framework consists of three components: 'Engagement'; 'Enhancement'; and 'Extension', interchangeably providing a benchmark for considering how technology tools are used. The Triple-E framework has been the focus of considerable research, which mostly emphasizes the following:

- At the core of any technology-enhanced lesson should be the learning goals (Linnenbrink & Pintrich, 2003).
- Meaningful use of technology in the classroom requires teachers to integrate technological affordances with pedagogical approaches for the specific subject matter to be taught (Mishra & Koehler, 2006).

- The importance of time-on-task active engagement (Kay & Lauricella, 2011; Wartella, 2015).
- The quality of using technology, rather than quantity (OECD, 2015).
- The type of use – avoiding ‘drill and practice’, which can have negative effects on learning outcomes, and integrating more real-world problem-solving and creation (Vaala, Ly, & Levine, 2015).
- Helping students to connect existing knowledge to new knowledge (Wartella, 2015).
- Co-use and/or joint-media engagement of technology devices and software (Darling-Hammond, Zielezinksi, & Goldman, 2014; The American Academy of Pediatrics, 2013; Zach, 2016).
- Significance of a ‘human’ as part of co-use (The American Academy of Pediatrics, 2013; Zach, 2016)
- Value-added strategies such as promoting student self-reflection, self-assessment and self-explanation (Means, Toyama, Murphy, Bakia, & Jones, 2009).
- Social aspects of learning through technology tools (Guernsey, 2012; Vaala et al., 2015).

ADDIE (Analyse, Design, Develop, Implement and Evaluate)

The ADDIE is an instructional design method used for educational and training programmes. It defines a series of steps to provide a standard strategy to better prepare and create materials (Shoemaker, 2009). The steps are ‘Analyse’, ‘Design’, ‘Develop’, ‘Implement’ and ‘Evaluate’ (McGriff, 2005). ADDIE is seen to be a strategic approach for implementing instructional design, and training technology-based tools. Therefore, this model has been linked to various instructional design models (Kurt, 2015). The descriptions of each step are illustrated as follows:

- Analysis: This step focuses on identifying learning problems, objectives and goals, the audience’s needs, prior knowledge, relevant characteristics, learning environment and the constraints for model developments (Branch, 2010, p. 24).

- Design: a systematic process of specifying learning objectives. In this step, detailed storyboards and prototypes present the look and feel, graphic user interface and some essential contents (Branch, 2010, p. 59).
- Develop: builds the model based on the analysis and design, generating the lesson plans and lesson material with media instructions, and supportive documents. This may include hardware (e.g. simulation equipment) and software (e.g. computer-based instruction) (Branch, 2010, p. 84).
- Implement: the actual delivery of the instruction using possible means including classroom-based, lab-based or computer-based. In this step, the major concerns are the effectiveness and efficiency of delivering the instructions so that students' understanding of the material and mastery of the learning objectives can be ensured (Branch, 2010, pp. 133–134).
- Evaluate: focuses on the effectiveness and efficiency of the instructions throughout the entire instructional design process. It could be conducted within steps and between them with formative and summative approaches (Branch, 2010, pp. 151–152).

RAT Model

RAT can be used by teachers to develop lesson plans for use on iPads, laptops and other forms of technology for evaluating their usefulness. It was developed with the standard defined in the International Society for Technology in Education (ISTE) (Kelly, 2016). This model majors on ensuring that ICT is regularly embedded into the classroom environment. Therefore, RAT and ISTE can assist teachers to evaluate their performance of using ICT in the classroom (Hughes, Thomas, & Scharber, 2006). The RAT model has three aspects and can be described as follows:

- Replacement: technology takes the place of traditional methods of teaching material.
- Amplification: technology increases efficiency and productivity without fundamental changes to the lesson.
- Transformation: technology completely transforms the lesson into instruction and learning.

Discussion of Comparing Different Models

While TPACK is at the heart of integration, it conceptually provides a practical tool to help teachers to envision what technology integration may look like in practice (Mishra & Koehler, 2006). Like other frameworks and models, such as SAMR, Triple-E and RAT, TPACK can be used to strategically define approaches and evaluations. However, it is different from them by focusing on individual teachers' knowledge, establishing by primary forms and domain knowledge, on the adoption of technology for their pedagogy for a subject matter (Learn NC article, 2015). For example, technology (TK) could involve diversification of hardware; pedagogy (TK) could relate to teachers' different styles and approaches to teaching; and content (CK) could provide a direct link to the subject matter. And TPACK further combines these primary forms of knowledge to define a specific domain of knowledge. For example, PCK concerns teachers' knowledge of teaching computer programming with a suitable pedagogical strategy to fulfil a common need for a pedagogy (PK) and subject content (CK). This systematic style makes TPACK highly focused on teachers' operationally oriented knowledge, from basic, and possibly goes further to a specific designed technological pedagogy.

On the other hand, the SAMR framework aims to establish a technological system to improve learners' performance. It adopts the concept of 'something could be done without technology' and focuses on diffusing educational technology to different levels, such as from entry to functional change. This framework indicates technical substitution but allows for slight improvement. However, this is only up to the level of modification. In this sense, SAMR is an essential planning tool helping the development of a technology integration strategy and offering guidance on process and reflection. Thus, the SAMR model is likely to be a tool measuring enhancement and transformation of using technology (Jude, Kajura, & Birevu, 2014). It makes SAMR suitable for use as a reference to evaluate the impact of the use of technology and is particularly suitable for pioneer projects in the use of technology, rather than TPACK, which focuses on operational and pedagogical developments.

The TIM framework is a valuable resource for guiding immense technologies at different achievement levels, identified as a matrix including 'Active', 'Collaborative', 'Constructive', 'Authentic' and 'Goal-Directed', where each is comprised with ‘Entry’, ‘Adoption’, ‘Adaptation’, ‘Infusion’ and ‘Transformation’. TIM provides details on how technology can be used with an active learning environment, and on forms of assessing students’ use of technology to undertake exercises. Unlike TPACK, the TIM framework does not provide an operational strategy to implement technology-based pedagogy, unlike SAMR, which provides a full-reference model to measure the achievement of technology integration in a subject-specific area together with the curriculum design.

The Triple-E framework is designed as a practical tool to bring together an instructional strategy of ‘Engaging’, ‘Enhancing’ and ‘Extending’ the learning goals, with a selection of cognitive tools in the current setting. Unlike other frameworks and models, Triple-E focuses on replacing traditional tools or seeking support from creative technologies, and the efforts are fully directed towards the learning tool rather than the built-in learning strategy (Triple-E framework, 2013). This is why Triple-E is often seen as a benchmark to find out whether students can meet their learning goals. Theoretically, Triple-E is similar to TPACK, both are pedagogy-practice oriented. However, unlike TPACK, this model highlights successful instructional design (Okojie, Olinzock, & Okojie-Boulder, 2006), not teachers’ capacity concerning the primary forms and domain knowledge. Therefore it does not emphasize the capability of using technological pedagogy.

ADDIE is a model developed from the ISD family (Instructional System Design) (Clark, 2015b), providing a systematic approach to instructional design. For example, this model consists of 90 steps grouped into five phases (analyse, design, develop, implement and evaluate) for the development of education and training programmes (Clark, 2015a). These processes are not specific to the pedagogical approach, but are subject-matter oriented. Therefore, this is likely to be a tool for evaluating on-the-job trainers’ performance and their adequacy in performing certain jobs or tasks with a technology-based platform (Soto, 2013).

RAT is like a simpler and practical version of the SAMR model, but it puts together the second and third levels of ‘augmentation’ and ‘modification’ in SAMR to be an amplification stage to avoid the ambiguity of augmentation and modification (Annaweisspol, 2016). It also adds a level of ‘replacement’ making it more effective if the school is already using technology to transform learning. In comparison to TPACK, RAT suffers from similar deficiencies to SAMR: for example, it does not focus on pedagogical issues in adopting technology.

2.14 Limitations of Technology Tool for Computer Programming

Previous developed models and tools for learning computer programming seem to be narrow in terms of specific programming languages, such as the Java, C/C++ and HTML: for example, SAMBA animation program (Duskis, Dey, Hartley, & Wagner, 2007), ViewJ model (Shanmugasundaram, Juell, Hill, & Nygard, 2007) and CADAL Quiz learning system (Sheard & Carbone, 2005). Specifically, for introductory programming, most supportive technological tools are likely to be developed based on providing a series of training and practically oriented exercises rather than facilitating learning and teaching with existing curricula (Järvinen, Tiusanen, & Virtanen, 2005; Woo, 2007). For applying technology for cognitive development, although some models claim to be cognitive tools and to provide a complete environment to improve learners' programming skills (Kunyanuth, Pubet, & Pattarapan, 2014; Pierson & Rodger, 2008; Vila, Beccue, & Doss, 2006), most of them are still designed for drill and practice, while the concept of cognitive development is rarely pursued (Ala-Mutka, 2005; Belland et al., 2008; Sanders & Thomas, 2007).

For the web-based model, although most were designed with cognitive artefacts such as simulation and animation techniques, they tend to be specific to program languages rather than focusing on the program semantics, as well as applying syntax-free approaches (T. Chan, 2005; Järvinen et al., 2005; Pierson & Rodger, 2008; Powers et al., 2006; Shanmugasundaram et al., 2007). Moreover, some of these models were only designed to focus on improving learners' skill at using the Integrated Development Environment (IDE) software to speed up their programming processes. They could not be identified as a learning tool to support pedagogical approaches (Schulte & Bennedsen, 2006).

Many cognitive tools were designed to enhance logical thinking and problem-solving. However, they tend to link scenarios that learners map to the context of programming by using object concepts, and poorly support students to explore their procedural thinking to understand the logics of program semantics (Graviss, 2005;

Powers et al., 2006; Shanmugasundaram, Juell, Jayasuriya, & Benson, 2004). Along with a few specific conceptual issues, problem-solving skills and the application of adaptive instruction approaches, these previously designed models seem to be more specific to providing an online environment for programming instead of being a learning tool. For example, the Logo and Stan programming tools actually provide a programming environment rather than learning how to program (Gilbert, Wilson, & Gupta, 2005; Kennedy & Shaughnessy, 2008 ; Mayer & Fay, 2013).

To conclude, little in the reviewed previous technology-based learning tools for computer programming is likely to focus on programming skills, memorizing the use of programming tools and using program languages to learn program semantics rather than understanding their use for program design. They do not specifically support novice learners to understand program design. Therefore, this cognitive development-based pedagogy is proposed to make up for these deficiencies.

2.15 The Pedagogy Design in this Project

This cognitive development-based pedagogy is scaffolded with the theories related to Vygotsky's Zone of Proximal Development (Shabani et al., 2010) and Piaget's Theory of Cognitive Development (PsychoHawks, 2010), and the subsequent works on pedagogical and instructional design theories discussed in Sections 2.9, 2.11 and 2.12. At the pedagogical level, there are three learning processes in this pedagogy defined by three major scaffolding functions: conceptual scaffolding, strategic scaffolding, and metacognitive scaffolding (Burns & Joyce, 2005; Sun-Ongerth, 2012). These learning processes include: (i) facilitating cognitive development on learning program semantics; (ii) helping students conceptually map their program semantic knowledge to the coding process; and (iii) providing exercises to enhance students' learnt knowledge. They are robustly supported by using the e-learning object to learn, as illustrated in Figure 2.1.

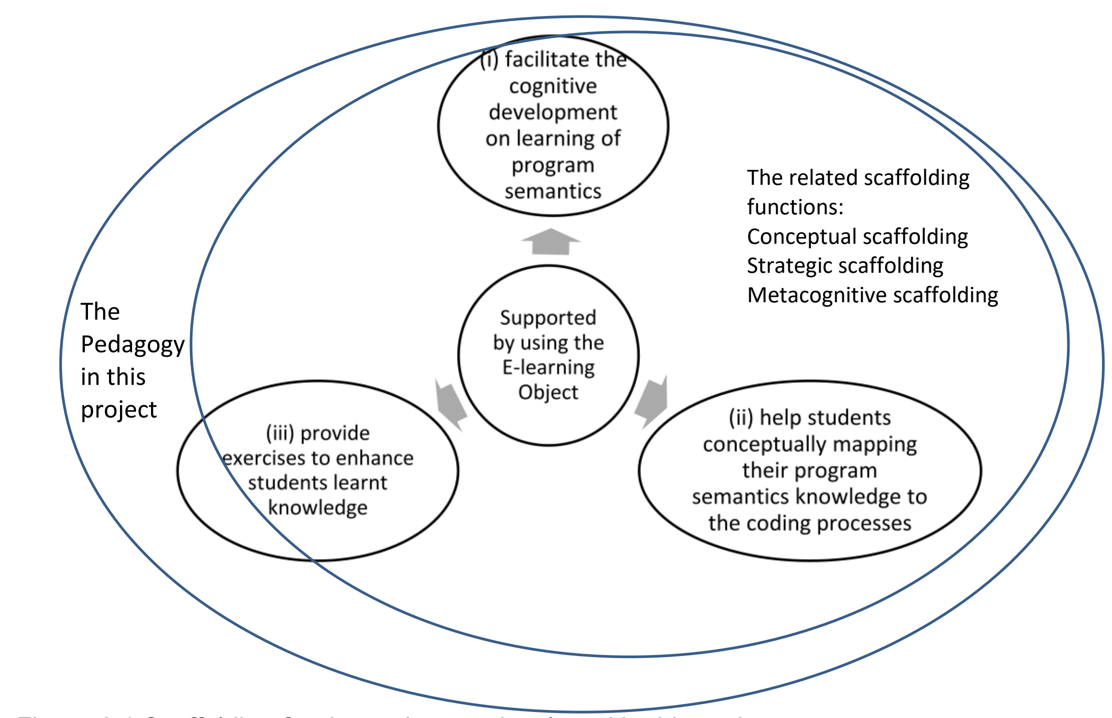


Figure 2.1 Scaffolding for the pedagogy developed in this project

As indicated in Figure 2.1, all the learning processes are supported by the e-learning object. The learning processes of (i) and (ii) were defined with conceptual scaffolding and strategic scaffolding. As discussed in Section 2.10, conceptual scaffolding focuses on making sense of the received information by reconstructing the information to be students' manageable forms, and helping them to map the received concept to knowledge context. Concerning their application to this pedagogy, the learning processes use the e-learning object to demonstrate the semantics of diverse program controls to students with an animated flowchart designed in the learning objects, and work accomplished with the animated code-based example helps students conceptually map the semantics of the program control presented by running the animated flowchart and the animation of code executions synchronously in front of the student. The detail of this process is discussed in Chapter 4.

For strategic scaffolding, this cognitive development-based pedagogy applied the e-learning object to improve a learning environment, and through these cognitive processes intends to enhance the pedagogical environment to be more structured, organized, and interactive styles between students and instructors. It also provides the means to these learning processes that can be guided by instructors in some aspect while students need for immediate feedback. Besides, by using the e-learning object, students can also further their studies on those learning contexts which cannot be discussed in detail during in class interactions (McLeod, 2014).

The improvement of the pedagogical environment also regards the use of the e-learning object to support two instructional methods that are suitable for focusing on learning program semantics and logical reasoning. They are the syntax-free approach and the use of program walk-through techniques to facilitate concept mapping. As discussed in Section 2.4, the syntax-free approach can be used to encourage students by primarily focusing on the concept of semantics in a program structure, while program walk-through for facilitating concept mapping is a self-exploration learning method that provides a strategy for students to independently construct their understanding of program design, and mapping to the coding process. As these instructional methods are independent of programming languages (Fidge & Teague, 2009), it means that the e-learning object can be used to support the cognitive processes in different programming languages. The detail of how these instructional methods are supported by the e-learning object is discussed in Section 3.6.

The learning process (iii) shown in Figure 2.1 is used to extend the learning process to practice and exercise after students acquire enough knowledge. This learning process requires students to work independently in evaluating their understanding on some prescheduled exercises, through self-regulating and reflections (Quintana et al., 2010). By using the e-learning object, this learning process provides a way for students to review and retry the online exercises as they need. This is a major learning process that aligns to the function in metacognitive scaffolding for determining whether students' cognitive presence on the program semantics is improved by using this cognitive development-based pedagogy (Zhang & Quintana, 2012).

The Use of E-learning Object in this Cognitive Development-based Pedagogy

The e-learning object designed for use in this project is an online learning tool applying the concept of strategic e-learning (Meng-Jung, 2009), as discussed in

Section 2.11. Concerning pedagogical issues, the design and use of the e-learning object follow the pedagogical features defined in e-learning pedagogy, as discussed in Section 2.12, including teaching presence, social presence and cognitive presence (Akyol & Garrison, 2011). This is explained below.

With regard to teaching presence, the e-learning object was used together with the course materials to improve teacher's instructional approaches. They can use the e-learning object to demonstrate the semantics of a program control through the animated flowchart and the code-based example to provide guidance on the learning processes and support the in-class activities and exercises.

Social presence promoted the scaffolding focuses on wider students' contexts by learning with the communities including the students and instructor (Psychology Notes HQ, 2018). For this cognitive development-based pedagogy, students have two major channels to interact with instructors and other students. First, as dicussed previously, this pedagogy still maintains a certain level of teacher presence, therefore students can work under suitable guidance and support. Secondly, the e-learning object is installed in the college's e-learning platform, the Blackboard System, on which they can post any questions, discuss with other students and review the learning contents at any time through the e-learning object.

Cognitive presence is a major feature concerning the design of this cognitive development-based pedagogy. It needs to be scaffolded with building up students' cognitive presences on the learning contexts. It applied an e-learning object designed by using an animated flowchart to visualize the semantics of a program control and a concept map to link the semantics to the coding processes. As discussed in Section 2.6, these two major information processing techniques provide advantages for information organization and reconstruction that convert the learning contents to learners' understandable and manageable style (Uwe, Rolf, Sabine, & Berthold, 2010).

2.16 Theoretical Frameworks of this Research Project

This section establishes two theoretical frameworks which are used to define the research processes of this project. They are the cognitive processing framework for learning computer programming and the TPACK framework, and are discussed in what follows.

The Cognitive Processing Framework

This framework provides a full spectrum of relating the cognitive processing steps and the cognitive attainments to computer programming instructional strategy. In respect to this research project, the requirement for this theoretical framework is to know which aspects need to use the e-learning object, and determine whether the cognitive attainments using the e-learning object match the levels of use in introductory programming.

Previous work in Table 2.2 and 2.4, Table 2.2 provides a strategy for novice learning of computer programming by breaking down the complexity to acquire knowledge incrementally, meeting the stepwise design in the scaffolding of a pedagogy (Butler & Morgan, 2007). While Table 2.4 provides detail of how these cognitive processing steps link to the attainment of cognitive domain levels defined by Bloom's Taxonomy, the cognitive processing framework therefore can be worked out by linking up the Table 2.2 to provide a unified model of cognitive processing steps and cognitive attainments with the instructional strategy of computer programming. This work is presented in Table 2.7.

Table 2.7 Modelling of cognitive processing framework for learning computer programming (The major stages of the cognitive tool supported in this pedagogy are indicated in bold. These also are major focuses in introductory programming.)

Cognitive processing steps	**Possible activities to attain the cognitive domain level(s)**	**Bloom's taxonomy cognitive domain**	**Computer programming instructional strategy**
Attention & Perception	Understand the meaning, translation, interpolation and interpretation of instructions and problems	**Knowledge Accretion**	**Problem Identification** e.g. - comprehensively knowing the programming problem - identify what input/output is
Repetition & Coding	State a problem in one's own words, or reconstruct the information to one's style		**Program design**, e.g. - Learning and understanding the semantics/concept of variables, the features of primitive data structure, keywords and complex data structure - learning and understanding the

			structure of simple statements and expressions such as I/O statement and logical expression - learning and understanding the features of primitive, nested and complex program controls
	The information is promoted to be knowledge	**Knowledge Understanding**	- elaborate the semantics and logics associated with the program - explain the meaning of a given program/algorithm
Coding & Storing	Use the knowledge to solve a domain-based problem and deepen the information to be solid knowledge	**Comprehension** & Application	**Program implementation** e.g. - coding the design with specific program language - write program to solve real-world problem
Storing & Retrieving	Reuse the knowledge on internal/external context	Analysis & Synthesis	Testing and debugging, e.g. - finding syntactical errors - review for semantics error - code modification - retesting and simple semantics errors
	Finding and modifying simple errors, Finding errors, Redesigning and redeveloping	Evaluation	Program redesign with the new requirements, e.g. - trigger a redevelopment cycle

To explain, Table 2.7 indicates that the learning phases of 'Program Design' need to involve the cognitive processing steps of 'Repetition & Coding', while attaining to the domain levels of 'Knowledge Accretion' and 'Knowledge Comprehension'. With regard to this cognitive development-based pedagogy, this linkage can be interpreted as 'if the use of e-learning object in this pedagogy needs to support the cognitive processing of "Program Design", it needs to be designed to support students being able to know the problem in their mind, and reconstruct the information to their understandable style, and eventually promote the information to be knowledge'. In a similar way, to support the 'Program Implementation', the 'use of the e-learning object in this pedagogy needs to help students to translate, comprehend and interpret information to new knowledge'.

As indicated, this cognitive processing framework shown in Table 2.7 identifies the cognitive processing for learning computer programming, and to its attainment, however it does not indicate the aspects to be supported by using the e-learning object. It is discussed in Section 3.2.

Evaluation of Teacher's Capacity: TPACK Framework

While the TPACK framework provides a systematic theoretical approach to design the stage 2 research, the second framework defined in this chapter is the TPACK framework. It is used to design the study on stage 2 which focuses on teachers' knowledge of using technology-based pedagogy. As discussed in Section 2.9, this project adopts the TPACK framework in the stage 2 study. The major advantages of using TPACK is explained as follows:

- TPACK assesses teachers' capacity to use technology through a set of fundamental knowledge types that teachers are required to hold. This tailored systematic approach to evaluate the spectrum of teachers' general skills is not present in other reviewed frameworks and models.
- TPACK is better than other frameworks or models, as it concerns teachers' operational features for implementing a pedagogy instead of evaluating the outcomes of using technology. This feature is important, as this study emphasizes teachers' acceptance rather than their view of using technology for this pedagogy.
- TPACK can be used to evaluate teachers' capacity for involvement rather than features applying to students. For example, in each integration phase TPACK requires teachers to understand how technology influences the pedagogy (the primary form of knowledge) through the related subject matter (domain knowledge) (Roblyer & Doering, 2010). This characteristic is rarely found in the reviewed frameworks or models. For example, the SAMR framework aims to establish a technological system in education to improve learners' performance; TIM is a resource to guide powerful technology at different levels but without considering operational strategy or the approach of technology integration; ADDIE is a strategic framework for instructional design based on ISD (Instructional System Design) recommendations, and is mainly used for developing training programmes; while RAT is a simple form of SAMR that is not designed to focus on teachers' capacity.
- Although both TPACK and Triple-E concern teachers' capacity for using technology, there are major deficiencies in applying Triple-E to this project. These are that Triple-E strategically brings 'Engagement' in instructional design,

driven by students' learning goals, and those do not concern teachers' capacities. Therefore, Triple-E does not meet the objective of the stage 2 study which is to address teachers' pedagogical ability.

- TPACK is a context-free framework and is not restricted to any specific subject. This permits freedom in questionnaire design. For example, some questions can be solely based on technological issues, and others on pedagogical issues. Therefore, the study can reflect the major characteristics of teachers' use of the e-learning object in their pedagogy.

2.17 Summary

This chapter focuses on discussing the outcome of a literature review relating to this research project. It started with discussion of diverse difficulties in novice programming. It found that most of these difficulties are likely related to poor design of pedagogies that focus on students' knowledge of program semantics. Motivated by this problem, this project developed a cognitive development-based pedagogy that focuses on improving students' cognitive presence on learning computer program semantics.

This proposed new pedagogy was designed as an application of educational technology, which is the concept of using technology for educational purpose. Educational technology initially was developed to facilitate learning by using the computer as Tool, Tutor and Tutee. This early developed model is called Taylor's three-T model which promotes the use of stand-alone computers to support teaching and learning. However, the revolution of the high-speed Internet changed the use of stand-alone computer-based learning to online learning. From this point, e-learning was quickly introduced to academia. E-learning was developed going through four generations. From a technological perspective, the differences between these generations are based on just providing online materials until to recent emergence of using complex modern technologies models termed KMS, LCMS and LMS. Learning object becomes an essential component for content design of this e-learning system. It is a piece of learning information, packaged with an object style.

When learning technologies were widely used for learning tool design, there were diverse technology-based learning models developed and used for supporting teaching and learning. However, previous work revealed a large number of these learning models and tools to be generally restricted to facilitate learning specific program languages, or provide students a drill and practice environment to learn how to program. For some others, they are likely used to facilitate students in speeding up the programming process. It indicates that few of these reviewed tools are used to support in-class activities with existing course materials, and focus of improving the scaffolding of a pedagogy.

This cognitive development-based pedagogy was designed by applying the concept of information processing. This concept defines the encoding process and, reconstructing a received learning information to be learners' understandable and manageable style, and finally convert this information to be learners' knowledge. This encoding process includes six major cognitive steps: 'Attention', 'Perception', 'Repetition', 'Coding', 'Storing' and 'Retrieve'. Each of these steps maps to different levels of cognitive attainment identified in Bloom's Taxonomy, including the domains of Knowledge, Comprehension, Application, Analysis, Synthesis and Evaluation. This is outlined in Table 2.4. This table provides information on the levels of cognitive attainment by the steps of cognitive processing. This work is used to identify which steps of cognitive processing need to be involved in the computer programming instructional strategy as well as supported by using the e-learning object.

The e-learning object is defined as a cognitive learning tool, as it is used to help students construct their knowledge on the semantics of program controls. It was designed by applying the concept of e-learning and learning object, grouping 31 learning objects and accessed via the course materials. It is mainly used to support the cognitive processes in the scaffolding of this pedagogy. The detail of the e-learning object design is discussed in Chapter 3.

Pedagogy is defined with practices and theories of teaching and learning. It is developed with different learning behaviours and psychological approaches, and has been discussed in psychology and pedagogical design for a long time. Skinner's *About Behaviorism* stated that learning is a process with positive and negative reinforcements, determined by the responses to stimuli. When cognitivism was closely connected with schools, Piaget suggested that learners need to go through four separate cognitive stages in a fixed order for the development of their cognitive skills. This theory defines cognitive development stages including a series of learning-related activities: observing, classifying, categorizing, attention, perception, interactivity, and reasoning. On the other hand, Vygotsky introduced a concept that cognitive development for learning can happen at diverse levels stemming from cognitive exploration, social interactions from guided learning with partners, instructors to co-construct knowledge, in the theory of 'zone of proximal development (ZPD)'. Both Bruner and Vygotsky used the term 'scaffolding' in describing classroom interaction, and the temporary assistance that teachers provided to students to facilitate their completing tasks and developing new understandings.

The pedagogy of this project was developed based on Vygotsky's concept of ZPD and is defined with three major learning processes: (i) facilitating cognitive development on learning program semantics; (ii) helping students conceptually map their program semantic knowledge to the coding process; and (iii) providing exercises to enhance students' learnt knowledge. These were developed following the functions of scaffolding which included conceptual scaffolding, strategic scaffolding, and metacognitive scaffolding. Conceptual scaffolding is used to provide an effective way for learning program semantics by using the e-learning object to make sense of the semantics knowledge and conceptually link to the coding processes. As indicated in Figure 2.1, they are the learning processes (i) and (ii), and are robustly supported by using the e-learning object. The learning process (iii) aligns with the metacognitive scaffolding to extend the learning process to practice and exercise after students acquire enough knowledge. Students are required to complete the exercises on their own. It could be also supported by reviewing the e-learning object.

Strategic scaffolding is pursued by using the syntax-free approach, and program walk-through instructional methods to improve the learning approaches. As discussed in Section 2.4, the syntax-free approach encourages students by primarily focusing on the concept of program semantics, while program walk-through is a self-exploration learning method that provides a strategy for students to construct their understanding to map the semantics to the coding process. Through support with the e-learning object, this scaffolding also improves the learning environment.

The use of an e-learning object for this cognitive development-based pedagogy is also subject to some concerns in e-learning-based pedagogy. These are teaching presence, social presence, and cognitive presence. For teaching presence, this pedagogy maintains in-class instructions delivered by instructors. For social presence, students can communicate with other students and instructors through the college e-learning system, while the e-learning object is installed together with the course materials (detail discussed in Chapter 3). For cognitive presence, this pedagogy applies the e-learning object to facilitate students learning cognitive skills.

There are two frameworks defined in this project. The first is used to define the cognitive processing steps required to learn computer programming in different learning phases. It is developed by mapping Table 2.2 to Table 2.6, as indicated in Table 2.7, to provide a full pedagogical spectrum of the cognitive processes to be designed in the pedagogy. These cognitive processes also define the needs of supporting the cognitive processes for the scaffolding in this pedagogy. The second theoretical framework applies the TPACK framework to determine whether the technological, pedagogical, and content features of this pedagogy meet with teachers' knowledge of using it. It is a major step to generalize this pedagogy to wider use in introductory programming. Although diverse models, such as SAMR, TIM, Triple-E, ADDIE and RAT, can be used for this purpose, it is found that the TPACK framework is more suitable for use for this purpose. It is because that the TPACK is a pedagogical, operationally focused and context-free framework that provides a systematic approach to understand teachers' capacity for primary form and domain knowledge, and based on them consisting of the whole knowledge framework.

Chapter 3 The E-learning Object

3.1 Introduction

Chapter 2 establishes a spectrum of the scaffolding of the pedagogy proposed in this research project. It also develops two theoretical frameworks defining the research approaches of this pedagogy. One is the cognitive processing framework which is used to identify the needs of cognitive attainments for the instructional strategy of computer programming, and based upon this information it identifies the scope in the scaffolding of this pedagogy that needs to be supported by using the e-learning object. This framework is achieved by mapping the cognitive processing steps required for every learning phase in the computer programming instructional strategy, and maps this process to the Bloom's Taxonomy cognitive domain levels. This is demonstrated in Table 2.7.

This chapter extends the work based on the cognitive processing framework, defined in Table 2.7, to show how the e-learning object is used in the scaffolding of this pedagogy. It is outlined in Table 3.1. With the work of identifying the role of using the e-learning object, this chapter also further discusses the design of the learning processes by following the functions of scaffolding. This is presented in Figure 2.1. Upon the completion of this framework, this chapter goes on to discuss the design detail of the e-learning object. This includes the processes of decontextualizing and contextualizing the context of program controls from the curriculum of introductory programming to establish the new learning context, which is suitable for using the e-learning object, and works out the granularity of this context as individual learning objects.

Following the descriptions of how the e-learning object applies the information processing concepts to design the animated flowchart and coding examples for individual learning objects, this chapter provides detail on the shape and size of the e-learning object. This includes how it is used together with the course materials, and supports the instructional approaches for class-based activities. Finally, this chapter discusses the outcome of an evaluation that focuses on a study as to whether the e-learning object meets the criteria used in this research project.

3.2 The Scope of the E-learning Object in Computer Programming Instructional Strategy

Table 2.7 defines a framework that links cognitive processing steps related to the instructional strategy for computer programming, and the attainments for these cognitive processing outcomes. However, this framework does not precisely indicate the aspects of these cognitive processing steps to be supported by the e-learning object, and with this information to identify the shape and size of the e-learning object in this cognitive development-based pedagogy.

To extends this cognitive processing framework based on previous works, there are many criteria need to be concerned. First of all, the use of the e-learning object within the course materials must be instructionally grounded, and the context designed in the e-learning object needs to prioritize a strategy to improve students' cognitive presences. Besides, the use of it has to relate on students' understanding the the semantics of different approaches for program design and development. In this sense, concerning the instructional strategy of computer programming, it is suggested that the e-learning object focuses on supporting the learning phases of 'Program Design' and 'Program Implementation'. As indicated in Section 2.6, these two learning phases are major sources of mentally taxing effort in novice programming, and also the use of cognitive artefacts provide a great contribution on the cognitive processing for this programming learning phases.

Based on this theoretical concept, the completed work to define the scope of using the e-learning object in respect of extending the pedagogical framework defined in Table 2.7 is elaborated in Table 3.1.

Table 3.1 Cognitive role of the e-learning object in this cognitive development-based pedagogy. The scope that is to be supported by the e-learning object is indicated in bold.

<table>
<tr><th>Cognitive Processing Steps</th><th>Bloom's taxonomy Cognitive domain</th><th>Computer programming instructional strategy</th><th>Scope supported by the e-learning object</th><th>Expected achievement of this scope</th></tr>
<tr><td>Attention & Reception</td><td rowspan="2">Knowledge accretion</td><td>Determine the programming problem, e.g.
- comprehensively know the problem
- identify what the inputs/outputs are to be</td><td>Not supported by the e-learning object</td><td>- analyse problem
- problem-solving skill
- knowledge is transferable to other program languages, as it is not dependent on language</td></tr>
<tr><td rowspan="2">Repetition & Coding</td><td rowspan="2">Program Design, e.g.
- understand the concept of variables
- understand the features of primitive data structure, keywords and complex data
- understand the structure of simple statements and expressions, such as I/O statements and logical expressions
- understand the features of primitive, nested and complex program controls
- elaborate the semantics and logics associated with the program
- explain the meaning of a given program/algorithm</td><td rowspan="2">- The essential part supported by the e-learning object, as far as the topics in introductory programming are concerned</td><td rowspan="2">- capture the concept of semantics and syntax
- understand the concept of use basic programming components
- able to design program
- use primitive, nested and complex program controls
- use semantics and the associated logics defined in the program/algorithm
- create a semantics model of a given program or algorithm mentally</td></tr>
<tr><td>Knowledge understanding</td></tr>
</table>

Cognitive Processing Steps	**Bloom's taxonomy Cognitive domain**	**Computer programming instructional strategy**	**Scope supported by the e-learning object**	**Expected achievement of this scope**
Coding & Storing	**Comprehension & application**	**Program implementation**, e.g. - code the design in a specific program language - write a program to solve a real-world problem	**- partially supported by the e-learning object in respect of:** **- implementing complex controls, handling the difficulties generated by tight coupling**	- knowledge of developing/implementing complex control - cope with tight coupling between semantics and syntax - able to transfer designed semantics to block of program codes - trace a block program developed by a specific language - relate the knowledge to next step, testing and debugging
	Analysis & synthesis	Testing and debugging, e.g. - finding syntactical errors - review for semantics errors - code modification - retesting and simple semantics errors	Not supported by the e-learning object; however, it could be used to refresh students' learnt knowledge by reviewing code-based examples to increase skill in debugging	- understand thoroughly the semantics, logics, and syntax of the tested and debugged program - advanced programming knowledge is required - isolated knowledge on testing needs to be established
Storing & Retrieving	Evaluation	Program evaluation and redesign, e.g. - final testing - trigger a redevelopment cycle	- unsupported by the e-learning object	- advanced programming techniques are required - generally, this is not in novice programming

Table 3.1 shows that the e-learning object involves in large part of the computer programming instructional strategy, including the phases of ‘Program Design’ and ‘Program Implementation’. According the curriculum of introductory programming (ACM Curriculum Committee on Computer Science, 2001), the topics including in these two learning phases are mainly relating to use of program controls, such as the decision, iterative, array and functional controls. Furthermore, studies discussed in Section 2.2 indicated that learning the semantics of these program controls is a major source of students’ learning difficulties, even if they have a good understanding of the syntax of the programming language (e.g. Farrell, 2007; Schulte & Bennedsen, 2006; Vihavainen, Airaksinen, & Watson, 2014). Therefore, the use of the e-learning object for these two learning phases in the instructional strategy of computer programming to be major advantages of this pedagogy.

Following with this concept, the framework designed in Table 3.1 uses the visualization techniques and concept mapping to achieve the cognitive processing steps ‘Attention & Perception’ and ‘Repetition & Coding’, and supports ‘Program Design’ and ‘Program Implementation’. While these two major learning phases matches the learning level of novice learners of computer programming. The advantages of using these techniques to supportive cognitive developments also has discussed in the Section 2.6. In respect to cognitive attainments, the framework defined in Table 3.1 identified it is aimed at completing the levels of ‘Knowledge Accretion’, ‘Knowledge Understanding’, and primarily ‘Comprehension’, as defined in Bloom’s Taxonomy cognitive domain model. They are highly specific to the needs of novice programming.

The last concern in Table 3.1 is whether the e-learning object can provide a significant effect across the whole range of learning processes of introductory programming. As suggested by the ACM Curriculum Committee on Computer Science (ACM Curriculum Committee on Computer Science, 2001), the topics of program control have a 70% weighting of the total contents of introductory programming. With this weighting, it is believed the e-learning object can significantly affect students’ learning outcomes. Furthermore, the knowledge of semantics and program controls is related to both procedural and object-oriented programming. This feature also is an essential property of this pedagogy while used in other introductory programming modules.

3.3 Decontextualization of the Context to Use the E-learning Object in Introductory Programming

As discussed in Chapter 2, learning object decontextualization is a conceptual process that extracts a specific learning context from a wider subject area, and contextualizes this context as a new learning domain that is capable of influencing the learning outcome on a subject area (Lukasiak et al., 2004). According to Table 3.1, the e-learning object does not support all learning phases in the instructional strategy of computer programming, but is focused on the program controls relating to ‘Program Design’ and ‘Program Implementation’. Therefore, the spectrum of the use of the e-learning object in the study programming module of this project is described in Table 3.2.

Table 3.2 The spectrum of decontextualizing the learning context of the study module to the e-learning object together with the module materials.

The learning contents in typical introductory programming	General approach of concept introduction	Suggested decontextualized context uses the e-learning object
Concept introduction & Use of the Editor/Compiler/Interpreter	Lecture notes	N/A
Keyword & Identifiers	Lecture notes	
Variables/Constant	Lecture notes	
Basic I/O operation	Lecture notes	
Decision Control	Lecture notes	Use the e-learning object If, if-else, nested if-else else if and switch accountManagement gradingAlgorithm charTest; find_max_min
Iterative Control	Lecture notes	Use the e-learning object: While, do-while, while nested if do-while nested if, nested while for, nested for find_averageMarks finding_prime factorial; sum_up_list
Array	Lecture notes	Use the E-learning object 1-D Array, Array with for control 2-D Array concept 2-D Array with for loop addingMatrix; sorting_list
File operations	Lecture notes	N/A
Function/Procedure	Lecture notes	Use the E-learning object calling and return parameter passing
Basic Class and Object (Java) Data Structure and ADT (C/C++)	Lecture notes	N/A

The characteristics of decontextualization of the pedagogical framework described in Table 3.1, to be supported by the e-learning object as discussed in Table 3.2, can be summarized as: the context supported by the e-learning object is significant to novice learners of computer programming; ideally this context is not specific to learning syntax of a programming language, therefore it is possible to reuse it in different programming languages; and this decontextualized contents can be broken down into finer topics and granularity as learning objects.

For the final characteristic, the granularity of the program controls to finer topic of learning objects is the process of broking down all program controls into 31 learning objects. Every learning object focuses on a specific topic and based of it classified as ‘primitive object’, ‘nested object’ and ‘complex object’. The primitive objects are used to deliver basic program controls, such as ‘if-then’ and ‘while-do’ constructs, while the nested objects are focused on delivering the nested controls, such as ‘if nested if’ or ‘while nested if’. The complex objects are selected program examples to enhance students’ concept of computer programming. They are used to deliver the concepts where this cannot be facilitated by primitive and nested objects separately. These learning objects are listed in Table 3.3, encoding with prefix LO and a number.

Table 3.3 The learning objects grouped in the e-learning object

Category of program control	**Learning object**		**Type of object**
	Object ID	**Name of control**	
Decision control	LO29	if	primitive
	LO29a	if-and	primitive
	LO30	if-else	primitive
	LO31	nested if-else	nested
	LO32	switch	primitive
	LO33	switch nested if	nested
Iterative control	LO34	while	primitive
	LO34a	while-or	primitive
	LO35	do-while	primitive
	LO36	while nested if	nested
	LO37	do-while nested if	nested
	LO9a	do-while nested if (pseudo)	nested
	LO38	nested while	nested
	LO39	for	primitive
	LO40	nested for	primitive
Array control	LO41	1D Array	primitive
	LO42	for with Array	primitive
	LO43	2D Array concept	primitive

	LO44	Nested 2D Array	nested
Functional control	LO17	calling and return	primitive
	LO18	parameter passing	primitive
Control with examples			
Decision/switch	LO45	accountManagement	Complex
Decision/switch	LO46	gradingAlgorithm	Complex
Decision/switch	LO47	charTest	Complex
Decision/switch	LO48	find_max_min	Complex
Loops	LO49	find_averageMarks	Complex
Loops/Decision	LO50	finding_prime	Complex
Loops	LO51	factorial algorithm	Complex
Loops	LO52	sum_up_list	Complex
Array	LO53	addingMatrix	Complex
Array	LO54	sortinglist	Complex

3.4 Uses of the E-learning Object in the Study Module of this Project

The pedagogy in this research project was firstly used in a module called 'Programming with C/C++ Language'. It is a year 1 programming module of a sub-degree programme in Computer Studies offered in The Hong Kong Institute of Education, School of Professional and Continuing Education. This module runs for 15 weeks, each involving a 2-hour lecture / demonstration / self-practice, 1-hour in-class activities and tutorial, and 3-hour out-of-class self-study. This is summarized in Table 3.4.

Table 3.4 General contents learnt in the module 'Programming with C/C++ Language'
Note: The No. of topics with the No. of 2–5 is mainly supported by the e-learning object.

Teaching Week	Topics (learning and self-practices + tutorial hours)	General contents
Week 1-2	Basic Programming Concept (4+2 hours)	Program identifiers; variables; data types and general skills of writing program
Week 3-4	Decision (4+2 hours)	if-else, nested if ; switch controls
Week 5-7	Iterative (6 + 3 hours)	while; do-while; for; nested for; for-while with inner if-else
Week 8-9	Array (4+2 hours)	1D Array; 2D Array; Use nested for cope with Array
Week 10-11	Methods (4+2 hours)	Define and Calling Method; flow of control; call by references/values
Week 12-13	File and String Operations (4+2 hours)	String object and Array; General file operations
Week 14-15	Object concept (4+2 hours)	Object and Class create and use; inheritance; encapsulation and polymorphism;
	Out-of-class Self-study (45 hours)	Self-study hours for complete assignment/exercise and exam preparation
	Total 30 learning/self-practices hours + 15 Tutorial/in-class activities + 45 out-of-class self-study hours = 90 hours	

Table 3.4 shows that the e-learning object is used in 33 learning and tutorial hours (e.g. topics 2–6) in a module of 45 contact hours. In this time, the e-learning object was mainly used to support the instructor to facilitate the learning contents demonstration and instructional delivery. For the 45-hour out-of-class self-study, the e-learning object mainly helped students to review the major concept and in completing the homework connected to the learnt topics. Therefore, with this usage of the e-learning object, significant effects could be provided in this module, including the lectures and demonstrations, as well as in-class self-work exercises. The learning activities in every lecture are summarized in Table 3.5.

Table 3.5 The learning activities in every weekly-based lecture

Time Used	Type of instructions	Instructional Activities
60	Demonstration	The lecturer presented the lecture notes supported by using the e-learning object to explain the concepts of the topics. It may use the links provided in the lecture notes to use the animated flowchart and code-based example step by step to explain the concept and semantics of the topic to students.
		15-minute break
15	Self-study	Students tried the animated flowchart and the code-based example by themselves to enhance their concepts of the semantics of program control.
30	In-class coding exercise	After the demonstration period, students were provided in-class exercises. They were normally required to complete these exercises independently based on limited help from instructor such as the given course resources (e.g. incomplete solution sample, hints and working guides, etc.). If students had any problems with working through the exercises, they could review the e-learning object anytime they wanted to. However, as this is in-class activities, students also could discuss with the instructor their problems.
60	Self-study and Practices	The 60 hours of self-study and practices were completed by students themselves based on instructor-assigned exercises. Students could use any given learning resources, including the e-learning object, to complete the assigned exercises.

3.5 The Design of Learning Objects

Every learning object presented in Table 3.3 was designed based on using the visual imagery, including dynamic graphics, animations, and simulations to visualize the learning concepts. As discussed in Chapter 2, by applying cognitive artefacts, the visual imagery allows users to absorb the information with the minimal cognitive load to facilitate information reconstruction (Doolittle, McNeil, Krista, & Scheer, 2005; Moreno & Park, 2010; Schatsky & Schwartz, 2015; Theiner, 2013). There are some major concept of visual imagery applied for e-learning object design and they are summarized in Table 3.6.

Table 3.6 Summary of cognitive modelling for the design of the e-learning object

Content creation	Modelling of cognitive artefacts in the e-learning object
Major styles	In respect of the major style the e-learning object is designed with the 2D animated flowcharts, and conceptually maps to code-based examples. The process of the animated flowchart is fully controlled by students.
Categories visualization techniques	Categories visualization techniques related to: (1) Animation modelling; (2) Learning objects modelling; and (3) View flows and process control. Students use animations, view process and process control through viewing the animations to capture the concepts of the learning content
Input style	Input style of the e-learning object relies on the animated flowchart and the code-based example to enhance students' understanding and the degree of accessing the information (Sims, 2000). Both teacher and student can control the e-learning object.
Human interface	Human interaction involves two major aspects. During class activities, the e-learning object could be used to affect the teacher's instructional approach and used by students to review the learnt concept during an out-of-class. The use of animated flowcharts is also familiar to teachers and students.
Delivery structure	The e-learning object is embedded in the institute's Blackboard, together with the course material. It is instructionally oriented, so delivery is initiated by teachers/instructors. However, as it is an online model, students can review the content in an out-of-class environment.
Major technology	Major technology includes: (1) The animated images with logical sequence connections. They are used to develop animated flowcharts and code-based examples. (2) The flows simulate the semantics. It is used to show the logics of the program controls (3) Web-based usability design via e-learning platforms. The learning objects are organized with hyper-structures. These technologies are effective cognitive artefacts and the evidence is significant for cognitive model design.

The layout design of all learning objects is consistently made up of three major parts. They are the animated flowcharts, code-based examples and three process control buttons, 'Click Process', 'Back' and 'Start Again'. Figures 3.1 and 3.2 present the typical design of two learning objects, which are the LO2 and LO44. They are designed to deliver the control topics of 'if-else control' and '2D array nested if control' respectively. The LO2 is a primitive object and LO44 is a complex object, as indicated in Table 3.3.

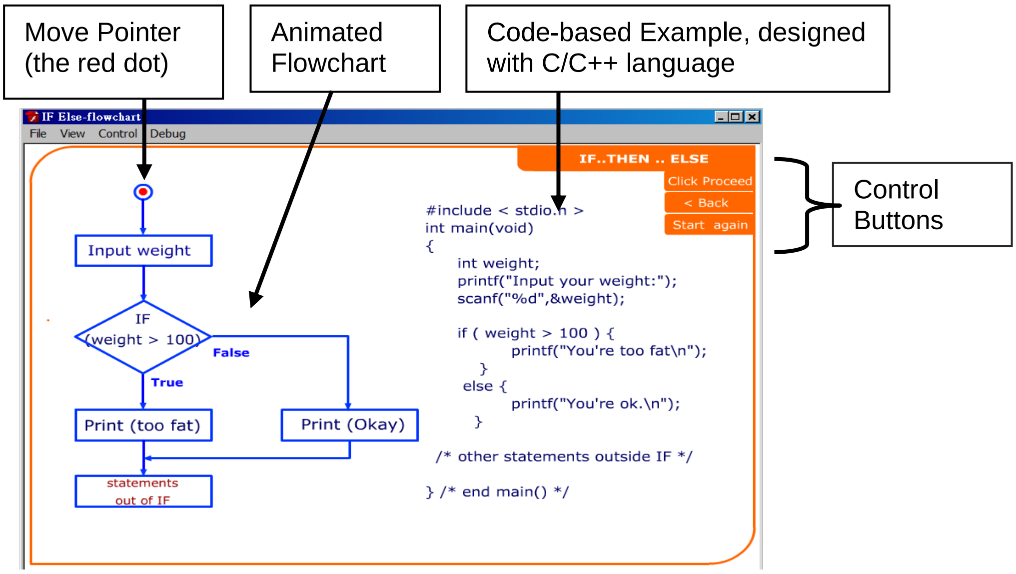


Figure 3.1 Layout of primitive object LO2, 'if-else control' with C language code-based example

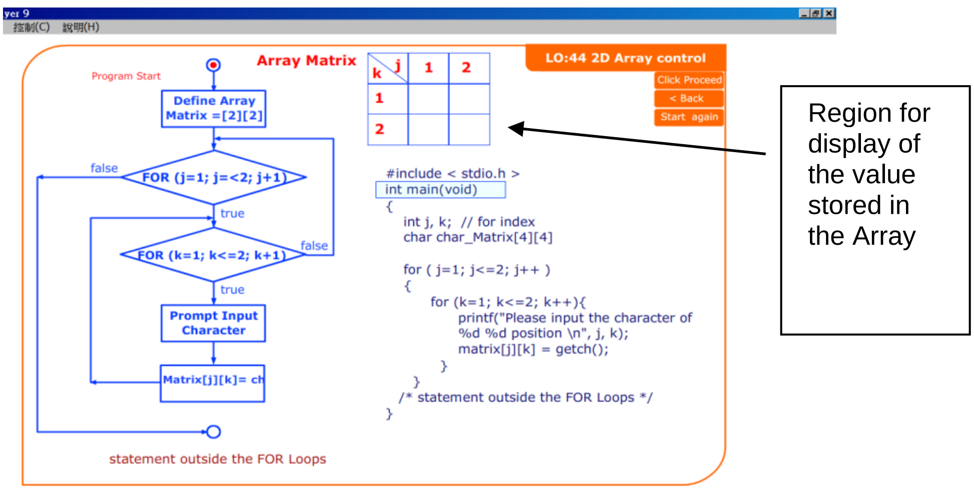


Figure 3.2 Layout of nested object LO44: Nested for a 2D Array, with C language code-based example

3.6 Instructional Approaches Supported by the E-learning Object

As discussed in Section 2.15, a feature of strategic scaffolding in this cognitive development-based pedagogy is the improvement of the instructional approach by using a syntax-free approach and program walk-through techniques to facilitate concept mapping. With regard to the syntax-free approach, the e-learning object used an animated flowchart to present the semantics of a program control. This animated flowchart does not execute automatically, but rather it is controlled by teachers or students through the 'Control Buttons', shown in Figure 3.1. This enables the teacher, or students using it in a step-by-step style, to be consistent with the instructional delivery in class interactions. It also possible for students to review the learnt contents during self-practices and exercises.

For the program walk-through technique, it is designed with the code-based example, as shown on the right-hand side of Figures 3.1 and 3.2. The facilitating of mapping students' concepts of semantics to the coding process is achieved with every step of the animated flowchart and code-based example being executed synchronously, while the execution processes are visualized with a moving 'red-dot' along with the arrow lines, that constructs the animated flowchart, to show the program execution status. For example, for each step, the red-dot will be moved to a 'symbol' of the flowchart to which the 'symbol' and the corresponding line of code in the code-based example are flashing to signal the logical status of the executed line of code in the program. Therefore, these synchronization processes between the animated flowchart and the code-based example highly possible to 'visualize' the 'logic flow' and map to line of codes of the program. Furthermore, each time the red-dot moves, it will carry a variable which shows up the current value of that variable to be used by the next step of the program. This crucially helps students to understand the runtime logical situations together with the hidden value carried by variables.

Figures 3.3 (a)–(k) illustrate the step-by-step instructional support with the learning object LO2 as an example. It is used to deliver the concept of topic 'if-then-else' in the decision program control.

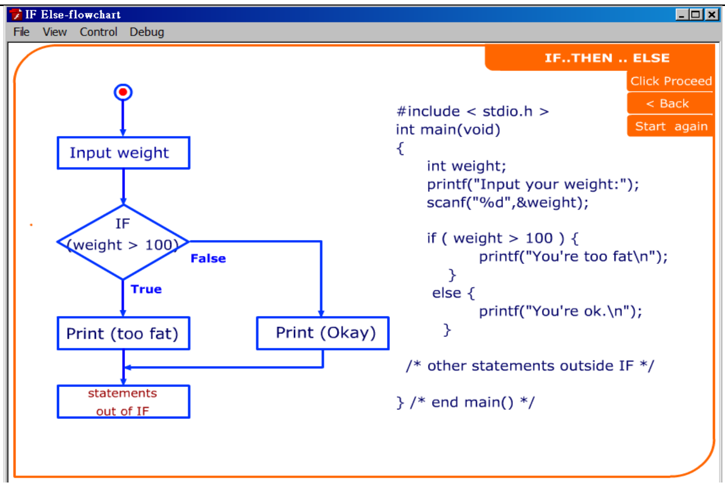


Figure 3.3(a) Step 0

The initial stage, when the learning object is loaded.

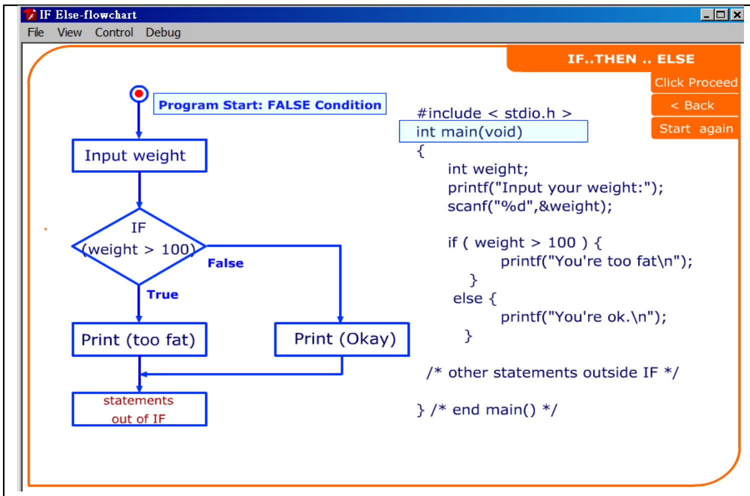
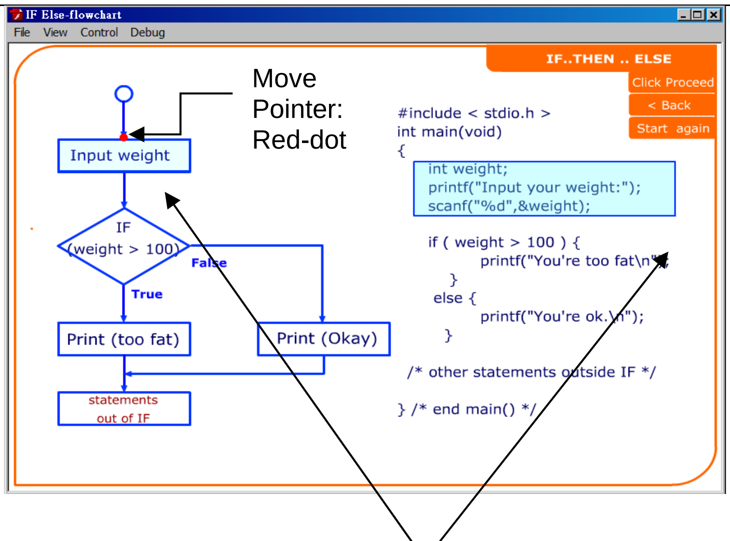


Figure 3.3(b) Step 1

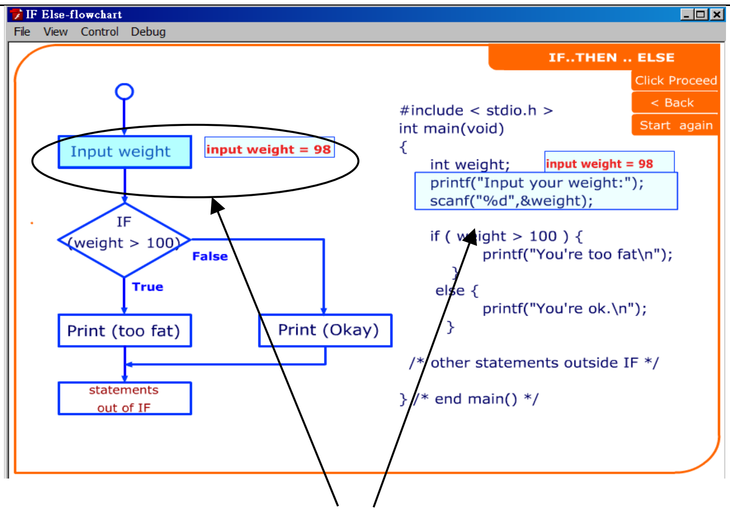
It shows the program start while it presents a false condition.



Highlighted and Flashing sychronously

Figure 3.3(c) Step 2

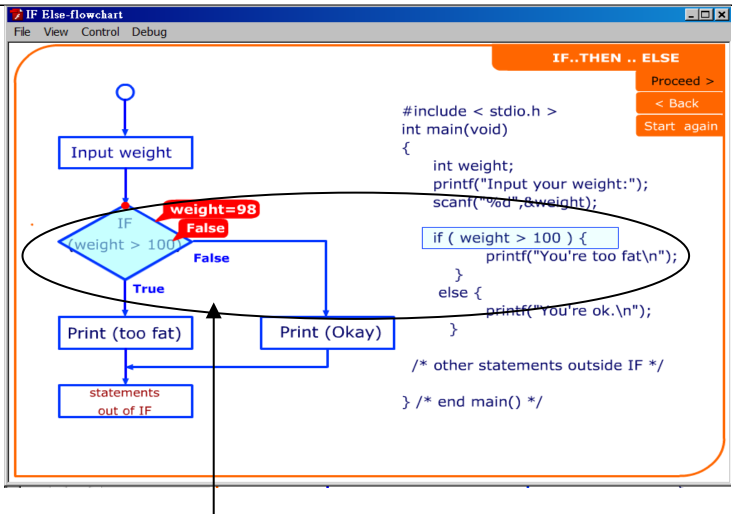
The control is flowed to get the value of 'weight'. In that time, the code-based example shown is processed by a series of I/O operations in Java language. In the meantime, the flowchart 'gets weight' and the corresponding lines of code in the code-based example are flashing. The 'move pointer' moves to the process 'get weight' along the arrow.



Highlighted and Flashing sychronously
The value of variable weight (=98) is

Figure 3.3(d) Step 3

Get the input weight = 98. As previous steps, the process 'get weight', the message 'input weight = 98', the corresponding symbol of the flowchart, the line of code in the code-based example is flashing.



Displays the test result

Figure 3.3(e) Step 4

Displaying the logical result of the test, ‘False’. Both the decision diagram and the line of code ‘if (weight > 100)’ are flashing. The ‘move pointer’ moves to the rhombus along the arrow.

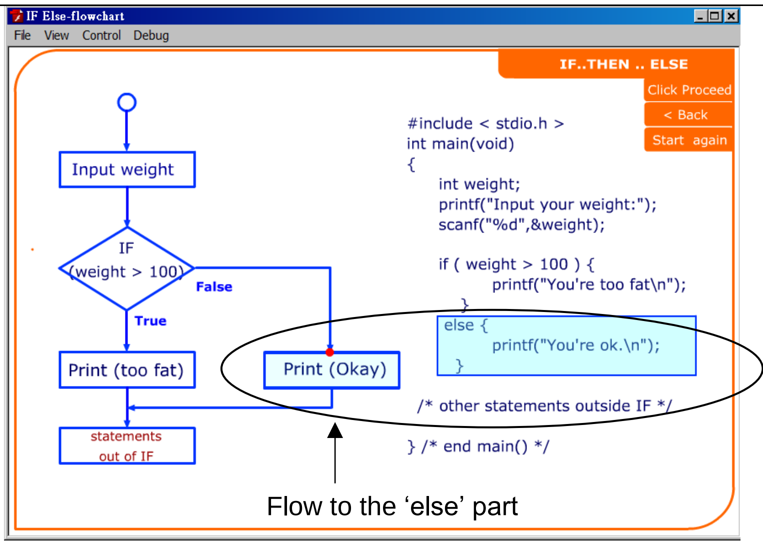
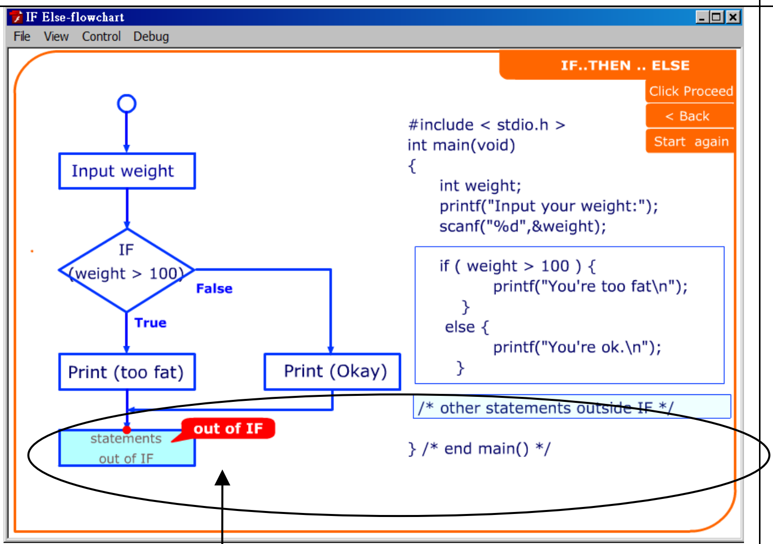


Figure 3.3(f) Step 5

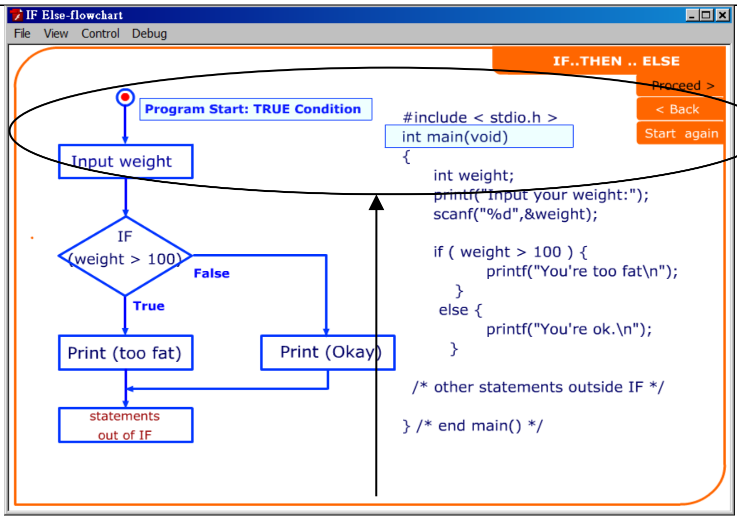
With the test return ‘False’, the flowchart shows that the program is flowed to ‘print okay’ along the ‘line false’. Similarly, the ‘move pointer’ moves to the process ‘Print (okay)’ along the arrow and the corresponding line of code confined by ‘else’ flashes.



Leave the else block

Figure 3.3(g) Step 6

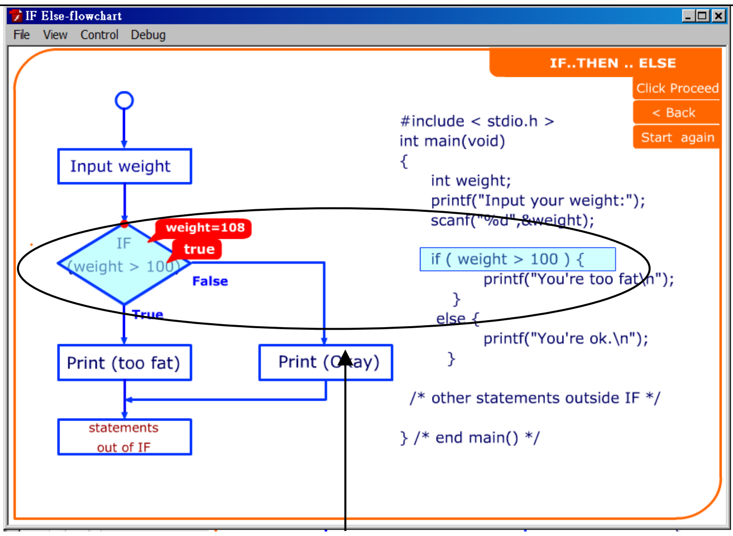
After processing the ‘else bock’, the program has flowed to the outside of the ‘if-else’ control. This information is displayed in messages and both the corresponding flowchart and code-based example flash accordingly.



Start again with true condition

Figure 3.3(h) Step 7

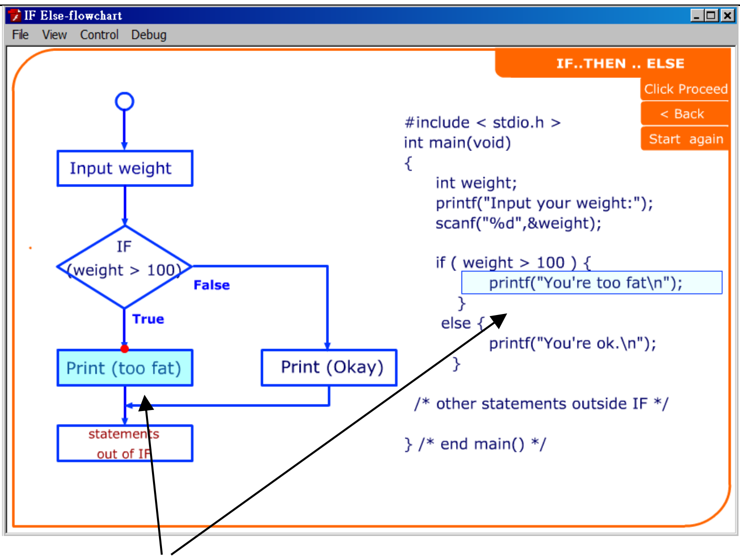
When the button ‘Click Proceed’ is clicked, the animation will be started again, this time with the true condition. This time the input weight is set to 108, over the condition of 100.



Display the true condition

Figure 3.3(i) Step 8

The process is as in previous steps, but the test result is true.



This time flow is to if part

Figure 3.3(j) Step 9

The program is flowed to the process ‘Print (too fat)’ along the ‘line true’. Similarly, the ‘move pointer’ moves along the arrow and is positioned at the correct process. Both the corresponding chart and code flash.

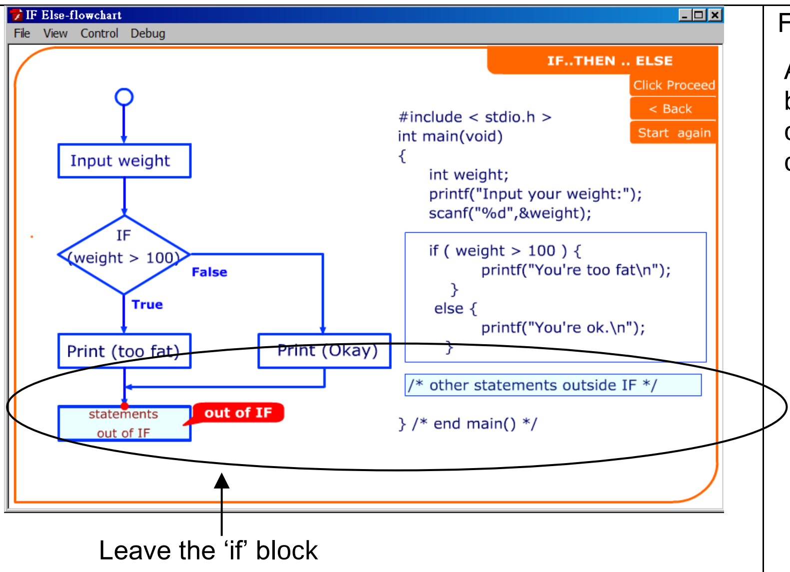


Figure 3.3(k) Step 10

After processing the 'if' block', the process is out of the 'if-else' control.

3.7 Delivery Strategy for the Learning Objects

The e-learning object is used with the course materials through the Blackboard system of the Hong Kong Institute of Education. The link for accessing the e-learning object is embedded in the corresponding lecture notes. Therefore, both students and teachers can use the e-learning object with just a 'click' from their lecture notes. Figures 3.4(a) and 3.4(b) indicate how the e-learning object is integrated into the Blackboard, and Figures 3.5(a) and 3.5(b) demonstrate how teachers access the learning objects through the lecture notes.

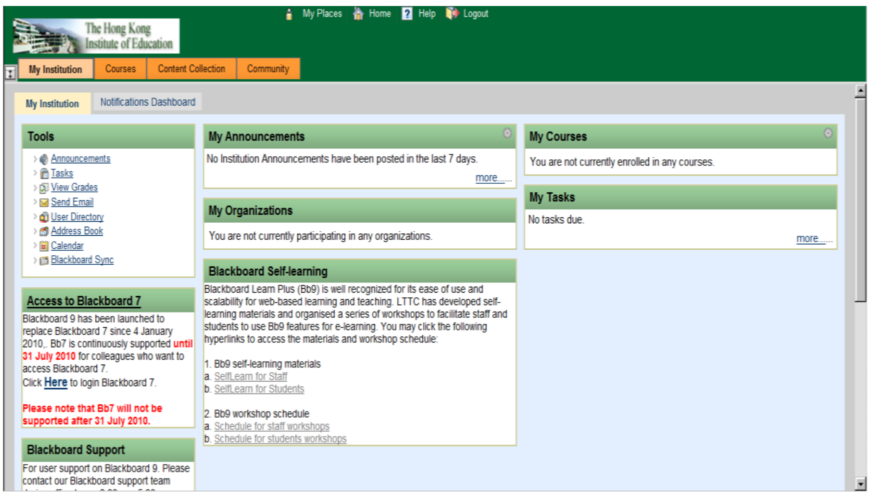
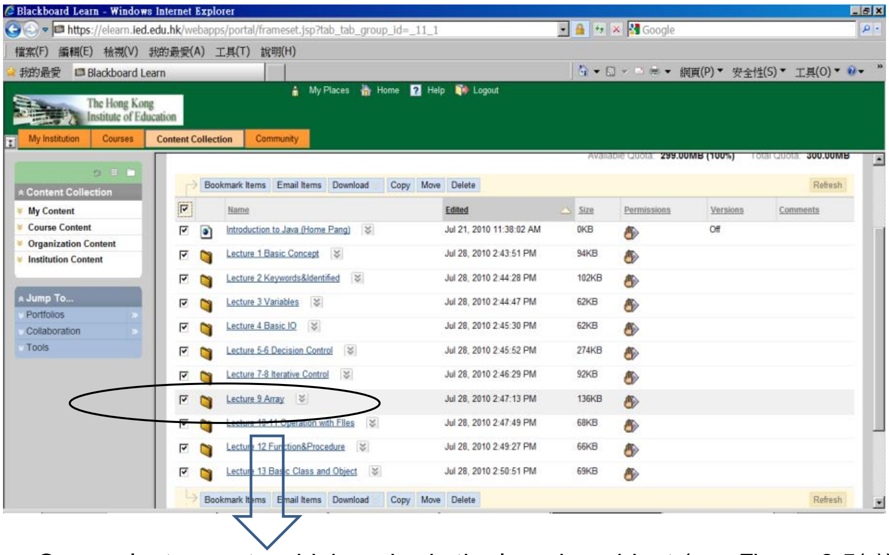
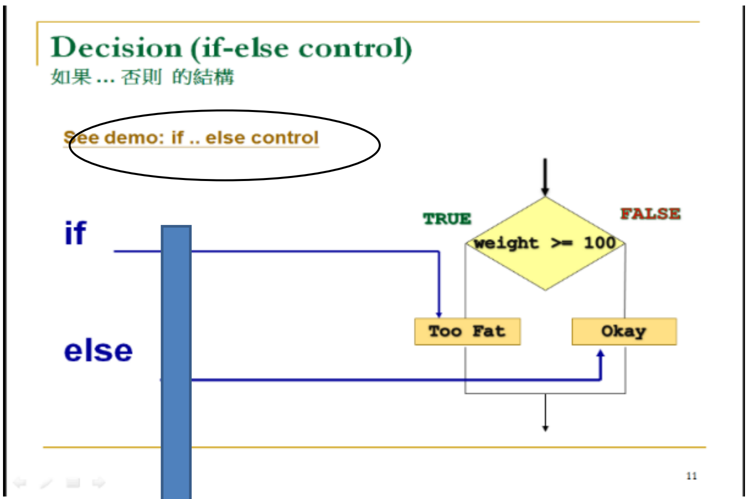


Figure 3.4(a) Blackboard system of Hong Kong Institute of Education. Students use it to access the studied programming module



Open a lecture note which embeds the learning object (see Figure 3.5(a))

Figure 3.4(b) Course material browsed in Blackboard. The e-learning object is embedded into all the corresponding lecture notes. Note this page is used by research group.



Link to open the learning object

Figure 3.5 (a)

The learning object if-else control is embedded in the course lecture notes, providing a Hyperlink to access. When the teacher teaching this topic is up to here in the class interaction, s/he is just required to click the link and use the learning object to demonstrate the semantics behind the if-else control.

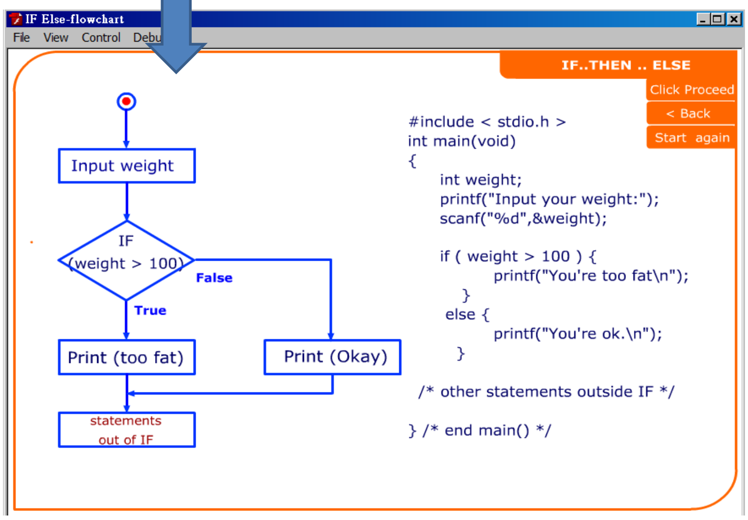


Figure 3.5 (b) Initial object

When the 'if-else' control object is loaded, the initial step of the learning object is presented (shown as step 0 in Figure 3.3a). Then the teachers can use the 'control process' to show how the control is executed in a step-by-step style.

There are many advantages to using Blackboard for delivering the e-learning object. First, Blackboard is a comprehensive and flexible e-learning platform while it is familiar to most teachers and students as it has been used in academic circles for a long time. Second, it can provide an effective way to set up the empirical study environment used by this research project. For example, it can separate out the content for both the research and control groups by only organizing the e-learning object in the research group that could be provide a consistent environment to both groups with the major difference is the use, or non-use the e-learning object. This approach highly minimizes the disruption to use the materials in research group. For example, with reference to Figure 3.5, in cases where this slide is used in control group, the link to open the learning object does not exist. The instructor therefore needs to explain normal pedagogical approach.

3.8 Scaffoldings and Learning Processes with the E-Learning Object

At the pedagogical level, the scaffolding of this cognitive development-based pedagogy was developed by conceptualizing the scaffolding functions of conceptual scaffolding, strategic scaffolding and metacognitive scaffolding to maximize students' learning outcomes through blending the instructor's helps and the student's independently learning in different cognitive processing steps for computer programming (Sarikas, 2018; Shabani et al., 2010). Defined by these scaffolding functions, there are three major learning processes involved in the scaffolding. While the associated theories of this scaffolding have been discussed in Section 2.5, and presented in Figure 2.1, this section goes further in explaining how the e-learning object works with this scaffolding. For convenience, Figure 2.1 is reproduced as Figure 3.6 and it is shown as below.

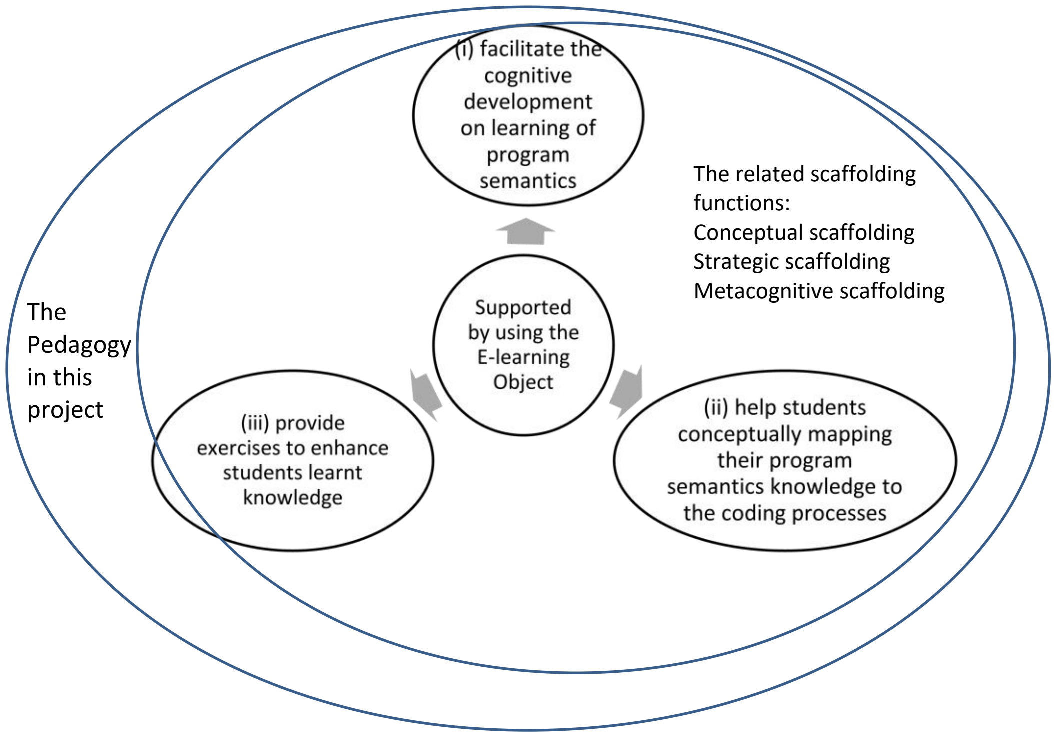


Figure 3.6 re-presents Figure 2.1 Scaffolding for the pedagogy developed in this project

Similar to Figure 2.1, Figure 3.6 shows that all the learning processes are associated with use of the e-learning object. However, they are supported at different points connected to different objectives defined in the scaffolding functions. This is explained as follows.

The learning processes of (i) and (ii) were developed following the conceptual scaffolding, to facilitate students' capturing knowledge of the learning contents. These processes mainly use the e-learning object in class interactions, accessing the learning objects through a link provided in the lecture notes as indicated in Figures 3.4 (a)–(b) and 3.5 (a)–(b). Upon displaying the e-learning object, the instructor then uses the animated flowchart and code-based example to demonstrate the logics of the relevant topic through the instructional approach as indicated in Figures 3.3 (a)–(k).

The learning process (ii), helping students to map the semantics concept to the coding process, is pursued by executing the animated flowchart and code-based example synchronously. Referring to Figures 3.3 (a)–(k) again, they show that the animated flowchart and the code-based example are presented together on the left- and right-hand sides of the learning object. While the animated flowchart is executed step by step, there is a ‘red-dot’, with a clearly visible valued variable moving along the flow-line of the chart, and the corresponding symbol in the flowchart and the line of code in the code-based example are flashing together in front of the students. This may help students conceptually mapping the concept of the logic and semantics of a program control to the coding detail they are holding in their mind.

The learning processes (i) and (ii) are also associated with the concept of strategic scaffolding. The supportive features of this scaffolding function for instruction methods, the syntax-free approach and program walk-through technique for concept mapping, have been discussed in Section 3.6. This use of an e-learning object to improve the pedagogical environment is also an approach designed to enhance the learning environment.

This scaffolding aligns with the design theory from Vygotsky’s Zone of Proximal Development (Shabani et al., 2010). Both learning processes (i) and (ii) are performed during in-class interactions, providing a means for students to learn collaboratively with their teachers and other students. Learning process (iii) extends the learning to provide a series of practices and exercises for students to complete independently. It is designed to test whether students can apply their knowledge to solving a programming problem while removing the direct support of instructors.

Besides, the scaffolding for this cognitive development-based pedagogy has many differences and similarities between using and not using the e-learning object in the learning processes. This feature provides a consistent learning environment to both the research and control groups in this research project, with the only difference being learning with or without the use of the e-learning object. For example, they learnt with the same curriculum of introductory programming and involved instructor support in learning processes (i) and (ii). The set of lecture notes they used only differed in having or not having the link to access the e-learning object: that is, in the control group, the instructor might use the static presentation to explain the concept, while the research group used the e-learning object. The detail of differences and similarities of use and non-use of the e-learning object in the scaffolding is summarized in Table 3.7.

Table 3.7 Differences and similarities of using and not using the e-learning object in the three major learning processes of the scaffolding

The major learning processes in the scaffolding	**Differences and similarities of**	
	Use of the e-learning object for in-class activities	**Non-use of the e-learning object for in-class activities**
(i) facilitate cognitive development on learning program semantics	The instructor presented an explanation of the animated flowchart to show the semantics of the learnt program control.	The instructor explains the topics with the lecture notes which present a static flowchart with the textual explanation.
(ii) help students conceptually mapping their program semantic knowledge to the coding process	Students are taught through mapping the concept shown in the animated flowchart using the coding example. The role of the instructor becomes that of facilitator.	The instructor relies on explaining some program example for understanding the coding process. However, it requires students to learn this context with prior knowledge of the programing language. It may mean students cannot solely focus on the semantics issue as they are required to cope with the concept of program semantics and syntax simultaneously.
(iii) provide exercise to enhance students' learnt knowledge	Students develop the program by directly using the IDE tools with reference to the e-learning object through the online learning platform	Students develop the program by directly using the IDE tools with reference to the course materials provided by the instructor.

3.9 Summarize the Advantages of Using the E-learning Object in the Pedagogy of this Project

There are many advantages of using the e-learning object for this cognitive development-based pedagogy. They are as follows:

- The e-learning object was designed by using the cognitive artefacts for information reconstruction, and presented in multisensory and interactive style. This strategy extends the possible types of information available to learners. These information visualization techniques have been seen to be effective for converting abstract concepts to be more understandable and manageable forms to learners (Brent et al., 2014; Lew, Sebe, Djeraba, & Jain, 2006; Ware, 2013).
- As discussed in Section 3.4 and Figures 3.3 (a)–(k), the strategy of using an animated flowchart to conceptually map the program semantics to the coding process, with the values of all essential runtime variables clearly visible so as to show the logical status of a program during runtime, can help students fill their

mental gaps between the syntax-free program semantics and the syntax-oriented coding process. It is a crucial step to facilitate students' cognitive presences in learning the program logics and semantical meanings (Fidge & Teague, 2009; Nguyen, 2008).

- The e-learning object provides an effective strategy to enable students to learn semantics while temporarily ignoring the syntax of a programming language. It helps first-time learners escape the trap of overfocusing on the programming language itself, while overlooking the program logic (Fidge & Teague, 2009). This strategy can also increase students' program walk-through ability and yields a better understanding of its meaning; improve their capacity of debugging; and provide meaningful information to them in explaining the program instead interpreting meanings driven by program key words (Farrell, 2007).
- The e-learning object is installed in the Blackboard system. This approach crucially provides students with an online, consistent environment to learn, while it is useful for using experimental research to study students' learning outcomes. It also does not require teachers to make too much effort in coping with the environmental differences between the research and control groups, and students have specific skills to access the e-learning object so that they can use it in out-of-class activities.

3.10 Evaluating the E-learning Object

Before using the e-learning object in the study module, a survey was conducted on teachers' professional comments to identify whether the e-learning object could be used in this research project. The process of this study is summarized in Table 3.8.

Table 3.8 Summary of procedural steps in evaluating the e-learning object

Step		Description
1.	Design and develop the e-learning object	Develop the pilot version of the e-learning object.
2.	Install the e-learning object to the pedagogy	Install the e-learning object onto the Blackboard system provided by the institute, set up trial passwords for teacher use.
3.	Provide briefing session for teachers on the coverage, design and usage of the e-learning object	Conduct a 20-minute briefing on using the e-learning object via the Blackboard, with an hour-long Q&A session.

4.	Provide study period to teachers	Allow teachers a period of a fortnight to evaluate the e-learning object. A survey questionnaire was delivered to teachers after the trial period. They were invited to respond to the questionnaire after the trial period.
5.	Proceed with the survey	Send questionnaire to teachers and collect their comments.
6.	Data analysis	Analyse the data and decide whether the e-learning object could be used in the pedagogy for future studies.

This survey concerned four major attributes of technology-based learning model design: ‘Pedagogical Issues’; ‘Content Design’; ‘Delivery Strategy’; and ‘Layout Design’. They were defined based on the key points provided in the ADDIE framework (Branch, 2010, pp. 151–152), as discussed in Chapter 2, and also are referenced the community of inquiry survey instrumentation for cognitive tool development for teaching and learning (Nurazian, Suzana, Haslizatul, & Ismassabah, 2007), and a standard questionnaire relating to learning effectiveness defined in ‘Reflective Faculty Evaluation: Enhancing Teaching and Determining Faculty Effectiveness’ (Jossey-Bass Higher and Adult Education Series) by Centra (1993). The rationale of defining these key points are discussed in the follows:

- Pedagogical Issues: Theoretically, most of the major features of ADDIE are related to technology-based pedagogy. For example, the key point ‘Design’ in ADDIE suggests that a cognitive model needs systematically to specify the learning objectives. The ‘Analysis’ recommends that the learning goals need to be identified. ‘Implementation’ suggests the integration of technology needs to enable the scaffolding to support cognitive processes.
- Content Design: The key point ‘Develop’ in ADDIE suggests that a cognitive model needs to be generated with enough structured lesson material, and that the media needs to be supported by any form of cognitive artefact (e.g. simulation, computer-based instructions) (Branch, 2010, p. 84). This point can link to the focus of using artefacts in the e-learning object’s content design (Soto, 2013).
- Delivery Strategy: The key point ‘Analysis’ in ADDIE suggests that the cognitive model needs to consider the appropriate delivery strategy and environment (Branch, 2010, p. 24).

- Layout design: The key point 'Design' in ADDIE means that the cognitive aspect needs to have a good appearance and a friendly-user interface graphically, in respect of these objectives (Branch, 2010, p. 84). It directly relates to how the layout of the learning object supports the cognitive processes (Jong, 2010).

The Survey

Apart from some statements concerning teachers' background, the questionnaire contained 14 statements and three open-ended questions as illustrated in Tables 3.9 and 3.10. The response type of these statements was a Likert scale of five points where 1='Strongly Disagree'; 2='Disagree'; 3='Neutral'; 4='Agree'; and 5='Strongly Agree'. Cronbach's alpha reliability coefficient based on teachers' responses was 0.907, which is a strong figure of reliability.

Table 3.9 Statement design in the questionnaire of teachers' survey

Statement	
Pedagogical Issue	
1	The e-learning object provides a good strategy to teach the program.
2	The e-learning object encourages students' involvement.
3	The e-learning object helps conceptually to map the semantic concept to coding process.
Content Design	
4	The topics in the e-learning object are significant to introductory programming.
5	The topics in the e-learning object are difficult to learn.
6	The topics in the e-learning object are difficult to teach.
7	The e-learning object suits its use as an instructional tool in introductory programming.
8	The e-learning object can be used with course materials.
Delivery Strategy	
9	Do you agree that e-learning object delivery with Blackboard and integrating it into the course material is an effective approach?
Layout Design	
10	The layout of the e-learning object is easy to follow.
11	The e-learning object can increase students' learning interest.
12	The e-learning object is well organized.
13	The e-learning object can attract students.
Overall	
14	Overall, I believe that the e-learning object can facilitate teaching and learning.

Table 3.10 Open-ended questions in Part B of the teacher questionnaire

	About the e-learning object, do you have any further comments on the following aspects?
i	Pedagogical issues
ii	Content design
iii	Delivery strategy
iv	Layout design

There were 12 teachers in this study, invited by the Hong Kong Institute of Education, School of Counting and Professional Education (HKIED SCPE), the Hong Kong Baptist University, School of Continuing Education (HKBU, SCE) and the Hong Kong Vocational Education (HKIVE), Sha Tin Campus. They all specialized in computer programming and had experience in teaching several programming languages, including the C/C++, Java, Flash Action Script, Java Script, JSP, PHP, Pascal and SQL, with a range of three to over 10 years' teaching experience. Although there were only 12 teachers involved in this survey, as it is a 'pre-study' with the aim of finding out whether any improvement to the e-learning object is necessary, teachers were only taking the role of consultant. In this sense, the involvement of this number of teachers can still be seen as adequate.

Survey Responses

All teachers responded to the survey; their backgrounds and comments are summarized in Tables 3.11 and 3.12.

Table 3.11 Summary of personal details of teacher subjects (n=12)

Particulars	Response type	Frequency
Teaching experience in IT & computing and computer science	4–6 years	2
	6–10 years	2
	over 10 years	8
Experience of teaching program language(s)	Java	12
	C/C++	12
	ASP.Net	7
	Visual Basic	8
	Others (e.g. SQL, PHP, HTML, etc.)	12
Experience of using technology to facilitate learning	Yes	10
	No	2
If having used it, the feeling of helpfulness	Very much	7
	A little	3
	As normal	2

Table 3.11 shows that there are eight teachers having more than ten years' experience, while for the remaining four, two have more than six years, and the other two have more than six years' teaching experience in IT, computing and computer science. All of them have to teach C/C++ and Java program languages, and most of them have experience of using technology and found it useful in teaching computer programming. Only two out of 12 teachers said there was no difference between use and non-use.

Table 3.12 Teachers' responses to statements

Statement		SDA	DA	N	A	SA	Mean (SD) (n=12)
Pedagogical Issue							
1	The e-learning object provides a good strategy to teach the program.			5	6	1	3.67(0.65)
2	The e-learning object encourages students' involvement.			2	6	4	4.00(0.74)
3	The e-learning object helps conceptually map the semantic concept to the coding process.		1	4	6	1	3.25(0.97)
Content Design							
4	The topics in the e-learning object are significant to introductory programming.			3	6	3	4.00(0.74)
5	The topics in the e-learning object are difficult to learn.			2	5	5	4.25(0.75)
6	The topics in the e-learning object are difficult to teach.			4	4	4	4.00(0.85)
7	The e-learning object suits use as an instructional tool in introductory programming.			2	5	5	4.25(0.75)
8	The e-learning object can be used with course materials.			5	5	2	3.75 (0.75)
Delivery Strategy							
9	Do you agree that e-learning object delivery with Blackboard and integration into the course materials is an effective approach?			3	7	2	3.92(0.67)
Layout Design							
10	The layout of the e-learning object is easy to follow.				6	6	4.50(0.52)
11	The e-learning object can increase students' learning interest.			3	6	3	4.00(0.74)
12	The e-learning object is well organized.			3	4	5	4.17(0.83)
13	The e-learning object can attract students.			4	6	2	3.83(0.72)
Overall							
14	Overall, I believe the e-learning object can facilitate teaching and learning.			2	6	4	4.17(0.72)

Table 3.12 shows the outcome of teachers' responses to the statements. Statements 1–3 concern pedagogical issues. For the outcomes, one teacher strongly agreed, and six teachers agreed that the use of an e-learning object is a good teaching strategy. The mean of this question is around the mid-point. However, it is of concern that five teachers gave a neutral response. Therefore, the result cannot be identified as strongly indicated. The result of Statement 2 is better than Statement 1. It concerns the topic design of the e-learning object. Ten of the 12 teachers indicated that the topics of the e-learning object are difficult. The mean is 4.00 and the result is strongly positive. Statement 3 evaluates whether the learning object helps to map concepts, from semantics to the coding process. A total of seven teachers strongly agreed or agreed. However, four teachers responded neutral, and the mean is 3.58, just past the mid-point, and just slightly more than half of the teachers felt that the e-learning object helps to map concepts. This is not strongly indicated.

Five statements (4–8) evaluated the content design of the e-learning object. In statement 4, of 12 teachers, nine teachers agreed or strongly agreed that the topics of the e-learning object are significant. The mean of this statement is 4.00. It may be concluded that most teachers think that the topics in the e-learning object are difficult. Statements 5 and 6 concern whether the content in the e-learning object is difficult to teach and learn. The outcome of Statement 5 is strongly positive, as 10 teachers strongly agreed or agreed. The mean of this statement is greater than 4.25. Similarly, in Statement 6 eight teachers strongly agreed or agreed that the content in the e-learning object is difficult to learn. The mean is 4.00 and may be concluded to be a positive result. In Statement 7, five teachers strongly agreed, and five teachers agreed that the content of the e-learning object is suitable for introductory programming. The mean is 4.25 and is strongly positive. The last statement had a positive result, that the e-learning object can be used with module material. The responses show that two teachers strongly agreed, and five teachers agreed. However, considering that the mean is 3.75, just greater than neutral, the views of the teachers are not consistent.

Regarding the delivery strategy, nine teachers strongly agreed, or agreed with Statement 9 that the e-learning object delivery via Blackboard is effective. The mean of 3.92 indicates that it is an effective delivery strategy.

The layout design was evaluated by Statements 10–13. In Statement 10, six teachers strongly agreed, and six teachers agreed that the e-learning object is easy to follow. With a mean of 4.50, the result is strongly positive. Three teachers strongly agreed, and six teachers agreed with Statement 11 that the e-learning object can increase students' interest. The mean of this statement is 4.00, so most teachers believed that students were interested in using the e-learning object to learn. Statement 12 evaluated if the e-learning object is well organized, and nine teachers strongly agreed or agreed. As the mean is over 4.17, the arrangement of the topics in the e-learning object is appropriate for use. Statement 13 concerns whether the e-learning object can attract students. The outcome is positive but not strongly conclusive, as only two teachers strongly agreed and six teachers agreed, while four teachers responded neutral. The last statement obtained teachers' comments on the overall design of the e-learning object. Four teachers strongly agreed, and six teachers agreed that the e-learning object can facilitate learning. The mean of this statement is 4.17, thus the result is highly positive. The result indicates teachers believe that the e-learning object can help students to learn.

In addition to the statements, teachers also provided useful information on the e-learning object through their responses to the open-ended questions. They are summarized in Table 3.13.

Table 3.13 Summary of responses to the open-ended questions (n=12). Note categorization by teachers' responses

Categorized comment (frequency)	
Pedagogical issues	Meet the criteria of use in introductory programming (7) Adding the concept of simple structure data type (4) Adding the pointer topic in C programming (4) It can facilitate learning but cannot be used independently (3) In pedagogical view, it is a good instructional tool (6)
Content design	The cognitive tool is particularly suitable for teaching computer programming (5) Can be used as a supportive tool only (2)
Delivery strategy	Appropriate as integrated into Blackboard (5) It needs a flexible strategy to integrate with the lesson materials (1)
Layout design	The design is interesting (7) Appreciate use of control button (5) The code-based example is helpful (6) Suggest providing textual explanation (2)

Others	Will use absolutely (1) Need more time on lecturing and it may bore students (2) Teachers suggest providing textual narrative (1) Teachers think the pedagogy may improve students' attitude to learning (2) E-learning object highly supports teaching (2) It is consistent with the pedagogy in teaching computer programming (2) It can facilitate learning but cannot be used independently (1) Pedagogy could be applied to both object and procedural programming (2) The use of code-based examples provides students with understanding of the control structure (5) Learning computer programming needs visual help. This pedagogy applies it (1)

As indicated, the responses in Table 3.11 can be concluded as highly positive. However, two responses need to be addressed: (i) two teachers said that the e-learning object could only be used as a supportive tool; and (ii) four teachers suggested adding the topics of the 'pointer' and 'data structure type' to the e-learning object. In respect of design features, the e-learning object is used with the course materials, supporting the cognitive processes in the scaffolding; it is not used independently to teach computer programming. Therefore, the design does not violate the use of an e-learning object as a major cognitive development supportive tool. As far as the second response is concerned, pointer is narrow as a topic in structural programming such as C language. However, one of the objectives of the research project is the wider use of this pedagogy across different introductory programming settings, such as procedural-first, object-first modules. In this sense, introduction of the pointer makes this pedagogy narrow by being used with only procedural-first modules.

3.11 Reflection of the Evaluation

As indicated, the key focus 'Pedagogy Issue' has slightly lower mean values ranging from 3.25 to 4.00 for all statements. However, most of the responses were over the mean values so it could be identified as positive in the pedagogical issues. For the other aspects, they were likely to be positive towards the design of the e-learning object. For example, the result of 'Content Design' has high-scale points, with the means reaching 4.00 and over in all statements. This indicates that most teachers agreed that the content in the e-learning object is both significant and difficult to learn. Furthermore, in using the e-learning object to improve the learning outcomes, teachers agreed it can match the chief aims of introductory programming in a normal setting. In their responses to the 'Delivery Strategy', teachers agreed that integrating the e-learning object to the course materials in the Blackboard system is an effective way of delivery.

For the ‘Layout Design’, the means of statements are ranged from 3.83 to 4.50. This can be identified as a strongly positive outcome. For example, the statements ‘The layout of the e-learning object is easy to follow’, ‘The learning objects are well organized’ and ‘The e-learning object can increase students’ learning interest’, having the mean of 4.17, and the last statement ‘The e-learning object can attract students’, show that teachers have a very positive overall impression of the e-learning object.

Teachers’ responses to the open-ended questions are also positive and indicate meeting the original design objectives. For example, regarding ‘Pedagogical Issues’, seven teachers indicated that the design met the criteria for use in introductory programming, and six teachers stated that the e-learning object is a good instructional tool. For the ‘Content Design’, five teachers stated that cognitive tool is useful for learning computer programming. For the delivery strategy, five teachers indicated that installing the e-learning to the Blackboard is appropriate. Regarding the ‘Layout Design’, the comments from teachers were highly positive, specifying that the design is interesting, and the code-based example is useful.

To conclude, most of the responses to the statements are positively indicated, and the open-ended questions are consistent with the statements’ responses. Therefore, by this evaluation, it is believed that the e-learning object is viable as an instructional tool in a normal setting for introductory programming. It meets the criteria of using it in this research project.

3.12 Summary

The conceptual framework of using the e-learning object is defined by extending the pedagogical framework achieved in Table 2.7 to identify the parts of the learning phases that need to be supported by using the e-learning object. This work is shown in Table 3.1, in which it is found that the e-learning object mainly supports the cognitive processes of ‘Repetition & Coding’ and ‘Coding & Storing’ for programming instructional activities of ‘Program Design’ and ‘Program Implementation’. These cognitive processes help students attainment in ‘Knowledge

Accretion' and 'Knowledge Understanding' to prepare for learning advance programming. This cognitive attainment meets with the needs for novice learners in computer programming.

The design of the e-learning object incudes a set of artefacts in visual imagery based on the fact it provides advantages in facilitating information reconstruction, and allowing users to absorb the information in manageable and understandable style. The types of visual imagery are illustrated in Table 3.6, which is the content creation and includes input style, human interface, delivery structure and major technology.

The context designed in the e-learning object is decontextualized from the curriculum of introductory programming, and forms a learning domain of program controls. To be successful for learning, this context needs to be highly supported by cognitive processes. Therefore, it is appropriate to use the e-learning object. This context then is granularity as 31 learning objects, and each of them focuses on a finer topic of program control. They are defined as primary objects, nested objects, and complex objects, as presented in Table 3.4. For each learning object, the cognitive processes are based on two main parts: an animated flowchart and a code-based example. They can be controlled by both the teachers and students. This design provides flexibility in using the learning objects in their own way. A typical process of a learning object is provided in Figures 3.4 (a)–(k). The e-learning object is used together with the course materials accessed from a link provided in the lecture notes. It is illustrated in Figures 3.5 (a) and (b). This approach can provide a consistent environment for students in the research and control groups with the only difference being whether the lecture notes are linked or not linked to the e-learning object.

At the pedagogical level, the e-learning object supports all the three major learning processes included in the scaffolding of this cognitive development-based pedagogy: (i) facilitate the cognitive development on learning program semantics; (ii) help students conceptually mapping their program semantic knowledge to the coding process; and (iii) provide exercises to enhance students' learnt knowledge. The relationship of these learning processes to this cognitive development-based pedagogy through the scaffolding functions is presented in Figure 3.6. It shows that the learning processes are designed based on three scaffolding functions including conceptual scaffolding, strategic scaffolding, and metacognitive scaffolding.

The major advantages of using the e-learning object can be summarized as follows: (1) it helps learners to establish a mental model of the learning context and facilitate information reconstruction; (2) it can provide instructors with a flexible learning environment by isolating the semantics and syntax issues to avoid cognitive overload; and (3) the design of the animated flowchart and code-based example helps students to fill their mental gap between the knowledge of program semantics by conceptually mapping this knowledge to the coding process.

A study on the e-learning object was performed to know whether it meets the requirements of use in this research project. It applied a survey that was designed by referencing the key points from the ADDIE framework, the community of inquiry survey instrumentation for cognitive tool development for teaching and learning, and a standard questionnaire relating to learning effectiveness. The study involves using 14 statements and three open-ended questions. They were designed to focus on four major key focuses: ‘Pedagogical Issues’; ‘Content Design’; ‘Delivery Strategy’; and ‘Layout Design’. For the outcome of this study, the responses to the statements and open-ended questions were likely positively indicated in all key focuses. Particularly, in the content and layout design, most related statements present high-value of mean scores reach or over 4.00. For the open-ended questions, the responses from teachers are generally consistent with all statements’ responses. With this outcome, it could be concluded that the e-learning object is highly appropriate for use in this research project. Upon to identify the e-learning object to be a major learning tool to support the pedagogical approaches defined in this project, it needs an empirical study to determine whether the e-learning object carries a positive learning outcome. This design of the research methodology is discussed in Chapter 4.

Chapter 4 Research Design and Methodology

4.1 Introduction

Since the outcomes on evaluating the e-learning object used for this research project are positively concluded in Chapter 3, this chapter changes the focus onto research methodology. This research project includes two stages of study. Stage 1 focuses on students' learning outcomes on studying modules in year 1 and 2, namely 'Programming with C/C++ Language' and 'Introduction to Java Programming' respectively. These modules were offered in The Hong Kong Institute of Education[2], School of Professional and Continuing Education (SCPE HKIED).

The objective of the stage 1 study is twofold. The first is to evaluate whether this cognitive development-based pedagogy significantly provides better learning outcomes to students, and the second is if a pedagogy focusing on facilitating cognitive developments for learning program semantics will be restricted to learning a particular programming language or not. Therefore, to fulfil the second part of the objective, this cognitive development-based pedagogy was only used in the year 1 module 'Programming with C/C++ Language' but did not use this pedagogy in the year 2 module 'Introduction to Java Programming'. While the organization of research and control groups are unchanged throughout the two years, therefore this approach could be used to evaluate whether students in the research group still had better learning outcomes than the control group in year 2 'Introduction to Java Programming' module.

Upon the pedagogy defined in this project is possible to improve students learning performance, it also requires knowing whether this outcome is connected to improvement of the cognitive presences on the learning context. For this reason, the study in stage 1 also focuses on three indicators of the cognitive outcomes on a series of cognitive processes: (1) students' attitudes toward learning computer

---

[2] Renamed the Education University of Hong Kong (EdUHK)

programming, (2) students' satisfaction with using the e-learning object, and (3) students' mental engagement with the pedagogical environment.

For the stage 2 study, it focuses on whether this cognitive development-based pedagogy matches teachers' knowledge and capacity for using it. This study provides an insight into whether this pedagogy could be generalized to wider use in introductory programming. As discussed in Chapter 2, among the diverse models that could be used for this objective, this research project applied the TPACK framework, as it is a multidimensional framework providing a systematic approach to evaluate teachers' knowledge, from primary forms, domain knowledge, and integrated as a framework.

Based on the objectives in these two stages of studies, six research questions are defined in this project. Stage 1 has five research questions Q1–Q5, where research questions Q1 and Q2 focus on students' learning outcomes in year 1 and year 2 respectively. Research questions Q3–Q5 concentrate on three indicators of cognitive outcomes while using this pedagogy to learn. The last research question, Q6, concerns of teachers' knowledge of using technology for teaching and learning, defined in the TPACK framework.

This research project applies the triangulation design methodology (Salkind, 2010), where the methodological and data triangulations have been considered in the studies of the research questions. This enables the interpretation of different views of this cognitive development-based pedagogy from students and teachers, matching the focuses of the research problems defined in this project, and proves beneficial in providing confirmation of findings, more comprehensive data, increased validity and enhanced understanding of studied phenomena.

The data collections included the use of quasi-experimental research, surveys, and rating scale and anchoring method survey using a seven-scale bipolar checklist. There are many data analysis methods used based on the appropriation for the research methods. They include the two-tailed paired t-test, two-tailed independent t-test, one-way repeated measure ANOVA, and descriptive method with mean analysis. They are summarized in Table 4.1.

Table 4.1 Summary of research methods for this research project

Stages	Research questions	Research methods	Data collection methods	Instrumenta-tions	Data analysis methods
1	Q1	Quasi-experimental	Quantitative	Pre-test and post-test (C/C++ programming)	Two-tailed paired t-test & independent t-tests
	Q2	Quasi-experimental	Quantitative	Post-test (Java programming)	Two-tailed paired t-test & independent t-tests
	Q3, Q4	Survey	Quantitative	Student questionnaire	Descriptive method and means analysis Two-tailed independent t-test
	Q5	Rating scale and anchoring method survey	Quantitative	7 scales rating bipolar anchoring specifications	One-way repeated measure ANOVA (GLM) & Two-tailed independent t-test
2	Q6	Survey	Qualitative Quantitative	Questionnaire open-ended questions	Descriptive method and mean analysis

4.2 Research Subjects

The stage 1 study included 51 students being assigned to either the research or control group according to the classes they belonged to. They were unchanged over the two study years. The research group learnt with the cognitive development-based pedagogy in the year 1 module ‘Programming with C/C++ Language’, but did not use this pedagogy in the year 2 module ‘Introduction to Java Programming’.

The stage 2 study concerned 52 experienced teachers specializing in Computer Science, Information Technology and Computer Studies. All teachers have good experience of teaching many programming languages. The research subjects for the two stages of study are summarized in Table 4.2.

Table 4.2 Summary of research subjects for the two stage studies.

* Note that the different number of subjects in years 1 and 2 is due to 'student attrition'

Attribute	**Number of research subjects**	
Year 1 Research (n=51)	Research group	Control group
Programming with C/C++ language	25 students	26 students
The ratio of males to females	21:4	23:3
Mean age of years	19.6 (SD=1.45)	20.1 (SD=2.20)
Year 2 Research (n=47)	Research group	Control group
Introduction to Java programming	23 students	24 students
The ratio of males to females	19:4	21:3
Mean age in years	19.9 (SD=2.13)	19.8 (SD=1.72)
Teacher study	52 IT/Computing teachers	

4.3 Research Questions

There were three main research questions, Q1, Q2 and Q6, defined in this study. Under the arch of research question Q1, three research questions, Q3, Q4 and Q5, were further defined. These research questions are listed as follows:

- Q1. Does the cognitive development-based pedagogy help students learn better in their first-year module 'Programming with C/C++ Language'?
- Q2. Do students who learnt with the cognitive development-based pedagogy in their first year still have better learning outcomes in their second-year module 'Introduction to Java Programming'?
- Q3. Is the exposure to learning with the cognitive development-based pedagogy associated with better attitudes towards learning computer programming?
- Q4. Do students feel that learning with the e-learning object provides them with a better approach to their learning?
- Q5. Is the exposure of learning with the e-learning object associated with better mental engagement in the pedagogical environment?
- Q6. Does the design of the cognitive development-based pedagogy match teachers' knowledge to use it, as suggested by the TPACK framework?

4.4 Procedural Steps

The procedural steps of this research are summarized in Figure 4.1. It provides an overview of the research concerning how it took place over two consecutive academic years in the stage 1 study. Each step of year 1 study concerned using the cognitive development-based pedagogy with the e-learning object, including investigation of students' learning outcomes, and the critical indicators for cognitive engagement on using this pedagogy. In year 2, this study was a continuous process to evaluate students' outcomes in their subsequence study. It is intended to provide evidence that the focuses of cognitive development in year 1 also provide positive effects on students' year 2 study, though they did not use the e-learning object in this year. The objective of this approach is to affirm that the capacity of this cognitive development-based pedagogy can be extended to learn different program languages. The teachers' survey was performed in year 2, defined as the stage 2 study, which was used to further understand teachers' capacity of applying technological pedagogy for their teaching. This outcome therefore is used to determine whether the scaffolding of this pedagogy meets their knowledge of using it. It is important using the outcome to know whether this cognitive development-based pedagogy could be generalized for wider use in introductory programming.

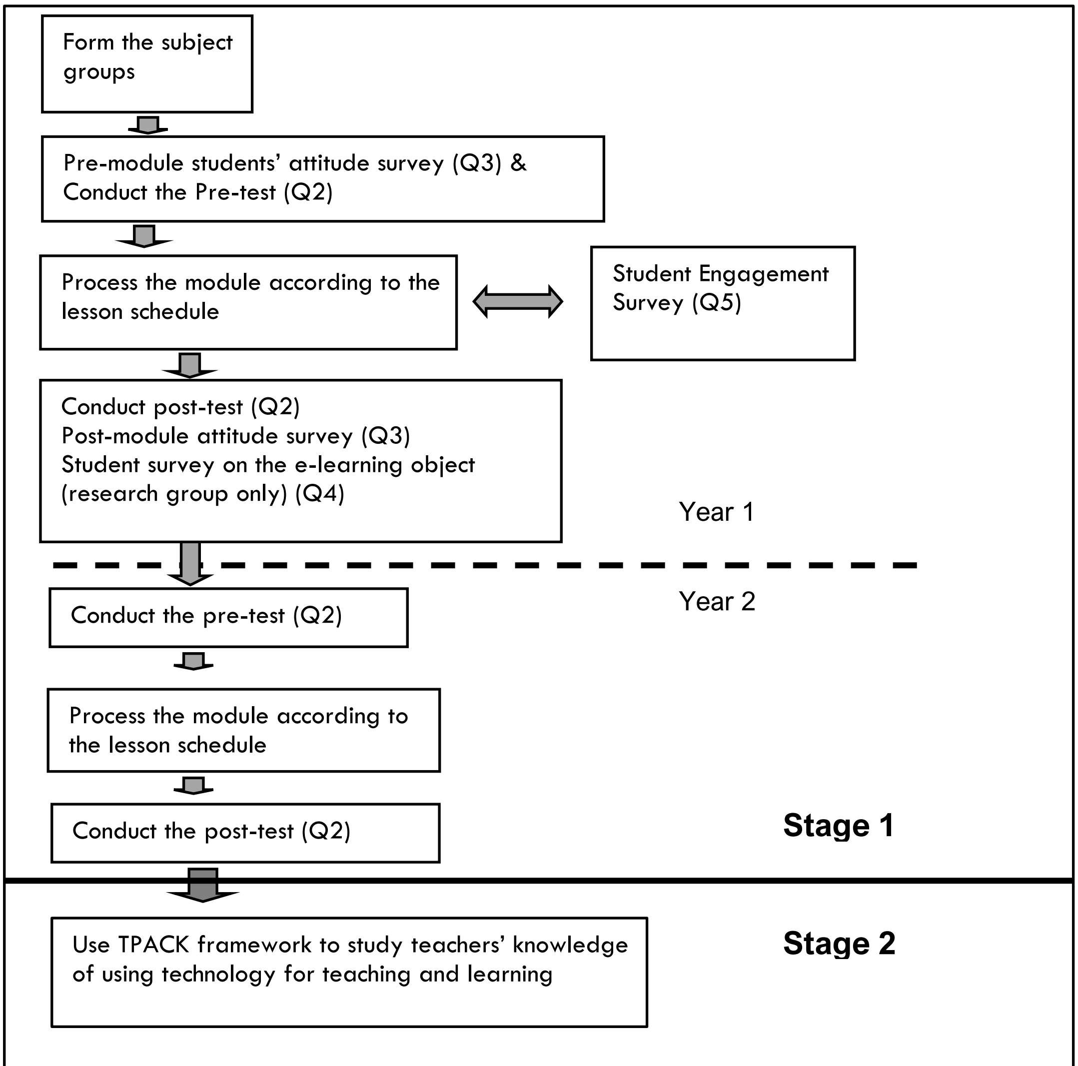


Figure 4.1 The procedural steps of this research project

4.5 Research Paradigm

This research project applies the triangulation design methodology (Salkind, 2010), where the methodological and data triangulations have been considered in the studies of all research questions. These quantitative and qualitative data help to obtain different but complementary views of the same research focuses that enrich the analysis process on the research questions, and provide a better understanding of the research problems. It brings together the differing strengths from the non-overlapping weaknesses of a single method to increase the validity and enhance the understanding of studied phenomena. Furthermore, the analysis with both the quantitative and qualitative evidence enables this study to interpret different views to provide a holistic picture of the research outcomes on using this cognitive development-based pedagogy.

Considering this research project focuses on the development of a cognitive development-based pedagogy, to evaluate its outcomes in respect of using it in different teaching classes, both the students’ and teachers’ views need to be considered. The use of triangulation design methodology matches the research problem as these different views contextually rely on the qualitative view from the teachers’ perspective of using this pedagogy, while there needs to be consistency with the students’ views on learning with this pedagogy. For example, using the e-learning object in the scaffolding, and their engagements on the learning environments evaluated in some research questions.

Besides, both types of data focused in this methodology could procedurally be collected together at the same time as the research. It could minimize the interruption of course progression in the research group, therefore being consistent with the pedagogical approach and class activities in the control group. These different types of data can also be analysed independently to evaluate the association of each data group, and used to validate and ascertain if they are complementary to each other by measuring the consistency of the qualitative and quantitative outcomes (Creswell, 2009).

The data collection methods for those research questions include the use of pre-test and post-test, surveys, self-evaluation checklist, and questionnaires with Likert scale statements and open-ended questions. The quantitative data focused on research questions Q1 and Q2, from the pre-test and post-test outcomes, as it can provide a wide range of nuanced data and an explanation of complex realities (Day et al., 2013), and is appropriate to test the research questions by focusing on the summative outcome upon the defined research variables (Lederman & Abell, 2014, p. 5). Quasi-experimental research was used, seeking to analysis students’ variance in the pre-test and post-test outcomes to test the significance of students’ progression (Lee & Smith, 2012).

The outcome of Q1 and Q2 is augmented, with quantized qualitative evidence from students’ responses on the surveys of Q3, Q4 and Q5. For these research questions, Q3 was used to investigate students’ change of learning attitudes in using the cognitive development-based pedagogy (Robert et al., 2009). Q4 is used to know whether the e-learning object satisfied students when they used it in their learning module based on some factors of using a technology-based cognitive learning tool (Brent et al., 2014; Jong, 2010; McGee, 2002; Pappas, 2014). A rating scale and anchoring method was used to explore Q5, (Flick, 2009, pp. 214, 216, 306) to focus on linear measures of students' learning processes through the whole module period (Field, 2012; Laerd.com, 2013b). This linear measuring approach allows use of the statistical technique ANVOA, as it is an effective way to repeatedly compare the variation of a research variable over time for understanding students' mental engagement with the learning environment (Lederman & Abell, 2014, 29).

Q6 seeks to obtain a wider range of teachers' views on using technology for their teaching processes by examining with the TPACK framework. It focuses on the quantized qualitative data with Likert scale statements and open-ended questions. Open-ended questions in this research question could be used to complement teachers' responses on the statements, and also provide a wider range of teachers' views on using technology-based pedagogy, which may not be considered in these statements.

With regard to the data analysis of these research methods, this research project mainly applied the two-tailed, paired t-test and independent t-test to compare the variance of some quantitative data at two different points in time, such as the change of learning attitudes between the students' pre- and post-module outcomes on research question Q3, and the means scores from the pre-test and post-test result in research questions Q1 and Q2. Moreover, this project also uses the descriptive with mean analysis method to analyse the quantized qualitative data such as students' satisfaction with using the e-learning object concerned in research question Q4.

Table 4.3 presents an overview of the research, data collection and data analysis methods applied in this research project, and the detail of the questionnaire design is discussed with the concerned research question individually in Section 4.6.

Table 4.3 Summary of the research, data collection and analysis methods

Stages	Research questions	Research Method/ Data collection methods	Data Analysis Methods
1	Q1	Quasi-experimental methods Quantitative with directly comparing the scores from pre-test and post-test papers	Two-tailed paired t-test & independent t-tests
	Q2	Quasi-experimental method Quantitative with directly comparing the scores from pre-test and post-test papers	Two-tailed paired t-test & independent t-tests
	Q3	Survey methods Quantitative data	Two-tailed paired t-test & independent t-tests
	Q4	Survey methods Quantitative data	Descriptive method and means analysis
	Q5	Rating scale and anchoring method survey Quantitative of qualitative evidences	One-way repeated measure ANOVA (GLM) & Two-tailed independent t-test Descriptive method and mean analysis
2	Q6	Survey method Quantitative and is justified by qualitative comments	Two-tailed paired t-test & independent t-tests

4.6 Research, Data Collection and Analysis Methods for Individual Research Question

This section provides detail discussion on individual research question from Q1 to Q6 respectively, as presented in Table 4.3. It includes the use, and reason of the research, data collection and analysis methods for different research questions.

<u>Q1. Does the cognitive development-based pedagogy help students learn better in their first-year module ‘Programming with C/C++ Language’?</u>

This research question applied a quasi-experimental research focusing on students’ learning outcomes on their year 1 module ‘Programming with C/C++ Language’. While the research group used the cognitive development-based pedagogy to learn, and the control group did not use it, this method can effectively control students’ learning process among the groups and compare their outcomes in a consistent way. These outcomes were determined by comparing the differences between the mean scores from a pair of identical pre-test and post-test papers. It used two-tailed paired t-test and independent t-test to evaluate whether the learning process gives significant effects to both groups, and whether the research group has significantly better learning outcomes than the control group at the end of the module.

The process of this research question is illustrated in Figure 4.2, and the test paper used in pre-test and post-test is attached in Appendix A. It had been commented on by the teachers who involved in evaluating the e-learning object. There was no particular concern on the standard and meaning of questions in the paper received.

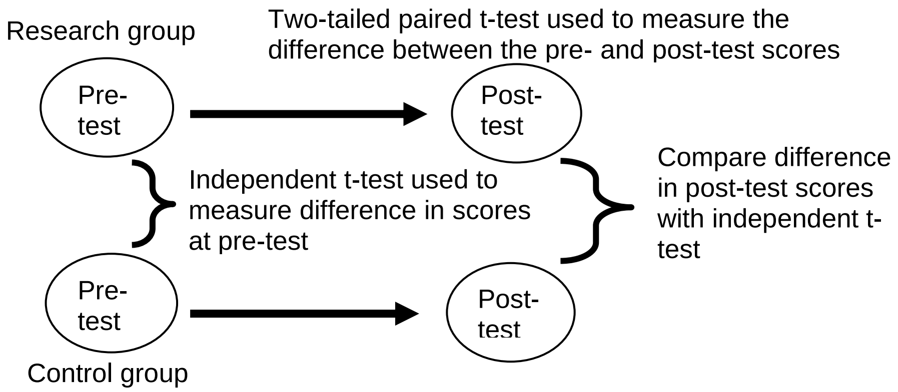


Figure 4.2 Process of quasi-experimental method for Research Questions Q1 and Q2

<u>Q2. Do students who learnt with the cognitive development-based pedagogy in their first year still have better learning outcomes in their second-year module ‘Introduction to Java Programing’?</u>

This research question also used the quasi-experimental research based on the advantages discussed in Q1. However, as Q2 is focuses on finding out whether the research group learning with the cognitive development-based pedagogy with the e-learning object in year 1 still had a better outcome in year 2 even though it did not learn with this pedagogy, therefore in this year, both the research and control groups did not use the cognitive development-based pedagogy to learn. Considering as the student groups were unchanged, and the research group only use the cognitive development-based pedagogy in year 1, while they were still have significantly better outcome in year 2, it could be conclude a pedagogy designed with cognitive development-based can give benefits on other programming language. The outcome of this research question can be used as evidence that a cognitive development-based pedagogy is not restricted by program language.

The test paper used for pre-test and post-test is designed with Java program language and is attached as Appendix B. Similarly, it had been commented on by teachers and there was not particular concern in the content and standard of this paper.

<u>Q3. Is the exposure to learning with the cognitive development-based pedagogy associated with better attitudes towards learning computer programming?</u>

This research question concerns of whether this pedagogy is connected to the improvement of cognitive presence, by studying whether students' learning attitude can be improved by learning with this pedagogy, as learning attitude is a major factor indicative of cognitive outcome through a series of learning processes (Brant, 2013; Buldua & Buldu, 2010).

It used survey method and the data collection is at the start and end of the module with a questionnaire. This questionnaire includes two parts: Part A has five statements which were designed to collect students' general background to identify if they are first-learners of computer programming. Part B contains 10 Likert scale statements with a five-point response type: 1='Strongly Disagree', 2='Disagree', 3='Neutral', 4='Agree' and 5='Strongly Agree', focusing on three attitudinal key factors of 'Confidence'; 'Importance'; and 'Liking'. They were defined by referencing 'The Jasper Project: Lessons in Curriculum, Instruction, Assessment, and Professional Development', published by Lawrence Erlbaum (Cognition and Technology Group at Vanderbilt, 1997), with modifications from the pool of general learning computer attitude questions (Mayer, 2013; Nurazian et al., 2007). The questionnaire had been commented on by teachers involved in evaluating the e-learning object. There was no major problem reported on the meaning and description of the statements.

As the statements were descriptively and meaningfully identical in pre- and post-module questionnaires, the variances between students' responses can be analysed by two-tailed paired and independent t-test to identify whether students' attitudes significantly changed.

The statements defined for this research question are presented in Table 4.4 (Part A) and 4.5 (Part B).

Table 4.4 Statements in Part A of Research Question Q3

1	What is your gender (male | female)?
2	Did you learn computer programming before this module, including learning by yourself (yes | no)?
3	If yes in Question 2, state what program language(s) you have learned.
4	Describe your skill in using computer (okay | good | very good).
5	Describe your skill in using the Internet (okay | good | very good).

Table 4.5 Statements defined in the attitude questionnaire

No.	Pre-module statement	Post-module statement	Key focus
1	Learning computer programming is fun	Learning computer programming is fun	liking
2	I look forward to learning computer programming	I look forward to learning advanced computer programming	liking
3	Learning computer programming is important to my study	Learning computer programming is important to my study	importance
4	I feel confident in learning computer programming	I feel confident in learning computer programming	confidence
5	I want to be a professional programmer	I want to be a professional programmer	importance
6	I feel nervous when I know I need to learn computer programming	I feel nervous when I know I need to learn computer programming	confidence
7	I am interested in learning computing	I am interested in learning computing	liking
8	I am not afraid of challenges	I am not afraid of challenges	confidence
9	I don't have any difficulty in using a computer	I don't have any difficulty in using a computer	confidence
10	I feel happy as I can learn computer programming in this course	I feel happy if I can learn advanced programming in other courses	importance

Q4. Do students feel that learning with the e-learning object provides them with a better approach to their learning?

Similar to research question Q3, Q4 also focused on a cognitive presence factor of students' satisfaction with using the e-learning object. It used a survey to collect students' comments on using this learning tool at the end of the module. There were 18 Likert scale statements with a five-point response type: 1='Strongly Disagree', 2='Disagree', 3='Neutral', 4='Agree' and 5='Strongly Agree'. They were designed by referencing some critical factors of using a cognitive learning tool to learn, and was focused on four key factors: 'Design'; 'Effectiveness'; 'Helpfulness'; and 'Motivation' (Brent et al., 2014; Jong, 2010; McGee, 2002; Pappas, 2014). The questionnaire was evaluated by the teachers who also commented on the questionnaire used in research question Q3. There was no major concern on the statements' descriptions and meanings. These statements are listed in Table 4.6.

Table 4.6 Statements in the student survey questionnaire

No.	Statement	Key focus
1	The e-learning object is helpful	helpfulness
2	With the e-learning object I feel the instructions are precise	effectiveness
3	I am frequently involved in the learning activities	motivation
4	I always feel the lessons are effective	effectiveness
5	I like to use similar cognitive tools in other programming modules	effectiveness
6	The learning process suits my own style	design
7	The e-learning object attracts me to learn	design
8	I feel comfortable with the e-learning object	design
9	The e-learning object supports my needs	helpfulness
10	I am happy to use the e-learning object to learn	motivation
11	I feel easy with the animated flowchart	design
12	I feel easy with the animated code-based example	design
13	The animation helps me to better understand the program controls	helpfulness
14	The animation is interesting	design
15	I often use the e-learning object in class exercises	motivation
16	I enjoyed the lesson	motivation
17	I could finish the class activities quickly	helpfulness
18	Overall, I felt the lesson is effective	overall

<u>Q5. Is the exposure of learning with the e-learning object associated with better mental engagement in the pedagogical environment?</u>

This research focused on whether students can learn while being mentally engaged to the pedagogical environment created by the e-learning object, and whether this can be transferred to an alternative learning way instead of restricted by this pedagogical environment. Considering a pedagogical environment can be varied depending on the learning contexts in different times, this study was conducted throughout the whole module with collecting students' responses to four milestones which were just finishing the topics of selective, iterative, array and functional controls. The variances between these milestones were analysed to identify whether students could stably engage with the pedagogical environment mentally. The general process of this study is illustrated in Figure 4.3, where M1, M2, M3 and M4 represent the milestones of selective, iterative, array and functional controls of this study.

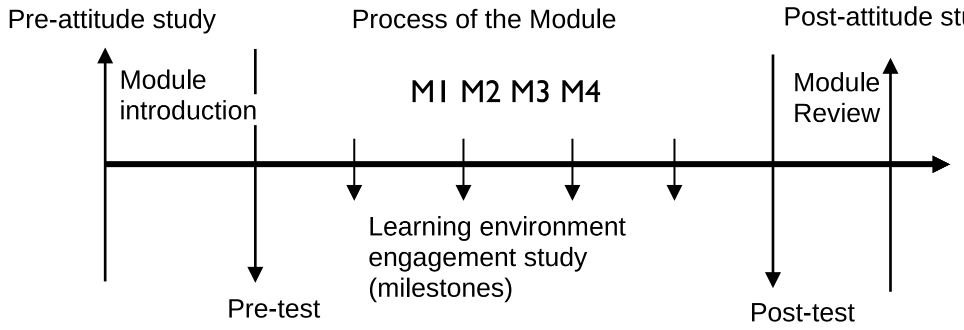


Figure 4.3 Milestones in the rating scale and anchoring method Survey

Note: This figure includes the pre-test–post-test, delivery of attitude

This research question used a rating scale and anchoring method survey to collect the quantized qualitative data regarding students' feelings based on four mental specifications: 'Boring – Stimulating'; 'Did Not Learn Much – Learned Much'; 'Not Engaged in Learning Process – Engaged in Learning Process'; and 'Not Much Work Done – Much Work Done', i.e. those which are deeply concerned with a pedagogical environment of computer programming (Tsai et al., 2013) and how cognitive presences can effect this environment (Garrison, 2006; Garrison & Cleveland-Innes, 2005). These specifications were defined in a seven scaling bipolar anchoring list. It is effective and practical for evaluating computer courses (Robert et al., 2009, pp. 248–254), as the bipolar rating scale allows students to make generic ratings without being quizzed or spending time completing a long survey by just ranking the specifications from 1 to 7 against the two polarities at every milestone. It can minimize the disruption to students.

Data was collected by the checklist and analysed with one-way repeated measure ANOVA. This data analysis method provides an effective strategy to repeatedly compare the variation of a variable from time to time (Field, 2012; Laerd.com, 2013b). The variances of these specifications therefore could be used to evaluate whether students could stably mentally engage with the pedagogical environment over time, rather than be restricted by learning contexts in different times. To find out whether the research group was better than the control group, there was a two-tailed independent t-test used to analyse whether the research group demonstrated a greater mental engagement with the environment established by this cognitive development-based pedagogy than the control group at all milestones.

Q6. Does the design of the cognitive development-based pedagogy match teachers' knowledge to use it, as suggested by the TPACK framework?

Research question Q6 focused on teachers regarding their knowledge of using technology for teaching and learning. As the teacher is the major implementer of a pedagogy (Sahin & Kelesoglu, 2011), this study intends to find out whether the design of this cognitive development-based pedagogy meets their knowledge, and therefore can be used to identify whether it can be accepted by teachers, or whether this new pedagogy is restricted by teachers' knowledge while using it in their teaching approach.

There were 25 Likert scale five-point response type statements, and two open-ended questions used to collect data. The questionnaire also was designed by referencing a research project evaluation in Teacher Teaching for the Future (TTF) (Jamieson-Proctor et al., 2013); and a study of works of literature about using ICT for classroom learning (Chai, Koh, & Tsai, 2010); and a pilot study that evaluated the reliability and validity of instruments for longitudinal studies on teachers' development on TPACK (Schmidt et al., 2009).

These studies indicate that the Technological Pedagogical and Content Knowledge (TPACK) framework provides a systematic approach to evaluate teachers' capacity of using technology for teaching and learning based on three primary forms of 'Technological Knowledge (TK)', 'Content Knowledge (CK)' and 'Pedagogical Knowledge (PK)', and follows up these primary forms to assess teachers' domain knowledge of 'Technological Pedagogical Knowledge (TPK), 'Technological Content Knowledge (TCK) and 'Pedagogical Content Knowledge (PCK), and finally integrates those components to evaluate teachers' capacity of implementing a technological-based pedagogy for a subject area.

The quantitative and qualitative data collected by the questionnaire were analysed with the descriptive methods connected with different response types. The statements and open-ended questions are presented in the following. They are organized by primitive forms, domain knowledge and the concerns of the whole TPACK framework.

Technological Knowledge (TK)

Four statements are defined to evaluate the primary form of 'Technological Knowledge' concerning teachers' knowledge on using technology for educational purposes, as listed in Table 4.7(a)

Table 4.7(a) Statements to evaluate teachers' knowledge of the primary form of 'Technological Knowledge (TK)'

1	I consider my knowledge of using technology for teaching to be: Expert/ Very Good/ Good/ Needs Improvement/Poor
2	My skill at using online platforms (e.g. WebCT; Moodle) for teaching is: Expert/ Very Good/ Good/ Needs Improvement/Poor
3	I know about a lot of different technologies, including the up-to-date. Fully Agree/ Agree/ Neutral / Not Agree/ Fully Disagree
4	I can learn how to use technology easily. Fully Agree/ Agree/ Neutral / Not Agree/ Fully Disagree

Content Knowledge (CK)

Three statements are defined to evaluate the primary form of 'Content Knowledge', concerning teachers' knowledge of computer programming, as listed in Table 4.7(b).

Table 4.7(b) Statements to evaluate teachers' knowledge of the primary form of 'Content Knowledge (CK)'

5	My knowledge of computer programming is: Expert/ Very Good/ Good/ Needs Improvement/Poor
6	Learning computer programming is difficult for most students. Fully Agree/ Agree/Neutral/ Not Agree/ Fully Disagree
7	I keep learning new programming languages to enhance my knowledge. Always/Usually/Sometimes/Seldom/Never

Pedagogical Knowledge (PK)

Four statements are defined to evaluate the primary form of 'Pedagogical Knowledge' on their ability to adopt different pedagogies as listed in Table 4.7(c).

Table 4.7(c) Statements to evaluate teachers' knowledge of the primary form of 'Pedagogical Knowledge (PK)'

8	I can organize effective classes. Always/Usually/Sometimes/Seldom/Never
9	I can use different teaching approaches for students' individual needs. Always/Usually/Sometimes/Seldom/Never

10	I can understand students' learning difficulties during teaching. Always/Usually/Sometimes/Seldom/Never
11	I can use different sources, including online materials, to facilitate teaching. Always/Usually/Sometimes/Seldom/Never

Pedagogical Content Knowledge (PCK)

Four statements are defined to evaluate the domain knowledge of 'Pedagogical Content Knowledge', concerning the congruous blend in pedagogy, and accepting alternative ways for teaching computer programming, as listed in Table 4.7(d).

Table 4.7(d) Statements to evaluate teachers' knowledge of the domain of PCK

12	Learning computer program semantics is more important than syntax. Fully Agree/ Agree/Neutral/ Not Agree/ Fully Disagree
13	Teaching computer programming requires an emphasis on cognitive development. Fully Agree/ Agree/Neutral/ Not Agree/ Fully Disagree
14	I can select appropriate materials for teaching computer programming. Always/Usually/Sometimes/Seldom/Never
15	I can use different strategies to teach different topics in computer programming. Always/Usually/Sometimes/Seldom/Never

Technological Content Knowledge (TCK)

Two statements are defined to evaluate the domain knowledge of 'Technological Content Knowledge' concerning teachers' ability to use technology for a subject of computer programming, as listed in the Table 4.7(e).

Table 4.7 (e) Statements to evaluate teachers' knowledge of the domain of TCK

16	Using information technology to learn semantic concepts of programming is: Very Important/Important/Moderate/Less Important/Not Important
17	Using information technology to facilitate cognitive development in teaching computer programming is: Very Important/Important/Moderate/Less Important/Not Important

Technological Pedagogical Knowledge (TPK)

Three statements are defined to evaluate the domain knowledge of 'Technology Pedagogical Knowledge', concerning teachers' knowledge to integrate diverse technologies into their pedagogy and how they use it to effect the teaching, as listed in Table 4.7(f).

Table 4.7(f) Statements to evaluate teachers’ knowledge of the knowledge domain of TPK

18	Encourage students to use technology to improve learning performance is: Very Important/Important/Moderate/Less Important/Not Important
19	Keeping using up-to-date information technology to facilitate teaching is: Very Important/Important/Moderate/Less Important/Not Important
20	I can use information technology to facilitate teaching. Always/Usually/Sometimes/Seldom/Never

TPACK framework

Five statements are defined to evaluate teachers’ knowledge concerning the TPACK framework through an integrated view of all primary forms and domain knowledge. These statements address the main features of integrating the technology, pedagogy and content as a knowledge framework, such as teachers’ knowledge of using the e-learning object in the developed new pedagogy, focusing on teachers’ views of using the cognitive model with different dimensions to learn computer programming, such as using animations, syntax-free approach and applying concept maps to realize the abstract aspects of program controls to the coding process. These statements are listed in Table 4.7 (g).

Table 4.7(g) Statements to evaluate teachers’ knowledge of the TPACK framework.

21	I like to use technology-based tools to help students to learn program semantics. Fully Agree/ Agree/Neutral/ Not Agree/Fully Disagree
22	I believe animated flowcharts help students’ understanding of the programme’s semantics concept. Fully Agree/ Agree/Neutral/ Not Agree/Fully Disagree
23	I like to use the learning object to facilitate teaching. Fully Agree/ Agree/Neutral/ Not Agree/Fully Disagree
24	I like to use online learning to facilitate teaching. Fully Agree/ Agree/Neutral/ Not Agree/Fully Disagree
25	I am open-minded about using diverse technologies in my teaching. Fully Agree/ Agree/Neutral/ Not Agree/Fully Disagree

Open-ended questions

There are two open-ended questions used to collect teachers’ qualitative comments on the key features of teachers’ knowledge of technological, pedagogical, and learning contents, where they are not defined in the statements. The outcome of teachers’ responses to these open-ended questions can also be used to evaluate whether they are consistent with the statements' responses. These questions are listed in Table 4.8.

Table 4.8 Open-ended questions regarding the TPACK framework

Q1.Do you have any suggestions on teaching approach to effective students' learning program semantics?
Q2.Please list the crucial points when you consider using technology for teaching computer programming.

4.7 Summary

This chapter discusses the methodology of this research project. It includes the research subject, questions, procedural steps, research paradigms and the data collection and analysis methods. The procedural steps of this project include two major stages as indicated in Figure 4.1. Stage 1 focuses on students' learning performance in their two academic years' studies, while Stage 2 focuses on teachers' knowledge of using technology for teaching and learning. There were 51 students and 52 teachers involved in this study. The students were assigned to either the research group or a control group according to their registered classes. The research group used the cognitive development-based pedagogy to learn the module 'Programming with C/C++ Language' in year 1 while the control group learnt this module as the normal approach. However, in the year 2 module 'Introduction to Java Programming', neither the research nor the control group learnt with the cognitive development-based pedagogy. This approach is intended to evaluate whether a cognitive development-based pedagogy is restricted by program language or not.

There were six research questions posed in this project, where research questions Q1 and Q2 are the main research questions, as they concern students' year 1 and year 2 learning outcomes, respectively. Research questions Q3 to Q5 are developed under research question Q1 as they cover the critical indicators of cognitive presence. While Q3 focuses on whether the pedagogy helps students explore a better attitude towards learning computer programming, research question Q4 focuses on students' satisfaction with learning with the e-learning object. Research question Q5 examines students' mental engagement in the pedagogical environment. These research questions provide an insight into the cognitive presence promoted by this cognitive development-based pedagogy. Research question Q6 relates to the stage 2 study which focuses on teachers' knowledge of integrating technology into their used pedagogy. It is intended to identify whether this pedagogy is consistent with teachers' knowledge of using technology, to ascertain whether it could be adopted by most.

The research design is based upon an approach where quantitative data are enriched by the collection of qualitative data. The choice of method allowed the gathering of data from multiple perspectives. This enables the interpretation in this study of different views from different stakeholders for a holistic picture of the use of the pedagogy with the e-learning object in Hong Kong.

Diverse research methods are used in different research questions. Quasi-experimental research was used in research questions Q1 and Q2, while quantitative data were collected with pre-test and post-test papers, and analysed with two-tailed paired-test and independent t-tests. Research questions Q3 to Q5 were mainly used to evaluate students' behavioral comments on the indicators of cognitive presences. They are the students' satisfaction with using the e-learning object, their learning attitude towards computer programming, and their mental engagement with the pedagogical environment. They are used as evidence that the outcome of research question Q1 is connected with the improvement of the cognitive presences on the learning contents.

The research question Q3 used two-tailed paired and independent t-tests to identify students' change of learning attitude at the start and end of the module. Q4 used the descriptive method with mean analysis to analyse students' responses to the questionnaire. This is an effective data analysis method for questionnaire responses. Q5 is focused on students' mental engagement with the pedagogical environment. It applied the rating scaling survey method and the data were collected with a seven-scale with bipolar anchoring checklists, and was analysed by one-way repeated measure ANOVA and two-tailed independent t-test. These methods are commonly used to evaluate the variation of the variables over several points in time.

Research question Q6 used 25 Likert scale five-point response type statements and two open-ended questions. They were designed by referencing the TPACK framework. It provides a systematic approach to evaluate teachers' knowledge on a set of well-organized attributes on teachers' operational features of technology-based pedagogy. The collected data were analysed with descriptive methods concerning the responses type. The pre- and post-test papers for both year 1 and 2 studies, and the questionnaires are attached as appendices. A summary of the methodology for this research project is presented in Table 4.1 and 4.3, while the outcomes of the data analysis are the focus of the next chapter.

Chapter 5 Data Collection and Analysis

5.1 Introduction

The methodology of this research project has been discussed in Chapter 4. In summary, it applied quantitative methods, including experimental research, cross-sectional survey methods, rating scale and anchoring method survey. Data were collected with pre- and post-tests, questionnaires, and a checklist with four specifications on seven scales with a bipolar anchoring list. They were analysed with the two-tailed t-test, independent t-test, one-way repeated measure ANVOA.

This chapter reviews the process and outcome of the data analysis presented via SSPS software. It is a reliable statistic software package, and can be used to supports a wide range of data analytical methods. Instead of focusing on the findings and goals from these studies, this chapter mainly provides descriptions of individual research questions organized in the two stages study. Stage 1 contains five research questions, and involves 51 students, randomly assigned in the research and control groups. Research questions Q1 and Q2 were used to evaluate students' learning outcome in year 1 and 2, respectively upon the research group use the pedagogy designed in this project. In addition to students' learning outcomes, many research questions were also used to focus on some critical cognitive indicators. They include students' learning attitude towards learning computer programming, which is the focus of research question Q3, their satisfaction with using the e-learning object to learn and their mental engagement with the learning environment established by the e-learning object, which are the focus of research questions Q4 and Q5, respectively. Stage 2 contains one research question, Q6, which evaluated teachers' knowledge of using technology in their pedagogy.

This study involved 52 teachers experienced in the subjects of Computer Science, Computer Studies, and Information Technology. This study was defined based on the TPACK framework, with the goal of finding out if the presented pedagogy met with teacher's capacity for using it, and reveals whether the pedagogy would be restricted by different teachers' capacity.

5.2 Stage 1 Study

Stage 1 included five research questions Q1–Q5, focusing on students' performance in learning with the e-learning object-based pedagogy.

Profile of the Research Subjects

There were four statements defined in the pre-module attitude questionnaire focusing on students' background. The result is presented in Table 5.1.

Table 5.1 Demographic details of student subjects in the student survey (n=51)

Attribute	**Response Type / Frequency**
Is this your first programming module?	Yes : No
Research group (n=25)	22 : 3
Control group (n=26)	25 : 1
How many computer languages learned?	None learned: 47
	2 languages: 2
	3+ languages: 2
Skill in using a computer	Very Good: Good: Okay
Research group (n=25)	9 : 14 : 2
Control group (n=26)	10 : 16 : 0
Skill in use of Internet	Very Good: Good: Okay
Research group (n=25)	9 : 16 : 0
Control group (n=26)	13 : 13 : 0

As indicated in Table 5.1, most students can be classified as novice learners. There are 45 out of 51 students who stated that it was their first-time learning computer programming. Regarding their skill in using computers and the Internet, almost all responded, 'Very good' or 'Good'. This shows that they had no problem in using the e-learning object via the institute's hardware and the Internet setting. Table 5.1 also indicates that there is no significant difference in the background of the research and control groups. Therefore, it is believed that their responses to subsequent research questions would not be affected by their background.

Data Analysis of Individual Research Questions

<u>Q1. Does the cognitive development-based pedagogy help students learn better in their first-year module 'Programming with C/C++ Language'?</u>

This research question directly evaluated students' learning outcomes by using the quasi-experimental research method with a pair of pre-test and post-test papers. The data was analysed by two-tailed paired and independent t-test. As discussed in Chapter 4, these methods can effectively control the research variables in a consistent way such as students' learning outcomes in the research and control group. Therefore, it is reliable for studying if the research and control groups statistically have significant differences in their learning outcomes. The result is presented in Table 5.2(a).

Table 5.2(a) Results of two-tailed paired t-test for the pre-test and post-test mean scores on the research and control group in year 1 study. (Programming with C/C++ language)

	Paired Differences							
				95% Confidence Interval of the Difference				
	Mean	SD	Std. Error Mean	Lower	Upper	t	df	Sig. (2-tailed) α=0.05
Research group (n=25) Pre-Score – Post-Score	-54.0	11.500	2.3000	-58.747	-49.253	-23.478	24	.000
Control group (n=26) Pre-Score – Post-Score	-37.9	17.537	3.4394	-45.045	-30.878	-11.037	25	.000

Table 5.2(a) indicates that the differences in the pre-test and post-test in both the research and control groups are statistically significant, with the t-values of t(24)=-23.478, p<0.001, md=-54.0, and t(25)=-11.037, p<0.001, md=-37.9, respectively. This shows that the teaching process has a significant effect on both groups.

Table 5.2 (b) presents the responses of the two-tailed independent t-test. As indicated, the Levene's test shows that the F-ratio of pre-score is (F=0.384 p=0.538) and post-score is (F=1.428 p=0.229), and both are 'Equal Variance Assumed' and adjustment is not required.

Table 5.2(b) Result of independent t-test for the pre-test and post-test mean scores of the research and control groups in year 1 study (Programming with C/C++ language)

		Levene's test for equality of variances		t-test for equality of means					95% confidence interval of the difference	
		F	Sig.	t	df	Sig. 2-tailed α=0.05	Mean difference	Std. error difference	Lower	Upper
Pre-Score	Equal variances assumed	.384	.538	-.852	49	.399	-1.647	1.934	-5.535	2.240
Post-Score	Equal variances assumed	1.482	.229	3.180	49	.003	14.390	4.524	5.297	23.483

The figures in Table 5.2(b) show that the difference in the pre-test mean scores of the research and control groups is not statistically significant, where the t-value is (t(49)=-0.852, p=0.399 (>0.05), md=-1.647). While the result of the post-test has a t-value of (t(49)=3.180, p=0.003 (<0.05), md=14.390), which shows the research group's learning outcome is significantly better than the control group.

Therefore, summarizing research question Q1, the paired t-test indicates the learning process in both the research and control groups is effective, and the difference on the pre-test mean scores is not significant but significant in the post-test, which shows that the research group has a significantly better learning outcome in the year 1 C/C++ programming module than the control group due to the group learning with the e-learning object-supported pedagogy.

<u>Q2. Do students who learnt with the cognitive development-based pedagogy in their first year still have better learning outcomes in their second-year module 'Introduction to Java Programming'?</u>

This research question also used the quasi-experimental research to evaluate students' learning in their year 2 module, 'Introduction to Java Programming'. Their mean scores of the pre-test and post-test were analysed with two-tailed paired and independent t-tests. However, by design, neither the research nor the control group used the e-learning object to support their learning process. The result is presented in Table 5.3(a).

Table 5.3(a) Result of two-tailed paired t-test on the mean scores of the pre-test and post-test in the research and control groups for the module 'Introduction to Java Programming'

	Paired Differences							
				95% confidence interval of the difference				
	Mean	SD	Std. error mean	Lower	Upper	t	df	Sig. (2-tailed) α=0.05
Research group (n=23) Pre-Score - Post-score	-47.80	15.250	3.410	-54.937	-40.662	-14.017	22	.000
Control group (n=24) Pre-score – Post-score	-30.00	15.111	3.561	-37.514	-22.485	-8.423	23	.000

Table 5.3(a) indicates that the differences in the pre-test and post-test in both the research and control groups show a statistically significant difference, with the t-values (t(22)=14.017, p<0.001, md=-47.80), and (t(23)=-8.437, p<0.001, md=-30.00), respectively.

Table 5.3(b) presented the result of two-tailed independent t-test. The Levene's test indicates that for both the research and control groups 'Equal Variance is Assumed', with the F-values (F=2.312, p=0.137) and (F=0.019, p=0.765), respectively, and adjustment is not required.

Table 5.3(b) Result of two-tailed independent t-test on the mean scores of the pre-test-to-pre-test and post-test-to-post-test in the research and control groups for the module of 'Introduction to Java Programming'

		Levene's Test for equality of variances		t-test for equality of means					95% confidence interval of the difference	
		F	Sig.	t	df	Sig. 2-tailed	Mean difference	Std. error difference	Lower	Upper
Pre-test score	Equal variances assumed	2.312	.137	1.020	45	.314	3.0222	2.9621	-2.985	9.029
Post-test score	Equal variances assumed	.091	.765	4.111	45	.000	20.8222	5.0651	10.549	31.094

The result of two-tailed independent t-tests shows that the pre-test scores are (t(45)=1.020, p=0.314 (>0.05), md=3.022), and the post-test scores are (t(45)=-4.111, p<0.001 (<0.05), md=20.822). Statistically, this indicates that the difference in the pre-test scores is not significant, while the difference in the post-test scores is significant. The result of the research group is significantly better than the control group.

Therefore, in summary, the paired t-test indicates the learning process in both the research and control groups is effective, and the difference on the pre-test mean scores is not significant but it is significant in the post-test, so it can be concluded that the research group has a significantly better learning outcome in the year 2 'Introduction to Java Programming' module, even though it does not use the e-learning object-based pedagogy.

Q3. Is the exposure to learning with the cognitive development-based pedagogy associated with better attitudes towards learning computer programming?

This research question used cross-sectional survey at the points of pre- and post-module. As discussed, this research method is effective for collecting data at a point in time, therefore it can be used to measure changes in students' learning attitude through the learning process. The pair of questionnaires are purposefully identically designed with the Likert scale five-point response type: 1= ‘Strongly Disagree’; 2= ‘Disagree’; 3= ‘Neutral’; 4= ‘Agree’; and 5= ‘Strongly Agree’. The data were analysed by two-tailed paired t-test and independent t-test. The students’ responses were analysed by Cronbach’s alpha reliability coefficient. It showed that the pre-module survey is 0.735 (n=51), and post-module survey is 0.763 (n=51). Both the figures indicate that the responses were highly reliable. The result of the paired t-test is described in Table 5.4(a), and the result of the two-tailed independent t-test is presented in Table 5.4(b).

Table 5.4 (a) Result of the two-tailed paired t-test for the attitude survey on students’ year 1 study

	Paired differences			95% onfidence interval of the difference				
	Mean	Std. eviation	Std. error mean	Lower	Upper	t	df	Sig. 2-tailed α=0.05
Research group	(n=25)							
Pair 1	-.280	1.458	.292	-.882	.322	-.960	24	.347
Pair 2	-.640	1.350	.270	-1.197	-.083	-2.370	24	.026
Pair 3	-.080	1.352	.270	-.638	.478	-.296	24	.770
Pair 4	-.560	.961	.192	-.957	-.163	-2.914	24	.008
Pair 5	-.600	1.291	.258	-1.133	-.067	-2.324	24	.029
Pair 6	.640	1.578	.316	.011	1.291	2.208	24	.054
Pair 7	-.800	1.399	.313	-1.455	-.145	-2.557	24	.019
Pair 8	-.900	1.683	.376	-1.688	-.112	-2.392	24	.027
Pair 9	-.700	1.490	.333	-1.397	-.003	-2.101	24	.049
Pair 10	-1.200	1.542	.345	-1.922	-.478	-3.479	24	.003
Control Group	(n=26)							
Pair 1	-.269	1.511	.296	-.880	.341	-.908	25	.372
Pair 2	1.269	1.511	.296	.659	1.880	4.282	25	.000
Pair 3	-.231	1.366	.268	-.782	.321	-.862	25	.397
Pair 4	.192	1.443	.283	-.390	.775	.680	25	.503
Pair 5	.038	1.661	.326	-.632	.709	.118	25	.907
Pair 6	-.538	1.702	.334	-.1.226	.149	-1.613	25	.119
Pair 7	-.056	1.474	.347	-.789	.678	-.160	25	.875
Pair 8	.167	1.543	.364	-.601	.934	.458	25	.653
Pair 9	-.056	1.474	.347	-.789	.678	-.160	25	.875
Pair 10	.722	1.406	.331	.023	1.421	2.179	25	.044

As indicated in Table 5.4(a), three statements in the research group do not show a significant difference statistically: Pair 1 (t(24)=-0.960, p=0.374, m=-0.280), Pair 3 (t(24)=-0.296, p=0.770, m=-0.080), and Pair 6 (t(24)=2.208, p=0.054, m=0.640). This result is better than in the control group, which has only two paired statements that indicate a statistically significant difference: Pair 2 (t=-4.282, p<=0.001, m=1.296) and Pair 10 (t=2.179, p=0.044, m=0.722). However, there is a positive mean in these statements, and this indicates that the change at the end of module resulted in a lower mean value, so it cannot be identified as a positive result.

Table 5.4(b) Result of independent samples t-test for attitude questionnaire of students' year 1 study of the research (n=25) and control (n=26) groups

Pair of statements		Levene's Test for Equality of Variances F	Sig.	t	df	Sig. 2-tailed α=0.05	Mean differ-ence	Std. error diff	95% confidence interval of the difference Lower	Upper
1	Equal variances assumed	.131	.719	-.058	49	.954	-.018	.317	-.655	.618
2	Equal variances not assumed	4.155	.047	6.333	46.888	.000	1.743	.275	1.189	2.297
3	Equal variances assumed	.021	.886	.3170	49	.753	.080	.253	-.428	.588
4	Equal variances assumed	.134	.716	2.666	49	.010	.762	.286	.187	1.336
5	Equal variances assumed	.695	.409	2.326	49	.024	.763	.328	.104	1.422
6	Equal variances assumed	.067	.796	-4.643	49	.000	-1.368	.295	-1.960	-.776
7	Equal variances assumed	.874	.356	1.926	49	.062	.617	.320	-.033	1.266
8	Equal variances assumed	1.048	.313	1.616	49	.115	.606	.375	-.154	1.366
9	Equal variances assumed	.401	.530	1.333	49	.191	.461	.346	-.241	1.163
10	Equal variances not assumed	9.733	.004	3.124	44.976	.004	1.011	.324	.34	1.678

For Table 5.4(b), the Levene's test for the responses is observed. It indicates that the pair of statements 2 and 10 of the survey at post-module cannot assume normality, with values of (F=4,155, p=0.047) and (F=9.733, p=0.004), respectively. After adjusting for the degree of freedom, the values of Pair 2 were (t(46.888), p<0.001, m=1.743) and Pair 10 (t(44.976), p=0.004, m=1.011). The results show a statistically significant difference. In addition to these two pair statements, three in the post-module survey are also statistically significantly different. They are Pair 4 (t(49)=2.666, p=0.01, m=0.762), Pair 5 (t(49)=2.326, p=0.024, m=0.763), and Pair 6 (t(49)=-4.643, p<0.001, m=-1.368) (note that the negative mean is due to the statement using negative wording). They all indicate that the research group had a better outcome in the relevant statement.

<u>Q4. Do students feel that learning with the e-learning object provides them with a better approach to their learning?</u>

This question focuses students' experience of using the e-learning object with four key focuses: 'Design', 'Effectiveness', 'Helpfulness' and 'Motivation'. Similar to research question Q3, it used a post-module cross-sectional survey for data collection.

As discussed in Chapter 4, survey is used in this study as it can focus precisely on sets of variables relating to the quality of the e-learning object. Data collection was quantified by using a questionnaire consisting of 18 Likert scale statements with a five-point response type: 1= 'Strongly Disagree'; 2= 'Disagree'; 3= 'Neutral', 4= 'Agree'; and 5= 'Strongly Agree'. The reliability of students' responses was analysed with Cronbach's alpha reliability coefficient. With a figure of 0.848 (n=25), this indicated that the responses were highly reliable. The outcome of individual key factors is discussed as follows. There are six statements defined to evaluate the key focus 'Design'. The result is presented in Table 5.5(a).

Table 5.5(a) Result of students' responses on the design of the e-learning object

SAD=Strongly Disagree; DA=Disagree; N=Neutral; A=Agree; SA=Strongly Agree (n=25)

Statement with Key Focus		SDA	DA	N	A	SA	Mean (SD)
Key focus: Design							
6	The learning process suits my own style	0	5	4	9	7	3.72 (1.10)
7	The e-learning object attracts me to learn	0	5	3	12	5	3.68 (1.03)
8	I feel comfortable with the e-learning object	0	1	3	15	6	4.04 (0.73)
11	I feel easy with the animated flowchart	0	5	7	9	4	3.48 (1.00)
12	I feel easy with the animated code-based example	0	2	6	13	4	3.76 (0.83)
14	The animation is interesting	1	3	3	13	5	3.72 (1.06)

As indicated, the means of all statements range from 3.68 to 4.04. It shows that students were generally satisfied with the design of the e-learning object. The details of responses are described as follows: total of 16 out of 25 students either 'agreed' or 'strongly agreed' (m=3.72) that the learning process suits their style. Similarly, 17 out of 25 (m=3.68) students said that the e-learning object encouraged them to learn, and up to 21 out of 25 (m=4.04) students felt comfortable with the e-learning object. This shows that they had no specific problem with using it. For the design of the animated flowchart and code-based example, 13 out of 25 students (m=3.48) and 17 out of 25 students (m=3.76) indicated that they were easy to use. Moreover, 18 out of 25 students (m=3.72) felt that the animation was interesting.

There are three statements defined to evaluate the key focus of 'Effectiveness'. The result is presented in Table 5.5(b).

Table 5.5(b) Result of students' responses on the effectiveness of the e-learning object

SAD=Strongly Disagree; DA=Disagree; N=Neutral; A=Agree; SA=Strongly Agree (n=25)

Key focus: Effectiveness		SDA	DA	N	A	SA	
2	With the e-learning object I feel the instructions are precise	2	4	5	4	10	3.64 (1.38)
4	I always feel the lessons are effective	0	4	4	12	5	3.72 (0.98)
5	I like to use similar cognitive tools in other programming modules	3	3	4	11	4	3.40 (1.26)

The mean values of these statements range from 3.40 to 3.72. This indicates that students felt that the e-learning object pedagogy is effective in general. The responses to individual statements are that 14 out of 25 students (m=3.64) indicated that by using the e-learning object the instruction was precise. Of these 14 students, 10 'strongly agreed'. This shows that the e-learning object can make the instruction effective in class learning. This outcome is consistent with the next statement, with which 17 out of 25 students 'agreed' or 'strongly agreed' that (m=3.72) the lesson was effective. However, the final statement 'I like to use similar cognitive tools in other programming modules' has the lowest mean value (m=3.40), and only 15 out of 52 students 'agreed' or 'strongly agreed'. This may be because students could not take in the concept of a similar cognitive tool since they did not have enough knowledge to define one.

There are four statements defined to evaluate the key focus of 'Helpfulness', as presented in Table 5.5(c).

Table 5.5(c) Result of students' responses on the helpfulness of the e-learning object

SAD=Strongly Disagree; DA=Disagree; N=Neutral; A=Agree; SA=Strongly Agree

	Key focus: Helpfulness	SDA	DA	N	A	SA	
1	The e-learning object is helpful	1	2	1	11	10	4.08 (1.08)
9	The e-learning object supports my needs	0	0	3	12	10	4.28 (0.68)
13	The animations help me to better understand the program controls	1	3	2	13	6	3.80 (1.08)
17	I could finish the class activities quickly	0	2	3	10	10	4.12 (0.93)

The mean values of these statements range from 3.80 to 4.28. This indicates that students generally felt that the e-learning object was helpful. The responses to the individual statements are that 21 out of 25 students 'agreed' or 'strongly agreed' (m=4.08) that the e-learning object was helpful. In respect of supporting their needs, it had the best mean value (m=4.28) of this key focus. This may show that the e-learning object can facilitate students' individual learning style. This result is important, as individual difference is commonly found in learning computer programming, causing great difficulty in teaching.

Regarding students' experience of using the animations (e.g. the animated flowchart and code-based examples), 19 out of 25 students 'agreed' or 'strongly agreed' (m=3.80) that it could help them to understand the program controls better. The result is positive, but the mean value is the lowest of all key focuses. Of the negative responses, 6 out of 25 students 'strongly disagreed', 'did not agree' or were 'neutral', so it may be concluded that approximately 6 students (24% of total responses) could not use the animated flowchart and code-based example to improve their understanding of the semantics of program controls. Some 20 out of 25 (m=4.12) students indicated that they finished the class activities quickly. In this sense, a large number of students in the research group felt that the e-learning object was helpful to their learning process.

There are four statements defined to evaluate the key focus of 'Motivation', and one statement related to students' overall impression of the e-learning object. The result is presented in Table 5.5(d).

Table 5.5(d) Outcome of students' responses on the motivation and overall impression of the e-learning object (Motivation)

SAD=Strongly Disagree; DA=Disagree; N=Neutral; A=Agree; SA=Strongly Agree

Key focus: Motivation		**SDA**	**DA**	**N**	**A**	**SA**	
3	I am frequently involved in the learning activities	2	4	3	7	9	3.68 (1.35)
10	I am happy to use the e-learning object to learn	1	0	8	9	7	3.84 (0.99)
15	I often use the e-learning object in class exercises	3	2	3	9	8	3.68 (1.35)
16	I enjoyed the lesson	1	1	4	11	8	3.96 (1.02)
Overall							
18	Overall, I felt the lesson delivery is effective	1	1	5	8	10	4.00 (1.08)

The mean values of these statements range from 3.68 to 4.00, showing a positive outcome. The responses to individual statements indicate that 16 out of 25 students 'agreed' or 'strongly agreed' (m=3.68) that they were frequently involved in the learning activities. This response does not indicate that students were motivated by or engaged in class interaction. This is consistent with the finding of the attitude survey that students felt nervous about learning computer programming, which could well result in low involvement in class activities.

Some 16 out of 25 students indicated ‘agree’ or ‘strongly agree’ (m=3.84) with the statement that they were happy to use the e-learning object to learn. This outcome is highly positive. However, this is not a conclusion drawn from a high mean value: one student responded, ‘strongly disagree’, although none responded, ‘not agree’. Moreover, 8 out of 25 students were ‘neutral’, and this is of concern since these students evidently ‘cannot find any benefit’ in using the e-learning object.

Of 25 students, 17 indicated ‘agree’ or ‘strongly agree’ (m=3.68) to the statement ‘I often use the object to work on the exercises’. This response is consistent with ‘I am happy to use the e-learning object to learn’, the outcome of which is not strongly positive. However, that can be explained by the e-learning object not being designed specifically to support exercises; moreover, some exercises are beyond the scope of the support which can be provided by the e-learning object.

The final statement of this key focus evaluated students’ overall feeling on class interaction: ‘I enjoyed the lesson’. It has the highest mean value, with 19 out of the 25 students stating ‘agree’ or ‘strongly agree’ (m=3.96). This outcome is similar to that of the final statement of the questionnaire: ‘Overall, I felt the lesson delivery is effective’. This is about students’ overall impression of the e-learning object, and 18 out of the 25 students indicated ‘agree’ or ‘strongly agree’ (m=4.00). This shows that students’ overall impression of using the e-learning object is very good.

<u>Q5. Is the exposure of learning with the e-learning object associated with better mental engagement in the pedagogical environment?</u>

This research question evaluated students’ engagement in the pedagogical environment through the whole learning process at points in time. As discussed in Chapter 4, it used a checklist of seven-point rating bipolar anchoring specifications for data collection, as it is an effective and practical method for evaluating computer courses, while allowing students to make generic ratings rather than be quizzed. This avoids them having to spend a long time completing the survey. The specifications in this checklist are: ‘Boring – Stimulating’; ‘Did Not Learn Much – Learned Much’; ‘Not Engaged in Learning Process – Engaged in Learning Process’; and ‘Not Much

Work Done – Much Work Done'. The data collection was at four milestones. It was defined upon finishing the topics of selective control, iterative control, array control and functional control. The detailed process is illustrated in Figure 4.3. The data were analysed with one-way repeated measure ANOVA, which is evidenced to be an effective measure of the variance in the specification over time. The difference between the research and control groups was evaluated with the two-tailed independent t-test to find out if there was significant difference between these two independent groups of data. The results are organized in the individual tables presented as follows.

Boring – Stimulating

The outcome of one-way repeated measure ANOVA for the specification 'Boring – Stimulating' at all milestones is presented in Table 5.6(a).

Table 5.6(a) Result of one-way repeated measure ANOVA on the specification of 'Boring-Stimulating' at all milestones for each group (Tests of within-subjects effects)

* degrees of freedom are corrected using Greenhouse-Geisser estimates of sphericity

Source		Type III Sum of Squares	df	Error (df)	Mean Square	F	Sig. α=0.05
Research group	Sphericity assumed (X2(5)=9.307, p=.098)	22.989	3	66	7.663	3.795	.014
Control group	Sphericity not assumed* (X2(5)=14.662, p=.012)	27.865	2.049	47.13:	13.598	8.793	.001

The observation of Mauchly's Test of Sphericity shows that the control group violated the sphericity test and the value is (Mauchly's W=0.509, $X^2(5)=14.662$, p=0.012 (>0.05)). This is corrected by Greenhouse-Geisser with sphericity (ε)=0.683 and SPSS indicates the corrected detail as Type III Sum of Squares=27.865, and df change to 2.049, mean square=13.598. With this correction, the outcome of the control group becomes (F(2.049, 47.132)=8.793, p=0.001(<0.05)). It shows a statistically significant difference (Field, 2012). Therefore, the C-control group was not mentally consistently engaged in the pedagogical environment across all the milestones.

For the research group, Mauchly's sphericity test ($X^2(5)=14.662$, p=0.098) indicates that sphericity is assumed. The F-value is (F(3, 66)=3.795, p=0.014 (<0.05)), indicating a statistically significant difference across the milestones. The figures in Table 5.6(a) therefore statistically indicated that neither the control nor the research group was mentally consistently engaged in the pedagogical environment, while ignoring whether they used or did not use the e-learning object to learn.

The result of the two-tailed independent t-test is presented in Table 5.6(b).

Table 5.6(b) Result of the two-tailed independent t-test for the milestones in the research to control groups in the specification of 'Boring – Stimulating'

M1 is the finishing point of selective control; M2 is the finishing point of iterative control; M3 is the finishing point of array control; and M4 is the finishing point of functional control

		Levene's Test for Equality of Variances		t-test for Equality of Means					95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. 2-tailed	Mean Difference (md)	Std. Error Difference	Lower	Upper
M1	Equal variances assumed	.053	.819	2.649	48	.011	.881	.333	.212	1.550
M2	Equal variances assumed	1.789	.188	2.783	46	.008	1.000	.359	.277	1.723
M3	Equal variances assumed	.795	.377	1.672	48	.101	.680	.407	-.138	1.498
M4	Equal variances assumed	3.642	.063	2.004	46	.051	.823	.411	-.004	1.649

The Levene's test is observed and shows that all milestones are 'Equal Variance Assumed'. The results at the milestones are: M1 (t(48)=2.649, p=0.011, md=0.881); M2 (t(46)=2.649, p=0.008, md=1.000); M3 (t(48)=1.672, p=0.101, md=0.680); and M4 (t(46)=2.004, p=0.051, md=0.823). This shows that the difference between the research and control groups at milestones M1, M2 and M4 is statistically significant, where p<0.05 with M4 is just over p=0.05. It may be concluded that the research group generally was significantly more mentally engaged in the pedagogical environment on the topics of selective control, iterative control, and functional control, but not indicated in array control.

In summary, neither the research nor the control group could mentally consistently engage to the specification 'Boring – Stimulating'. That is, in all control topics, students felt bored rather than stimulated. This outcome is consistent with many studies, as discussed in Chapter 2, which directly indicate that computer programming is one of the most boring subjects in Computer Studies. This is of concern, as boredom is a major adverse factor in learning programming. However, there is a better outcome when comparing the research and control groups at the end of the module. It shows that the research group had superior mental engagement to the control group in all topics apart from array control.

Did Not Learn Much – Learned Much

The outcome of one-way repeated measure ANOVA for the specification 'Did Not Learn Much – Learned Much' is presented in Table 5.7(a).

Table 5.7(a) Result of one-way repeated measure on the specification 'Did Not Learn Much – Learned Much' across of all milestones (Tests of within-subjects effects)

Source		Type III Sum of Squares	df	Error (df)	Mean Square	F	Sig. α=0.05
Research group	Sphericity assumed (X2(5)=4.507, p=.479)	8.865	3	69	2.955	2.585	.060
Control group	Sphericity assumed (X2(5)=8.624, p=.125)	14.365	3	69	4.788	3.048	.034

Mauchly's Test of Sphericity shows that it is not necessary to correct the degree of freedom, as neither the research nor the control group violate the sphericity assumption with the values of ($X^2(5)=4.507$, $p=0.479$) and ($X^2(5)=8.624$, $p=0.125$), respectively.

The result in Table 5.7(a) indicates that the research group was consistently mentally engaged in the pedagogical environment across all milestones. The F-value is ($F(3, 69)=2.585$, $p=0.060$ (>0.05)). However, it was not shown in the control group, of which the F-value of ($F(3, 69)=3.048$, $p=0.034$ (<0.05)), which indicates the difference is statistically significant. Therefore, the control group could not be consistently mentally engaged in the pedagogical environment.

The result of the two-tailed independent t-test for all milestones is presented in Table 5.7(b).

Table 5.7 (b) Result of the two-tailed independent t-test for the specification of 'Did Not Learn Much – Learned Much' on research groups to control groups on each Milestone

M1 is the finishing point of selective control; M2 finish is the finishing point of iterative control; M3 the finishing point of array control; and M4 the finishing point of functional control

		Levene's Test for Equality of Variances		t-test for Equality of Means					95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference (md)	Std. Error Difference	Lower	Upper
M1	Equal variances not assumed	4.313	.043	6.997	43.625	.000	1.955	.279	1.392	2.518
M2	Equal variances assumed	.006	.940	3.053	46	.004	1.208	.396	.412	2.005
M3	Equal variances assumed	.651	.424	4.715	48	.000	1.720	.365	.987	2.453
M4	Equal variances assumed	.011	.916	4.106	48	.000	1.520	.370	.776	2.264

Levene's test is observed in Table 5.7(b). Since it shows that the milestone M1 (F=4,313, $p<0.05$) does not assume normality, the degree of freedom (df) is adjusted to 43.625. Therefore, the t-values of all milestones are M1 (t(43.627)=6.997, $p<0.001$, md=1.9555); M2 (t(46)=3.035, p=0.004, md=1.208); M3 (t(48)=4.751, $p<0.001$, md=1.720); and M4 (t(48)=4.106, $p<0.001$, md=1.520). It indicates that the differences between all milestones are statistically significant, showing that the research group was more mentally engaged in the pedagogical environment across all milestones than the control group.

In summary, the one-way repeated measures ANOVA statistically indicates that the research group was consistently mentally engaged to the specification 'Did Not Learn Much – Learned Much' across all milestones, but it does not show in the control group. Moreover, the two-tailed independent t-test indicates that the research group had superior mental engagement in this specification. Therefore, the outcome of this specification strongly shows that the e-learning object can facilitate students' 'Learned Much'.

<u>Not Engaged in Learning Process – Engaged in Learning Process</u>

The outcome of one-way repeated measure ANOVA for the specification 'Not Engaged in Learning Process – Engaged in Learning Process' is presented in Table 5.8(a).

Table 5.8(a) Result of one-way repeated measure for the specification of 'Not Engaged in Learning Process – Engaged in Learning Process' on the research and control groups across of all milestones (Tests of within-subjects effects)

Source		Type III Sum of Squares	df	Error (df)	Mean Square	F	Sig. α=0.05
Research group	Sphericity assumed (X2(5)=5.163, p=.397)	12.913	3	66	4.304	2.143	.103
Control group	Sphericity assumed (X2(5)=8.526, p=.130)	17.448	3	69	5.816	4.817	.004

Mauchly's Test of Sphericity shows that it is not necessary to correct the degree of freedom, as neither the research nor the control group violates the sphericity assumption with the values of ($X^2(5)=4.163$, $p=0.397$) and ($X^2(5)=8.528$, $p=0.130$), respectively.

The result of Table 5.8(a) indicates that the research group was consistently mentally engaged in the 'Learning Process' across all milestones, with the F-value ($F(3, 69)=2.143$, $p=0.103$ (>0.05)). However, it did not show in the control group, where the F-value was ($F(3, 69)=4.817$, $p=0.004$ (<0.05)). This indicates that the e-learning object can create a better steady pedagogical environment in 'Engaged to Learning Process'.

The result of the two-tailed independent t-test is presented in Table 5.8 (b). The Levene's test is observed and indicates that all milestones have 'Equal Variance Assumed', so it is not necessary to adjust the degree of freedom. The t-values of these milestones are M1 ($t(48)=3.224$, $p=0.002$ (<0.05), md=1.080); M2 ($t(48)= 1.503$, $p=0.140$, md=0.667); M3 ($t(48)= 1.702$, $p=.095$, md=0.640); and M4 ($t(48)=3.224$, $p=0.011$ (<0.05), md=1.042). Statistically, the result shows that the research group was significantly more mentally engaged in the 'Learning Process' at both milestones M1 and M4 than those in the control group. However, the difference between them at M2 and M3 is not statistically significant.

Table 5.8(b) Result of the two-tailed independent t-test for the milestones on the ‘Not Engaged in Learning Process – Engaged in Learning Process’ on the research groups to control groups on each milestone

M1 is the finishing point of selective control; M2 finish is the finishing point of iterative control; M3 is the finishing point of array control; and M4 is the finishing point of functional control

		Levene's Test for Equality of Variances		t-test for Equality of Means					95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
M1	Equal variances assumed	.728	.398	3.244	48	.002	1.080	.333	.411	1.750
M2	Equal variances assumed	1.290	.262	1.503	46	.140	.667	.444	-.226	1.559
M3	Equal variances assumed	.701	.407	1.702	48	.095	.640	.376	-.116	1.396
M4	Equal variances assumed	2.081	.156	2.637	46	.011	1.042	.395	.247	1.837

In summary, although the one-way repeated measure ANOVA shows that the research group was consistently more mentally engaged to the specification ‘Not Engaged in Learning Process – Engaged in Learning Process’, the two-tailed independent t-test failed to show the research group had significantly better mental engagement than the control group at milestones M2 and M3. This is of concern, as the topics of M2 (iterative control) and M3 (array control) are major learning components in introductory programming. If students are not mentally engaged in the learning process, it may seriously affect their overall performance.

Not Much Work Done – Much Work Done

The outcome of one-way repeat measure ANOVA for the specification ‘Not Much Work Done – Much Work Done’ is presented in Table 5.9(a).

Table 5.9(a) Result of repeat measure on the specification of ‘Not Much Work Done –Much Work Done’ for each of the groups (Tests of within-subjects effects)

Source		Type III Sum of Squares	df	Error (df)	Mean Square	F	Sig. α=0.05
Research group	Sphericity assumed (X2(5)=8.268, p=.142)	9.478	3	66	3.159	1.702	.175
Control group	Sphericity assumed (X2(5)=5.224, p=.387)	14.115	3	69	4.705	4.001	.011

Mauchly's Test of Sphericity shows that it is not necessary to correct the degree of freedom, as neither the research nor the control group results violate the sphericity assumption with values of ($X^2(5)=8.268$, p=0.142) and ($X^2(5)=5.224$, p=0.387), respectively.

Statistically the result shows that the research group was consistently mentally engaged to 'Much Work Done' across all milestones, with the F-value as (F(3, 66)=1.702 p=0.175 (>0.05)). However, it was not shown in the control group where the F-value was (F(3, 69)=4.001 p=0.011 (<0.05)). This indicates that the e-learning object can create a better steady pedagogical environment to help students' 'Much Work Done'.

The result of two-tailed independent t-test is presented in Table 5.9(b).

Table 5.9(b) Result of the two-tailed independent t-test for the milestones on the research and control groups for the specification of 'Not Much Work Done – Much Work Done'

M1 is the finishing point of selective control; M2 finish is the finishing point of iterative control; M3 is the finishing point of array control; and M4 is the finishing point of functional control

		Levene's Test for Equality of Variances		t-test for Equality of Means					95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
M1	Equal variances assumed	1.070	.306	4.413	48	.000	1.330	.301	.724	1.936
M2	Equal variances assumed	.937	.338	2.908	46	.006	1.125	.387	.346	1.904
M3	Equal variances assumed	.254	.617	2.059	48	.045	.880	.427	.021	1.739
M4	Equal variances no assumed	4.164	.047	3.464	36.941	.001	1.423	.411	.591	2.255

The Levene's test is observed and it shows that the milestone M4 does not assume normality. The degree of freedom (df) is adjusted to 36.941 and gives the result of M4 as (t(36.941)=3.464, p=0.047 (<0.05), md=1.423). It is statistically significantly different. For the other milestones, the t-values are: M1 (t(48)=4.413, p<0.001, md=1.330; M2 (t(46)=2.908, p=0.006, md=1.125); and M3 (t(36.941)=3.464, p=0.001, md=0.880). All figures indicate statistically significant difference. Therefore, this result shows that the research group presented better mental engagement in the pedagogical environment than the control group.

In summary, the repeated measure ANOVA statistically indicates that the research group was consistently mentally engaged to the specification ‘Not Much Work Done – Much Work Done’ across all milestones. However, it did not show this in the control group. Moreover, the two-tailed independent t-test indicates that the research group could more mentally engage to this specification than the control group. It shows a strong positive outcome that the e-learning object helps students’ achievement in ‘Much Work Done’.

With the outcomes of all specification, it can be concluded that research question Q5 is strongly positive in all specifications across the four milestones, except ‘Boring – Stimulating’, and students tend to be ‘Not Engaged in Learning Process’ in the topics of ‘iterative control’ and ‘array control’. These concerns are further discussed in Chapter 6.

5.3 Stage 2 Study

There is only one research question defined in this stage, and it is to evaluate if the proposed pedagogy could be adopted by most teachers. In Chapter 2, it was found that there were many possible frameworks and models that could be used, but the TPACK was selected based on its characteristics, such that it gives a high degree of freedom in using a systematic approach to evaluate teachers’ capacity with the primary forms to domain knowledge.

<u>Q6. Does the design of the cognitive development-based pedagogy match teachers’ knowledge to use it, as suggested by the TPACK framework?</u>

This research question focuses of a set predefined attributes on teachers’ capacity for using technology for teaching extracted from the TPACK framework. This study used a cross-sectional survey because it can confine teachers’ responses relating to the characteristics defined in this pedagogy. There were 25 Likert scale five-point response type statements plus two open-ended questions used to gather teachers’ comments, providing both quantitative and qualitative data. They were defined by referencing related literature as discussed in Chapter 4. The outcome of teachers’ responses is presented in the remainder of this section.

Teachers' Profile

A total of 52 teachers were involved in this survey, and their background is presented in Table 5.10.

Table 5.10 Teachers' background (n=52)

Question	**Response type/frequency:**
What is your highest academic qualification?	PhD/Doctoral degree: 10 (24%) MPhil/Master's degree: 42 (76%) Bachelor: 0 Subdegree: 0
How many years is your teaching experience?	over 20 years: 25 (48%) 15–19 years: 10 (19%) 10–14 years: 7 (13%) 5–9 years: 3 (6%) below 5 years: 7 (14%)
Have you taught any computer programming subject? (n=52)	Yes: 45 (87%) No: 7 (13%)

Table 5.10 shows that all teachers are experienced education professionals . All teachers at least have a master's degree, and 10 of them have finished a PhD or doctoral degree. Regarding teaching experience, 17 have from 10 to 19 years, and only 10 teachers have less than 10 years' teaching experience. Regarding teaching computer programming, 45 of the 52 teachers, amounting to 87% of the total, have experience of this subject area. Therefore, it can be concluded that all teachers in this study have solid experience and knowledge of teaching computer programming, and their academic qualifications meet the requirements for teaching tertiary and undergraduate computing subjects. They are eligible to be focused on by this research.

Primary Forms of Knowledge

The primary forms of knowledge include 'Technological Knowledge', 'Pedagogical Knowledge' and 'Content Knowledge'. The evaluation of these primary forms is discussed in what follows.

Technological Knowledge (TK)

The primary form was evaluated by four statements. The result is presented in Table 5.11.

Table 5.11 Teachers' responses on the primary form of TK

1	I consider my knowledge of using technology for teaching to be: (n=52)	Expert	Very Good	Good	Needs Improvement	Poor
		6	28	5	13	0
2	My skill of using online platforms (e.g. WebCT) for teaching is: (n=52)	Expert	Very Good	Good	Needs Improvement	Poor
		9	13	17	7	6
3	I know about a lot of different technologies, including the up-to-date. (n=52)	Fully Agree	Agree	Neutral	Not Agree	Fully Disagree
		2	28	18	4	0
4	I can learn how to use technology easily. (n=52)	Fully Agree	Agree	Neutral	Not Agree	Fully Disagree
		5	34	13	0	0

The responses to the primary form 'Technological Knowledge' can be concluded as positive in general. This is evidenced by 6 out of 52 teachers indicating that their knowledge of using technology for teaching is 'expert' and 33 out of 52 teachers 'very good' or 'good', which amounts to 75% of the total responses. Regarding skills in using online learning platforms, 9 out of 52 teachers responded 'expert', while 30 out of 52 teachers identified themselves as 'very good' or 'good', which is 75% of the total responses. However, it is of concern that 7 out of 52 teachers identified their skills as 'poor' and 'need some improvement'. This is not the majority, but it is unexpected, as the use of online platforms for teaching is very common in post-secondary schools. Perhaps, there needs to further consideration strategically of how to promote among secondary teachers the advantages of online learning in order to increase their interest in using it.

In respect of teachers' knowledge relating to 'knowing about a lot of different, up-to-date technologies', the results indicate that just 2 out of 52 teachers 'fully agreed' and 28 out of 52 teachers 'agreed', which is just over half (57%) of the responses. With 18 out of 52 indicating 'neutral', this result cannot convincingly suggest that teachers often look for new technologies and are motivated to use different kinds of technology. However, it can be shown that most teachers are able to learn the use of new technology, shown by there being 39 out of 52, up to 75%, who 'fully agree' or 'agree', and none who 'fully disagreed', so they have the confidence to learn how to use new technology without difficulty. Of concern is that there were 13 out of 52 teachers who responded 'neutral', which may be due to there being not too many suitable new technologies for teachers to select, or their knowledge of using technology is still to improve.

To conclude the primary form ‘Technological Knowledge’, the outcome of teachers’ responses is most positive in all four statements. Unexpectedly, up to 13% of teachers indicated poor skill in using online platforms. This figure is of concern, and it may be required that they be encouraged to use the pedagogy in this project. Further, however, teachers’ knowledge and motivation to learn how to use new technology for supporting teaching can be seen to be in a good shape with responses where more than 50% of the teachers agreed. However, there are still some concerns, with complex reasons including teachers’ knowledge and lack of suitable new technology. The detail of this needs to be looked into further in the future.

Content Knowledge (CK)

This primary form was evaluated with three statements. The result is presented in Table 5.12.

Table 5.12 Teachers’ responses of the primary form of CK

5	My knowledge of computer programming is: (n=52)	Expert	Very Good	Good	Needs Improvement	Poor
		9	22	15	6	0
6	Learning computer programming is difficult for most students. (n=52)	Fully Agree	Agree	Neutral	Not Agree	Fully Disagree
		16	30	3	3	0
7	I keep learning new programming languages to enhance my knowledge. (n=52)	Always	Usually	Sometimes	Seldom	Never
		9	12	20	8	3

Table 5.12 indicates that most teachers have very good knowledge of computer programming. It can be evidenced by that, out of 52 teachers, 9, 22 and 15 responded that their knowledge of computer programming was ‘expert’, ‘very good’ and ‘good’, respectively. This is up to 88% of the total responses. However, it is of concern that 6 out of the 52 teachers stated that their subject knowledge was ‘need to improve’. This may be due to the appearance of diverse modern programming languages that have been introduced in academia, such that teachers may not have enough time to pursue them. It is consistent with the last statement, where just 21 out of 52 teachers, reaching not even half of the total, indicated that they ‘always’ or ‘usually’ learn new programming languages for teaching purposes. With just 20

teachers indicating ‘sometimes’, it is clear that teachers are generally not motivated to explore new program languages by themselves. For statement 6, further, 46 out of 52 teachers ‘fully agreed’ or ‘agreed’ that learning computer programming is difficult for students. This result is consistent with them feeling that teaching computer programming is difficult, as indicated in a lot of studies.

To conclude the primary form ‘Content Knowledge’, teachers’ knowledge and experience of teaching computer programming, most teachers in this study showed very good content knowledge and agreed learning computer programming is difficult. With this insight, teachers may likely search for an alternative effective way to diversify scaffolding for a pedagogy.

Pedagogical Knowledge

This primary form was evaluated by four statements. The result is presented in Table 5.13.

Table 5.13 Teachers’ responses to primary form of PK

		Always	**Usually**	**Sometimes**	**Seldom**	**Never**
8	I can organize effective classes (n=52)	3	43	4	0	2
9	I can use different teaching approaches for students’ individual needs (n=52)	0	30	20	2	0
10	I can understand students’ learning difficulties during teaching (n=52)	10	26	16	0	0
11	I can use different sources, including online materials, to facilitate teaching (n=52)	2	37	11	2	0

Table 5.13 indicates a highly positive result in this primary form. This is indicated by 46 out of 52 teachers who ‘always’ and ‘usually’ organize effective classes. By adding the four teachers who indicated ‘sometimes’, up to 96% of responses are positive. On the question of supporting individual needs during class, 30 out of 52 teachers responded ‘usually’ and 20 responded ‘sometimes’, so up to 96% of responses indicated that teachers could address individual needs in class. Moreover, most teachers indicated that they were aware of students’ learning difficulties, as of the 52 teachers, 10 responded ‘always’, 26 ‘usually’ and 16 ‘sometimes’. No teacher indicated ‘seldom’ or ‘never’, so this outcome is 100% positive. These two statements importantly indicate that the teachers in this study

have very good pedagogical knowledge. Concerning the use of online resources, 39 out of 52 teachers responded ‘always’ or ‘usually’. By adding the 11 teachers who responded ‘sometimes’, this is up to 96% positive responses. This shows that teachers like to use online resources in their teaching. In view of the fact that the e-learning object is delivered online, it may be likely that it be used by most teachers seeing it as an online resource. This feature gives an advantage to wider use of this pedagogy in introductory programming.

To conclude the primary from ‘Pedagogical Knowledge’, the indicated outcome is strongly positive. This means that teachers have a very good knowledge of implementing pedagogy for teaching computer programming.

Domain Knowledge

Domain knowledge as defined in TPACK systematically combines the primary forms specific to a domain, which are used to identify whether teachers can further establish their knowledge, to a domain, from their primary knowledge. The domains in TPACK include the ‘Pedagogical Content Knowledge’, ‘Technological Content Knowledge’ and ‘Technological Pedagogical Knowledge’. The evaluation of these domains specific to teachers’ knowledge is reported below.

Pedagogical Content Knowledge (PCK)

This domain knowledge was evaluated by four statements. The result is presented in Table 5.14.

Table 5.14 Teachers’ responses to the domain of PCK

12	Learning computer program semantics is more important than syntax. (n=52)	Fully Agree 18	Agree 23	Neutral 11	Not Agree 0	Fully Disagree 0
13	Teaching computer programming requires emphasis on cognitive development. (n=52)	Fully Agree 14	Agree 29	Neutral 9	Not Agree 0	Fully Disagree 0
14	I can select appropriate materials for teaching computer programming. (n=52)	Always 10	Usually 23	Sometimes 9	Seldom 6	Never 4
15	I can use different strategies to teach different topics in computer programming. (n=52)	Always 5	Usually 24	Sometimes 10	Seldom 6	Never 7

Table 5.14 shows that 41 out of 52 teachers, about 79% of total responses, ‘fully agreed’ or ‘agreed’ that learning program semantics is more important than learning syntax, and 43 out of 50 teachers, up to 82% of the total responses, ‘fully agreed’ or ‘agreed’ that teaching computer programming needs to emphasize cognitive development. Moreover, no response to these two statements stated ‘not agree’ or ‘fully disagree’. This indicates that all teachers are aware of the importance of learning semantics and that, because of this, the pedagogy needs to emphasize cognitive development. This result is consistent with the objectives of this project which intends to develop a pedagogy to support students’ cognitive development for learning program semantics.

For the last two statements, just over average responses are found. As shown in Table 5.14, 33 out of 52 teachers indicated that they can ‘always’ or ‘usually’ select appropriate materials. Adding the nine responses of ‘sometimes’, a total of 42 teachers, amounting to 80% of the total, generally like to select appropriate materials for their teaching. However, it is of concern that 10 out of 52 teachers, about 19%, responded ‘seldom’ or ‘never’. They seem unlikely to search for new technology to support teaching computer programming. It may discourage them from using the e-learning object as an alternative. Fortunately, there are still up to 80% responses that indicated positive. Regarding the last statement, it is more important to this domain knowledge, as it evaluates whether teachers would adopt the pedagogy as a different strategy to teach computer programming. The result shows that 29 out of 52 teachers responded ‘always’ and ‘usually’. Adding the 10 teachers who responded ‘sometimes’, this amounts to 75% of the total responses, providing a positive result which is also consistent with the evaluation on previous responses that teachers can select appropriate materials.

To conclude the domain knowledge of ‘Pedagogical Content Knowledge’, the outcome is positively indicated in most statements, except the last two. It shows that there are about 20% and 25% of total responses indicating teachers were unwilling to select and use different pedagogies to enhance learning, respectively. They seem likely to adhere to their present, familiar scaffolding and pedagogy instead of searching for a more suitable, alternative way to teach.

Technological Content Knowledge (TCK)

This domain knowledge was evaluated by two statements. The result is presented in Table 5.15.

Table 5.15 Teachers' responses regarding the domain of TCK

		Very Important	Important	Moderate	Less Important	Not Important
16	Using information technology to learn semantic concept of programing is: (n=52)	15	19	16	0	2
17	Using information technology to facilitate cognitive development in teaching computer programming is: (n=52)	13	17	20	0	2

Table 5.15 shows there are 34 and 30 out of 52 teachers, over 50% of total responses, stating that the use of technology to learn program semantics and to facilitate cognitive development, respectively, is 'very important' or 'important'. With less than 4% of total responses saying that it was 'not important', the outcome can be seen highly positive. However, it is a concern that about 42% of responses, i.e. 16 and 20 teachers, saying 'moderate', so there is still some uncertainty in teachers' views on the benefits of using technology.

To conclude the domain knowledge of 'Technological Content Knowledge', the outcome is positively indicated. Specifically, if adding the responses of 'moderate', it shows that up to 95% of responses indicated that the use of technology to support teaching computer programming is valuable. However, if 'moderate' is not a favoured response, there are some uncertainties still in teachers' views on using technology to support the teaching of computer programming.

Technological Pedagogical Knowledge (TPK)

This domain knowledge was evaluated by three statements. The result is presented in Table 5.16.

Table 5.16 Teachers' responses to the domain of TPK

18	Encourage students to use technology to improve learning performance is: (n=52)	Very Important 8	Important 30	Moderate 13	Less Important 1	Not Important 0
19	Keeping using up-to-date information technology to facilitate teaching is: (n=52)	Very Important 7	Important 26	Moderate 17	Less Important 2	Not Important 0
20	I can use information technology to facilitate teaching. (n=52)	Always 1	Usually 42	Sometimes 7	Seldom 2	Never 0

Table 5.16 shows 38 out of 52 teachers, up to 73% of the total responses, indicated that encouraging students to use information technology to improve their learning outcome is 'very important' or 'important'. Regarding teachers' willingness to use technology, 33 out of 52 teachers, up to 63% of the total, said that it is 'very important' or 'important' to keep learning up-to-date technology, and of the 52 teachers, 43 indicated that they 'always' or 'usually' use technology to facilitate teaching, and seven responded 'sometimes'. Teacher' are very willing to use technology for teaching and supporting students' learning.

To conclude the domain knowledge of 'Technological Pedagogical Knowledge', the outcomes are highly positive in all aspects, since just one teacher responded that it was 'less important' to encourage students to use technology to improve their learning performance, and two responded that it was 'less important' to keep using up-to-date technology to facilitate teaching, and two responded that they 'seldom' used technology to facilitate teaching. However, it is of concern that the responses of 'moderate' in the first two statements, whereby of all the 52 teachers, 13 and 17 teachers, about 25% of the responses for each statement, indicated they did not strongly engage in encouraging students to use information technology, or to learn, and keep learning, up-to-date technology for their teaching. It shows that there is still a relatively large number of teachers unwilling to use technology for their teaching and learning processes, and this outcome needs to be considered by this research project since it uses an e-learning object as a major cognitive tool.

Technology Pedagogical and Content Knowledge (TPACK)

Lastly, suggested by TPACK, is the integration of all domain knowledge as a framework focus on teachers' knowledge on all aspects defined concerned by the primary forms, and domain knowledge. For this study, five statements were used. The result is presented in Table 5.17.

Table 5.17 Teachers' responses to the TPACK framework

		Fully agree	Agree	Neutral	Not agree	Fully disagree
21	I like to use technology-based tools to help students to learn program semantics. (n=52)	15	26	11	0	0
22	I believe animated flowcharts help students' understanding of the program semantics. (n=52)	17	28	7	0	0
23	I like to use learning objects to facilitate teaching. (n=52)	12	30	10	0	0
24	I like to use online learning to facilitate teaching. (n=52)	20	28	4	0	0
25	I am open-minded about using diverse technologies in my teaching. (n=52)	11	33	8	0	0

Table 5.17 shows 41 out of 52 teachers, up to 78% of the total, 'fully agreed' or 'agreed' that they like to use technology to help students to learn. Regarding the use of the animated flowchart, 45 out of 52 teachers, amounting to 86% of the total, 'fully agreed' or 'agreed' that it can help students to learn. Some 42 out of 52 teachers, amounting to 80% of the total, and 48 out of 52 teachers, up to 92% of total responses, 'agreed' or 'fully agreed' that they like to use learning objects and online learning platforms to support pedagogies, respectively. The result of the final statement is that 44 out of 52 teachers, up to 84% of the total, 'fully agreed' or 'agreed' that they are open-minded about using diverse technologies.

To conclude the statements focusing on an integral view of teachers' knowledge, the use of technology, the animated flowchart, and learning objects to help students to learn program semantics, these figures are able to show that the evaluation results are highly positive, such as indicated by no teachers responding 'do not agree' or 'fully disagree' and about 21% responses of these statements being 'neutral', while up to 70% of responses indicate that they 'agree' or 'fully agree' with all aspects of this framework.

Therefore, by the evaluation defined with the TPACK, from teachers' primary knowledge, domain knowledge to the framework, it is believed that the pedagogy suggested by this project can be adopted by most teachers, as it has been shown that the objectives and design match teachers' needs and are consistent with their knowledge of using technology for teaching and learning. This outcome supports a goal of this research project which is how to generalize this pedagogy for wider use in introductory programming. This will be further discussed in the next chapter.

Open-ended Questions

There were two open-ended questions used to gather qualitative data in respect of teachers' concerns on issues connected to, or not mentioned in, the statements. A total of 10 teachers provided comments in response to these open-ended questions, as summarized in Table 5.18.

Table 5.18 Teachers' responses to the open-ended questions. (n=52)

Q1. Do you have any suggestions on teaching approach to effective students' learning program semantics?

Q2. Please list the crucial points when you consider using technology for teaching computer programming.

	Sample 1
Q1	1. Under the concepts of a language; 2. Dynamic behaviour of a program; 3. Focus on data structures; 4. Historical background of computer programming.
Q2	1. It needs to be easy to use 2. Online-based 3. Freeware 4. Focus on crucial topics such as class and object
	Sample 2
Q1	A clear step-by-step explanation of the semantics and meaning behind each coding
Q2	I feel the use of technology to support teaching is very important The technology should be focused and possible to share between students The context needs to be specific to students' difficulties, not only in using technology
	Sample 3
Q1	It needs to increase students' problem-solving skills. It also needs to increase student motivation
Q2	Aptitude of students to use IT Ability and appropriate pedagogy of teachers to use IT as a measure

	Sample 4
Q1	Focusing on logical flow of program Lets students understand the importance of program design Not over-focused on program language
Q2	The technology needs to be easy to use and encourage exploration and learning Focus on cognitive developments Must be able to be used by students at home Mobile-oriented
	Sample 5
Q1	Focus on logical flow of program Lets students understand the importance of program design Not over-focused on program language
Q2	I have concerns about the effectiveness of that technology; time spent on learning to use that technology
	Sample 6
Q1	Focus of cognitive development Ability and appropriate pedagogy for teachers to use IT as a measure Provide strategy of a clear step-by-step explanation of those semantics and meaning behind each coding
Q2	The tool needs to support the crucial features, not all features, as that may waste time in some areas Instruction still needs to be the main approach, such that the tool only be used as a supplementary tool
	Sample 7
Q1	Teacher & student relationship
Q2	Nil
	Sample 8
Q1	Prepare additional materials specific to semantics, to be provided in the pedagogy Do not use IDE and use program editor
Q2	1. Attitudes of teachers 2. Prior knowledge of using technology 3. Facilitates support
	Sample 9
Q1	Focus of problem-solving; logical issues and procedural concept Select suitable language to support these features Do not learn IDE
Q2	Technology needs to be easy to use and facilitate teaching
	Sample 10
Q1	Use simple program language, e.g. Pascal, Basic, etc. … to introduce semantics and transfer this concept to more complicated program languages, such as object-oriented
Q2	Focus on crucial topics such as class and object If the content is focused in my module That do not require too much time to learn, easy to use

The comments provided in the open-ended questions are generally consistent with the objective and design of this project. This shows that the pedagogy meets the needs of most teachers. However, among the positive comments there are still some concerns. For example, the response rate is only about one quarter of the total number of teachers involved in the TPACK study, so it may be not enough to draw a conclusive outcome independently. Besides, some of the responses on these open-ended questions not directly relate to this pedagogy. However, considering most responses are direct and useful for drawing a conclusion, the responses on the open-ended questions are still meaningful for providing information when making a 'cross-check' with the statements, and insight to find out whether any essential comments are not found in the statements. Furthermore, some negative comments and suggestions need to be responded to accordingly, as provided in Table 5.19. They are organized by individual 'Sample' as presented in the comment list table, Table 5.18.

Table 5.19 Description of fulfilment on teachers' responses in the open-ended questions.

	Sample 1/Description of fulfillment
Q1	The pedagogy is conceptually focused, as it uses an animated flowchart as well as a visualization of code-based example to link the concept to coding process But it does not relate to data structure concept, as it is not focused on introductory programming.
Q2	The e-learning object is online, and the use of flowchart approach is familiar to most computing teachers. However, it does not focus on class and object.
	Sample 2/Description of fulfilment
Q1	The pedagogy applies concept maps that map the logic of program control to coding process step by step through the animation.
Q2	The pedagogy is focused on the crucial features of program controls, according to the recommendation in the curriculum of introductory programming.
	Sample 3/Description of fulfilment
Q1	The use of animation is an approach to increasing students' motivation as research evidences are positively indicated by this outcome.
Q2	A study of students' satisfaction in using the pedagogy is included in this research project. It gives a positive outcome in this aspect.
	Sample 4/Description of fulfilment
Q1	Those points suggested by teachers are the basic approach of the pedagogy design. For example, the pedagogy uses amination to show the logic of program control, and also maps to the logic of coding process by using code-based examples. It highly emphasizes program design and uses (the animated flowchart) a syntax-free approach.
Q2	The e-learning object is relatively useful as it adopts the animated flowchart design and uses a widely applied program language as the code-based example. It uses visual imagery techniques to help students' cognitive development, and applies suitable artefacts and animation to realize the abstract concepts. The pedagogical model is online and can be accessed by web pages. Students can use and review the learnt materials out-of-class via the Internet. However, there is a limitation on this response in that it is not available mobile.

	Sample 5/Description of fulfilment
Q1	The pedagogy applies a concept map that maps the logic of program control to coding process step by step through the animation. The pedagogy also uses amination to show the logics of program control, and maps to the logic of coding process by using the code-based example. It highly emphasizes program design and uses (the animated flowchart) a syntax-free approach.
Q2	The e-learning object is integrated into a normal setting introductory programming environment and used together with the course materials. It does not require the teachers to cope with additional technical issues as it is not a separate, independent learning tool.
	Sample 6/Description of fulfilment
Q1	The pedagogy uses visual imagery techniques to help students' cognitive developments, and applies suitable artefacts and animation to realize the abstract concepts. It supports instructional approach as in a normal setting introductory programming course, since it is a common approach used in computer programming courses. However, there is a limitation against this response in that it does not provide interactive exercises.
Q2	The concept relates to this response being a crucial design of this pedagogy. It intends to focus on a major topic likely to generate major difficulties to students and is closely linked to cognitive development. This topic is diverse program controls. In this sense, the pedagogy basically provides a degree of freedom to the teachers who can design use of the e-learning object in their own pedagogy. It is also noted that the e-learning object from the suggested pedagogy can be integrated in a normal setting introductory programming course.
	Sample 7/Description fulfilment
Q1	No responses in this area, since no interactive features highlighted in this pedagogy.
Q2	Nil
	Sample 8/Description fulfilment
Q1	The e-learning object itself is an additional material but it is closely linked to the course material.
Q2	The e-learning object has been studied by computing teachers in a pilot study. The outcome of this study can be defined as positive.
	Sample 9/Description fulfilment
Q1	The approach of this pedagogy is to use animated flowcharts and it is fundamentally using a syntax-free approach. It does not engage in any Integrated Development Environment (IDE) software or platform.
Q2	The adoption uses an animated flowchart design and uses a widely applied program language as the code-based example. It uses visual imagery techniques to help students' cognitive development, and applies suitable artefacts and animation to realize the abstract concepts. These features are believed to facilitate teaching.
	Sample 10/Description of fulfilment
Q1	The approach of this pedagogy is to use animated flowcharts and fundamentally it uses a syntax-free approach. It uses code-based examples executed synchronously with the flowchart to map conceptually the semantics of program control to coding process. It applies C/C++ programming as coding example and traces the students' performance on learning Java programming to find the capacity for transfer to another program language (i.e. Java). The outcome of the follow-up study is highly positive, so we may conclude this pedagogy is not limited to C/C++ program language.

Q2	Program control is an important and common topic in introductory programming. This pedagogy is believed to be applicable to a wide range of modules provided in different IT/computing modules. It is easy to use, together with the normal materials, therefore teachers can use it with their prepared materials by linking the animation to suitable presentations. It does not take much time to use it. Animated flowcharts are also familiar to most IT/computing teachers. However, as discussed, the e-learning object does not link to the concept of class and object.

5.4 Summary

This chapter focuses on data collection and analysis of this research project. The discussion is separated into two stages. There are five research questions, Q1 to Q5, in Stage 1. Research question Q1 evaluated students' performance in their first-year module 'Programming with C/C++ language'. Data analysis indicates that the pedagogy significantly improves students' learning performance. As the e-learning object is designed to facilitate cognitive development on program semantics, it is interesting to establish if this area can be 'reused' across different program languages. Research question Q2 was defined for this objective, and focuses on whether students who used the e-learning object in year 1 also had a better outcome on their year 2 module 'Introduction to Java Programming'. To know if the outcome of research question Q1 relates to cognitive development, this study also defined three research questions, Q3 to Q5, which are focused of three major cognitive presence indicators. They are the students' attitude towards learning computer programming; their satisfaction with using the e-learning object; and their engagement in the pedagogical environment.

The data collection in Q1 and Q2 was based on a pair of pre- and post-test papers. For Q3, it applied a pair of pre- and post-module questionnaires with 10 statements focused on three major attitudinal attributes: 'Liking', 'Confidence' and 'Importance'. The collected data for these research questions were analysed with two-tailed paired t-test to measure the variance of mean scores for Q1 and Q2, and the change of attitudes in Q3 between the beginning and end of module, and used two-tailed independent t-test to find out whether the research group had a significantly better outcome than the control group at the end of the module. Although there is not a strong positive outcome showing students' learning attitude was changed significantly, the significantly better outcomes in learning in year 1 and 2 programming modules can be indicated in the study.

Students' satisfaction with using the e-learning object was evaluated by Q4. Data were collected through a post-module questionnaire on the key factors of 'Design', 'Effectiveness', 'Helpfulness' and 'Motivation'. By analysis with the descriptive method, it indicated that the outcome is positive in most features of the key factors. The last cognitive presence indicator studied in Q5, which used a checklist, contains four rating seven scales bipolar anchoring specifications of Boring – Stimulated'; 'Did Not Learn Much – Learned Much'; 'Not Engaged in Learning Process – Engaged in Learning Process'; and 'Not Much Work Done – Much Work Done' for data collection. The survey was conducted at four predefined milestones with both the research and control groups, upon finishing each topic of program control. The variance in these specifications across the milestones was analysed with one-way repeated measure ANOVA, and the groups' differences at every milestone were analysed with two-tailed independent t-test. The data analysis outcome indicates that the research group generally had a superior and more consistent engagement in the pedagogical environment. However, this cannot be concluded on the response of the specification 'Boring – Stimulating'; this may imply that learning computer programming is 'really boring' for most novices. Also, the study cannot indicate using the e-learning object has a better engagement to the learning process than not using it in the topics of iterative and array control. This problem deserves further attention for recommendation of using this pedagogy as these topics are crucial in introductory programming.

Stage 2 of this project has one research question, Q6. It aims to establish if the pedagogy can match computing teachers' capacity to use technology in their teaching, so that the pedagogy established by this project could be generalized to wider use in introductory programming without being restricted in its implementation by different teachers. To analyse teachers' capacity using IT for teaching, this research question applied the TPACK framework because of its advantages in focusing on teachers' pedagogical knowledge and how it is used in their teaching, with a systematic approach to integrate these elements into an evaluation framework. Specifically, the evaluation began from primary forms including 'Technological Knowledge (TK)', 'Content Knowledge (CK)' and 'Pedagogical Knowledge (PK)', and extended to domain knowledge including 'Pedagogical Content Knowledge (PCK)', 'Technological Pedagogical Knowledge

(TPK)' and 'Technological Content Knowledge (TCK)', and eventually integrating as a framework specific to this pedagogy. There were 25 statements and two open-ended questions used to collect data. These were analysed by the descriptive method with mean analysis. The analysis outcome can be concluded to be positive for most of the chief elements, with some strongly positive, indicated by responses such as 'very important' or 'important' and in the context of: (i) the use of IT to support students to achieve a better learning outcome in program semantics; (ii) the use of technology to support cognitive development in learning program semantics; and (iii) learning program semantics as more important than syntax. Statements to which teachers indicated that they 'agreed' or 'strongly agreed' included: (i) the use of animated flowcharts to help students to understand the semantic of program controls; (ii) the use of technology-based tools to help students to learn computer programming; (iii) they have no difficulty in using learning objects and online learning platforms; and (iv) they are open-minded about using diverse technologies in teaching. The analysis outcomes and the findings of this study are discussed in the next chapter.

Chapter 6 Discussion

6.1 Introduction

This short chapter focuses on reporting the findings which are drawn from the analysis outcomes of the research questions discussed in Chapter 5. There are seven findings raised from use of this cognitive development-based pedagogy, some of which, however, are specific to use of the e-learning object. These findings are important for indicating the contributions of applying a cognitive learning tool to learning computer programming, instead of focusing on pedagogical approach. The discussions in this chapter are mainly organized by individual finding to concur with the defined objectives of this project. Deficiencies from using this cognitive development-based pedagogy are also provided. These deficiencies may lead to some recommendations for using it, or provide hints for future studies in the field of related pedagogical issues, and effective approaches for learning computer programming. However, the discussion of recommendations and future studies are the focus of Chapter 7.

6.2 Discussion of the Findings

There are seven major findings drawn from the outcomes of the research questions, as discussed in Chapter 5. They are presented individually, as Findings 1–7.

Finding 1: This cognitive development-based pedagogy significantly improves students' learning in their first-year module 'Programming with C/C++ Language'

This finding is drawn from a result of research question Q1 as presented in Table 5.2 (a) and (b). In summary, it shows that the paired t-test indicated the teaching processes are significant in both the research group ($p<0.001$, $m=-54.0$) and the control group ($p<0.001$, $m=-37.9$). While the two-tailed independent t-test indicated that the learning outcome of the research group on post-test ($p=0.003$, $m=0.14.39$) was significantly better than the control group, it did not indicate as much in the pre-test ($p=0.399$, $m=1.647$). Therefore, it could be concluded that this cognitive development-based pedagogy using the e-learning object can significantly improve students' learning in their year 1 module 'Programming with C/C++ Language'.

Finding 2: This cognitive development-based pedagogy is not limited to benefiting learning of the C/C++ program language

This finding is drawn from a result of research question Q2 as presented in Table 5.3 (a) and (b). As indicated in those tables, the two-tailed paired t-test showed the teaching processes in both the research ($p<0.001$, $m=-47.8$) and the control groups ($p<0.001$, $m=-30.0$) are effective. While the two-tailed independent t-test indicated the learning outcome of research group on post-test ($p<0.001$, $m=20.822$) was significantly better than the control group, it did not indicate as much in pre-test ($p=0.314$, $m=3.022$). Therefore, it finds that the research group still has significantly better learning outcomes on their year 2 module ‘Introduction to Java Programming’.

Further to this outcome, in consideration of year 2, neither the research nor the control group used the cognitive development-based pedagogy to learn, but instead learnt with the normal approaches. As this study was focused on ‘Introduction to Java Programming’ and students in the research group had used the cognitive development-based pedagogy in learning the module ‘Programming with C/C++ programming language’, it is possible, with regard to these outcomes, to show that this cognitive development-based pedagogy does not only give positive effects on learning C/C++ programming language, but also in other program languages by using the improved cognitive skills from year 1 study.

In this sense, the use of the e-learning object in this new pedagogy to improve cognitive processing with scaffolding, as adopted in this research project, can be redesigned and used in different program languages, for the e-learning object concerned aspects across computer programming. These aspects are closely connected to the programming phases of ‘Program Design’ and ‘Program Implementation’, as identified in Table 3.1. Therefore, this finding is possible to provide evidence on the wider usage of this cognitive development-based pedagogy to diverse introductory programming modules. This, importantly, supports one of major contributions of this research project. It is discussed in the next chapter.

Finding 3: The proposed pedagogy provides students with better attitudes towards learning computer programming, although this is not strongly indicated

This finding is drawn from the positive outcomes of three major attitudinal factors of ‘Liking’, ‘Importance’ and ‘Confidence’ as indicated in research question Q3. This question focuses on students' change of learning attitude upon finishing the module. The detail of the outcome is presented in Table 5.4 (a) and (b), and is summarized in Table 6.1.

Table 6.1 Summary of significant/non-significant statements to the attitude questionnaire of students' year 1 study

Statements (key factors)	Paired t-test		Independent t-test
	Research group	Control group	Research to control
Pair 1 (liking) Learning computer programming is fun	non-significant	non-significant	non-significant
Pair 2 (liking) I look forward to learning computer programming / I look forward to learning advance computer programming	significant	significant	significant
Pair 3 (importance) Learning computer programming is important to my study	non-significant	non-significant	non-significant
Pair 4 (confidence) I feel confident in learning computer programming	significant	non-significant	significant
Pair 5 (importance) I want to be a professional programmer	significant	non-significant	significant
Pair 6 (confidence) I feel nervous when I know I need to learn computer programming	non-significant (0.54)	non-significant	significant
Pair 7 (liking) I am interested in learning computing	significant	non-significant	non-significant
Pair 8 (confidence) I am not afraid of challenges	significant	non-significant	non-significant
Pair 9 (confidence) I don't have any difficulty in using a computer	significant	non-significant	non-significant
Pair 10 (confidence/importance/liking) I feel happy as I can learn computer programming in this course / I feel happy as I can learn advanced programming in the other courses	significant	significant	significant

Looking at the columns of two-tailed paired t-test, the research group generally is significantly better than the control group in most statements. This could be explained with most attitudinal factors, such as the statements in pairs 4, 6 ($p=0.054$, where p is just over 0.05) 8 and 9. It concludes that the research group had a significant improvement in 'Confidence', and also the pairs of 2, 7 and 10, related to 'Liking', indicated students 'feel' more forward-looking (Pair 2) and 'happy' (Pair 10) to learn advanced programming, presenting it as more interesting to learn (Pair

7). Regarding ‘Importance’, the pairs of 5 and 10 indicate that the research group had significant change with hoping to be professional programmers (Pair 5), and feeling that learning advanced programming was important (Pair 10).

However, positive responses are not shown in the pair 3 ‘Learning computer is important to my study’, and similarly in the pairs 1 and 6, as students still felt learning computer programming was not ‘fun’ (Pair 1), though they were interested; and still felt ‘nervous’ (Pair 6), while they wanted to be professional programmers. For the control group, it shows that only two pairs, 2 and 10, presented significant change through the learning process. The pedagogy for the control group does not show a significant effect on improving students’ learning attitudes, except ‘look forward to learn advance programming’ (Pair 2) and ‘feel happy to learn advance programming’ (Pair 10). The unwelcome result from pairs 3, 1 and 6 in the research group’s paired t-test may be due to computer programming being mentally taxing (e.g. Chan, 2008; Piteira & Costa, 2013). This may be a concern because, if students are not aware of the importance, do not feel interest, and are nervous about learning computer programming, it may become a problem for their future study.

Considering the result of independent t-test, it is only statements from half the pairs, those of 2, 4, 5, 6 and 10, significantly indicating that the research group is significantly better than the control group. This outcome cannot be concluded as positive. Further, considering the pairs 7, 8 and 9, it shows that the research group had significant change in paired t-test, but it cannot be shown to be better than the control group. From these outcomes it is only possible to conclude that the e-learning object provides positive effects on students feeling more interest in (Pair 7), and less difficulty with and fear of (Pair 9 and Pair 8), learning computer programming than the students who do not learn with the new pedagogy using the e-learning object.

In summary, this cognitive development-based pedagogy with the e-learning object can significantly give positive effects on the aspects of ‘Importance’, ‘Liking’ and, specifically, ‘Confidence’. With this change, students can significantly improve their confidence, outlook and motivation to be professional programmers, as indicated in the result of pairs 1 and 4. However, the use of pre- and post-module studies in this research question presents a limitation, while attitude needs to be measured in an ongoing process. Only comparing with two points at the beginning and end of the module may not be enough to deal with students' attitudes presented in the learning process, activities, and their engagement to the learning environment. However, this finding only intends to provide evidence on the outcome of research question Q1, which is engaged to cognitive development, and it provides evidence on students' learning performance connected to improving cognitive processes in the scaffolding of this pedagogy. Therefore, it is safe to provide supplementary information to this research objective. This is to be further discussed in the next chapter.

Finding 4: In general, students are satisfied that the e-learning object provides a better approach to their learning.

This finding is drawn from the results of research question Q4, focusing on students' satisfaction with using the e-learning object to learn based on four key focuses: 'Design'; 'Effectiveness'; 'Helpfulness'; and 'Motivation'. It used survey method to evaluate these variables, and the research process is discussed in Chapter 4, while the outcomes are presented in Tables 5.5(a)–(d). In conclusion with high means scores in all the key focuses at the values of: 'Helpfulness' is 3.80 to 4.28; 'Design' is 3.48 to 4.04; 'Motivation' is 3.68 to 3.96; and 'Effectiveness' is 3.40 to 3.72, the outcome could be concluded students are satisfied with learning with the e-learning object in the newly developed pedagogy. This finding is important for providing evidence on research question Q1 indicating the result is connected to improving cognitive development because satisfaction with using a learning tool in a pedagogical environment is also a major indicator of the cognitive presences.

Finding 5: Students' learning with e-learning is associated with better mental engagement in the pedagogical environment in most learning topics

This finding is drawn from research question Q5, focusing on students' engagement in the pedagogical environment through the learning process. The outcomes of data analysis are presented in Tables 5.6(a)–(b), 5.7 (a)–(b), 5.8 (a)–(b), 5.9(a)–(b) and 5.10(a)–(b). This research question used a checklist with anchoring bipolar response type on four mental specifications: 'Boring – Stimulating'; 'Did Not Learn Much – Learned Much'; 'Not Engaged in Learning Process – Engaged in Learning Process'; and 'Not Much Work Done – Much Work Done', to collect data at four milestones throughout the course. The variances between these milestones are analysed by one-way repeated measure ANOVA. It is a reliable and effective analysis tool to measure the variance of variables over different times.

As for the outcomes, aside from the specification ‘Boring – Stimulating’ ($p=0.014$), the variations between the milestones are significant. The other specifications of ‘Not Learned Much – Learned Much’ ($p=0.06$), ‘Not Engaged in Learning Process – Engaged in Learning Process’ ($p=0.103$) and ‘Not Much Work Done – Much Work Done’ ($p=0.175$) are able to indicate that students learn with the cognitive development-based pedagogy generally more stably and mentally engaged to the pedagogical environment. Comparing the research and control groups, similarly except for ‘Boring – Stimulating’ in M3 ($p=0.101$) and M4 ($p=0.051$) and ‘Not Engaged in Learning Process – Engaged in Learning Process’ in M2 ($p=0.140$) and M3 ($p=0.095$), it is found that this pedagogy provides a better mental engagement to the pedagogical environment than the control group.

In summary, this finding indicates that the research group can be stably mentally engaged to the pedagogical environment created by using the e-learning object at most milestones, but it does not find that in the control group. This means that the learning outcome in the research group, as indicated in research question Q1, is connected to improving cognitive development by using the e-learning object in the scaffolding (see Figure 2.1). However, the study on research question Q5 also raises a concern that students are likely to feel bored when learning computer programming, as indicated in the specification ‘Boring – Stimulating’, and they were not very engaged with the learning processes on iterative and array controls, as indicated in the specification ‘Not Engaged in Learning Process – Engaged in Learning Process’. This result may indicate learning computer programming itself is really ‘boring’, particularly in learning iteration and array. It also is consistent with the outcomes of the statements of pairs 7, 8 and 9 of research question Q3, which indicated that students were not interested in learning computer programming, and that it is mentally taxing.

Finding 6: The pedagogy in this research project matches teachers' knowledge regarding the TPACK framework

This finding is drawn from the results of research question Q6, focusing on whether the design of this pedagogy can match teachers' knowledge of using technology-based pedagogy in their teaching process. This study applied survey method and the questionnaire was defined based on the TPACK framework, of which data was analysed from primary forms, to domain knowledge, and integrated with a view of the framework. The advantages of this research method were discussed in Chapter 4. The outcome of data analysis is presented in Tables 5.12–5.17. This research question reveals that the responses to the primary forms are sufficient to show most teachers can prepare well for using technology for teaching and learning. As indicated by the figures, more than 70% of the total responses indicate that teachers' technological knowledge is at expert level, teachers are able to learn technology easily, and they possessed very good knowledge of using new technology. Moreover, half of them know a great deal about different technologies. For the knowledge of subject content and pedagogy, 80% felt their knowledge to be very strong and identified themselves as expert, confident to conduct effective classes. Some 70% of the total could understand students' individual needs, and more than half were able to use different pedagogical approaches.

Regarding the domain knowledge, the figures indicate that of the 52 teachers about 80% understand that the teaching of program semantics needs to emphasize cognitive development. More than half of the teachers can select appropriate strategies and materials to enhance their own pedagogy. For the use of technology, the figures show that more than 60% have enough knowledge to use technology for cognitive development, and the same proportion can use technology effectively to support cognitive development. A similar proportion can select suitable technologies to support different strategies and topics in computer programming. They well understand the need to encourage students to use technology to learn.

For the view of the TPACK framework, including the use of animated flowchart and learning objects to help students learning program semantics, the result is highly positive, such that 78% of the total 'fully agreed' or 'agreed' that they like to use technology to help students to learn. As much as 86% of the total 'fully agreed' or 'agreed' that it can help students to learn; and 80% to 92% of total responses 'agreed' or 'fully agreed', respectively, that they like to use the learning objects and online learning platforms to support pedagogies. There is also up to 84% of the total responses which 'fully agreed' or 'agreed' that they are open-minded about using diverse technologies.

In summary, the outcomes of research question Q6 conclude that most teachers process the knowledge recommended by TPACK to use technological pedagogy in their teaching. With an insight into primary forms and domain knowledge, it finds that the chief features of integrating the e-learning object to the scaffolding of this cognitive development-based pedagogy match teachers' knowledge of use it. It indicates that the use of this cognitive development-based pedagogy will not be limited by teachers' technological, pedagogical, and content knowledge as defined in the TPACK framework, when used in introductory programming.

Finding 7: Teachers' comments from the open-ended questions in the TPACK questionnaire are generally consistent with their responses on the Likert scale statements.

Teachers' comments in response to the two open-ended questions in the TPACK survey are listed in Tables 5.18 and 5.19. As discussed in Chapter 4, this qualitative data is used to compare with the quantized qualitative data collected in the statements of the questionnaire in research question Q6. These two open-ended questions are also defined based on the TPACK framework, just as the statements in the questionnaire. As for the outcome, a lot of consistent comments and responses are found. This included that teachers indicated that a pedagogy needs to proceed step by step to explain program semantics, focusing on problem-solving, program logics, data flow and design, increasing students' motivation and avoiding overfocusing on the program language's syntactical issues. The many suggestions were also consistent with the responses of the statements, including that a pedagogy needs to emphasize cognitive development to increase students' cognitive skills for understanding the concept of programming by using simple programming language, and that, while it is used, technology needs to match students' style of use, be easy to control, and be used alongside course materials. The context of using technology is not trivial, because it needs to maintain instructor–student interactions, support essential learning contents, and therefore make it possible to improve students' learning outcomes. All of these comments are consistent with the outcomes of the statements of the questionnaire of this research question, and they can be used to determine whether teachers would accept this cognitive development-based pedagogy in their teaching processes. This detail is further discussed in Chapter 7.

6.3 Summary

This chapter provides a discussion of the findings which are drawn from the data analysis outcomes on the research questions discussed in Chapter 5. It shows that most outcomes of these research questions are positively indicated. Finding 1 shows that this cognitive development-based pedagogy using an e-learning object can significantly improve students' learning performance, and finding 2 shows that this pedagogy is not restricted to learning C/C++ programming language, as the research group still has significantly better learning outcomes than the control group in the year 2 module, 'Introduction to Java Programming', while neither the research nor the control group learnt with the new pedagogy, and the groups' organization did not change through the two years. This outcome possibly shows that a pedagogy focus on cognitive developments can give value to different programming languages.

Finding 3 shows that the pedagogy can positively change students' attitudes towards learning computer programming, specifically on their confidence and outlook on becoming professional programmers, but it does not yet show this cognitive development-based pedagogy can significantly improve students' learning attitudes, as only half of the statements indicated the research group is significantly better than the control group. Finding 4 shows that students were generally satisfied that the e-learning object can help them to have a better learning approach. Finding 5 indicates that the research group has a better mental engagement to the pedagogical environment such that students are not restricted by the learning environment while using the e-learning object. However, this better mental engagement cannot be shown on 'Boring – Stimulating' and 'Not Engaged in Learning Process – Engaged to Learning Process'. It indicates that students still felt bored even when they learn with this pedagogy, specifically during the topics of 'iterative control' and 'array control’. Findings, 3, 4 and 5 provide evidence for the outcome of research question Q1, in which students’ learning outcome is significantly improved due to the improvement of the cognitive presences in the scaffolding.

For the last two findings, 6 and 7, it can be concluded that teachers generally have the knowledge suggested by the TPACK framework to use technology in their teaching. Therefore, the scaffolding design of this cognitive development-based pedagogy, introducing the e-learning object as a major learning tool, matches their needs and knowledge of using technology. Moreover, the findings also present the idea that this pedagogy is not restricted by different teachers’ knowledge, or by different programming languages when using it.

Chapter 7 Conclusion and Recommendations

7.1 Introduction

Chapter 6 identifies seven findings from the outcome of the research questions. This last chapter extends the discussions to focus on the implications from these findings, and links them to the aims and goals of this research project. It first draws the conclusions from those findings, organizing them into a concise summary, supported by references back to the detail of previous chapters, and relating the discussions to the four major contributions of this research project in an integrated style. These contributions are: (1) the e-learning object used in this pedagogy can be used to improve students' learning performance in computer programming; (2) the project provides evidence that a pedagogy focusing on cognitive development can be used to improve students' learning performance without being limited by programming languages; (3) development of a cognitive development-based pedagogy can be widely used in introductory programming without being limited by teachers' knowledge and programming languages; and (4) learning with this cognitive development-based pedagogy builds up students' problem-solving skills with applications to different subject areas.

With these four contributions, there are 12 recommendations which critically provide guidance and information to teachers on the use of this cognitive development-based pedagogy. As the last chapter of this research project, it proposes some future studies that are developed based on the outcomes of the data analysis and findings in Chapters 5 and 6, which are not connected with this project. The beginning of this last chapter provides a review of the background and rationale of this research project.

7.2 Background and Rationale

This project was prompted by a wide range of reasons based on the 'Problem Statement' presented in Chapter 1 and the research evidence presented in the literature review, specifically Sections 2.2 'Difficulties of Introductory

Programming'; 2.9 'Pedagogy'; 2.10 'Scaffolding'; and 2.14 'Limitation of Technology Tool for Computer Programming'. These discussions reveal the importance of using technology for cognitive development in learning computer programming (Linn & Dalbey, 2010; Nguyen, 2008; Pappas, 2014; Shanmugasundaram et al., 2007; Sternberg & Zhang, 2009), and also found that it is closely connected to the design of cognitive processes to support the scaffolding (Farrell, 2007; Jyotsana & Ajay, 2016; Kinchin, 2014). With this range of needs, this project proposed an alternative pedagogy for teachers which is focused on cognitive development by introducing a learning tool, namely 'e-learning object', to support the cognitive processes in the scaffolding of this cognitive development-based pedagogy. The major features of this pedagogy, different from others, are that it applies cognitive artefacts to visualize the abstract concept of semantics by using an animated flowchart with conceptual mapping to a corresponding animated code-based example. The detail of this design is provided in Section 3.5. This pedagogy applied three major functions of scaffolding to support the learning process: conceptual, strategic, and metacognitive scaffolding. In addition, it used the syntax-free approach to motivate students to focus on program semantics instead of an overemphasis on the syntactical issues of a program language. With this approach, the theme of this project is multidimensional and fundamental to applying cognitive theory to the field of computer programming, while also applying to the educational aspects of Computer Science.

7.3 Conclusions and Contributions

This section links all the findings, discussed in Chapter 6, to the contributions of this research project. The discussions are organized by individual contribution, referencing the discussions provided in Chapter 6.

The e-learning object used in this pedagogy can be used to improve students' learning performance in computer programming

This contribution links to finding 1 and is addressed by research question Q1. It shows that the use of an e-learning object in this cognitive development-based pedagogy can significantly improve students' learning performance. As the e-learning object is the major learning tool used to support the scaffolded cognitive processes in this pedagogy, it can be concluded that this e-learning object provides a contribution to producing a better learning outcome on learning C/C++ programming language.

The project provides evidence that a pedagogy focusing on cognitive development can be used to improve students' learning performance without being limited by programming languages

This contribution is addressed by findings 1, 3, 4 and 5; and matches the design objectives of this cognitive development-based pedagogy, whereby this pedagogy focuses on improving students' cognitive presences in the semantics of program controls, applying this knowledge across different programming languages. By reviewing these findings, as discussed in Chapter 6, finding 1 shows that students learning with this cognitive development-based pedagogy can significantly improve their learning outcome, while findings 3, 4 and 5 are able to indicate that this better learning outcome is closely connected to improving the cognitive processes in the scaffolding, with using the e-learning object. Besides, revisiting finding 2 shows that, while students learnt with the cognitive development-based pedagogy in the year 1 module, 'Programming with C/C++ Language', they still performed better in the year 2 module, 'Introduction to Java Programming', although they did not use the cognitive development-based pedagogy. Considering this pedagogy gives students a significantly better outcome in the year 1 C/C++ programming module while also providing a better performance on the year 2 Java programming module, and together with findings, 2, 4 and 5 which identify that the results are linked to the outcome of cognitive developments, this outcome could be conclude this contribution as this pedagogy is focused on cognitive development, and provide better performance in learning computer programming in different programming languages.

However, there are some limitations when applying findings 3 and 5 to this contribution. For finding 3, as discussed in Chapter 7, it cannot be concluded that students have a significantly better learning attitude than those do not use this pedagogy, although they have significantly improved through the learning process. Similarly, in finding 5, it cannot be concluded that students' learning is tiresome and their interest in learning computer programming has been improved. Besides, this finding also points out that students still engage poorly with the learning processes of the iterative and array controls. This negative result may weaken this contribution. For this reason, there are recommendations focusing on these features provided in Section 7.4.

Development of a cognitive development-based pedagogy can be widely used in introductory programming without being limited by teachers' knowledge and programming languages

This contribution links to findings 2, 6 and 7, and is addressed by research questions Q2 and Q6. Finding 2 concludes that the research group which only uses this cognitive development-based pedagogy in the year 1 module 'Programming with C/C++ Language' still had a significantly better learning outcome in the year 2 module 'Introduction to Java Programming'.

Finding 6 concludes that this pedagogy indeed matches teachers' knowledge and capacity. For example, most teachers agree that the design of the e-learning object is consistent with their teaching approaches, such as the use of a syntax-free approach to facilitate semantics concepts and map these concept to the coding process, as presented in Figures 3.4(a)–(k), where both are applied visualization techniques that help reconstruct the learning contents into students' mentally manageable styles. With regard to the layout design, most teachers said it was easy to use, they were familiar with it and were likely to use it with the course materials as it can directly control the interactions with students in class activities. With regard to the learning object design, teachers said it can be selectively used as the learning object is easily added to and dropped from the course materials. For the acceptability, most teachers have high expectations of using an alternative approach to solve specific students' difficulties. This cognitive development-based pedagogy is an ideal alternative approach.

All of these features from teachers' responses in finding 6 are consistent with finding 7, the open-ended questions defined in the TPCAK questionnaire. For example, teachers indicated that a pedagogy needs step-by-step procedures to explain program semantics, dealing with problem-solving, program logics and data flow. This expectation is consistent with the e-learning object design by using the process control (see Figures 3.4(a)–(k)). Many teachers also said they wanted a pedagogy that is able to direct the teaching to facilitate cognitive development, strategically helping students to learn program design, emphasizing procedural concept by using simple language, and, if using technology, it should not be trivial and easy to use.

As discussed in previous contributions, this new pedagogy improves students' learning and is linked to improving students' cognitive presences, and it indicates that these contributions are not specific to a particular program language. Therefore, in summarizing findings of 2, 6 and 7, this cognitive development-based pedagogy is not limited to learning C/C++ program language but can be applied to a wider range of programming languages. It also matches teachers' knowledge of using technology for teaching and learning. Therefore, together with the specific features supporting in the scaffolding, indicated in findings 3, 4 and 5, this cognitive development-based pedagogy can very likely be generalized for use with introductory programming as an alternative way of teaching.

Learning with this cognitive development-based pedagogy, builds up students' problem-solving skills with applications to different subject areas

Problem-solving skills are linked to understanding the types and domains associated with a problem, and accordingly develop a solving strategy (Morrosty, 2017). Diverse related studies indicate a binary relationship between problem-solving skills and learning computer programming, and also point out that both are closely connected to cognitive development for humans' mental operations and logical reasoning (Bouzid, 2015; Jamone et al., 2015; Kunimune & Niimura, 2014; Wang & Chiew, 2018). By revisiting the pedagogical framework defined in Tables 2.2, 2.4, 2.7 and 3.1, this new pedagogy focuses on cognitive development by linking a problem-solving model to cognitive processes for the instructional strategy of learning computer programming. Therefore, because this cognitive development-based framework was developed based on the theories that define the pedagogical framework, this cognitive development-based pedagogy can be extended to facilitate students to build up their problem-solving skills (Brant, 2013; Fidge & Teague, 2009; Lau & Yuen, 2009), and apply them to different subject areas.

7.4 Recommendations

There are many deficiencies and features pointed out in the findings which may require recommendations for using this cognitive development-based pedagogy. They are listed as follows:

- Recommendation 1: This cognitive development-based pedagogy should apply to normal setting introductory programming with possible integration of the e-learning object into the course materials. The context of this module needs to be consistent with the curriculum of Computer Science year 1 study (CS1) or similar training programmes.
- Recommendation 2: This cognitive development-based pedagogy can contribute to the learning process, emphasizing the addressing of improvements in cognitive processes in the scaffolding, and is highly focused on using syntax-free language to learn computer programming.
- Recommendation 3: This cognitive development-based pedagogy is highly focused on improving the semantics knowledge of the four basic program controls: selective control, iterative control, array control and functional control. Therefore, to apply this pedagogy, it is recommended that these topics can significantly affect the outcome of the learning processes.
- Recommendation 4: In respect of the learning environment, this cognitive development-based pedagogy is best used in those programming modules that does not rely on using Integrated Development Environment (IDE) to learn how to program.
- Recommendation 5: This cognitive development-based pedagogy is specific to C/C++ programming language, as the code-based example in the e-learning object was designed by this programming language (e.g. see Chapter 3). Although the finding 2 indicated that it can be used in other programming languages and deliver a better performance, however, if it is used in a non-C/C++ program language, a suitable revision, or substitution, of the animated coding example may be required. For example, revise the code-based example to use pseudocode instead of C/C++ program language.

- Recommendation 6: For use of this cognitive development-based pedagogy in 'non-C/C++ programming modules' as discussed in recommendation 5, alternative approaches may be: use a set of different learning objects which are suitable for using in the scaffolding by referencing the framework of using the e-learning object as discussed in Table 3.1, to support the instructional approach.
- Recommendation 7: As implied in findings 1 and 2, and based on recommendations 5 and 6, this cognitive development-based pedagogy can be applied to both procedural and object-oriented programming modules. However, for object-oriented programming, it may require some additional efforts on decontextualizing the topics which are associated with object-oriented concepts from the curriculum to define a meaningful context. Therefore, the outcome may vary depending on the curriculum defined in the module.
- Recommendation 8: In order to increase teachers' understanding of using this cognitive development-based pedagogy, it is strongly recommended that a two-week period is provided for them to familiarize themselves with the delivery strategy of using the e-learning object, and with how to integrate the e-learning object into their own course materials.
- Recommendation 9: The TPACK survey shows that teachers are not strongly in favour of encouraging students to use technology to learn. Therefore, it is recommended that teachers, who wish to use this cognitive development-based pedagogy, need to design a strategy with some major characteristics of this pedagogy that can encourage students using technology before the module commences.
- Recommendation 10: Finding 3 indicates that this cognitive development-based pedagogy cannot provide definitive evidence to show that it significantly improves students' attitudes towards learning computer programming. As attitude is a major factor in the success of junior level programming, it is recommended, in using this pedagogy, that teachers consider additional strategies to improve students' learning attitude.
- Recommendation 11: As the outcome of mental engagement indicates that this cognitive development-based pedagogy cannot improve students' boredom and

poor engagement with learning the iterative and array controls, it is recommended, upon using this pedagogy, that teachers include a series of practices which are focused on enhancing these topics in students' learning.

- Recommendation 12: Theoretically, this cognitive development-based pedagogy can be used to improve students' problem-solving skills. However, in spite of this concept being discussed in some reviewed studies, it has not been proved by research evidence. Therefore, when using this pedagogy to claim that it improves students' problem-solving skills, it is recommended that a pertinent study to evidence this goal is considered and appropriately pursued.

7.5 Future Work

Future work stemming from the findings is recommended to concentrate on the role of pedagogical approaches and teachers to enhance the limitations and deficiencies in this study. This project does not detail the issues of teachers' culture and behaviour in using this cognitive development-based pedagogy. Future studies may be designed focusing on these missing features. With regard to this pedagogy applying the C/C++ program language to design the coding example, although findings of this project indicated that this pedagogy could be transferred to other program languages, it may need some further modifications when applying it to other languages. Therefore, a further study may be focused on understanding teachers' knowledge of revamping and redesigning the cognitive learning tool, as the e-learning object, where this knowledge is not reflected in the TPACK framework.

Besides, as indicated in findings 3 and 5, the cognitive processes cannot improve students' poor learning attitude and low interest in learning advanced programming. Also, they cannot consistently engage in the learning environment without feeling that computer programming is boring. Considering that this cognitive development-based pedagogy is learning content-focused, a future study may investigate how it could be redesigned to improve students' learning attitude without being based on a narrow learning context. For example, a study could focus on the process rather than the learning context of a given curriculum to find out which process (e.g. self-learning, assessment) needs more cognitive support and can increase students' interest in learning. This could focus on improving the pedagogical environment in which the learning repertoire may be effectively applied, or search for appropriate strategic tools to facilitate supporting practices to improve mental engagement.

Moreover, based on the problems of low engagement with learning process for the iterative and array controls indicated in finding 5, a future study might include a strategy to revise the pedagogy: for example, its cognitive strategy or delivery approach, to reduce students' boredom with learning computer programming. Lastly, as the TPACK survey showed, a large proportion of teachers seek support in learning modern program languages, so a future study on this area is strongly recommended. It might include pedagogical issues and revise the e-learning object, such as using other coding examples to transfer the concept from a procedural style to an object style, and from C/C++ program language to another modern programming language.

7.6 Summary

This research has a wide range of aspects, including how to effectively teach novices computer programming by improving students' cognitive presences. It specifically focuses on program design and mapping the design to the coding process. To support these needs, this project designed a cognitive development-based pedagogy. It is characterized by using a cognitive learning tool, namely an e-learning object, to realize the abstract concepts in the semantics of the four basic program controls. They are the selective, iterative, array and functional controls.

There are four major contributions achieved in this project. They are the use of e-learning object to improve students' learning outcomes in computer programming as the e-learning object is a major learning tool in this cognitive development-based pedagogy. This is evidenced by the outcome of finding 1 directly. This project also provides evidence that a pedagogy focusing on cognitive development can be used to improve students' learning performance without being limited by programming languages. This is concluded from findings 1, 3, 4 and 5, which indicate that students' better learning outcome is directly linked to the improvement of the learning process with some major factors of cognitive development. This includes students' learning attitude, satisfaction with using the e-learning object to learn, and mental engagement with the pedagogical environment. The third contribution is the development of a cognitive development-based pedagogy for wide use in introductory programming without being limited by teachers' knowledge and programming languages. This is evidenced by findings 2, 6 and 7. These findings conclude that this cognitive development-based pedagogy is not restricted to specific program languages, or by different teachers' knowledge of using them. The last contribution is that learning with this cognitive development-based pedagogy builds up students' problem-solving skills, with applications to different subject areas. This is implicitly defined by research indicating that programming is a meaningful tool for learning problem-solving skills. Because this cognitive development-based pedagogy is built based on a problem-solving model to work out the related cognitive processing steps, and map those steps to the instructional strategy of learning computer programming, this pedagogy is not only specific to learning how to program but also to facilitating the building up of students' problem-solving skills which can be used in other subject areas.

The recommendations in this chapter mainly concern the delivery of essential information to those who wish to apply this pedagogy to their teaching and learning processes, and provides supplementary information for using it. These are mainly discussed in recommendations 1–8. The deficiencies and unwelcome outcomes from the findings also provide appropriate recommendations for minimizing the effects of using this pedagogy, as indicated in recommendations 9–12. The last part of this final chapter provides a discussion of possible future study. There are many major features that have been found and need further attention regarding the pedagogical approach and technological issues of using the e-learning object and this cognitive development-based pedagogy.

References

Abid, A., Farooq, M. S., Farooq, U., Abid, K., & Shafiq, M. (2015). A Strategy for the Design of Introductory Computer Programming Course in High School. *Journal of Elementary Education, 25*(1), 145–165.

ACM Curriculum Committee on Computer Science. (2001). Curriculum 2001: Computer Science: Communications of the ACM (November 2001) and IEEE CS (December 2001).

Adalbert, G. S. R., Jignesh, M. P., & Richard, H. (2018). *Is More Active Always Better for Teaching Introductory Programming?* Paper presented at the Learning and Teaching in Computing and Engineering (LaTICE).

Akyol, Z., & Garrison, D. R. (2011). Understanding cognitive presence in an online and blended community of inquiry: Assessing outcomes and processes for deep approaches to learning *British Journal of Educational Technology, 42*(2), 233–350.

Ala-Mutka, K. (2005). Problems in Learning and Teaching Programming: Institute of Software Systems, Tampere University of Technology, Finland.

Alber, R. (2014). 6 Scaffolding Strategies to Use With Your Students. *TEACHING STRATEGIES.* https://www.edutopia.org/blog/scaffolding-lessons-six-strategies-rebecca-alber

Alfeld, P. (2011). G. Polya, How to Solve It. 19 NOv 2019, from https://www.math.utah.edu/~pa/math/polya.html

Annaweisspol. (2016). Teaching effectively with technology: TPACK, SAMR, RAT. Retrieved 15 July, 2018, from https://micool.org/updates/blog/2016/06/09/teaching-effectively-with-technology-tpack-samr-rat/

Apiola, M., & Tedre, M. (2012). New perspectives on the pedagogy of programming in a developing country context. *Computer Science Education, 22*(3), 285–313.

Armstrong, P. (2010). Bloom's Taxonomy. Retrieved 20 Nov 2019, from https://cft.vanderbilt.edu/guides-sub-pages/blooms-taxonomy/

Belland, B., Walker, A., Kim N, & Lefler, M. (2017). Synthesizing Results From Empirical Research on Computer-Based Scaffolding in STEM Education: A Meta-Analysis. https://www.ncbi.nlm.nih.gov/pubmed/28344365

Belland, B. R., French, B. F., & Ertmer, P. A. (2008). Validity and Problem-Based Learning Research: A Review of Instruments Used to Assess Intended Learning Outcomes. *Interdisciplinary Journal of Problem-based Learning, 3*(1), Article 5. Available at: http://docs.lib.purdue.edu/ijpbl/vol3/iss1/5

Belland, B. R., Kim, C. M., & Hannafin, M. J. (2013). A Framework for Designing Scaffolds That Improve Motivation and Cognition. *Educational Psychologist, 48*(4), 243–270.

Bevier, M. L. (2010). Revision of Bloom's taxonomy of educational objectives, Longman, 352 p. *Journal of Geoscience Education, 58*(5), 252–261.

Bills, D. P., & Bile, J. A. (2005). The role of Programming in IT. *Communication of ACM, 32*(1).

Bodemer, D., & Faust, U. (2006). External and mental referencing of multiple representations. *Computers in Human Behavior, 22*(1), 27–42.

Boulos, M., Maramba, I., & Wheeler, S. (2006). Wikis, blogs and podcasts: A new generation of web-based tools for virtual collaborative clinical practice and education. *BMC Medical Education, 6*(41). http://www.biomedcentral.com/content/pdf/1472-6920-6-41.pdf#search=%22wikis%20blogs%20and%20podcasts%20boulos%20.pdf%22

Bouzeghoub, A., Defude, B., Duitama, F., & Lecocq, C. (2006). A Knowledge-Based Approach to Describe and Adapt Learning Objects. *International Journal on E-Learning, 5*(1), 95–102.

Bouzid, A. (2015). Problem Solving & Decision Making: Tools, Techniques &Method: A Guide to Learning Effective Method, Powerful Tools and Techniques for Problem Solving and Decision Making Retrieved 12 May, 2017, from https://www.udemy.com/problem-solving-tools-techniques-methods/

Boyle, T. (2006). *The Design and Development of Second Generation Learning Objects.* Paper presented at the Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications Orlando, Florida.

Branch, R. M. (2010). *Instructional Design: The ADDIE Approach* New York, NY: Springer.

Brant, W. A. (2013). *Mental Imagery and Creativity: Cognition, Observation and Realization*. Hamburg: AkademikerVerlag.

Branun, T. (2008). Making a Choice: The Perceptions and Attitudes of Online Graduate Students. *Journal of Technology & Teacher Education, 19*(1), 63–92.

Breivik, J. (2016). Critical thinking in online educational discussions measured as progress through inquiry phases: A discussion of the cognitive presence construct in the Community of Inquiry framework. . *International Journal of E-Learning & Distance Education, 31*(1), 1026.

Brent, R., Laura, E., & Donna, W. (2014). Cognitive Tool. In M. Orey (Ed.), Emerging perspectives on learning, teaching, and technology (Vol. May 2017): Department of Educational Psychology and Instructional Technology, University of Georgia

Retrieved from http://epltt.coe.uga.edu/index.php?title=Cognitive_Tools.

Brown, E. L., Krasteva, A., & Ranieri, M. (Eds.). (2016). *E-Learning and Social Media: Education and Citizenship for the Digital 21st Century* (1st ed.). Charlotte, NC: Information Age Publishing Inc.

Brown, H. (2012). In order to be you have to be: Modeling a constructivist approach for teacher candidates. *Brock Education, 21*(2), 36–52.

Bruce, K. B. (2005). Controversy on how to teach CS 1: a discussion on the SIGCSE-members mailing list. *37*(2), pp. 111–117. https://pdfs.semanticscholar.org/343d/b188a18a4f08d8343e1f298fe49cd3790d11.pdf

Bruner, J. S. (1979). On Knowing: Essays for the Left Hand. 78.

Buldua, M., & Buldu, N. (2010). Concept mapping as a formative assessment in college classrooms: Measuring usefulness and student satisfaction. *Procedia Social and Behavioral Sciences, 2*, 2099–2104.

Burns, A., & Joyce, H. d. S. (Eds.). (2005). *Teachers' voices 8: Explicitly supporting reading and writing in the classroom*. Sydney National Centre for English Language Teaching and Research, Macquarie University.

Butler, M., & Morgan, M. (2007). *Learning challenges faced by novice programming students studying high level and low feedback concepts*. Paper presented at the Proceedings ascilite Singapore conference paper retrieved from

Cakiroglu, U., Baki, A., & Akkan, Y. (2012). The Effects of Using Learning Objects in Two Different Setting. *The Turkish Online Journal of Educational Technology, 11*(1), 181–191.

Carlston, D. E. (Ed.). (2013). *The Oxford Handbook of Social Cognition*. New York, NY: Oxford University Press.

Caspersen, M. E., & Bennedsen, J. (2007). *Instructional design of a programming course: a learning theoretic approach*. Paper presented at the the third international workshop on Computing education research Atlanta, Georgia, USA proceeding retrieved from http://localhost/phdWeb/2009-B/InstructionDesignForProgramming.pdf

Castelli, V., Bergman, L., Lau, T., & Oblinger, D. (2010). Sheepdog, parallel collaborative programming-by-demonstration. *Knowledge-Based Systems, 23*(2), 94–109.

Centra, J. A. (1993). *Reflective Faculty Evaluation: Enhancing Teaching and Determining Faculty Effectiveness (Jossey-Bass Higher and Adult Education)*. San Francisco, CA: Jossey Bass.

Cetin, I., & Andrews-Larson, C. (2016). Learning sorting algorithms through visualization construction. *Computer Science Education, 26*(1), 27–43.

Chai, C. S., Koh, J. H. L., & Tsai, C.-C. (2010). Facilitating Preservice Teachers' Development of Technological, Pedagogical, and Content Knowledge (TPACK) *Educational Technology & Society, 13*(4), 63–73.

Chan, I. T. (2008). Issue and Difficulties in Teaching Novice Computer Programming. In M. Iskander (Ed.), Innovative Techniques in Instruction Technology, E-learning, E-assessment, and Education (pp. 199–204). B.V: Springer Science + Business Media Retrieved from http://books.google.com.hk/books?id=ngQH8S7Zx0gC&pg=PA200&lpg=PA200&dq=%E2%80%9CCloseness+of+mapping%E2%80%9D+is+a+description+of+the+closeness+of+the+programming+language&source=bl&ots=PplrJGb7Z5&sig=6ljX-lTOSNxO0S_lWfVrVkCEHFM&hl=zh-

TW&ei=52mLStviL8zIkAWI_I0h&sa=X&oi=book_result&ct=result&resnum=1#v=onepage&q=&f=true.

Chan, T. (2005). *Teaching Computer Programming Online with JAVA Applets*. Paper presented at the Proceedings of World Conference on the WWW and Internet Conference Paper retrieved from

Changing Mind. (2016). Inductive Reasoning. Retrieved 31 July, 2017, from http://changingminds.org/disciplines/argument/types_reasoning/induction.htm

Chen, C. (2010). Information visualization. *Wiley Interdisciplinary Reviews: Computational Statistics, 2*(4), 387–401.

Churchill, D. (2007). Towards a useful classification of learning objects. *Educational Technology Research And Development, 2007, 55*(5), 479–497.

Churchill, D. (2012). Presentation design for 'conceptual model' learning objects. *British Journal of Education Technology, 45*(1), 136–148.

Clark, D. (2015a). ADDIE Timeline. Retrieved 1 Aug, 2018, from http://www.nwlink.com/~donclark/history_isd/addie.html

Clark, D. (2015b). Instructional System Design: The ADDIE Model A Handbook for Learning Designers. *Performance Juxtaposition Site.* Retrieved 1 Aug, 2018, from http://www.nwlink.com/~donclark/hrd/sat.html

Clifford, M. (2016). WHAT BLENDED LEARNING IS – AND ISN'T. https://www.blendedlearning.org/what-blended-learning-is-and-isnt/

Collis, B., & Strijker, A. (2003). Re-usable learning objects in context. *Internal Journal on E-Learning: Corporate, Government, Healthcare, & Higher Education, 2*(4), 5–16.

Conole, G. (2017). E-Learning Pedagogy Part 2: The Cognitive Perspective. Retrieved 12 July, 2017, from http://blog.lambdasolutions.net/e-learning-pedagogy-part-2-the-cognitive-perspective

Costley, C., Elliott, G. C., & Gibbs, P. (2010). *Doing Work Based Research: Approaches to Enquiry for Insider-Researchers*. London: Sage.

Creswell, J. W. (2009). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches* (3rd ed.). Thousand Oaks, CA: Sage.

Creswell, J. W. (2013). *Qualitative Inquiry & Research Design: Choosing among Five Approaches* (3rd ed.). Thousand Oaks, CA: Sage.

CUREE. (2012). Evaluation of Learning Away: Hypothesis 5: Teachers' pedagogical skills. London: Paul Hamlyn Foundation.

Dahn, I. (2006). A Metadata Profile to Establish the Context of Small learning Objects: The Slicing Book Approach. . *International Journal on E-Learning.* , *5*(1), 59–66.

Darling-Hammond, L., Zielezinksi, M. B., & Goldman, S. (2014). Using Technology to Support At-Risk Students' Learning. *SCOPE.*

Day, C., Sammons, P., & Gu, Q. (2013). Combining Qualitative and Quantitative Methodologies in Research on Teachers' Lives, Work, and Effectiveness: From Integration to Synergy *Educational Research, 37*(6), 330–342.

Dede, C. (2014). *THE ROLE OF DIGITAL TECHNOLOGIES IN DEEPER LEARNING* DEEPER LEARNING RESEARCH SERIES, Retrieved from https://jfforg-prod-prime.s3.amazonaws.com/media/documents/The-Role-of-Digital-Technologies-in-Deeper-Learning-120114.pdf

Dehnadi, S. (2009). *A cognitive study of learning to program in introductory programming courses.* (PhD), Middlesex University.

Dewey, J. D. (1933). *How We Think: A Restatement of the Relation of Reflective Thinking to the Educative Process.* Boston, MA: D. C. Heath & Co Publishers.

Dol Aher, S. (2015). *An Animated Flowchart with Example to Teach the Algorithm Based Courses in Engineering*. Paper presented at the 2015 IEEE Seventh International Conference on Technology for Education (T4E), Warangal, India. conference paper retrieved from https://ieeexplore.ieee.org/document/7395617

Doolittle, P. R., McNeil, A. L., Krista, P. T., & Scheer, S. B. (2005). Multimedia, Cognitive Load Pedagogy. In S. Mishra & R. C. Sharma (Eds.), *Interactive Multimedia in Education and Training*. Hershey, PA: Idea Publishing Group.

Driscoll, M. P. (2002). How People Learn (and What Technology Might Have To Do with It). . ERIC The Education Resource Center. Available on http://www.ericdigests.org/2003-3/learn.htm.

Duskis, S., Dey, A., Hartley, S., & Wagner, C. (2007). JSAMBA's java version of the SAMBA animation program. Retrieved July 18, 2004, from http://www.cc.gatech.edu/gvu/softviz/algoanim/jsamba/

Eid, C., & Millham, R. (2012). Which Introductory Programming Approach Is Most Suitable For Students: Procedural Or Visual Programming? *American Journal Of Business Education, 5*(2), 173–178.

Emerging Technology. (2000). Learning Theory. Retrieved April 20, 2004, from http://www.emtech.net/learning_theories.htm

Epignosis LLC. (2014). *E-learning Concepts, Trends, Application.* Retrieved from https://www.talentlms.com/elearning/what-is-elearning

ESCOT. (2000). Educational software components of tomorrow website Retrieved March 23, 2004, from Available: http://butterfly.ctl.sri.com/escot/

Fajardo Contreras, W., Gibaja Galindo, E., Marín Caballero, E., & Marín Caballero, G. (2006). An Intelligent Tutoring System for a Virtual E-learning Center. *Current Developments in Technology-Assisted Education (2006).* Retrieved May 14, 2004, from http://www.formatex.org/micte2006/pdf/768-772.pdf

Farrell, J. (2007). *Programming Logic and Design, Introductory* (5th ed.). Boston, MA: Cengage Learning.

FASTIGGI, W. (2014). The SAMR Model. *LEARN TO USE TECHNOLOGY AND USE TECHNOLOGY TO LEARN.* Retrieved 15 July, 2018, from https://technologyforlearners.com/the-samr-model/

Fidge, C., & Teague, D. (2009, 01 January). *Losing their marbles: syntax-free programming for assessing problem-solving skills.* Paper presented at the Eleventh Australasian Conference on Computing Education Wellington, New Zealand.

Field, A. (2012). Discovering Statistics: Repeated Measure ANOVA. 1–21.

Flener, P., & Schmid, U. (2010). Inductive Programming. In C. Sammut & G. I. Webb (Eds.), *Encyclopedia of Machine Learning* (pp. 537–544). New York, NY: Springer.

Flick, U. (2009). *An introduction to qualitative research* (4th ed.). London: Sage.

Forehand, M. (2005). Bloom's taxonomy: Original and revised.. In M. Orey (Ed.), Emerging perspectives on learning, teaching, and technology. . Retrieved June, 2017, from http://projects.coe.uga.edu/epltt/

Fullan, M., & Langworthy, M. (2014). *A Rich Seam: How New Pedagogies Find Deep Learning*. London: Pearson.

Gannon-Leary, P., & Fontainha, E. (2007). Communities of practice and virtual learning communities: benefits, barriers and success factors. Retrieved September 14, 2008, from http://www.elearningeuropa.info/files/media/media13563.pdf

Garner, S. (2003). *Learning Resources and Tools to Aid Novices Learn Programming.* Paper presented at the Informing Science Conference. conference paper retrieved from

Garrison, D. R. (2006). Online community of inquiry review: Social, cognitive, and teaching presence issues. http://files.eric.ed.gov/fulltext/EJ842688.pdf

Garrison, D. R. (2017). COGNITIVE PRESENCE AND CRITICAL THINKING. http://www.thecommunityofinquiry.org/editorial5

Garrison, D. R., & Arbaugh, J. (2007). Researching the community of inquiry framework: Review, issues, and future directions. *The Internet and Higher Education, 10*(3), 157–172.

Garrison, D. R., & Cleveland-Innes, M. (2005). Facilitating Cognitive Presence in Online Learning: Interaction Is Not Enough. *The American Journal of Distance Education, 19*(3), 133–148.

Gaytan, J. (2008). Using On-Demand Help Features Effectively in Interactive Learning Environments: A Literature Review. *Type II Applications.* https://www.tandfonline.com/doi/abs/10.1300/J025v23n01_14

Gilbert, J., Wilson, D., & Gupta, P. (2005). Learning C With Adam. *International Journal on E-Learning, 4*(3), 337–350.

Gomes, A., & Mendes, A. (2007). *Learning to program – difficulties and solutions.* Paper presented at the International Conference on Engineering Education ICEE 2007. conference paper retrieved from

Graviss, K. (2005). *Online supplemental instruction and assessment of OOP (object-oriented programming) concepts.* Paper presented at the Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education Montreal, Canada Conference Paper retrieved from

Grow, G. (1996). *Serving the Strategic Reader: Reader Response Theory and Its Implications for the Teaching of Writing*. Paper presented at the Qualitative Division of the Association for Educators in Journalism and Mass Communication, Atlanta.

file:///D:/PhD/Middlesex%20PhD/References/2017/MayBefore/GrowCognitive.pdf

Guernsey, L. (2012). Can your preschooler learn anything from an iPad App? Slate. http://www.lisaguernsey.com/articles&volume=1&issue=can-preschoolers-learn-anything-from-an-app

Halford, G. S., & Andrews, G. (Eds.). (2010). *Information -Processing Models of Cognitive Development.* Hoboken, NJ: John Wiley.

Havenga, & Maria, H. (2009). *An investigation of students' knowledge, skills and strategies during problem solving in object oriented programming.* University of South Australia, Australia. Retrieved from http://uir.unisa.ac.za/handle/10500/2416

Heersmink, R. (2013). A taxonomy of cognitive artifacts: function, information, and categories. *Review of Philosophy and Psychology, 4*(3), 465–481.

Hobbs, D. (2002). A Constructivist Approach to Web Course Design, A Review of the Literature. *International Journal on E-Learning, 1*(2), 60–65.

Holman, E., Ong, J., Domeshek, E., & Mohammed, J. (2002). *Flowcharts as a basic for procedural trainnin and task support.* Paper presented at the World Conference on Educational Multimedia, Hypermedia and Telecommunications 2002, Denver, Colorado.

Hughes, J., Thomas, R., & Scharber, C. (2006). Assessing Technology Integration: The RAT – Replacement, Amplification, and Transformation – Framework. In C. Crawford, R. Carlsen, K. McFerrin, J. Price, R. Weber & D. A. Willis (Eds.), *Society for Information Technology & Teacher Education International Conference 2006* (pp. 1616–1620). Orlando, Florida, USA.

Huitt, W., & HUmmel, J. (2006). Piaget's Theory of Cognitive Development *Educational Psychology Interactive: Cognitive Development.* Retrieved 14 Nov, 2019, from http://chiron.valdosta.edu/whuitt/col/cogsys/piaget.thml

IMS. (2006). Alignment of IMS Learning Resource Meta-data with IEEE Learning Object Metadata, IMS Global Learning Consortium Web site. Retrieved December 12, 2007, from http://www.imsglobal.org/metadata/index.html

ISMAIL, M. N., NGAH, N. A., & UMAR, I. N. (2010). Instructional strategy in the teaching of computer programming: A need assessment analyses. *Journal of Educational Technology, 9*(2), 125–131.

Israel, S. E., & Duffy, G. G. (Eds.). (2011). *Handbook of Reseaerch on Reading Comprension*. New York, NY: Taylor & Francis.

Iwuchukwu, M. O. (2016). Literature Review and Use of Benjamin Bloom's Taxonomy. *International Journal of Humanities and Cultural Studies, 2*(3), 436–444.

Jamieson-Proctor, R., Albion, P., Finger, G., Cavanagh, R., Fitzgerald, R., Bond, T., & Grimbeek, P. (2013). Development of the TTF TPACK Survey Instrument. *Australian Educational Computing, 27*(3), 26–35.

Jamone, L., Saponaro, G., Alexandre Antunes, Ventura, R., Bernardino, A., & Santos-Victo, J. (2015). *Learning Object Affordances for Tool Use and Problem Solving in Cognitive Robots. In AIRO@ AI* IA (pp. 68–82).* Paper presented at the AIRO@ AI* IA. Journal retrieved from

Järvinen, H., Tiusanen, M., & Virtanen, A. (2005). *Convit, a Tool for Learning Concurrent Programming.* Paper presented at the World Conference on E-Learning in Corp., Govt., Health., & Higher Ed. Conference Paper retrieved from

Jonassen, D., Howland, J., Marra, R. M., & Crismond, D. (2010). How Does Technology Facilitate Learning? , May 2017, from https://www.education.com/reference/article/how-does-technology-facilitate-learning/

Jonassen, D. H., & Churchill, D. (2004). Is There a Learning Orientation in Learning Objects? *International Journal on E-Learning., 3*(2), 32–41.

Jong, T. d. (2010). Cognitive load theory, educational research, and instructional design: some food for thought. *Instructional Science, 38*(2), 105–134.

Jude, L. T., Kajura, M. A., & Birevu, M. P. (2014). Adoption of the SAMR Model to Asses ICT Pedagogical Adoption: A Case of Makerere University *International Journal of e-Education, e-Business, e-Management and e-Learning, 4*(2), 106–115.

Jyotsana, G., & Ajay, S. (2016). Concept Map: A visual Learning Tool. *The International Journal of Indian Psychology, 4*(1), 188–193.

Kannan, S., A. Muthu, K., & Jennifer, P. (2015). Effectiveness in Teaching 'Introductory Programming' for the Students of Engineering and Technology: The Need of Hour. *Indian Journal of Science and Technology, 8*(31), 1–5.

Kay, R. H., & Lauricella, S. (2011). Unstructured Versus Structured Use of Laptops in Higher Education *Journal of Information Technology Education Innovations in Practice, 10*(2011).

Kaya, e., & Akdemir, A. S. (Eds.). (2016). *COGNITIVE LEARNING THEORIES: Approach and Model.* Ankara: Çözüm Eğitim Yayıncılık. .

Kedar, S., & Thakare, S. (2009). *Principles of Programming Languages* (4th ed.). Pune: Technical Publications Pune.

Keengwe, J., Onchwari, G., & Wachira, P. (2008). The Use of Computer Tools to Support Meaningful Learning. *Association for the Advancement of Computing In Education Journal, 16*(1), 77–92.

Keith, N., Aidan, M., & Susan, B. (2015). *Examining the role of cognitive load when learning to program*. Paper presented at the 3rd International Workshop on Eye-Movements in Programming, Joensuu, Finland.

Kelly. (2016). The RAT Model. Retrieved 20 July, 2018, from https://instructionaldesignbykelly.wordpress.com/2016/03/01/the-rat-model/

Kelly. (2017). The Evolution of Bloom's Taxonomy and How it Applies to Teachers Today. Retrieved 15 May, 2018, from https://instructionaldesignbykelly.wordpress.com/author/kellynwymer/

Kennedy, C. K., & Shaughnessy, M. F. (2008). Q & A with Ed Tech Leaders Interview with Allan Collins. *Journal of Education Technology, May-June*, pp. 49–59.

Keogh, R., & Pearson, J. (2011). Mental Imagery and Visual Working Memory. *6*(12), pp.1–8. https://doi.org/10.1371/journal.pone.0029221

Kinchin, I. M. (2014). Concept Mapping as a Learning Tool in Higher Education: A Critical Analysis of Recent Reviews. *The Journal of Continuing Higher Education, 62*(2014), 39–49.

Klett, F. (2007). *Advance Technology Standards for Learning and performance and Their Impact*. Paper presented at the 8th Internation Conference on Information Technology Based Higher Edcuationa and Training., Kumamoto City, Japan. Conference Paper retrieved from

Kodat, R. (2007). Jean Piaget's theory of cognitive development. Retrieved April 24, 2007, from http://www.essortment.com/all/jeanpiagettheo_rnrn.htm

Koehler, M. J., & Mishra, P. (2009). What is technological pedagogical content knowledge? *Contemporary Issues in Technology and Teacher Education, 9*(1), 60–70 Association for the Advancement of Computing in Education (AACE).

Konza, D. M. (2012). Researching in schools: Ethical issues. . *International Journal of the Humanities, 9*(6), 77–86.

Kunimune, H., & Niimura, M. (2014). Preliminary Evaluation of a Problem-posing Method in Programming Classes. *Procedia Computer Science, 35*, 794–802.

Kunyanuth, K., Pubet, K., & Pattarapan, R. (2014). Developing an Adaptive Web-Based Intelligent Tutoring System using Mastery Learning Technique. *Social and Behavioral Sciences, 19*(1), 686–691.

Kurt, S. (2015). ADDIE Model: Instructional Design. *Educational Technology.* Retrieved 12 July, 2018, from https://educationaltechnology.net/the-addie-model-instructional-design/

Laerd.com. (2013a). Independent t-test for two samples. *Statistic Laerd.com.* Retrieved 1 Jan, 2018, from https://statistics.laerd.com/statistical-guides/independent-t-test-statistical-guide.php

Laerd.com. (2013b). Repeated Measures ANOVA. *Statistic Laerd.com.* Retrieved 1 Jan, 2018, from https://statistics.laerd.com/statistical-guides/repeated-measures-anova-statistical-guide.php

Lahtinen, E., Ala-Mutka, K., & Järvinen, H.-M. (2005). A Study of the Difficulties of Novice Programmers. *SIGCSE Bulletin archive, 37*(3), 14–18.

Lau, W. F., & Yuen, H. K. (2009). Toward a framework of programming pedagogy. In M. Khosrow-Pour (Ed.), *Encyclopedia of Information Science and Technology* (2nd ed., Vol. VIII, pp. 3772–3777). Hershey, PA: Idea Group Inc.

Learn NC article. (2015). Critique of TPACK Framework and The SAMR Model. *Emerging Web and Mobile Technology.* Retrieved July 1, 2018, from https://gradgirl149.wordpress.com/2015/09/02/critique-of-tpack-framework-and-the-samr-model/

Lederman, N. G., & Abell, S. K. (2014). *Handbook of Research on Science Education, Edition: Volume II Quantitative Research Designs and Approaches* Vol. II.

Lee, J. (2007). The effects of visual metaphor and cognitive style for mental modeling in a hypermedia-based environment. *Interacting with Computers, 19*(5–6), 614–629.

Lee, S., & Smith, C. A. M. (2012). Criteria for Quantitative and Qualitative Data Integration: Mixed-Methods Research Methodology. *Journal of Computers, Informatics, Nursing, 30*(5), 251–256.

Lew, M. S., Sebe, N., Djeraba, C., & Jain, R. (2006). Content-based multimedia information retrieval: State of the art and challenges *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP), 2*(1), 1–9.

Linn, M. C., & Dalbey, J. (2010). Cognitive consequences of Programming Instruction: Instruction, Access, and Ability. In E. Soloway & J. C. Spohrer (Eds.), *Studying the Novice Programmer* (pp. 57–83). New York, NY: Taylor & Francis.

Linnenbrink, E. A., & Pintrich, P. R. (2003). The role of self-efficacy beliefs in student engagement and learning in the classroom. *Reading and Writing Quarterly, 19*, 119–137.

Liu, J., & Cheng, X. (2008). An Evaluation of the Learning of Undergraduates Using E-Learning in a Tertiary Institution in China. *International Journal on E-Learning, 7*(3), 427–447.

Ljuboevic, D., Cook, J., & Boyle, T. (2004). *Towards Contextual Metadata to Increase Learning Object Technology Effectiveness.* Paper presented at the World Conference on Educational Multimedia, Hypermedia and Telecommunications, Lugano, Switzerland. Conference Paper retrieved from

Louis Cohen et al. (2011). Research methods in education. *Birtish Journal of Eduational Technology, 42*(5).

Lu, Z.-L., & Dosher, B. A. (2007). Cognitive psychology. *Scholarpedia, 2*(8), 27–69.

Lukasiak, J., Agostinho, S., Burnett, I., Drury, G., Goodes, J., Bennett, S., . . . Harper, B. (2004). A Framework for the Flexible Content Packaging of Learning Objects and Learning Designs. *Journal of Educational Multimedia and Hypermedia., 13*(4), 465–481.

Lynch, M. (2016). 20 Top Virtual Reality Apps that are Changing Education. *Best of Best List.* Retrieved 17 July, 2017, from http://www.thetechedvocate.org/20-top-virtual-reality-apps-that-are-changing-education/

MarioD. V. (2017). Cognitive Learning: A Guide to Types of Learning. Retrieved 20 Nov, 2019, from https://blog.cognifit.com/cognitive-learning-an-education-guide-to-types-of-learning/

Mason, R., & Cooper, G. (2013). Mindstorms robots and the application of cognitive load theory in introductory programming. *Computer Science Education, 23 1023*(4), 296–314.

Mayer, R. E. (Ed.). (2013). *Teaching and Learning Computer Programming: Multiple Research Perspectives*. Mahwah, NJ: Lawrence Erlbaum Associates Inc.

Mayer, R. E., & Fay, A. L. (2013). Learning LOGO: A cognitive Analysis. In R. E. Mayer (Ed.), *Teaching and Learning Computer Programming: Multiple Research Perspectives*. New York, NY: Routledge.

McGee, P. (2002). *Cognitive Tools in Web-based Learning Environments: Implications for Design and Practice.* Paper presented at the World Conference on Educational Multimedia, Hypermedia and Telecommunications 2002 Denver, Colorado, USA.

McGriff, S. J. (2005). *Instructional System Design (ISD): Using the ADDIE Model*. Instructional Systems, College of Education.

McLeod, S. (2014). Lev Vygotsky: Simply Psychology. *Developmental Psychology.* Retrieved May, 2017, from https://www.simplypsychology.org/vygotsky.html

McLeod, S. (2018). Information Processing. https://www.simplypsychology.org/information-processing.html

Means, B., Toyama, Y., Murphy, R., Bakia, M., & Jones, K. (2009). *Evaluation of evidence-based practices in online learning: A meta-analysis and review of online learning studies.* USA: Retrieved from http://www2.ed.gov/rschstat/eval/tech/evidencebased-practices/finalreport.pdf.

Meng-Jung, T. (2009). The Model of Strategic e-Learning: Understanding and Evaluating Student e-Learning from Metacognitive Perspectives *Journal of Educational Technology & Society, 12*(1), 34–48.

MERLOT. (2000). Multimedia educational resource for learning and on-line teaching website Retrieved March 23, 2006, from Available: http://www.merlot.org/

Mhashi, M. M., & Alakeel, A. (2013). *Difficulties Facing Students in Learning Computer Programming Skills at Tabuk University.* Paper presented at the 6th WSEAS World Congress.

Michael, d. R. (2008). *Teaching Programming Strategies Explicitly to Novice Programmers.* (Doctor of Philosophy Dissertation), University of Southern Queensland.

Mishra, P., & Koehler, M. J. (2006). Technological pedagogical content knowledge: A framework for integrating technology in teacher knowledge. *Teachers College Record, 108*(6), 1017–1054.

Mnguni, L. E. (2014). SpringerPlus: The theoretical cognitive process of visualization for science education (Publication no. doi: 10.1186/2193-1801-3-184). Retrieved June 2017, from US National Library of Medicine National Institutes of Health https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4000355/

Mogharreban, N., & Guggenheim, D. (2008). Learning Pod: A New Paradigm for Reusability of Learning Objects. *Interdisciplinary Journal of E-Learning and Learning Objects, 4*(2008), 303–315.

Mohorovičić, S., & Strčić, V. (2011). *An Overview of Computer Programming Teaching Methods*. Paper presented at the Central European Conference on Information and Intelligent Sysyem. Retrieve form: http://archive.ceciis.foi.hr/app/index.php/ceciis/2011/schedConf/presentations

Moreno, R., & Park, B. (2010). Cognitive Load Theory: Historical Development and Relation to Other Theories. In J. L. Plass, R. Moreno & R. Brunken (Eds.), *Cognitive Load Theory* (pp. 9–29). New York, NY: Cambridge University Press.

Morrosty, P. (2017). *Pedagogy for Creative Problem-solving*. New York, NY: Routledge.

Mullier, D., Hobbs, D., & Moore, D. (2002). Identifying and using hypermedia browsing patterns. *Journal of Educational Multimedia and Hypermedia, 11* (1), 1031–1050.

Nadir, Ç., Yavuz, E., & Mehmet, Ş. (2016). CHAPTER 3: COGNITIVE LEARNING THEORIES. In K. Zeki & A. Selçuk (Eds.), *LEARNING AND TEACHING : THEORIES, APPROACHES AND MODELS* (2 nd ed.). Ankara, Türkiye: Çözüm Publishing.

Neji, M., & Ammar, M. B. (2007). Agent-based Collaborative Affective e-Learning Framework. *Electronic Journal of e-Learning 5*(2), 123–134.

Newton, D., & Ellis, A. (2006). *A model for e-learning integration*. Paper presented at the World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education Hawaii, USA. Conference Paper retrieved from

Nguyen, T. (2008). *Student Mental Models, Working Memory, and Cognitive Overload during a Spreadsheet Activity.* Paper presented at the Society for Information Technology and Teacher Education International Conference 2008 Las Vegas, Nevada, USA.

Nurazian, M. D., Suzana, B., Haslizatul, F. M. H., & Ismassabah, I. (2007). *Development of Instruments for Measuring Learning Attitudes Towards Computer Programming*. Paper presented at the National Conference on Programming Science (ATUR 07), Pacific KLIA, Selangor.

OECD. (2015). Students, Computers and Learning: Making the Connection (Online Database). from OECD Publishing http://dx.doi.org/10.1787/9789264239555-en

Okojie, M. C. P. O., Olinzock, A. A., & Okojie-Boulder, T. C. (2006). The pedagogy of technology integration. . *Journal of Technology Studies, 32*(3), 66–71.

Olubunmi, S., & Adesope, O. (2007). *Using Cognitive Artifacts for Learning.* . Paper presented at the World Conference on Educational Multimedia, Hypermedia and Telecommunications 2007 paper retrieved from

Oswalt, A. (2010). Intelligence: The Measurement Of Cognitive Capabilities. Retrieved 23 Dec, 2017, from https://www.mentalhelp.net/articles/intelligence-the-measurement-of-cognitive-capabilities/

Pahlevi, M. R. (2017). Teachers' Pedagogical Knowledge and the Teaching Profession. In Susanto (Ed.), (Vol. 5(2)): State University of Surabaya.

Pang, K. M. (2003a). *An odyssey of e-learning model design*. Paper presented at the Society for Information Technology & Teacher Education 14th International Conference, Albuquerque, USA. Short Paper retrieved from https://www.learntechlib.org/p/18471/

Pang, K. M. (2003b). *A Survey on the use of web-based learning resources by Hong Kong secondary teachers in classroom teaching.* Paper presented at the Information Technology & Teacher Education 14th International Conference, Albuquerque, USA.

Pang, K. M. (2007). *A Model of E-learning Object for the Learning of Program Semantic.* . Paper presented at the 8th International Conference on Information Technology Based Higher Education and Training, Kumamoto City, Japan.

Pang, K. M., & Au, W. K. (2002). *E-learning and HK schools.* Paper presented at the International Conference on Computer Education 2002, Auckland, New Zealand.

Pappas, C. (2014). Instructional Design Models and Theories: Information Processing Theory *Instructional Design.* Retrieved July, 2017, from https://elearningindustry.com/information-processing-theory

Pappas, C. (2016). Developing Reusable Learning Objects: 9 Characteristics To Consider *E-learning Design and Development.* Retrieved 10 July, 2017, from https://elearningindustry.com/developing-reusable-learning-objects-characteristics-consider

Pappas, C. (2017). The Top 6 Animated Video Software In The eLearning Market (2017 Update) *eLearning Software.* Retrieved 12 July, 2017, from https://elearningindustry.com/the-top-6-animated-video-software-in-the-elearning-market

Peach-Squibb, J. (2013). Constructivist Learning. http://edutechwiki.unige.ch/en/Constructivist_learning

Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., . . . Paterson, J. (2007). A survey of literature on the teaching of introductory programming. *ITiCSE-WGR' 07: Working group reports on ITiCSE on Innovation and technology in computer science education, 36*(4), 204–223.

Pedroni, M. (2003). *Teaching Introductory Programming with the Inverted Curriculum Approach.* Diploma thesis, Department Computer Science, ETH Zurich. Retrieved from http://74.125.153.132/search?q=cache:1jOM9mwO_t0J:se.ethz.ch/projects/michela_pedroni/Report.pdf+Teaching+Introductory+Programming+with+the+Inverted+Curriculum+Approach&cd=1&hl=zh-TW&ct=clnk&gl=hk

Penney, S. R. f. (2014). image of Bloom's Digital Taxonomy as a pyramid with images of possible web tools that could be used to reach each level. Retrieved 12 May, 2018, from http://faculty.indstate.edu/spenney/bdt.htm

Perkins, D. N., Hancock, C., Hobbs, R., Martin, F., & Simmons, R. (1986). Conditions of learning in novices programmers. *Journal of Educational Computing Research, 2*(1), 37–55.

Perkins, D. N., Schwartz, S., & Simmons, R. (1985). Instructional strategies for the problem of novice programmers. In R. E. Mayer (Ed.), *Teaching and learning computer programming: Multiple research perspective* (pp. 153–178). New York, NY: Rougledge.

Pierson, W., & Rodger, S. H. (2008). The JAWAA 2.0 homepage: java and web based algorithm animation. Retrieved July, 24, 2004, from http://www.cs.duke.edu/csed/jawaa2/

Piteira, M., & Costa, C. (2012). *Computer programming and novice programmers*. Paper presented at the International Conference on Information Systems and Design of Communication.

Piteira, M., & Costa, C. (2013). *Learning computer programming: Study of difficulties in learning programming*. Paper presented at the CM International Conference Proceeding Series.

Powers, K., Cooper, S., Goldman, K., Carlisle, M., McNally, M., & Proulx, V. (2006). *Tools for teaching introductory programming: What works?* Paper presented at the 37th SIGCSE technical symposium on Computer science education.

Prieto, A. (2016). 5 Steps to Solving Programming Problems. Retrieved 26 July, 2018, from https://dev.to/aprietof/5-steps-to-solving-programming-problems--502d

PsychoHawks. (2010, September 2015). Theories of cognitive development: Jean Piaget. *Making the Psychology Simple for Everyone.* Retrieved 1 August, 2017, from https://psychohawks.wordpress.com/2010/09/05/theories-of-cognitive-development-jean-piaget/

Psychology Notes HQ. (2018). Vygotsky's Sociocultural Theory of Cognitive Development. *Online Resources for psychology students.* Retrieved 8 Nov, 2019, from https://www.psychologynoteshq.com/vygotsky-theory/

Puentedura, R. R. (2013). SAMR: Moving from enhancement to transformation. Retrieved 12 June, 2018, from http://www.hippasus.com/rrpweblog/archives/000095.html

Queirós, R. (Ed.). (2014). *Innovative Teaching Strategies and New Learning Paradigms in Computer*. Porto: IGI Global.

Quintana, C., Zhang, M., & Krajcik, J. (2010). A Framework for Supporting Metacognitive Aspects of Online Inquiry Through Software-Based Scaffolding. *Educational Psychologist, 40*(4), 235–244.

Rafae, l. H. B., Lars, B., & Mehdi, D. (2006). A Survey of Programming Languages and Platforms for Multi-Agent Systems. *informatica, 30*(2006), 33–44.

Robert, M. G., Floyd, J. F., C, M. P., James, M. L., Eleanor, S., & T, R. (2009). *Survey Methodology* (2nd ed.). Hoboken, NJ: John Wiley.

Roblyer, M. D., & Doering, A. (2010). *Integrating Educational Technology into Teaching* (5th Ed.). Boston, MA: Pearson Education, Inc.

Romrell, D., Kidder, L. C., & Wood, E. (2014). The SAMR Model as a Framework for Evaluating mLearning. *Journal of Asynchronous Learning Network, 18*(2).

Rosenberg, M. (2001). *E-learning: Strategies for delivering knowledge in the digital age*. New York, NY: McGraw-Hill.

Ruman, M., & Prakasha, G. S. (2017). *Application of Technology Integration Matrix (TIM) in teaching and learning of Secondary School Science subjects* IOSR Journal Of Humanities And Social Science (IOSR-JHSS).

Sahin, I., & Kelesoglu, A. (2011). DEVELOPMENT OF SURVEY OF TECHNOLOGICAL PEDAGOGICAL AND CONTENT KNOWLEDGE (TPACK) *The Turkish Online Journal of Educational Technology, 10*(1).

Salkind, N. J. (2010). *Research Design: Encyclopedia of Research Design*. London: Sage.

Sanders, K., & Thomas, L. (2007). *Checklists for grading object-oriented cs1 programs: concepts and misconceptions.* Paper presented at the Innovation and technology in computer science education, New York.

Sarikas, C. (2018). Vygotsky Scaffolding: What It Is and How to Use It. Retrieved 28 Nov, 2019, from https://blog.prepscholar.com/vygotsky-scaffolding-zone-of-proximal-development

Schatsky, D., & Schwartz, J. (2015). *REDESIGNING WORK in an era of COGNITIVE TECHNOLOGIES*. Deloitte Review (DUP1203_DR17).

Scherer, R., & Siddiq, F. (2016). The transfer effects of training computer programming skills on related cognitive skills. PROSPERO 2016:CRD42016051756. http://www.crd.york.ac.uk/prospero/display_record.asp?ID=CRD42016051756

Scherer, R., Siddiq, F., & Sánchez Viveros, B. (2018). The cognitive benefits of learning computer programming: A meta-analysis of transfer effects. *Journal of Educational Psychology*.

Scherer, R., Tondeurb, J., & Siddiqc, F. (2017). On the quest for validity: Testing the factor structure and measurement invariance of the technology-dimensions in the Technological, Pedagogical, and Content Knowledge (TPACK) model. *Computer & Education, 112*(2017), 1–17.

Schmidt, D. A., Baran, E., Thompson, A. D., Mishra, P., Koehler, M. J., & Shin, T. S. (2009). Technological Pedagogical Content Knowledge (TPACK): The Development and Validation of an Assessment Instrument for Preservice Teachers. *International Society for Technology in Education, 42*(2), 123–149.

Schulte, C., & Bennedsen, J. (2006). *What do teachers teach in introductory programming?* Paper presented at the Proceedings of the 2006 international workshop on Computing education research ICER '06. proceeding retrieved from http://localhost/phdWeb/acmpaper/WhatTeachIntordProg.pdf

Settle, A., Vihavainen, A., & Miller, C. S. (2014). *Research Directions for Teaching Programming Online*. Paper presented at the The International Conference on Frontiers in Education: Computer Science and Computer Engineering, Las Vegas.

Shabani, K., Khatib, M., & Ebadi, S. (2010). Vygotsky's Zone of Proximal Development: Instructional Implications and Teachers' Professional Development. *English Language Teaching,, 3*(4), 237–248.

Shank, P. (2005). The value of multimedia in learning. Retrieved 6 November, 2008, from http://www.adobe.com/designcenter/thinktank/valuemedia/

Shanmugasundaram, V., Juell, P., Hill, C., & Nygard, K. (2007). *Cognitive Model for Learning Java.* Paper presented at the World Conference on Educational Multimedia, Hypermedia and Telecommunications 2007.

Shanmugasundaram, V., Juell, P., Jayasuriya, N., & Benson, J. (2004). *Development of ViewJ – a Visualization Builder for Object Oriented Programming Development Environment.* Paper presented at the World Conference on Educational Multimedia, Hypermedia and Telecommunications Lugano, Switzerland.

Shapiro, A. M., Sims-Knight, J., O'Rielly, G. V., Capaldo, P., Pedlow, T., Gordon, L., & Monteiro, K. (2017). Clickers can promote fact retention but impede conceptual understanding: The effect of the interaction between clicker use and pedagogy on learning. *Computer & Education, 111*(2017), 44–59.

Sheard, J., & Carbone, A. (2005). *CADAL Quiz: Providing support for self-managed learning?* Paper presented at the World Conference on the WWW and Internet San Antonio, Texas.

Shoemaker, D. (2009). About Instructional Design. Retrieved 12 July, 2018, from http://www.instructionaldesignexpert.com/index.html

Skinner, B. (1974). *About behaviorism*. New York, NY: Vintage.

Snape, P., & Fox-Turnbull, W. (2011). Perspectives of authenticity: implementation in technology education. *International Journal of Technology and Design Education, 23*, 51–68.

Sooriamurthi, R. (2009). *Introducing Abstraction and Decomposition to Novice Programmers*. Paper presented at the 14th ACM-SIGCSE Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE-2009, Paris, France. ACM Article retrieved from

Soto, V. J. (2013). Which instructional design models are educators using to design virtual world instruction? *Journal of Online Learning & Teaching, 9*(3), 364 –375.

Spector, J. M., Lockee, B. B., Smaldino, S., & Herring, M. (Eds.). (2013). *Learning, Problem Solving, and Mindtools: Essays in Honor of David H. Jonassen*. London: Routledge.

Sternberg, R. J., & Zhang, L. F. (Eds.). (2009). *Perspectives on thinking, learning, and cognitive styles*. Mahwah, NJ: Lawrance Erlbaum Associates, Inc.

Sun-Ongerth, Y. (2012). *Exploring Novice Teachers' Cognitive Processes Using Digital Video Technology: A Qualitative Case Study.* Georgia State University Retrieved from http://scholarworks.gsu.edu/msit_diss/108/

Takaya, K. (2008). Jerome Bruner's Theory of Education: From Early Bruner to Later Bruner. *interchange, 39*(1), 1–19.

Tastle, W. J., White, B. A., & Shackleton, P. (2005). E-learning in higher education: the challenge, effort, and return on investment. . *International Journal on E-Learning, 4*(2), 241–251.

Teague, D. (2011). *Pedagogy of Introductory Computer Programming: A People-First Approach.* (Master of Information Technology), Queensland University of Technology, Austrialia.

The American Academy of Pediatrics. (2013). *Children, adolescents, and the media. Pediatrics 132, 958-961. doi: 10.1542/peds.2013–2656.*

Theiner, G. (2013). *Transactive Memory Systems: A Mechanistic Analysis of Emergent Group Memory*. Manuscript.

Thramboulidis, K. C. (2003). A Constructivism-Based Approach to Teach Object-Oriented Programming. *Journal of Informatics Education and Research, 5*(1), 1–14.

TIM Development Team University of South Florida. (2015). The Technology Integration Matrix: A project of the Florida Center for Instructional Technology. *TIM.* Retrieved 1 July 2018, from https://fcit.usf.edu/matrix/

Tolman, E. C. (1948). Cognitive maps in rats and men. *Psychological Review, 55 (4)*, 189–208. http://psychclassics.yorku.ca/Tolman/Maps/maps.htm

Triple-E framework. (2013). About Triple-E Framework. Retrieved 30 July, 2018, from https://www.tripleeframework.com/

Tsai, C.-W., Shen, P.-D., & Fan, Y.-T. (2013). Research trends in self-regulated learning research in online learning environments: A review of studies published in selected journals from 2003 to 2012. *British Journal of Educational Technology, 44*(5), E107–E110.

Tuğba, B. (2016). How to perpare teachers to teach with technology: Technology teachers' case. *The Many Faces of TPACK/Computer Education Teacher Education.* Retrieved 20 Jan, 2017, from https://en.wikibooks.org/wiki/The_Many_Faces_of_TPACK/Computer_Education_Teacher_Education

Tuparov, G., Tuparova, D., & Tsarnakova, A. (2012). Using Interactive Simulation-Based Learning Objects in Introductory Course of Programming. *Social and Behavioral Sciences, 46*(2012), 2216–2280.

Tutkun, O. F., Guzel, D., Koroğlu, M., & Ilhan, H. (2012). Bloom's Revized Taxonomy and Critics on It. *TOJCE: The Online Journal of Counselling and Education, 1*(3), 23–30.

Uwe, K., Rolf, P., Sabine, S., & Berthold, M. (2010). Developing and evaluating a strategy for learning from animations. *Learning and Instruction, 20*(5), 424–433.

Vaala, S., Ly, A., & Levine, M. (2015). *Getting a Read on the App Stores: A market scan and analysis of children's literacy apps*. New York, NY: Joan Ganz Cooney Center.

Varlamis, I., & Apostolakis, I. (2006). The Present and Future of Standards for E-Learning Technologies. *Interdisciplinary Journal of Knowledge and Learning Objects, 2*(2006).

Verenikina, I. M. (2010). *Vygotsky in Twenty-First-Century research*. Paper presented at the Proceedings of ED-MEDIA 2010--World Conference on Educational Multimedia, Hypermedia & Telecommunications (pp. 16–25), Toronto, Canada. https://www.learntechlib.org/primary/p/34614/.

Vihavainen, A., Airaksinen, J., & Watson, C. (2014). *A systematic review of approaches for teaching introductory programming and their influence on success*. Paper presented at the the tenth annual conference on International computing education research Glasgow. http://dl.acm.org/citation.cfm?id=2632349

Vila, J., Beccue, B., & Doss, D. (2006). *Active Server Page Programming Tutorial*. Paper presented at the Society for Information Technology and Teacher Education International Conference Orlando, Florida. paper retrieved from

Vygotsky, L. S. (1978). *Mind in society.* Cambridge, MA: Harvard University Press.

Wagner, K. (2008). What makes programming so difficult - and can we make it easier? . *Karsten Wagner's Blog.* Retrieved December 20, 2008, from http://kawagner.blogspot.com/2008/05/whats-makes-programming-so-difficult.html

Wang, Y., & Chiew, V. (2018). Thinking, Problem Solving, Cognition, Second Edition. *Cognitive System Research, 2*, 81–92.

Ware, C. (2013). *Information Visualization: Preception and Design* (3rd Ed.). Waltham, MA: Elsevier.

Wartella, E. (2015). Educational apps: What we do and do not know. *Psychological Science in the Public Interest, 16*(1), 1–2.

Watson, J. B. (2013). *Behaviorism*. New York, NY: W. W. Norton.

Welsh, J. L., Harmes, J. C., & Winkelman, R. (2011). Tech tips: Florida's Technology Integration Matrix. *Principal Leadership, 12*(2), 69–71.

White, C. (2005). A Theory of the Relationships between Cognitive Requirements of Computer Programming Languages and Programmers' Cognitive Characteristics. *Journal of Information Systems Education, 13*(1).

Wiley, D., Waters, S., Dawson, D., Lambert, B., Barclay, M., Wade, D., & Nelson, L. (2004). Overcoming the Limitations of Learning Objects. *Journal of Educational Multimedia and Hypermedia, 13*(4), pp. 507–521.

Williams, R. (2004). Context, content and commodities: E-learning objects. *Electronic-Journal of E-learning, 2*(2), 305–312, #ISSN 1479-4403. http://www.ejel.org/volume-2/vol2-issue2/v2-i2-art7-williams.pdf

Winnips, J. C. (2001). *Scaffolding by design. A model for www learner support.* (PhD dissertation), University of Twente, Netherlands.

Woo, C., & Choi, J. (2007). *WALTS: Web-based Adaptive programming Language Tutoring System.* . Paper presented at the World Conference on Educational Multimedia, Hypermedia and Telecommunications 2000(1), Denver, Colorado.

Wood, D., Bruner, J., & Ross, G. (1976). The role of tutoring in problem solving. *J Child Psychol Psychiatry, 17*(2), 89–100.

Zach, E., & Barr, R. (2016). The role of interactional quality in learning from touch screens during infancy: Context matters. *Frontiers in Psychology*. https://doi.org/10.3389/fpsyg.2016.01264

Zhang, M., & Quintana, C. (2012). Scaffolding strategies for supporting middle school students' online inquiry processes. *Computers & Education, 58*(1), 181–196.

Glossary

Animated flowchart: within the scope of this project, a computer flowchart which is developed with animation. It can animate the logic flow of a designed flowchart.

Augmented Reality (AR): an interactive experience of a real-world environment where the objects that reside in the real world are ‘augmented’ by computer-generated perceptual information, sometimes across multiple sensory modalities, including visual, auditory, haptic, somatosensory and olfactory.

Blackboard System: a web application that allows an institute or college to offer courses online and gives instructors the ability to post supplemental material for hybrid and face-to-face courses. You can log onto Blackboard and access your class materials anywhere you have an Internet connection.

Bloom’s Taxomony: developed by Benjamin Bloom with collaborators Max Englehart, Edward Furst, Walter Hill and David Krathwohl in early 1956. Its purpose was to categorize educational goals as a model by focusing on the cognitive processing outcome of a learning activity and it was described as Taxonomy of Educational Objectives.

Computer-based model: a program that is designed to simulate what might or what did happen in a situation. They are used in many ways including in business, science, physics, biology and astronomy. For use in educational purposes, it also refers to a computer-based learning model.

Concept map: a type of graphic organizer used to help students organize and represent knowledge of a subject. Concept maps begin with a main idea (or concept) and then branch out to show how that main idea can be broken down into specific topics.

Concept visualization: a general term that describes any effort to help people understand the significance of concept by placing it in a visual context by using specific artefacts such as graphics and animation.

Cognitive artefacts: elements used to maintain, display and operate information as a presentation functions affecting human’s cognitive performance. These elements can be different types of technologies available for information processing. They could be multimedia, 2D and 3D animations. Many cognitive tools are designed with modern technologies such as visual reality (VR) and augmented reality (AR).

Cognitive development: a field of study in neuroscience and psychology focusing on a child’s development in terms of information processing, conceptual resources, perceptual skill, language learning and other aspects of the developed adult brain and cognitive psychology. Cognitive development is defined in adult terms as the emergence of ability to consciously cognize and consciously understand and articulate their understanding.

Cognitive indicator: an indicator to show the cognitive performance of a cognitive system. This cognitive performance includes and aims to study knowledge, memory and reasoning variables. A cognitive indicator can be identified with degree of cognitive attainment (or level).

Cognitive modelling: defining a set of reachable levels of cognition in a learning or pedagogical model of a specific subject. It relates to a series of cognitive development steps to achieve while highly connected to information processing. It relies on major steps to achieve a certain cognitive level.

Cognitive load: the effort being used in the working memory. Cognitive load theory differentiates cognitive load into three types: intrinsic, extraneous and germane. Intrinsic cognitive load is the effort associated with a specific topic, extraneous cognitive load refers to the way information or tasks are presented to a learner, germane cognitive load refers to the work put into creating a permanent store of knowledge, or a schema.

Cognitive presence: a presence that requires the observation of one's own learning and how to build and confirm meaning. The extent to which learners can construct and confirm meaning through sustained personal reflection and discourse. Cognitive presence is central to successful student learning. The quality of cognitive presence reflects the quality and quantity of critical thinking, collaborative problem-solving and construction of the meaning.

Cognitive processes: a series of processes that synthesizes the cognitive elements from different perspectives and contexts of cognition in a developing field to transfer existing knowledge to new knowledge (Mnguni, 2014). It is a mental action of acquiring knowledge through experience and senses.

Cognitive tools: generalizable computer tools that are intended to engage and facilitate cognitive processing. They help learners with complex cognitive learning activities and critical thinking. These tools are learner-controlled in the sense that they construct their knowledge themselves using the tools rather than memorizing knowledge. From this perspective, computer systems are 'partners' that stimulate learners or groups of learners to make maximum use of their cognitive potential.

Cloud-based learning: online learning that takes place on the cloud – a virtual space that is not tied to any one computer. There are various cloud-based e-learning systems available, and they bring with them a whole host of benefits for the classroom at all educational levels. Here are seven benefits of cloud-based e-learning.

Debugging: the process of detecting and removing existing and potential errors (also called 'bugs') in a software code that can cause it to behave unexpectedly or crash. To prevent incorrect operation of a software or system, debugging is used to find and resolve bugs or defects.

E-learning: 'a networked learning platform which makes learning contents capable of instant updating, storages/retrieval, distribution, sharing of instruction or information.

E-learning object: an online model consisting of a group of 31 learning objects, each of which is used to deliver a topic of program controls. The e-learning object in this project is used as a cognitive learning tool supporting the cognitive processes in the scaffolding of the pedagogy developed in this project.

HKCAAVQ: The Hong Kong Council for Academic Accreditation (HKCAA) was a statutory body established under the HKCAA Ordinance (Chapter 1150) in 1990. During the 1990s, HKCAA conducted accreditation exercises for institutions such as Hong Kong Baptist University, City Polytechnic of Hong Kong, Lingnan University, Open Learning Institute and the Hong Kong Polytechnic.

Inductive reasoning: a logical process in which multiple premises, all believed to be true or found to be true most of the time, are combined to obtain a specific conclusion. Inductive reasoning is often used in applications that involve prediction, forecasting or behaviour.

IDE: an integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of a source code editor, built automation tools and a debugger. Most of the modern IDEs have intelligent code completion.

Intelligent Agent (IA): an autonomous entity which acts, directing its activity towards achieving goals (i.e. it is an agent), upon an environment using observation through sensors and consequent actuators (i.e. it is intelligent). Intelligent agents may also learn or use knowledge to achieve their goals. They may be very simple or very complex.

Introductory programming: a programming module designed as the first programming module for Computer Science, and Computer Studies programmes. This module was endorsed by the Associate of Computer Machinery Curriculum Committee on Computer Science as early as 1960. It was designed to be focused on novice learners of computer programming, providing them with the basic knowledge and skills of computer programming.

KMS: Knowledge Management System (KMS) was a commercial second-generation hypermedia system, originally created as a successor for the early hypermedia system ZOG. KMS was developed by Don McCracken and Rob Akscyn of Knowledge Systems, a 1981 spin-off from the Computer Science Department of Carnegie Mellon University in Pittsburgh. The purpose of KMS was to let many users collaborate in creating and sharing information within large, shared hypertext, and from the very beginning the system was designed as a true multi-user system.

Learning Agent: a tool in artificial intelligence (AI) that is capable of learning from its experiences. It starts with some basic knowledge and is then able to act and adapt autonomously, through learning, to improve its own performance.

Learning cognition: ‘the mental action or process of acquiring knowledge and understanding through thought, experience, and the senses’. It encompasses many aspects of intellectual functions and processes such as attention, the formation of knowledge, memory and working memory, judgment and evaluation, reasoning and ‘computation’, problem-solving and decision-making, comprehension and production of language. Cognitive processes use existing knowledge and generate new knowledge.

LMS: a learning management system (LMS) is a software application for the administration, documentation, tracking, reporting and delivery of educational courses, training programmes or learning and development programmes. The LMS concept emerged directly from e-learning. Although the first LMS appeared in the higher education sector, the majority of the LMSs today focus on the corporate market. LMSs make up the largest segment of the learning system market. The first introduction of the LMS was in the late 1990s. Learning management systems were designed to identify training and learning gaps, utilizing analytical data and reporting. LMSs are focused on online learning delivery but support a range of uses, acting as a platform for online content, including courses, both asynchronous based and synchronous based. An LMS may offer classroom management for instructor-led training or a flipped classroom, used in higher education, but not in the corporate space.

Learning object: in 2006, the Institute of Electrical and Electronics Engineers (IEEE) quality assurance and standards body defined learning objects as ‘any entity, digital or non-digital, which can be used, re-used or referenced during technology-supported learning’. With this concept, this project defines a learning object as a piece of self-contained, digitized learning information, which can be used and reused in a specific, related learning context internally and externally.

M-learning: also known as mobile learning, this is an educational system. Mobile learning supports, with the help of mobile devices, a continuous access to the learning process. This can be on appliances like your phone, laptop or tablet. You can learn wherever and whenever you want.

Mental operation: the operation that affects mental contents. Initially, operations of reasoning have been the object of logic alone. Pierre Janet was one of the first to use the concept in psychology. Mental operations have been investigated at a developmental level by Jean Piaget, and from a psychometric perspective by J. P. Guilford.

Object-first: use of an object-oriented program (e.g. Java) as the first language for introductory programming. In Computer Studies courses, there is controversy over the use of object-first or procedural-first for introductory programming.

Predicate: in mathematical logic, commonly understood to be a Boolean-valued function P: X→ {true, false}, called the predicate on X. However, predicates have many different uses and interpretations in mathematics and logic, and their precise definition, meaning and use will vary from theory to theory.

Procedural-first the use of procedural program language (e.g. C language) as the first program language for introductory programming. In Computer Studies courses, there is controversy over the use of procedural-first or object-first for introductory programming.

Program control: how a program makes decisions or organizes its activities. Program control typically involves executing particular code, based on the outcome of a prior operation or a user input. In computer programming, there are five basic program controls that are specific to introductory programming. They are sequence control, selective control, iterative control, array control and functional control. This thesis focuses on the later four basic controls.

Program semantics: in programming language theory, semantics is the field concerned with the rigorous mathematical study of the meaning of programming languages. Semantics describes the processes a computer follows when executing a program in that specific language. This can be shown by describing the relationship between the input and output of a program, or an explanation of how the program will be executed on a certain platform, hence creating a model of computation.

Program walk-through: a method for assessing a given algorithm line by line. In learning processes, program walk-through requires the learner to understand the program semantics at a very detailed level. It is intended to enable them to identify problems in the program design.

Pseudocode: uses the structural conventions of a normal programming language, but is intended for human reading rather than machine reading. Pseudocode typically omits details that are essential for machine understanding of the algorithm, such as variable declarations, system-specific code and some subroutines.

Scaffolding: termed by Wood, Burner and Ross (1976) in their early works that described the activities between learners and tutors when assisting the learner to accomplish a difficult task with independence. This term has become widely used in education describing the large variety of practices of pedagogy. There are three major functions of scaffolding applied in the design of the pedagogy in this project. They are conceptual scaffodling, strategic scaffolding and metacognitive scaffolding.

Syntax-free approach: focuses on teaching program semantics with non-programming languages, strategies and mindtools in order to reduce learners' cognitive load of learning by escaping the syntactical issues and learn with IDE software

TPACK. The Technological Pedagogical Content Knowledge (TPACK) is a framework that identifies teachers' knowledge of using technology in their own pedagogy. The TPACK can be used to evaluate the dynamics and transactional relationship between the combination of content, technology and pedagogy owned by a teacher.

Technology integration: the concept of using technology for educational purposes. This concept is evolved with the belief that technology helps learners master the abstractions of a principle and skills through the visualization process. It helps learners grasp the complexity of learning context by varied presentations with alternative forms.

Virtual reality (VR): a technology that makes it possible to experience anything, anywhere, anytime. It is the most immersive type of reality technology and can convince the human brain that it has really existed.

Appendix A: Pre-test and post-test paper for evaluating C/C++ programming

Questions 1-6 Based on Segments A – C answering the questions

Segment A	Segment B	Segment C

if(test_1){
statement-1;}
else {
if (test_2){
statement-2;}
else {
statement-3;}}
	if(test_1){
	statement-1; }
	statement-3;
	if (test_2) {
	statement-2; }
		if(test_1) {
		statement-1; }
		else{
		if (test_2){
		statement-2;}}
		statement-3;

(1)The logical structure of Segment A and B is identical.	Yes □ No □ I do not know □
(2)The logical structure of Segment A and C is identical	Yes □ No □ I do not know □
(3)The logical structure of Segment B and C is identical	Yes □ No □ I do not know □
(4)In the segment A, if both test_1 and test_2 are FALSE, statement-1 will be processed.	Yes □ No □ I do not know □
(5)In the segment B, if both test_1 and test_2 are FALSE, no statements will be processed	Yes □ No □ I do not know □
(6)In the segment C, if both test_1 and test_2 are FALSE, statement-3 will be processed.	Yes □ No □ I do not know □

Questions 7-10

(7)The logical result of if (x==5+y*2) and if (x==(5+y)*2) is identical	Yes □ No □ I do not know □
(8) In general, do-while control always executes one loop more than while control	Yes □ No □ I do not know □
(9) In general, if control can be replaced by switch-case control	Yes □ No □ I do not know □
(10) Basically, for control is recommended to be used on those looping structure that has a known number of loops.	Yes □ No □ I do not know □

Questions 11-21

(11) If I need to write a program to read a list of unknown number of characters, I will use the _____________ program control to do it.
if–else□ switch-case□ while□ do-while□ for□ I do not know□

(12) To simulate the result of flipping a coin, I will use the______________ program control to handle.
if–else□ switch-case□ while□ do-while□ for□ I do not know□

(13) I will use the ______________ data type to store the value of the 52 cards in a suit
Array□ 2D-Array□ String□ Integer□ Character□ I do not know□

(14) I will use the ______________ program control to initiate the values of a suit of card
if–else□ switch-case□ while□ do-while□ for□ I do not know□

(15) Which of followings control is the most appropriate to use for displaying three identical messages?
if–else□ switch-case□ while□ do-while□ for□ I do not know□

(16) Which of followings control is the most appropriate to use for developing the expression of 「Factorial」?
if–else switch-case□ while□ do-while□ for□ I do not know□

(17) Read the statements and answer the questions

```
if (x == 1)
x = 12;
else
x =13
```

If x=12; the last value of x is: 1 □ 12 □ 13 □ I do not know□
If x=1; the last value of x is: 1 □ 12 □ 13 □ I do not know□

(18) How many 'Hello' message(s) will be displayed by the following code segment?

```
int k=0;
while(k<=3){
k=k+1;
printf("Hello");
```

1 □ 2 □ 3 □ 4 □ 5 r□ I do not know□

(19) How many "Hello" message(s) will be displayed by the following code segment?

```
int k=0;
do{
k=k+1;
printf("Hello");
} while (k <= 3)
```

1 □ 2 □ 3 □ 4 □ 5 r□ I do not know□

(20) How many "Hello" message(s) will be displayed by the following code segment?

```
for (int k=1; k<=10; k++)
k=k+1;
printf("Hello");
```

1 □ 2 □ 3 □ 4 □ 5 □ I do not know□

(21) What will be the last value of x, y, and z in the following code segment?

```
int main (String args[]) {
 int x=4;
 int y=1;
 int z = x+ y;
 int y=z;
 cout >> “x= %d, y= %d, z= %d”, x, y, z;}
```

x=4 y=1 z=1□ x=4 y=5 z=5□ x=4 y=4 z=1□ x=4 y=1 z=0□
x=0 y=0 z=0□ I do not know□

Question 22 Based on the following code segment to answer this question

```
char oper;
switch(oper){
 case “+”: x=x+y;
 case “-“ : x=x-y;
 default: break; }
```

If x=4, y=5 and oper= “*”, what will be the last value of x?
1 □ 2 □ 3 □ 4 □ 5 r□ I do not know□

If x=7, y=4 and oper= “-”, what will be the last value of x?
1 □ 2 □ 3 □ 4 □ 5 r□ I do not know□

Question 23 Based on the following code segment to answer this question

```
pic= new char[6][6];
int i, j;
for(i=0; i<6; i++){
for(j=0; j<6; j++){
if (i==j || i==0 || i==5)
pic[i][j]='A';
else
pic[i][j]='B'; } }
```

pic[1][1] = A□ B□ I do not know□
pic[1][5] = A□ B□ I do not know□
pic[5][1] = A□ B□ I do not know□

Question 24 Please circle “four” error in the following code segment

```
public function FixError{
 int a;
 double d=0.0; double e;
 string s;
 printf(“a, e and s is” a+e+s); }
```

Question 25 Based on the following code segment to answer the questions 25(a) and 25(b)

```
int x=0; int input[];
do{ printf(x);
 x=x+2
} while (x==3);
```

25(a) What is the problem in this code segment?

25(b) Can you suggest a method to solve the problem?

Question 26 Based on the following code segment, answer Questions 26(a) to 26(c)

```
int main () {
int i, j, k; int x=0;
char a = “7”;
String str-2 = “2000”;
i = 2;
j = i * 2;
k = 10 – j * 2;
int a = (int) str-1;
double b = int str-2;
if (k >= i){
x=cube(k);}
else{
x=square(k);}
printf(x);
x=square(a);
}

public int square(int y){
return y*y;}
public int cube(int y){
return y*square(y);}
```

(26a) How many variables are in the program? ______________________________.
(26b) What is the value that k holds just after the “line 8”? ___________________________.
(26c) What is the value that x holds just after the “line 12”? __________________________.
(26d) What is the value that x holds just after the “line 15”? __________________________.
(26c) Do you think the statement in “line 10 is valid? __________________________. Why?

Appendix B: Pre-test and post-test paper for evaluating Java programming

Question 1 State and Fix the problem of following codes

```
public class MyGame {
            public static void main(String args[]) {
                        playGame();
 }
playGame() {
System.out.println(args);
            }
}
```

Question 2 Study the following Java program and conclude what will be printed?

2a.

Java program code: a = b-- + --c;	a	b	c
Before Execution	0	4	2
After Execution	?		

2b.

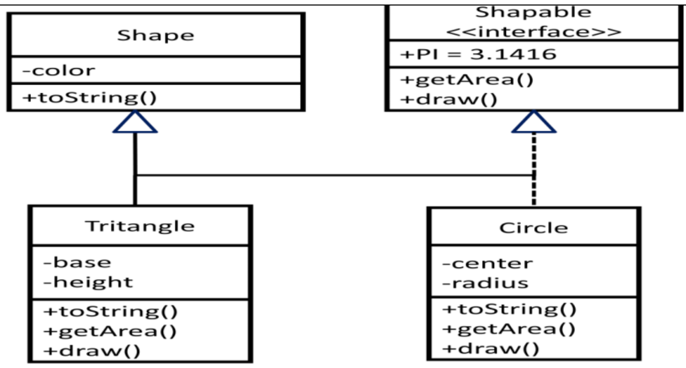
Java program code: a = --b - c++;	a	b	c
Before Execution	0	3	8
After Execution	?		

Question 3 Use for-loop to rewrite the following codes

```
 3a int num1 = 1;
while (num1 <= 1) {
             num1 = num1 * 1;
num1 += 1;
            }
3b int num2 = 10
while (num2 <= 20) {
                        num1 = num1 * num2;
num2 = num2 + 1;
            }
```

Question 4 Given the following class diagram, please complete the TestShape to provide the following output.

Shape of color="red"
Triangle of base=4 and height=5, subclass of Shape of color="blue"
Area is 10.0
Triangle is drawn!
Circle of radius=6.0 center(x,y)=(4.0,5.0), subclass of Shape of color="yellow"
Area is 113.0976
Circle is drawn!




```
public class TestShape {
 public ______ void main(String[] args) {
 ______ s1 = new ______ ("red");
 System.out.println(s1);
 System.out.println();
 _________ s2 = new ________ ("blue", 4, 5); // upcasting
 _________ s3 = new ________ ("yellow", 4, 5, 6); // upcasting
 _________ ss = {s2, s3};
 for (int i=0; i < ss.______; i++) {
             System.out.println(ss___); // Polymorphism
 System.out.println("Area is " + _____.getArea());
             _____.draw(); // Polymorphism
 }
 } }
```

Question 5 State and correct the problem of following code segments.

(5a) loop 20 times

```
            private static void loop1() {
             for(int i = 0; i < 20; i--) {
             System.out.println("In loop "+i); }
             System.out.println("Out of loop");
                        }
```

(5b) loop 10 times

```
            private static void loop2() {
             for(;;) System.out.println("In loop");
             System.out.println("Out of loop");
             }
```

(5c) loop 10 times

```
            private static void loop3() {
                        int i = 0;
                        do {
                                    System.out.println("In loop "+i);
                        } while(i < 10);
                        System.out.println("Out of loop");
            }
```

(5d) loop until i is equal to j

```
            private static void loop6(){
                        Random r = new Random();
                        int i = 0;
                        int j = 10;
                        while (false){
                                    i = r.nextInt(10);
                                    j = r.nextInt(1000);
                                    System.out.println("i = "+i+" and j = "+j);
                                                if (i==j)break;
                                    }
                                    System.out.println("out of loop"); }
```

(5e) loop until i is equal to j

```
	private static void loop7(){
		Random r = new Random();
		int i = 0;
		int j = 10;
		while (true){
			i = r.nextInt(j);
			if (i==j)break;
			System.out.println("i = "+i+" and j = "+j);
			}
		System.out.println("out of loop"); }
```

(5f) loop until i is equal to j

```
	private static void loop7(){
		Random r = new Random();
		int i = 0;
		int j = 10;
		while (true){
			i = r.nextInt(j);
			if (i==j)break;
			System.out.println("i = "+i+" and j = "+j);
		}
		System.out.println("out of loop");
		}
```

(5g) Search a String array for a particular String. If the search String is found return true, else return false.

```
	private static boolean loop12(String[]a, String find) {
			int i;
			for (i=0; i<a.length && a[i]!=find; i++);
			return (a[i]==find);
		}
```

Question 6 Write the output of following code

```
public class T {
public static void main(String[] args) {
x=4;
for(int j=1;j<=x;j++) {
for(int i=0;i<ZZZ;i++)
System.out.print("*");
System.out.println();
}
 }
```

Question 7 Use Array structure to rewrite this program.

```
class LargestNew { public static void main(String[] args)
{
Scanner in = new Scanner(System.in);
System.out.print("How many numbers will you enter? "); i
nt n=in.nextInt();
System.out.println("Enter "+n+ "Numbers ");
int x=in.nextInt();
int largest=x;
for(int i=0;i<n-1;i++)
{
x=in.nextInt();
if (x>largest)
largest=x;
 }
System.out.println("largest is " + largest);
}}
```

Question 8 State at least four true values of I, j and x of the following Boolean express.
(i==0 || i==x-1 || j==0 || j==x-1)

Question 8 Consider the following program:

```
public class D {
public static void main(String[] args) {
Scanner in =new Scanner(System.in); System.out.print("Enter Number>"); int x=in.nextInt(); int
t=0; if (t==0 && t==3) System.out.println("hello"); }
}
```

Which of the following describes its behaviour when the user enters 3?
(a) The program will output hello
(b) The program will output nothing at all
(c) The program will output 0
(d) The program will not compile
(e) None of the above.

Question 9 if the user enters 0, the program does not behave well. Correct it.

```
class Input1New {
public static void main(String[] args) {
Scanner in = new Scanner(System.in);
System.out.print("How many numbers do you want to enter? ");
int k=in.nextInt();
int [ ] num = new int[k];
System.out.print("Now type in the numbers");
for (int i=0;i<k;i++)num[i]=in.nextInt();
for (int i=0;i<k;i++)System.out.println(num[i]); }
}
```

Appendix C: TPACK survey questionnaire

QUESTIONNAIRE

This survey focuses on evaluating teachers' general knowledge of using IT to teach computer programming. There are 25 statements and 2 opening questions defined in this questionnaire. It may last to 20-25 minutes to respond. Please click the appropriate checkbox ☒ for those statements and write your comments to all opening questions in the textbox provided.

PEROSNAL DETAIL

1. What is your highest academic qualification?
☐ PhD/ Doctoral Degree ☐ MPhil/Master's Degree ☐ Bachelor's Degree ☐ Sub-Degree

2. How many years of your teaching experience?
☐ 20 years OR over ☐ 15-19 years ☐ 10-14 years ☐ 5-9 years ☐ below 5 years

3. Have you taught any computer programming subject?
☐ Yes ☐ No

PART A: Please Select the appropriate option by click the checkbox once to ☒

1 I consider my knowledge of using information technology for teaching to be:
☐ Expert ☐ Very Good ☐ Good ☐ Needs Improvement ☐ Poor

2 My skill of using online platform (e.g. WebCT; Moodle) for teaching is:
☐ Expert ☐ Very Good ☐ Good ☐ Needs Improvement ☐ Poor

3 I know about a lot of different technologies, including the up-to-date.
☐ Fully Agree ☐ Agree ☐ Neutral ☐ Not Agree ☐ Fully Disagree

4 I can learn how to use information technology easily.
☐ Fully Agree ☐ Agree ☐ Neutral ☐ Not Agree ☐ Fully Disagree

5 My knowledge of computer programming is:
☐ Expert ☐ Very Good ☐ Good ☐ Needs Improvement ☐ Poor

6 Learning computer programming is difficult for most students.
☐ Fully Agree ☐ Agree ☐ Neutral ☐ Not Agree ☐ Fully Disagree

7 I keep learning new programming languages to enhance my knowledge.
☐ Always ☐ Usually ☐ Sometimes ☐ Seldom ☐ Never

8	I can organize effective class. ☐ Always ☐ Usually ☐ Sometimes ☐ Seldom ☐ Never
9	I can use different teaching approaches for students' individual needs. ☐ Always ☐ Usually ☐ Sometimes ☐ Seldom ☐ Never
10	I can understand students learning difficulties during teaching. ☐ Always ☐ Usually ☐ Sometimes ☐ Seldom ☐ Never
11	I can use different sources, including online materials, to facilitate teaching. ☐ Always ☐ Usually ☐ Sometimes ☐ Seldom ☐ Never
12	Learning computer program semantics is more important than syntax. ☐ Fully Agree ☐ Agree ☐ Neutral ☐ Not Agree ☐ Fully Disagree
13	Teaching computer programming requires emphasis on cognitive development. ☐ Fully Agree ☐ Agree ☐ Neutral ☐ Not Agree ☐ Fully Disagree
14	I can select appropriate materials for teaching computer programming. ☐ Always ☐ Usually ☐ Sometimes ☐ Seldom ☐ Never
15	I can use different strategies to teach different topics in computer programming. ☐ Always ☐ Usually ☐ Sometimes ☐ Seldom ☐ Never
16	Using information technology to learn semantic concept of programming is: ☐ Very Important ☐ Important ☐ Moderate ☐ Less Important ☐ Not Important
17	Using information technology to facilitate cognitive development in teaching computer programming is: ☐ Very Important ☐ Important ☐ Moderate ☐ Less Important ☐ Not Important
18	Encourage students to use technology to improve learning performance is: ☐ Very Important ☐ Important ☐ Moderate ☐ Less Important ☐ Not Important
19	Keeping use up-to-date information technology to facilitate teaching is: ☐ Very Important ☐ Important ☐ Moderate ☐ Less Important ☐ Not Important
20	I can use information technology to facilitate teaching. ☐ Always ☐ Usually ☐ Sometimes ☐ Seldom ☐ Never
21	I like to use technology-based tool to help students to learn program semantics. ☐ Fully Agree ☐ Agree ☐ Neutral ☐ Not Agree ☐ Fully Disagree

22 I believe animated flowcharts help student understanding the program semantics concept.
☐ Fully Agree ☐ Agree ☐ Neutral ☐ Not Agree ☐ Fully Disagree.

23 I like to use learning object to facilitate teaching.
☐ Fully Agree ☐ Agree ☐ Neutral ☐ Not Agree ☐ Fully Disagree.

24 I like to use online learning to facilitate teaching.
☐ Fully Agree ☐ Agree ☐ Neutral ☐ Not Agree ☐ Fully Disagree.

25 I am open mind about using diverse technologies in my teaching.
☐ Fully Agree ☐ Agree ☐ Neutral ☐ Not Agree ☐ Fully Disagree.

PART B. Comments:

1. Do you have any suggestions on teaching approach to effective learning program semantics?

2. Please list the crucial points when you consider using technology for teaching computer programming.

End of Questionnaire. Thanks!

Appendix D: Attitude survey

QUESTIONNAIRE

Pre-Module Attitude Survey

This survey focuses on evaluating students attitude toward to learn computer programming Including 5 statements focus of personal detail, there are total 15 statements defined in this questionnaire. It may last to 15-20 minutes to respond. Please click the appropriate checkbox ☒ for your choice.

PART A: PEROSNAL DETAIL

1. What is your gender?
☐ Male ☐ Female

2. Did you learn computer programming before this module, including learning by your self?
☐ Yes ☐ No

3. If yes in question 2, state what program language(s) you have learned?

____

4. Describe your skill of using computer programming?
☐ okay ☐ good ☐ very good

5. Describe your skill of using the Internet?
☐ okay ☐ good ☐ very good

PART B: Please Select the appropriate option by click the checkbox once to ☒

1	Learning computer programming is fun. ☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree
2	I look forward learning computer programming ☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree
3	Learning computer programming is important to my study. ☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree
4	I feel confidence in learning computer programming. ☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree
5	I what to be professional programmer. ☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree
6	I feel nervous when I know I need to learn computer programming. ☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree
7	I am interested in learning computing. ☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree
8	I am not afraid challenges. ☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree
9	I don't have any difficulty using a computer. ☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree
10	I feel happy as I can learn computer programming in this course. ☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree

End of Questionnaire. Thanks!

QUESTIONNAIRE

Post-Module Attitude Survey

This survey focuses on evaluating students attitude toward to learn computer programming Including 5 statements focus of personal detail, there are total 15 statements defined in this questionnaire. It may last to 15-20 minutes to respond. Please click the appropriate checkbox ☒ for your choice.

Please Select the appropriate option by click the checkbox once to ☒

1 Learning computer programming is fun.
☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree

2 I look forward learning advance computer programming
☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree

3 Learning computer programming is important to my study.
☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree

4 I feel confidence in learning computer programming.
☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree

5 I what to be professional programmer.
☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree

6 I feel nervous when I know I need to learn computer programming.
☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree

7 I am interested in learning computing.
☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree

8 I am not afraid challenges.
☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree

9 I don't have any difficulty in using a computer.
☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree

10 I feel happy if I can learn advance programming in other courses.
☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree

End of Questionnaire. Thanks!

Appendix E: Student survey on using the E-learning object

QUESTIONNAIRE

Post-Module Survey: Satisfaction of Using E-learning

This survey focuses on evaluating students' view upon use the e-learning object to learn. There are 18 statements defined in this research. It may last to 20~25 minutes to respond. Please click the appropriate checkbox ☒ for your choice.

PART B: Please Select the appropriate option by click the checkbox once to ☒

1 The e-learning object is helpful.
☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree

2 With the e-learning object, I feel the instructions are precise.
☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree

3 I am frequently involved in the learning activities.
☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree

4 I always feel the lessons are effective.
☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree

5 I like to use similar cognitive tools in other programming modules.
☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree

6 The learning process suits my own style.
☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree

7 The e-learning object attracts me to learn.
☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree

8 I feel comfortable with the e-learning object.
☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree

9 The e-learning object supports my needs.
☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree

10 I am happy to use the e-learning object to learn.
☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree

2 | Page

11 I feel easy with the animated flowchart.

☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree

12 I feel easy with the animated code-based example.

☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree

13 The animation helps me to better understand the program controls.

☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree

14 The animation is interesting.

☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree

15 I often use the e-learning object in class exercises.

☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree

16 I enjoyed the lesson.

☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree

17 I could finish the class activities quickly.

☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree

18 Overall, I felt the lesson is effective.

☐ 5=Strongly Agree ☐ 4=Agree ☐ 3=Neutral ☐ 2=Disagree ☐ 1=Strongly Disagree

End of Questionnaire. Thanks!