# HINDRANCES TO LEARNING TO PROGRAM IN AN INTRODUCTORY PROGRAMMING MODULE

by

THOMAS SELAKANE MAROKANE

submitted in accordance with the requirements
for the degree of

MAGISTER TECHNOLOGIAE

in the subject of

INFORMATION TECHNOLOGY

at the

UNIVERSITY OF SOUTH AFRICA

Supervisor: Professor Ian Sanders

2017

RESEARCH SUMMARY

*Introductory programming failure rate among students is high worldwide, including in South Africa. The failure rate remains a subject for investigation due to a high number of students who find learning to program difficult. This study evaluates factors that contribute to high failure rates in an introductory programming module at University of South Africa. The study evaluates curriculum, programming syllabus, and personal factors to evaluate reasons for high failure rates. Quantitative and qualitative research approaches are used to identify learning hindrances.*

*The research results show that personal factors are the leading contributing factors, followed by the curriculum and then the programming syllabus. Personal factors relate to time, personal reasons, and commitments; curriculum involves tutorials; and programming syllabus factors are linked to programming concepts and application. The findings have implications for how teaching and learning in introductory programming can be improved. The study provides recommendations for improvement and future studies.*

**Keywords:** Learn to program; introductory programming; higher learning; personal factors; students; teaching; learning; curriculum; programming; challenges; failure; hindrances; educators; lecturers; mixed methods; programming syllabus; module; factors; tutorials

DECLARATION OF THE WORK

I declare that “hindrances to learning to program in an introductory programming module” is my own work and that all the sources that I have used or quoted have been indicated and acknowledged by means of complete references.

I further declare that I have not previously submitted this work, or part of it, for examination at Unisa for another qualification or at any other higher education institution.

Signature Date

ACKNOWLEDGEMENTS

My sincere appreciation and thanks to:

My supervisor Professor Ian Sanders for his patience, guidance from the very first day, the support in every critical step of the research and more importantly the invaluable knowledge he has imparted to allow me to complete this research.

Ms van Heerden for the immense support, time and assistance she has afforded me without any hesitation. The opportunity to work with her allowed me to gain insight into the topic of the study that I could not have gained on my own.

Unisa College Research and Ethics Committee for granting me permission to conduct the survey and interviews with students and lecturers at Unisa.

Unisa students of 2014 Introduction to Interactive Programming ICT1512-14-S1 and ICT1512-14-S2 who provided me with valuable feedback through the questionnaire survey conducted during the study period.

## Table of Contents

LIST OF TABLES

LIST OF FIGURES

# GLOSSARY OF TERMS

Throughout the dissertation, several terms are used that provide context to a particular subject. The terms are defined alphabetically below.

**Challenge:** something that requires a great level of mental effort in order to be completed successfully.

**Course:** recognition for study credit towards an academic qualification at a higher education institution.

**Curriculum:** the guideline of the academic content covered by the educators for students undergoing a particular programme or qualification. It covers what academic content should be taught.

**Distance learning universities:** universities providing distance learning that focuses on teaching methods with the objective of delivering teaching instructions to students who are physically absent in a traditional educational environment such as a classroom.

**Educators:** affiliated members of a higher learning institution who provide education or academic instruction including lecturers, tutors, and other elected members of staff.

**Experience:** skill or knowledge resulting from practical interaction with or observation of an event or facts.

**Higher learning institution:** a higher education and research institution which offers academic degrees.

**Hindrance:** an obstacle that delays or inhibits a desired action.

**Learners:** people who are learning at a university or other higher education institution.

**Learning:** an act of acquiring skill or knowledge through experience or studying or instruction.

**Lecturers:** qualified university educators who teach introductory programming language.

**Module:** similar to course but more specific to Unisa in the case of this study.

**Program:** a set of computer instructions to perform a specific task.

**Programming:** a process of writing a sequence of computer instructions using a specific programming language to perform certain tasks.

**Programming language:** an artificial or high-level language for writing computer programs.

**Students:** people who are learning at a university or other higher education institution.

**Syllabus:** a set of documents that contain topics taught in a specific subject. A syllabus is formulated by teachers unlike a curriculum, which is defined by the institution.

**Teachers:** an affiliated member of a higher learning institution who provides education or academic instruction.

**Teaching:** the activities of imparting skill or knowledge to students.

**Traditional universities:** universities that offer education, where an institution focuses on imparting education to students who are gathered in a traditional classroom, typically on the university's campus.

# 1 INTRODUCTION

Computer programming provides a way to design, develop, and manage computer programs with the objective of instructing a computer to carry out specific activities in order to yield desired behaviours as perceived by the end user. The process of computer programming often needs expertise in the application and management of computer programs in use. Learning to program is universally a challenging and difficult task (Robins, Rountree and Rountree, 2003; Gomes and Mendes, 2007). It remains unclear why globally some students find it easy to learn and pass an introductory programming course while other students have difficulties in learning to program easily or quickly (Jenkins, 2002). Few students find learning to program easy; for this reason, there are high failure rates. Programming is a technique that needs critical thinking and translation of abstract concepts into real-life application (Winslow, 1996). As a result, students who are either generally unable to effectively comprehend and translate abstract concepts or those with limited exposure to programming have difficulties in applying programming practically (Winslow, 1996).

## 1.1 PROBLEM STATEMENT

Studies at several higher education institutions show that the failure rate globally is as high as 32.3% (Bennedsen and Caspersen, 2007; Watson and Li, 2014). Researchers indicate the difficulties introductory programming students have when learning to program and highlight how students find the course the least interesting of all their courses (Hagan, Sheard and Macdonald, 1997; Eckerdal, 2006; Ben-Ari, 2015; Dasuki and Quaye, 2016). The higher failure rates have been for four decades a focus of interest by computer technology educators and researchers. Many are still intrigued by the high number of students who still find computer programming difficult to understand and work with (Tinto, 1975; Roddan, 2002; Robins, Rountree and Rountree, 2003; Bergin and Reilly, 2005; Kinnunen and Malmi, 2008; Derus and Ali, 2012; Schoeman, 2015). As a result, learning institutions continue to explore better ways of teaching students to effectively learn how to program.

Watson and Li (2014), based on the original study by Bennedsen and Caspersen (2007), revisited the failure rates in introductory programming from across the world. The revised study involved analysis of pass rate data from applicable articles and a systematic assessment of introductory programming literature. The data set containing the pass rate data included 161 introductory programming courses from 51 institutions across 15 different countries. Watson and Li's (2014) study, depicted in Figure 1.1 by year and Figure 1.2 by country, indicates a mean global pass rate of 67.7%, which aligns to the report by Bennedsen and Caspersen (2007). The mean global failure and dropout rate is 32.3%, with the South Africa failure rate sitting at around 45%.

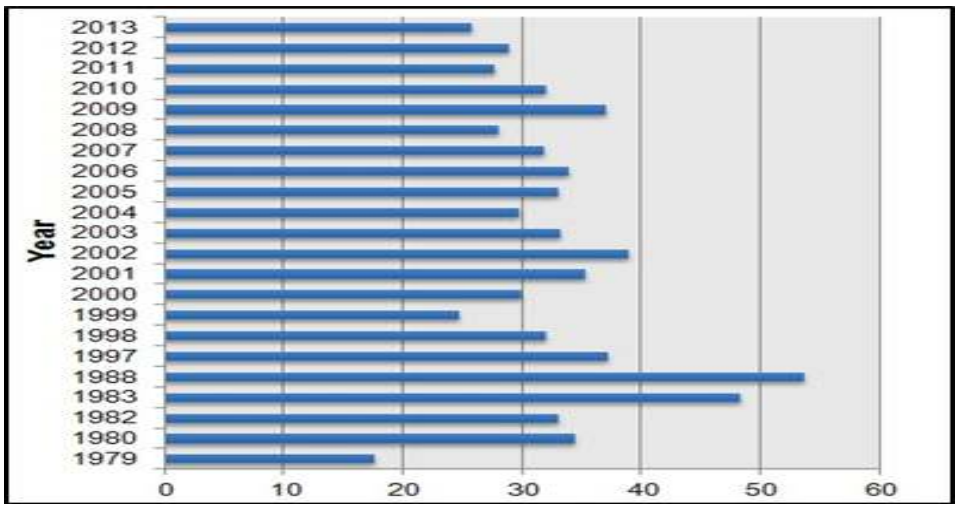


**Figure 1.1: Mean percentage of non-passing students by year**

Source: Watson and Li (2014)

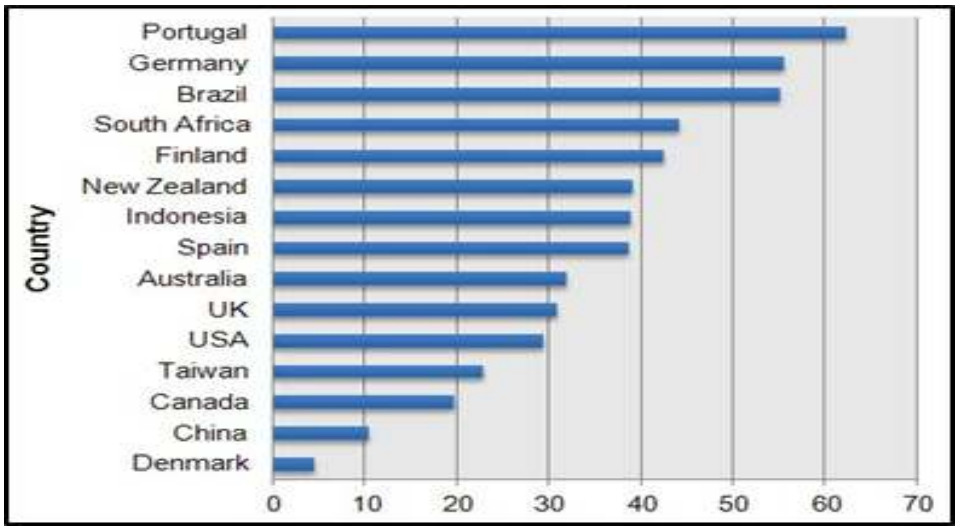


**Figure 1.2: Mean percentage of non-passing students by country**

Source: Watson and Li (2014)

Studies by Bennedsen and Caspersen (2007) and Watson and Li (2014) show that levels of failures have been high for a long period and that the challenge is still a factor experienced by current students. University of South Africa (Unisa), an open distance education university, is among higher learning institutions with a high failure or dropout rate in the introductory programming course (Watson and Li, 2014), something echoed by Goosen and Van Heerden (2013), indicating that the pass rate for "Introduction to Interactive Programming" (ICT1512) at Unisa is very low. Schoeman (2015) also suggests pass rates of as low as 28% at Unisa for the first-year programming module COS1511 as described in Appendix G.

## 1.2 BACKGROUND TO THE STUDY

In the 25 years since the early 90s, countries that have seen significant and sustainable economic growth have built their economies on technology innovation. This strategic practice is quite evident in countries such as South Korea, United States of America, India, and all Scandinavian countries. Higher learning institutions form a crucial component of the development and advancement of any country's technology innovation through educational and research programmes. As a result, it is imperative that institutions of higher learning provide quality education to allow technology custodians to not only keep the country

running but also make it more innovative and competitive globally. Having competition in technology innovation entails having the best computer technology scientists with rigid computer skills (Code, 2016). Programming is one of the most vital skills necessary for the development of technology developers and innovators (Code, 2016). Programming skills are pivotal for students in computer technology, science, and engineering. The importance of these skills in the three fields of practice is referred to in the study by Hwang, et al. (2012).

Increasing the number of programming students who pass programming at Unisa is paramount (Goosen and Van Heerden, 2013; Schoeman, 2015). Academics have been involved in finding better ways to find answers to low success rates. Govender and Grayson (2008) highlight that the performance of students in programming at Unisa has been identified as a matter of concern. In the past, studies have been conducted at Unisa, and the review of the literature shows how involved the focus has been on education styles, philosophy, and the curriculum. It would appear that very limited attention on programming curriculum was given to students in particular. The researcher believes that it all starts with students – what they can do, are prepared to do, and generally conduct towards learning to program. Failure to completely comprehend the factors affecting the students' ability to do well would continue to compromise the overall success of the students, institutions, and the country.

The study used the undergraduate module Introduction to Interactive Programming (ICT1512) at Unisa to conduct the study on the high failure rate at the university. Unisa offers the module ICT1512 as an undergraduate module in the National Diploma in Information Technology (Unisa, 2016). The module has specific outcomes and teaches the students programming using JavaScript as the programming language. The module study is offered over 14 weeks, and during this study period, the students are expected to spend around 8 hours per week studying the module, completing various assessments, including a small practical project, and assignments before they qualify for an examination. Provided below are the key requirements, outcomes, and deliverables of the Introduction to Interactive Programming (ICT1512) module at Unisa derived from the module course outline (Unisa, 2016).

**myUnisa**

It is the online portal used by the university to engage the students on various academic activities including administrative matters programme and course-related engagements. In

the case of Introduction to Interactive Programming (ICT1512), enrolled students can use the portal to receive communiques, receive teaching instructions, and interact with educators and other students through online forums or e-mails (Unisa, 2016).

**Module Outcome**

Upon successfully studying all the theory and completing all the practical exercises and hands-on projects in this module, students will be able to (Unisa, 2016):

- show that they understand problem statements provided by users in various industries. The module content shows the students' use of JavaScript, mathematics, and English to design, develop, and apply end user programs.
- utilise programming principles in the development of a functional program using JavaScript object-orientated methods, event-based graphical user interfaces as well as decision-making, array and looping structures.
- develop functional programs according to a client's requirement specification using web design tools.
- use JavaScript to implement objects designed using the web design tool through the application of user-defined methods, object-orientated designs, graphical user interfaces, functions and classes while program exceptions are managed.

**Study Period**

The module is offered over a semester of 14 weeks including 2 hours allocated for the exam. The recommended time to spend on the module is 8.25 hours per week or 1.18 hours per day (Unisa, 2016).

**Key Assumptions**

Students (Unisa, 2016):

- possess basic computer skills;
- indicate an understanding of the current topics in information and communication technology;
- can take responsibility for their own progress and adapt to the learning environment without any assistance;
- have the ability to learn largely from material written in English; and

- have regular access to both a personal computer and internet access from the first week of the semester.

**Syllabus**

Programming module consisting of nine chapters focusing on the use of JavaScript-based programs (Unisa, 2016). Students learn web page development using HTML, using object-orientated programming with JavaScript, and general programming code management including design, development, error handling, testing, and application (Unisa, 2016).

**Assignments**

The students are required to successfully complete three assignments during the study period (Unisa, 2016).

**Assessments**

Formative assessments take place through the study period (Unisa, 2016). A summative assessment takes place during the examination period (Unisa, 2016).

Formative Assessment:

- Self-Assessment
- Assignment 1 – multiple choice questions
- Assignment 2 – design and develop web pages based on user specifications
- Assignment 3 – participation in the online blog

Summative Assessment:

- Examination Paper – theoretical and application questions
- Examination Project – application of all course outcomes

The students are required to complete a small project by creating a website for a legitimate business of their choice, as part of their summative assessment for the module (Unisa, 2016).

**Discussion Forums and Blogging**

The online forum is a requirement for the module, and certain hours of the module are allocated to the time spent on the forum (Unisa, 2016). The forum is facilitated by the

university as focus groups to post and discuss activities of the module as concerned, which is the programming module in this case (Unisa, 2016). The benefit of the online discussion forum is that students have access to other students and can interact with other students to share information and their experiences (Unisa, 2016). Other benefits are that the forums serve as a communication platform among the peers and a tool for educators to have visibility into students' academic activities (Unisa, 2016).

Students are also expected, as part of their formative assessment, to form and maintain a shared blog and then diarise the activities they have covered (Unisa, 2016).

**Teaching Staff**

Lecturers: Provide assistance with academic work related to the module (Unisa, 2016).

Tutors: Students are given e-tutors who are part of the teaching team (Unisa, 2016). The e-tutor operates in a virtual classroom environment to support students and stimulate discussions.

## 1.3 SIGNIFICANCE OF THE STUDY

The study coincides with the age of digital information where the demand for more faster and increasingly complex computer services is on the rise. Consumers in all part of the globe rely more on technology than ever before regardless of location and time. This demand drives the demand for people who can write computer programs required for managing intelligent and reliable computer applications. Therefore, it becomes imperative for learning institutions, particularly in South Africa, to produce not only the best computer programmers but also adequate programmers in order to remain competitive. The current high dropout rate and failure rate in South Africa, however, undermines the efforts to produce qualified computer programmers at the level of the experts. The study is significant because it contributes towards the understanding of various causes for the high failure or dropout rate in first-year programming students at Unisa.

The outcomes of the study could add to the existing research-based knowledge for additional studies in South Africa on this and similar topics, since only a limited number of studies have been conducted. The study might also provide a foundation for reference to the global

research community due to the type of approach utilised in this study, which entails the use of various combinations of factors emanating from the curriculum, programming syllabus, and personal factors, which allow for subjective and objective investigations into hindrances to learning to program.

In addition, the research outcomes are expected to provide recommendations to Unisa on various potential improvements that can be implemented to bridge the gaps discovered in the study. This is done in order to enhance the quality of education provided by the institution to introductory programming students.

## 1.4 RATIONALE FOR THE STUDY

For many years, programming courses have quite regularly had high failure rates or high dropout rates. The phenomenon around the high number of students who fail or drop out of programming courses is complex, with various contributing factors. The high failure rates have left many educators as well as researchers wondering, despite many years of research (Mead, et al., 2006).

There is general agreement in literature that learning to program is a difficult task (Jenkins, 2002). Students are confronted with many challenges that have been referred to earlier as being inherent in the curriculum, the result of the complex nature of the programming syllabus, and personal reasons. The curriculum is the overall educational content defined by the institution for the overall development of the students and is the same for all teachers. The syllabus is defined by the teachers as part of the course and relates to a particular subject.

Based on the researcher's own assessment of literature and discussions with the lecturer of the modules being studied, the researcher presents the following three key areas of investigation:

- features of the curriculum that prove to be challenging to the students and those that can be key in assisting the students to succeed in the module;
- features of the programming module syllabus that affect learning to program; and

- personal factors that influence the performance of introductory programming students.

There are key differences between the curriculum and the syllabus (Surbhi, 2015; Pediaa, 2016). For clarifying the differences between the two entities, as used in the study, a summary is provided. The curriculum represents the academic content covered by the educators for students undergoing a particular program or qualification. It covers what should be taught and how it should be taught (Pediaa, 2016). The programming syllabus consists of the outline for and documents covered in a particular module. Unlike the programming syllabus, which is formulated by teachers, the curriculum is defined by the institution. The curriculum can affect the outcome of the programming syllabus, since it is the overall governing entity for learning (Surbhi, 2015; Pediaa, 2016). The learning curriculum generally involves the students, lecturers, and the institution. Teachers administering a specific module have no or limited influence in the general administration activities of the institution, qualification programs, and other matters relating to policies and procedures. The view of the study is that these challenges cannot necessarily be managed as part of the programming module syllabus. As a result, it may be better to form a different category of factors emanating from the curriculum. It is believed that if the two entities, curriculum and programming syllabus, are isolated for the study, such approach will compromise the ability to pinpoint sources of problematic areas.

These three areas of investigation are discussed in more detail in the next sections.

### 1.4.1 HIGHER LEARNING CURRICULUM

The curriculum in higher learning is paramount to the students' performance and academic outcomes (Department of Higher Education and Training, 2015). As a result, both learners and educators should embrace the curriculum in place to ensure the success of all participants that form part of the curriculum at the institution.

Institutions of learning are therefore expected to adopt the best curriculum starting with the overall approach the institution takes for offering education to the degree or diploma

students are enrolled for (DHET, 2015). Every module has specific requirements that every student must follow once they register for them to succeed (Unisa, 2016).

Although students may be well aware of the curriculum, continuous support from both the institutions and educators form part of the key components required for students to succeed (Sanderson, Phua and Herda, 2000; Pinar, 2012). It should be expected that every student has a preferred way of learning (Dunn and Dunn, 1992). While some students may prefer to study without any support, others may prefer to receive constant support from fellow students or educators (Dunn and Dunn, 1992).

Educators constitute the third part of the triangle, along with the students and the institution, to ensure that the curriculum produces the results that enable students to improve performance in institutional learning. Educators have a responsibility to completely adopt the curriculum to ensure students' optimal success (DHET, 2015). Since the curriculum is the vehicle for learning at any institution (DHET, 2015), it is crucial to investigate factors involved in ensuring the success of the students.

### 1.4.2 STUDENTS LEARNING TO PROGRAM

Past studies have indicated that students have several difficulties in learning to program (McCracken, et al., 2001; Wilson and Shrock, 2001; Jenkins, 2002; Bennedsen and Caspersen, 2007). The major difficulty experienced by beginners is to use basic building blocks to construct a program (Lahtinen, Ala-Mutka and Järvinen, 2005; Caspersen and Kolling, 2009). Several studies indicate that students regularly perform well during formal assessments but retain very little knowledge after the completion of their studies (Robins, Rountree and Rountree, 2003; Lister, et al., 2004; Butler and Morgan, 2007). Jenkins (2002) brings out that programming is a skill rather than a body of knowledge. This skill and its associated activities must be carried out with the view to contributing to the program, which is the end product (Ben-Ari, 2015).

It is therefore important to find better ways of introducing programming students to concepts that not only help them gain knowledge of what programming is about but more importantly

how this knowledge can be moulded into a skill that can be applied in real-life situations in order to solve a problem.

### 1.4.3 PERSONAL FACTORS AFFECTING STUDENTS' ABILITY TO LEARN

First-year introductory programming students are faced with direct and indirect factors that have an impact on their study performances (Wilson and Shrock, 2001; Jenkins, 2002; Simon, et al., 2006). The factors encompass curriculum-related challenges (including the need to speedily adapt to new ways of learning compared to high school), the teaching style, and the pace at which they need to learn (Tinto, 1987; Roddan, 2002; Butler and Morgan, 2007). Other factors have to do with time management, motivation, aptitude, and cognitive factors (Jenkins, 2002; Kinnunen and Malmi, 2006). These factors taken together could prove very challenging and overwhelming for most first-year students.

## 1.5 RESEARCH QUESTIONS

The high failure rate among introductory programming students at Unisa is a problem that needs attention (Bennedsen and Caspersen, 2007; Watson and Li, 2014). There is a gap in understanding hindrances that lead to this issue (Jenkins, 2002). This study asks questions to provide solutions to the problem identified. The study consists of one main research question and three supporting or secondary questions to help categorise the contributing factors into relevant areas. The outcome of the findings based on the three research questions will help answer the main research question.

### 1.5.1 MAIN RESEARCH QUESTION

The main question forms the basis of the effort to understand various factors that contribute to the hindrances to learning programming:

- What are the factors that contribute to learning hindrances experienced by programming students at Unisa?

### 1.5.2 SECONDARY SUPPORTING QUESTIONS

These questions are key anchors of the study because the outcomes of the three questions were used to answer the key primary question. The three research questions were as follows:

- What are the hindrances related to the university course curriculum?
- What are specific challenges relating to the programming syllabus?
- What are personal factors that have an impact on learning?

## 1.6 AIM OF THE STUDY

Through the evaluation and interpretation of the responses to the research questions, the research aimed to:

- understand general learning challenges faced by students;
- uncover hindrances specific to learning to program that students experience;
- understand challenges associated with the curriculum that contribute to learning barriers; and
- recommend learning and teaching strategies that may form part of the curriculum in an introductory programming module.

## 1.7 OUTLINE OF THE STUDY

The study consists of seven chapters, a complete list of references, and appendices.

Chapter 1 discusses the problem statement, aim of the study, background to the study, significance of the study, rationale of the study, the research questions, and the aim of the study.

Chapter 2 reviews literature to provide perspective into the high rate of failure among introductory programming students at institutions of higher learning. The review of the literature also provides insight into the academic and personal hindrances first-year

programming students experience. This chapter also sets out the challenges relating to programming as a subject and the curriculum adopted by various institutions.

Chapter 3 describes the methodology used in the study. The methodology includes the philosophical view brought into the study and the influence it has on the study, the research approach and the research strategy. Furthermore, covered as part of the methodology are the research methods, research time frame, and techniques for data collection and analysis.

Chapter 4 sets out the quantitative data analysis of the responses from the survey and the presentation of the results. The data analysis and presentation of the results are categorised into the curriculum, the programming syllabus, and personal factors.

Chapter 5 presents the qualitative data analysis of the responses from the survey and the presentation of the results. The data analysis and presentation of the results are categorised into the curriculum, the programming syllabus, and personal factors.

Chapter 6 presents both the quantitative and qualitative data analysis of the responses from the survey and the presentation of the results. The data analysis and presentation of the results are categorised into the curriculum, the programming syllabus, and personal factors.

In Chapter 7, the results are interpreted from the quantitative and qualitative data, and the variations and converging aspects of the study are presented. The outcome of the chapter highlights the hindrances to learning to program in an introductory programming module. In this way, the chapter presents responses to answer the main research question.

In conclusion, Chapter 8 presents the findings of the study and the recommendations based on the findings. The chapter also highlights the limitations of the study including the challenges associated with research formulation, data collection and analysis, and diverging outcomes. Suggestions for future studies and concluding remarks are also given.

The outline of the chapters of the study is illustrated in Figure 1.3.

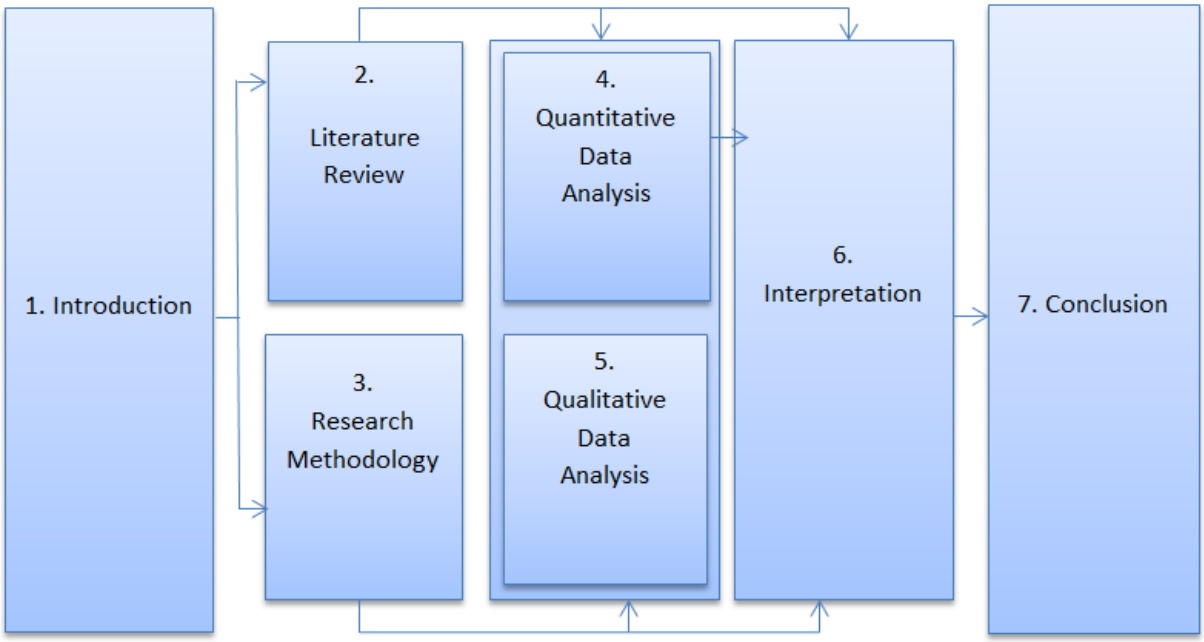


**Figure 1.3: The outline of the study chapters**

A review of literature pertinent to this study follows in the next chapter.

# 2 LITERATURE REVIEW

The previous chapter provided an introduction and background to the study. This chapter presents a summary of various studies carried out in the past on factors that affect introductory students' ability to succeed in programming. This forms the basis of the main research question of the study, which asks, "What are the factors that contribute to learning hindrances experienced by programming students at Unisa?" The chapter further explores gaps in previous related studies and sets out how this study seeks to address the gaps and also expand on prior studies undertaken. The review of literature centres mainly around previous studies on the three secondary research questions relating to the actual programming subject, the curriculum set by the university, and personal factors as outlined in Section 1.6. The chapter starts by providing an overview of contributing factors in Section 2.1. Section 2.2 discusses the curriculum which covers institutional education, the curriculum programme, how educational material and teaching strategy affect students, and finally, educational learning. The reviews of personal factors that affect learning are discussed in Section 2.3, while factors associated with the programming syllabus are outlined in Section 2.4. Finally, a summary of the chapter is set out in Section 2.5.

## 2.1 OVERVIEW OF CONTRIBUTING FACTORS

The review of literature starts by providing a general review and discussion of various factors that impact on learning to program. It is important to understand the scale of what previous studies have covered and subsequently organise the findings in order to formulate a structural approach for this study. The questions found in Chapter 1 are grouped to allow the systematic organisation of related factors, since the literature shows that the reasons for the difficulties are vast and varied (McCracken, et al., 2001; Wilson and Shrock, 2001; Jenkins, 2002; Bennedsen and Caspersen, 2007).

Many studies (Byrne and Lyons, 2001; Boyle, Carter and Clark, 2002; Rountree, Rountree and Robins, 2004; Bennedsen and Caspersen, 2007; Sarpong, Arthur and Amoako, 2013; Watson and Li, 2014) indicate that it is a combination of factors that lead to students' failure to learn to program. Researchers continue to conduct studies on introductory programming with some focusing on the programming aspect of the course (Giangrande, 2007; Koulouri, Lauria and Macredie, 2015; Schoeman, 2015). Other researchers provide the programming learning challenges linked to the educational curriculum approach (Vihavainen, Paksula and Luukkainen, 2011). Some researchers provide insight into personal factors affecting students' performance in learning to program (Xenos, Pierrakeas and Pintelas, 2002; Simon, et al., 2006; Kinnunen and Malmi, 2008; Sarpong, Arthur and Amoako, 2013). Additionally, studies by Xenos, Pierrakeas and Pintelas (2002) and Sarpong, Arthur and Amoako (2013) cover a wide range of factors that contribute to students' failure, which include personal, financial, and educational factors.

To derive a comprehensive plan to ensure the success of the students in programming, a broader assessment of the contributing factors influencing the ability to learn to program is necessary. Factors contributing to students' failure come from many disciplines (Tinto, 1987; Chmura, 1998; Kinnunen and Malmi, 2006; Mhashi and Alakeel, 2013) requiring a multi-disciplinary approach. The varying results highlight the complex nature of what really affects students' ability to succeed and supports the notion that learning to program is a complex matter to comprehend (Matthíasdóttir and Geirsson, 2011).

To effectively manage and refine contributing factors, this study is based on three broad categories: personal, curriculum, and the programming syllabus.

### 2.1.1 CURRICULUM

The first research question poses the question, "What are the hindrances related to the university course curriculum?" The question allows for the grouping of previous studies that show that teaching and learning approaches by the institutions are pivotal to the success in learning to program. The review of literature indicates that various aspects of the curriculum need to be improved to help students succeed in programming (Robins, Rountree and Rountree, 2003; Kinnunen and Malmi, 2008). The improvements include the course design

(Oliva and Gordon, 2012), ample support by the educators (Pinar, 2012), tools (Powers, et al., 2006; Derus and Ali, 2012), and teaching strategy and assessment criteria (Robins, Rountree and Rountree, 2003). Jenkins (2002) even suggests that programming should never be offered until the second year. The most common understanding is that for students to succeed, they need to be ready in many areas starting with personal preparedness to the ability to learn at the expected curriculum level (Tinto, 1987; Conley, 2014). Jenkins (2002, p.53) highlights that "If students struggle to learn something, it follows that this thing is for some reason difficult to learn". This statement implies that understanding the student's situation is paramount to learning. It is therefore important to explore curriculum factors that have a direct impact on students' performance in introductory programming.

### 2.1.2 PROGRAMMING SYLLABUS

The programming syllabus in the study describes the content specific to the programming module. Several studies have focused on the actual programming subject and associated syllabus to highlight challenges specific to learning, development, and application of programs (Butler and Morgan, 2007; Giangrande, 2007; Koulouri, Lauria and Macredie, 2015; Schoeman, 2015). In a study relating to programming, Bennedsen and Capersen (2007, p.111) indicate that "Learning to program is notoriously considered difficult". In the past four decades, learning to program has been a topic of paramount concern in introductory programming (Tinto, 1975; Kember, 2001; Winn, 2002; Bergin and Reilly, 2005; Bennedsen and Caspersen, 2007; Goosen and Breedt, 2012; Watson and Li, 2014; Schoeman, 2015). Studies of Jenkins (2002) and Matthews (2014) support the view that the ability to program requires multiple skills.

Some studies provide an in-depth assessment of the content of the programming subject (Reges, 2006; Schulte and Bennedsen, 2006; Schoeman and Gelderblom, 2016). Giangrande (2007) highlights the importance of looking at multiple aspects that included the type of language, topics to cover, and methodology to use. Other studies reveal that the difficulties associated with programming are not only linked to learning but also to teaching (Robins, Rountree and Rountree, 2003; Mhashi and Alakeel, 2013). Others even view the use of appropriate tools during teaching as an effective way of teaching computer programming (Powers, et al., 2006; Derus and Ali, 2012; Essa, 2016).

This study categorises these studies through the evaluation of the second question of the research, “What are specific challenges relating to the programming syllabus?”

### 2.1.3 PERSONAL FACTORS

Researchers indicate that students’ personal factors play a pivotal role in the success of learning to program (Jenkins, 2002; Winn, 2002; Kinnunen and Malmi, 2008). Boyle, Carter and Clark (2002) assert that students’ success is attributable to personal drive, attitude, and general approach to education rather than prior academic achievements or programming experience. Other researchers argue that prior experience in programming contributes positively towards the success of students in introductory programming (Hagan and Markham, 2000; Byrne and Lyons, 2001; Wilson and Shrock, 2001; Derus and Ali, 2012).

Other studies show that a link exists in student success between self-efficacy (Baldwin, 2016), personal factors (Rogalski and Samurçay, 1990; Wilson and Shrock, 2001; Jenkins, 2002; Simon, et al., 2006; Watson and Li, 2014), students’ motivation (Jenkins, 2002; Alaoutinen and Smolander, 2010), and course outcome expectations (Rountree, Rountree and Robins, 2002; Rountree, Rountree and Robins, 2004; Gomes and Mendes, 2007; Kinnunen, et al., 2007). Wilson and Shrock (2001) conducted a broader investigation that covered 12 determinants for students’ success. These determinants include personal drive, gender, interest, previous education, and programming proficiency. One result of Wilson and Shrock’s (2001) study was that the students’ perception (comfort level) of the difficulty of the programming course was the factor most associated with success. The findings from the studies relate to the research question, “What are the personal factors that have an impact on learning?” which seeks to establish personal factors affecting programming students at Unisa.

The next three sections of this chapter focus, in detail, on specific literature relevant to the three areas of programming syllabus, curriculum, and personal factors affecting learning to program. Factors from the three areas affect learning to program. The three areas of study are derived from the three secondary research questions given in Chapter 1, which will

ultimately answer the main research question, “What are the factors that contribute to learning hindrances experienced by programming students at Unisa?”

## 2.2 CURRICULUM

Curriculum is the first area to be explored based on the research questions asked in the study. In the study, the curriculum refers to the lessons, means, and materials that form part of an institution with an objective to achieve predefined educational outcomes for a specific programme or course. The curriculum may involve the skills and knowledge learners are anticipated to acquire (Nkomo, 2000; Ornstein and Hunkins, 2016). The curriculum involves the learning objectives students are expected to meet, the teaching instructions educators give, the modules that educators offer, as well as the tests and assignments learners undertake (Hsi and Soloway, 1998). The course materials such as study guides, textbooks, videos, articles, and presentations are part of the curriculum, and so are the tests, exams, and assessments used to evaluate learners (Threlkeld and Brzoska, 1994; Mock, 2003; DHET, 2015). A curriculum requires proper formulation and management (Oliva and Gordon, 2012) and does not involve a list of activities to be undertaken as part of the educational programme (Coles, 2003).

In this study, curriculum consists of four aspects:

- Institutional education: the nature and form of education being offered (DHET, 2015).
- Curriculum programme: the guide for both the educators and students on how to perform various functions. The functions are not specific to any module or subject but are the same for all educators and students across the institution concerned (Surbhi, 2015; Pediaa, 2016).
- Educational materials: the tools and media required for learning and teaching (Threlkeld and Brzoska, 1994; Mock, 2003; DHET, 2015).
- Teaching and learning strategy: the structured and principled ways used by educators for teaching and for the acquisition of knowledge by the students in formal education (Egan, Sebastian and Welch, 1991; Mayes and Fowler, 1999).

### 2.2.1 INSTITUTIONAL EDUCATION

Institutional learning can be categorised into different ways that provide interaction between educators and learners including open distance learning (ODL) and contact learning or face-to-face learning (DHET, 2015). Since the study involves Unisa, which is exclusively a distance learning-based university (Unisa, 2016), significant emphasis will be placed on ODL in this section.

The South African Department of Higher Education and Training (DHET) has compiled a comprehensive report on distance education in South African universities (DHET, 2015). The department believes that although progress has been made in the deployment of distance education, there is still much improvement required to allow full exploitation of the benefits associated with distance learning in higher education. The benefits of distance learning are widely publicised in literature (Wedemeyer, 1981).

The department has highlighted important points relating to ODL in South Africa. The DHET (2015) statistics in Table 2.1, adapted from the department, show that Unisa accounts for 90.1% of the total number of students enrolled in ODL in South Africa. In the field of science, engineering, and technology, the university accounts for 92.7% of the total number of ODL education enrolments in all public higher institutions as summarised in Table 2.1 and detailed in Appendix F. The higher number of students represented in this field for Unisa indicates the level of demand for the institution in all fields of study.

**Table 2.1: Distance learning enrolment by major field for 2014**

| Students Enrolled for Distance Learning Per Major Field of Study (2014) | All other Public Higher Education Institutions | University of South Africa | University of South Africa (%) of all Institutions |
|---|---|---|---|
| Science, engineering and technology | 27421 | 25417 | 92.7% |
| Business management | 58692 | 57413 | 97.8% |
| Education | 48060 | 34781 | 72.4% |
| All other humanities and social science | 73342 | 69431 | 94.7% |
| **Total** | 207515 | 187042 | 90.1% |

Source: DHET (2015)

Further in-depth assessment of the report (DHET, 2015) shows that the proportionally high number of enrolments at Unisa relative to other public higher education institutions is also prevalent in undergraduate studies as well. The university had the largest number of enrolled undergraduate certificates, diplomas, and degrees in 2014 (45% of 605 589) in South Africa. The department (DHET, 2015) has also noticed, since 2009, an increase in young students enrolling with Unisa, which asserts that there is a growing interest in distance education among those entering higher education.

In conclusion, ODL offers access to education to many students that have not had the opportunity to enrol for contact learning at various higher education institutions (Bosman and Frost, 1996). In South Africa, there is a significant demand for higher education ODL, including from young people (DHET, 2015). Pityana (2009) highlights challenges pertaining to ODL that include concerted expectations by the national government from institutions of higher education to increase throughput rates. The expectation presents further challenges on the institutions offering higher education to enrol more students. Unisa is among several universities that offer distance education in the country (DHET, 2015). The university accounts for 87.9% of all distance learning enrolments in South Africa. The proportionally high number of students enrolling at Unisa presents a challenge to the institution in dealing with the vast number of students, not only in computing courses but also across all academic disciplines (DHET, 2015). However, it does not mean that the quality of education needs to be compromised (Morrow, 2007).

### 2.2.2 CURRICULUM PROGRAMME

The programme of the curriculum is essential to learning and teaching (Whittington, 1987; Smithson, 2012; DHET, 2015; UNESCO, 2017). It provides a structured way for the institution and educators to provide appropriate support to the students for the purpose of learning (Biggs, 1999; Smithson, 2012). The programme also assists the students in understanding what is expected of them and how they can solicit proper support from the institution and educators. Egan, Sebastian and Welch (1991) and Pinar (2012) indicate that students benefit greatly from a well-designed curriculum programme. The curriculum programme provides the ability for the students, educators, and institution to function in

unison (Smithson, 2012). It requires systematic development, implementation, and maintenance (Pinar, 2012).

Further review of the literature indicates that the curriculum programme includes the development and management of the curriculum institutional objectives and goals (Nkomo, 2000; Ornstein and Hunkins, 2016), activities covered for the course (Nkomo, 2000), a student support system involving the institution and educators (Sanderson, Phua and Herda, 2000), and formation or facilitation of student communities (Tinto, 1997; The United Nations Educational, Scientific and Cultural Organization(UNESCO), 2017).

Another important aspect of the curriculum is the induction (York, Bollar and Schoob, 1993; Bers and Younger, 2014; Martzoukou and Kemp, 2016), which ensures the students are aware of what the responsibilities of higher learning entail. When students make a transition from high school to a higher learning institution, they find many aspects of learning to be different (Tinto, 1987; Roddan, 2002; Butler and Morgan, 2007). New students have to familiarise themselves to a different learning environment (Honey and Mumford, 1982; Furnham, 1995), different teaching styles (Dunn and Dunn, 1992; Mayes and Fowler, 1999), a relatively faster pace of teaching and learning (Jenkins, 2002), and reduced contact with their educators and the level of attention from their educators (Butler and Morgan, 2007). The students, however, find ways of managing these challenges (Tinto, 1987). Researchers have studied factors that affect first-year students' learning once the students enter higher learning institutions after high school (Tinto, 1975; Kember, 2001; Winn, 2002; Derus and Ali, 2012; Bers and Younger, 2014). These studies indicate that students generally struggle to adapt to the rapid transition and change in the learning environment. According to Tinto (1987), around 41 in 100 learners will leave an institution of higher learning within the first two years, that is, before acquiring the qualification they originally sought. Three-quarters of these students leave in the first year. The high attrition rates among students are also highlighted by Watson and Li (2014) and Schoeman (2015).

Institutions of higher learning need to close the gap between high school and themselves by identifying the challenges faced and needs required by the students as part of the academic programme (York, Bollar and Schoob, 1993; Sheard, et al., 2014). The rapid change in learning experience presents the question of how programming students at Unisa manage

this challenging situation in order to succeed in learning to program, given that the study period for an introductory programming module is 14 weeks long.

### 2.2.3 EDUCATIONAL MATERIALS

The curriculum learning materials include a computer and access to the Internet, prescribed books, study guides, and tutorials as requirements (Threlkeld and Brzoska, 1994; Mock, 2003; DHET, 2015). Educational materials provide the most efficient way to issue teaching instructions for learning (Mock, 2003; Vihavainen, Paksula and Luukkainen, 2011). In the case of distance learning education, these materials are even more crucial to success in learning because of the limited contact between educators and learners (Sheard and Carbone, 2007). Matthíasdóttir and Geirsson (2011) found that teachers' recordings of the lectures are the most useful material to work on, followed by study materials issued by the lecturers. Notes taken in class are the least preferred materials for learning to program. Kinnunen and Malmi (2008) highlight that students use study materials before asking for help from others as a strategy of resolving difficult issues during programming.

Literature shows the importance of learning materials within the curriculum (Matthíasdóttir and Geirsson, 2011) and also the critical role the materials have both for learning and teaching if developed relative to the curriculum requirements of the students (Martins, 2012). Learning materials are used by students for knowledge and skills acquisition, as well as for preparation when they perform various activities and are confronted by challenges or to achieve a particular goal (Rowntree, 1992; Bennedsen and Caspersen, 2007; Sheard, et al., 2014). Learning materials also have an influence on the learning ability of the students in learning to program (Keegan, 1990; Sheard, et al., 2014).

### 2.2.4 TEACHING AND LEARNING STRATEGY

In addition to the educational materials discussed in Section 2.2.3, the teaching strategy in ODL is another pillar of the curriculum that is essential to learning (Egan, Sebastian and Welch, 1991; Friedman and Fisher, 1998; Sheard, et al., 2014). In distance learning, the teachers have limited or no ability to interact with students in contact classes (Butler and Morgan, 2007). As a result, it is important to acknowledge that the strategy required for

teaching demands a different type of setup compared to the setup for contact-based learning (Ranko-Ramalli and Rakoma, 2012). Rossett (2002) and Chipere (2017) state that online learning has benefits but requires great dedication and resources. Online learning needs to be managed properly through proper design of learning materials, adequate support for the students, and with participants in mind (Threlkeld and Brzoska, 1994; Rovai and Downey, 2010).

Mayes and Fowler (1999) and Derus and Ali (2012) indicate that teachers should be aware that students will have different learning strategies; a great focus on students is pivotal; and often one-on-one discussions with students may be necessary. Other strategies for improved performance are effective communication (Holmberg, 1985; Sheard, et al., 2014), better use of technology such as online discussions, and computer-based teaching (Keith, 1999; Kitahara, Westfall and Mankelwicz, 2011; Goosen and Breedt, 2012). Study materials and institutional deadlines for all deliverables and feedback are also needed (Egan, Sebastian and Welch, 1991). According to Oblinger (2003), universities have difficulties in devising ways of managing the diversity among students for learning. The review of literature in the study highlights the importance of recognising different learning preferences and styles (Honey and Mumford, 1982; Dunn and Dunn, 1992; Furnham, 1995; Zander, et al., 2009; Seyal, et al., 2015) to be considered when institutions derive teaching and learning strategies.

Learning to program is often viewed as an isolated activity, but it involves a process of progressive growth and reassessment (Vihavainen, Paksula and Luukkainen, 2011). The process of progressive growth during learning involves the continuous enrichment of understanding (Rumelhart and Norman, 1978). Li, Chen and Tsai (2008) define learning style as a learner's preferred method of observing, perceiving, and understanding information in different forms. It is therefore relevant for educators to consider students' learning styles when teaching students to program.

The review of literature indicates that students' learning styles influence the outcome of learning to program (Dunn and Dunn, 1992; Honey and Mumford, 1992; Furnham, 1995; Dunn and Griggs, 2003; Coffield, et al., 2004; Sayel, et al., 2015). Dunn and Dunn (1992) bring out that each student has a favoured learning style to learn and keep new and complex information. Other models (Honey and Mumford, 1992; Furnham, 1995; Dunn and Griggs, 2003; Coffield, et al., 2004; Zander, et al., 2009) outline different elements of learning styles

that have an influence on teaching and learner achievements. Honey and Mumford (1992) indicate that there are four types of learners: activists, who are disciplined and perfectionists; theorists, who are optimistic and open-minded; pragmatists, who are problem-solvers; and experimental or reflectors, who are thoughtful and cautious. Coffield, et al. (2004) indicate that learners' learning styles evolve significantly during the transition from childhood to adulthood. Learners have an array of preferences: (1) strong preferences, if encouraged, will lead to enhanced learning outcomes for the learner; (2) moderate preferences may need intervention to enhance learning; and (3) unindicated preferences because they are not relevant to the learners. In other cases, success is dependent upon the learner's level of interest or external factors. The general observation is learners have specific ways of responding to instructional methods.

In summary, the key components of the curriculum as defined at the beginning of this section of the literature review are institutional education, curriculum programme, educational material, as well as the teaching and learning strategy. These entities collectively define the curriculum of higher learning as defined in the study and provide a structured review of literature on how the curriculum enables learning and teaching, particularly in introductory programming. All defined areas of the curriculum have been discussed to provide a context on how the areas individually and collectively affect the curriculum. Institutional education has been reviewed in the context of ODL because of its obvious relevance to the study. Also covered is the importance of educational materials, since they provide one of the most preferred tools that students use. Students use educational materials to acquire knowledge and address difficult situations or solve specific problems in learning. The view on how students obtain and apply knowledge in learning and also have various learning preferences was discussed as well. All these factors provide a fundamental enabling role for the success of the learning curriculum and have an impact on how students learn, particularly programming.

The curriculum in the study is one of the three areas of the study formulated from the research question, "What are the hindrances related to the course university curriculum?" The response to this question shows how the curriculum alone and the curriculum working with the two other areas – that is, the programming syllabus and personal factors – influence the success rate in programming students at Unisa.

The next section discusses the influence the programming syllabus has on learning to program and the relationship the programming syllabus has with both the curriculum and personal factors.

## 2.3 PROGRAMMING SYLLABUS

The previous section provided a review of the literature on curriculum factors and personal factors. The curriculum in the study is the basis for the formulation of the teaching and learning in higher education institutions, particularly at Unisa. The review of literature highlighted the role the curriculum plays in supporting programming students. The review also revealed the personal factors related to an individual that affect learning to program in programming students. In the case of programming students, the curriculum provides the structure for educators and learners as defined by the institutions that the programming syllabus can work.

This section reviews and discusses the literature on the programming syllabus. First, a general overview of programming is provided. The programming syllabus is different from the curriculum because unlike the curriculum, it is formulated by the teachers and not by the institution. It contains documents that cover topics taught in a specific module and, in this case, an 'introduction to programming' module. Once the elements of programming are outlined, teaching to program is discussed. This is followed by learning to program, which entails the construct of programming and the management and development of writing computer programs. Ultimately, elements of programming in practice are covered to provide insight into how learning to program translates into knowledge and skills to resolve problems or perform specific tasks by students. The study sought to understand factors relating to the programming syllabus that affected the success rate in first-year students at Unisa. The evaluation of such factors is derived from the research question, "What are specific challenges relating to the programming syllabus?"

### 2.3.1 TEACHING TO PROGRAM

The review of the curriculum teaching strategy in Section 2.2.4 indicates that effective teaching at the curriculum level is important. Teaching at the curriculum level entails the

development and management of multitudes of areas so that students receive appropriate support and education, especially in distance learning education. This section focuses on properties specific to teaching to program rather than teaching at curriculum level, which covers institution-related interventions.

When teaching introductory programming students to program, it is expected that students will in large have no or very limited knowledge of programming (Pedroni, Oriol and Meyer, 2009). Goosen, Mentz and Nieuwoudt (2007) state that there is a significant difference in the needs, knowledge, and abilities of entry-level programmers compared to expert programmers. Novice programmers generally focus on context rather than the overall program (Kessler and Anderson, 1987) and spend limited time on planning. Giangrande (2007) points out that teaching should consider the methodology, topics, and language. In the teaching of introductory programming to students, the teaching approach follows the basic steps, which are design, develop, debug, and test with the aim of performing a specific task or solving a problem (Du Boulay, 1986; Schulte and Bennedsen, 2006; Butler and Morgan, 2007; Kinnunen, 2009). Teaching starts with simple low-level functions such as syntax, which enables programs to be constructed. Teaching ends with complex programming concepts such as objects or procedures that allow the programmer to manage complex tasks (Butler and Morgan, 2007). The poor formulation of teaching plans and learning tools leads to poor performance in learning to program as suggested by Derus and Ali (2012).

The selection of the appropriate programming language is also a subject for discussion. Lister, et al. (2006) disclose that there are different views on whether object-orientated- or structured programming should be offered in introductory programming. Schulte and Bennedsen's (2006) study showed that 52% of universities in the study used Java despite the fact that the language is seen as one of the most difficult programming languages to learn. Goosen, Mentz and Nieuwoudt (2007) indicate that it is crucial for entry-level programmers to obtain a foundation in general programming and theoretical concepts and being educated in the language encourages the application of problem-solving skills. Koulouri, Lauria and Macredie's (2015) study supports this suggestion by indicating that the use of a simple general-purpose programming language such as Python appears to support the students in learning programming concepts. The use of effective tools has been highlighted as matter for consideration (Gross and Powers, 2005). The study also reveals that students' performance

improves when they are exposed to problem-solving prior to programming. The study by Ali, Kohun and Coraopolis (2005) highlight that problem-solving in technological subjects such as programming should be the goal. Butler and Morgan (2007) reveal that irrespective of the programming approach adopted, the development environment that prevails, and the language used, the students will still have difficulties combining the logical reasoning steps and the abstract concepts in programming.

Bloom's Taxonomy (Bloom, et al., 1956) is one of the widely adopted models to describe and group the levels of cognitive complexity in learning that involves logical reasoning and abstract concepts. The taxonomy consists of three educational categories:

- Psychomotor: involves manual and physical skills
- Cognitive: is concerned with mental concepts
- Affective: covers feelings and attitude

Individual categories are divided into hierarchies of objectives. In this section, the cognitive category is discussed further because of its relevance to the programming syllabus related to learning to program. The taxonomy model consists of six different levels, as represented in Table 2.2, with knowledge at the lowest level of the hierarchy. Knowledge level has to do with learners memorising the information being studied. The higher the level in the taxonomy, the more the mental engagement of the learner is required. At the top of the pyramid is evaluation, which involves the formulation, development, and composition of ideas. The taxonomy approach was used by Oliver, et al. (2004) in the field of computer technology to compare the cognitive difficulty level of courses. These authors found that introductory programming academic programmes showed a high level of cognitive demands.

**Table 2.2: Cognitive levels of Bloom's Taxonomy**

| Level | Category | Description |
|---|---|---|
| 6 | Evaluation | Test on the ability to evaluate ideas |
| 5 | Synthesis | Test on the ability to relate knowledge from several areas and use of old ideas to create new ones |
| 4 | Analysis | Test on the ability to understand the information and translate it into a different context |
| 3 | Application | Test on the ability to apply the information in a concrete situation; questions should be resolved using skills and knowledge |
| 2 | Comprehension | Test on the ability to understand the information and translate it into a different context |
| 1 | Knowledge | Test on the observation and recollection of information acquired |

Source: Bloom, et al. (1956)

What is important to note is that teaching to program requires step-by-step activities that begin with elementary concepts such as syntax and moves towards very complex functions such as procedures and objects (Bloom, et al., 1956; Butler and Morgan, 2007). It is for this reason that various elements of programming would have different levels of difficulty (Bloom, et al., 1956). The more complex the programming tasks, the greater the level of mental engagement required (Bloom, et al., 1956). Porter and Calder (2004) have outlined the relationship between the different levels of Bloom's Taxonomy and programming tasks as described in Table 2.3. The table shows that learning the concepts is less difficult than working with various building blocks in programming. Understanding the problem and deriving the solution are even more difficult. The highest level of working with programs is looking for alternatives or assessing the best ways to solve problems or perform certain tasks. What this means is that introductory programming students will find developing programs that can solve problems difficult (Robins, Rountree and Rountree, 2002). It is even more difficult to evaluate options for managing exceptions such as program runtime errors (Matthíasdóttir and Geirsson, 2011; Schoeman 2015), coding-related issues, and finding alternative ways to perform tasks (Kinnunen and Malmi, 2008). It is therefore important for teachers to elevate the level of support to the students as the students start to learn very complex programming functions (Butler and Morgan, 2007).

**Table 2.3: Bloom's Taxonomy levels vs. Programming tasks**

| Level | Bloom's Taxonomy levels | Programming Tasks |
|---|---|---|
| 6 | Evaluation | Looks at alternatives |
| 5 | Synthesis | Formulates the solution |
| 4 | Analysis | Understands the problem |
| 3 | Application | Flows, semantics |
| 2 | Knowledge | Tools, constructs, syntax |
| 1 | Comprehension | Linked to concepts |

Source: Porter and Calder (2004)

### 2.3.2 LEARNING TO PROGRAM

Many computer programming educators will agree that one of the contributing factors to high failure in introductory programming students is that most students feel that learning to program is a not an easy task (Lahtinen, Ala-Mutka and Järvinen, 2005). Programming is a difficult task and is generally viewed by many introductory students as challenging (Buck and Stucki, 2001; Jenkins, 2002; Mahmoud, Dobosiewicz and Swayne, 2004; Mead, et al., 2006; Bergin and Reilly, 2005; Butler and Morgan, 2007). Other authors hold the same view about the significant number of students who have difficulties succeeding in programming (Bennedsen and Caspersen, 2007; Watson and Li, 2014; Schoeman, 2015).

Literature has documented the ability of introductory programming students to write (McCracken, et al., 2001; Lahtinen, Ala-Mutka and Järvinen, 2005) and read (Lister, et al., 2004) programs. A review of the results of code writing and reading tests showed that the students performed poorly in both evaluations (McCracken, et al., 2001; Lister, et al., 2004). The link between the ability to read written programming code and the ability to program is written in past studies. Chmura (1998) and Ala-Mutka (2004) indicate that students who are able to comprehend or read text perform well in learning to program. Schoeman (2015) also points out that students obtain skills in programming by learning to read a code and then to explain a code. Only after being able to explain code would students be able to write code.

The next section looks into various elements that form part of learning to program. The content of the section provides insight into what learning to program entails.

#### 2.3.2.1 PROGRAMMING DIFFICULTIES

Computer programming students must acquire knowledge about programming before they can start writing programs to solve specific problems or complete certain tasks (Winslow, 1996). Rogalski and Samurçay (1990) state that obtaining and building programming knowledge is vastly complex. Derus and Ali (2012) and Ma, et al. (2008) indicate that obtaining and building programming knowledge requires cognitive thinking, mental depiction of programs, design, development, and testing. Robins, Rountree and Rountree (2002) describe the difficulties of working with aspects of computer programming concepts. Programming concepts require knowledge of programming constructs (Robins, Rountree and Rountree, 2003; Butler and Morgan, 2007), which include the design, development (using variables, loops, array, conditions), deployment, and derivation of mental models (Ma, et al., 2008) to resolve the problem. Winslow (1996) brings out that such knowledge generally remains at a distance and cannot be grasped holistically by the introductory programming students. According to Du Boulay (1986), the activities that form part of learning to program include:

- structures that entail plans based on the above;
- general orientation, which means the aim and use of programs;
- the machine, which represents the computer as it manages the execution of programs;
- notational representation, the semantics, and syntax of a given programming language; and
- pragmatics, which entail the skills required to plan, develop, debug, and test programs.

Dann, et al. (2006) suggest that the syntax for the programming language, difficulties identifying program results during runtime, and limited understanding of design technique are some of the challenges novice programmers experience. According to Renumol, Janakiram and Jayaprakash (2010), novice programmers often make programming syntax-

related errors or basic programming mistakes such as using functions or variables before declaring them. Vogts, Calitz and Greyling (2010) relate the difficulties to a combination of factors that often take place simultaneously for the students to learn. The learning happens in a way not familiar to the students, which is in the syntax associated with the new programming language while learning to use the programming development environment. The various difficulties highlighted in literature make it necessary to ask the question, "What is the most effective learning approach to learning to program?"

#### 2.3.2.2 LEARNING APPROACH

This section discusses the learning process that is particular to the programming syllabus, given that often learning is unique in individual courses. In the field of programming, the choice of relevant learning approach gains is important given the difficulties described in the previous section.

Booth (1992) highlights that introductory programming students' experiences of learning to program can be grouped into four categories:

- Learning a programming language
- Learning to write codes in a programming language
- Learning to solve problems using programs
- Becoming part of the programming community

The first two areas are specific to computer coding, and the third area involves using relevant techniques to resolve problems or accomplish specific tasks. The last area is concerned with interaction with peers, instructors, and clients. Oliva and Gordon (2012) indicate that the learning approach adopted affects the students' learning experiences. Learning experiences affect the ability to learn to program (Hawi, 2010). The learning experience is thus pivotal to the approach individual students adopt when learning to program (Govender and Grayson, 2008). Students can only be successful in learning to program if they understand the programming concepts that provide a basis to practical computer programs (Winslow, 1996; Ismail, AzilahNgah and Umar, 2010). The study

sought to evaluate how the learning approach adopted by the students affected their learning outcomes in programming.

#### 2.3.2.3 PROGRAMMING CONCEPTS

Programming concepts are essential to the fundamental development of basic knowledge in programming, particularly among introductory programming students (Derus and Ali, 2012). The key basic concepts of programming are (1) tools, (2) variables, (3) data structures, (4) control structures, and (5) syntax. Learning a programming language involves the ability to understand the syntax, semantics, procedures, variables, and structures (Butler and Morgan, 2007). One also needs to have coding skills and basic computer literacy skills (Winslow, 1996; Yeh, et al., 2010). These concepts are generally difficult for introductory programming students to comprehend. Winslow (1996) indicates that students learn syntax and semantics independently but are generally unable to combine the two into a working program. This study investigated the challenges associated with the programming concepts by asking students questions about having difficulties with the programming syntax, the tools, and development environments.

### 2.3.3 PROGRAMMING IN PRACTICE

Many studies indicate that the majority of students lack the ability to apply basic programming concepts, problem-solving, and practical programming skills (Winslow, 1996; Jenkins, 2002; Lahtinen, Ala-Mutka and Järvinen, 2005; Wiedenbeck and Labelle, 2004). Several reasons are possible, including limited in-depth programming knowledge and challenges dealing with very complex programming functions such as objects, arrays, decisions, and algorithms (Butler and Morgan, 2007). The amount of time required to learn to program could also be a hindrance. Winslow (1996) highlights that learners require at least 10 years to learn to program to a level where they can, as experts, practically apply the lessons learned. When students learn to program, various elements are taught, from a conceptual viewpoint, yet the application of these elements is very hard for introductory students (Robins, Rountree and Rountree, 2003; Lahtinen, Ala-Mutka and Järvinen, 2005; Butler and Morgan, 2007). It should therefore be expected that the introductory

programming students would have difficulties building programs that can solve problems or perform a certain task.

Section 2.4 provides a review of literature on personal factors affecting introductory programming students' performance in learning to program.

## 2.4 PERSONAL FACTORS

The previous sections in this chapter have focused on the curriculum and programming syllabus to provide insight into the institutional setup and specific programming-related aspects of the module formulated to support both teaching and learning. The effort the students put into their studies has an influence on the outcome of their studies (Simon, et al., 2006; Mhashi and Alakeel, 2013). Personal commitments and reasons the students have could affect their learning performance (Tinto, 1987; Kinnunen and Malmi, 2006; Simon, et al., 2006). The study therefore explored personal factors associated with individual learners that might inhibit performance in learning to program. Personal factors covered in this section are prior learning, aptitude and cognitive factors, personal commitments, and personal reasons.

The factors are evaluated in the study by asking the question, "What are the personal factors that have an impact on learning?" This question becomes more relevant in the context of Unisa, given that the university offers ODL to the largest number of students from different backgrounds in South Africa (Section 2.2.1), with the majority studying part-time and having other commitments such as employment and other personal commitments (DHET, 2015). Section 2.2.1 also indicates that Unisa has a significant number of young learners that enrol at the institution unprepared for the learning ahead. The combination of the profile of the learners and the ODL model provide different challenges for the institution in contact based higher learning institutions (DHET, 2015). The study sought to understand the personal factors experienced by the learners that affect their learning to program. The focus of the study, though, is limited to key personal factors that are deemed relevant to learning to program at Unisa.

### 2.4.1 PRIOR LEARNING

Many authors indicate that previous education, skills, or experience in science subjects and programming has an important role in determining the success of students in learning to program (Byrne and Lyons, 2001; Boyle, Carter and Clark, 2002; Stephenson, et al., 2005). Prior exposure deemed relevant in programming includes a high school education in mathematics and science (Byrne and Lyons, 2001; Bennedsen and Caspersen, 2005; Kori, et al., 2015; Qahmash, Joy and Boddison, 2015), lessons in computer technology (Byrne and Lyons, 2001), or adequate involvement in the actual practice of programming (Stephenson, et al., 2005). Other authors posit that mathematics has no relevance for students' success in programming (Chmura, 1998; Boyle, Carter and Clark, 2002; Ventura, 2005). Stephenson, et al. (2005) found that many first-year computer technology students lack experience in programming – a factor deemed relevant in programming by Byrne and Lyons (2001). Hagan and Markham (2000) affirm that students with enough prior exposure to at least one programming language perform substantially well during assessments.

This study evaluated the relationship between prior learning and success in learning to program (Kori, et al., 2015). The students were asked if they were proficient in computer literacy and if they had been exposed to a certain level of programming before. The link between both mathematics and science and performance in programming is not explored in the study, to limit the scope of the survey.

### 2.4.2 APTITUDE AND COGNITIVE FACTORS

Jenkins (2002) points out that there is nothing implicitly difficult about learning to program, but it is merely because students lack aptitude. Davy and Jenkins (1999) conducted a study to assess the link between aptitude and the outcome in programming. The outcomes indicated that no link exists between the two entities – something echoed in the study by Tukiainen and Mönkkönen (2002). Other tests on aptitude were conducted by Mazlack (1980), but the outcomes were inconclusive. Jenkins (2002) indicates that there is no reliable method to assess aptitude for programming. If it cannot be proven that a relationship between programming and aptitude exists, then the focus must be turned to cognitive aspects

of learning. Cognitive factors might aid in the understanding of challenges associated with learning to program (Jenkins, 2002; Ma, et al., 2008).

Cognitive factors that may affect learning to program are learner's motivation and learning style (Jenkins, 2002; Eccles and Wigfield, 2002). Programming students may need a specific level of motivation (Alaoutinen and Smolander, 2010) or some form of learning style to find learning to program easy. Students with inappropriate motivation or who use an incorrect learning style are likely to have difficulties learning to program (Jenkins, 2002; Roddan, 2002; Kinnunen and Malmi, 2006). The link between cognitive factors and learning to program can be established by assessing learning styles adopted and students' motivation (Jenkins, 2002).

The review of learning styles covered in Section 2.2.5 shows that educators need to be aware of the fact that different students adopt different learning styles. Educators have to provide support to the students in the best possible way based on the students' learning preferences (Winn, 2002). Learning styles were also evaluated in Section 2.2.4 by assessing if students adopted styles that would enable them to adjust well to the perceived short time required to learn to program. The next paragraph reviews and discusses motivation and the influence it has on learning to program.

Motivation has featured often in various studies as the determinant for the failure rates in introductory programming (Isroff and Del Soldato, 1998; Jenkins, 2002; Winn, 2002; Bennett, 2003; Kinnunen and Malmi, 2006; Kori, et al., 2016). Kinnunen and Malmi (2006) and Kori, et al. (2016) found that lack of motivation was one of the main reasons for high dropout rates in first-year computer students. Winn (2002) and Sheard, et al. (2014) highlight various factors that affect a student's level of motivation, such as personal situations, other commitments, and demanding situations. These factors, singly and in combination, can result in students dropping out of their studies, especially in the first year of higher education (Kinnunen and Malmi, 2006). The factors appear to be more prevalent in ODL institutions, as students generally have several commitments (Govender and Grayson, 2008). This study explored if motivation was the chief factor for the failure rate in programming at Unisa. Students were asked to indicate if they simply lacked the motivation to study.

### 2.4.3 PERSONAL COMMITMENTS

Personal commitments could have a negative impact on the learning outcomes of the student (Jenkins, 2002; Simon, et al., 2006; Mhashi and Alakeel, 2013). Authors highlight that there are students who are able to manage both the family and employment commitments and study demands, whereas others have difficulties doing so (Winn, 2002; Bennedsen and Caspersen, 2007; Watson and Li, 2014).

Time management is also a factor in learning that is generally a result of other multiple personal reasons (Tinto, 1987; Kinnunen and Malmi, 2006) and, if not properly managed, might impede performance in learning. The time factor is even more relevant for students who are learning to program, since programming requires extensive time for study and practice (Kinnunen and Malmi, 2006; Kinnunen and Malmi, 2008; Mhashi and Alakeel, 2013). Programming exercises are ranked as the most negative factor for time management in introductory programming (Kinnunen and Malmi, 2006). Some of the reasons for ineffective time management given in the study by Kinnunen and Malmi (2006) are expressed in the view that the course required more time than most students expected. Additionally, personal and work commitments took up much time. Xenos, Pierrakeas and Pintelas (2002) state time management as the main reason for students not completing introductory programming. In fact, what was observed in Winn's (2002) study was that some students were still unable to spend more time on their studies despite the few personal commitments that they had.

### 2.4.4 PERSONAL REASONS

Several personal reasons that result in failure rates in introductory programming have been discovered (Kinnunen and Malmi, 2006; Mhashi and Alakeel, 2013). There are studies that link personal reasons to challenges in general learning and to the high failure rate in the subject. The personal reasons provided are manifold (Lenning, Beal and Sauer, 1980; Glossop, 2002; Bennett, 2003; Kinnunen and Malmi, 2008) and include financial difficulties (Bennett, 2003), low self-esteem (Bennett, 2003), family issues (Kinnunen and Malmi, 2008), perceived learning challenges, and dissatisfaction with the course (Ramist, 1981;

Simon, et al., 2006). This study focused largely on perceived learning challenges faced by students to confine the focus to factors directly affecting learning to program.

## 2.5 SUMMARY

The review of literature provided insight into the high failure rates in introductory programming students. Since the study attempts to understand "the factors that contribute to learning hindrances experienced by programming students at Unisa" based on the main research question, the review of literature was categorised into three different areas of learning. These were the curriculum, the programming syllabus, and students' personal factors. The three areas are based on the three secondary research questions formulated to support the main question. As a result, the literature review was based on the three questions, which are (1) "What are the hindrances related to the university course curriculum?" (2) "What are the specific challenges relating to the programming syllabus?" and (3) "What are personal factors that have an impact on learning?" The review of literature is therefore grouped into the evaluation of factors emanating from (1) the curriculum, (2) the programming syllabus, and (3) student's personal factors.

The review of the literature showed that there was a significant increase in enrolment in ODL institutions, with Unisa accounting for 87.9%. Literature showed that this increase in enrolment number should not compromise the quality of education being offered by the institutions. Educational materials have a significant influence on the outcomes of students' performance in learning. The lecture recordings and study guides are the most preferred form of educational materials. The preference of the learning materials by the students remains to be explored in the case of programming at Unisa. Literature also indicated that students are confronted with an elevated level of academic pressure due to the new style of learning and teaching they are exposed to during their first two years in institutions of higher learning, with the majority of dropouts or failures taking place in the first year. The study notes in literature reviewed that teaching and learning strategy is effective when formulated in line with students' learning preferences and styles.

The review of the programming syllabus-related studies highlights that many students find learning to program difficult. Learning programming involves problem-solving abilities and

abstract mental models. Literature further reveals that introductory programming students should be taught in general-purpose programming languages that encourage problem-solving. For students to learn to program, they need to learn basic programming building blocks, which include syntax, control structures, data structures, tools, and variables. Learning to program evolves through different phases, and learning to program requires adequate time to be able to apply what one has acquired in practice.

The review of the personal-factor-related studies uncovered that prior learning, the challenges associated with the transition from high school to higher learning institutions, and cognitive and aptitude factors affected the ability to learn to program. Literature highlights that prior learning in programming, mathematics, and science is significant for the performance in learning to program, while other studies hold that no link exists. Students who have some exposure to mathematics, science, and computer technology courses before undertaking a programming course generally do better in learning to program (Byrne and Lyons, 2001; Bennedsen and Caspersen, 2005; Kori, et al., 2015; Qahmash, Joy and Boddison, 2015).

Additional to personal factors covered, aptitude is highlighted as important to learning to program, but there are very limited findings that link the two. With limited findings relating to aptitude, literature indicated that researchers need to focus their attention on cognitive factors such as learning styles and motivation. These two factors have been found to affect ability to do well in learning to program. Personal commitments and personal reasons also contribute to the factors affecting learning to program. Additional personal factors that have an impact on learning to program include aptitude, motivation, and personal commitments that include family and employment. In addition, personal reasons such as family issues and financial difficulties often contribute to the challenges faced by introductory students in learning to program.

Discussions from literature on the three areas of the curriculum, the programming syllabus, and students' personal factors are evaluated in the context of programming students at Unisa. The evaluation expands on the literature where gaps exist by covering the need to conduct the study on the hindrances to learning to program in introductory programming students by evaluating (1) the impact the university curriculum has on students' ability to succeed in learning to program, (2) the impact the programming syllabus has on students' ability to

succeed in learning to program, and (3) the impact students' personal factors have on students' ability to learn to program.

The chapter that follows focuses on the research methodology used in this study.

# 3 RESEARCH METHODOLOGY

## 3.1 INTRODUCTION

The preceding chapter reviewed literature relevant to this study. This chapter sets out the research philosophy, research approach, research strategy, and research methods and techniques for directing the study in a way that answers the research question(s) and fulfils the research objective(s). The chapter outlines the research philosophy adopted for the research, the research approach formulated, the research strategy, and techniques and procedures used for gathering and interpreting data.

The various aspects of the research methodology outlined in this chapter are derived from the research model formulated by Saunders, et al. (2012). The model illustrates the stages that must be considered when developing a research approach. When observed from the outside, the 'research onion' in Figure 3.1 depicts stages of the research process in the form of an onion consisting of various layers that represent the research philosophies, approaches, strategies, choices, time horizons, as well as techniques and procedures. These layers must be peeled when developing a research approach. The research onion layers are the building blocks of the research methodology for this study and also form the basis for the overall approach of the remaining chapters of the study during the data collection, analysis, and presentation of the results of the study.

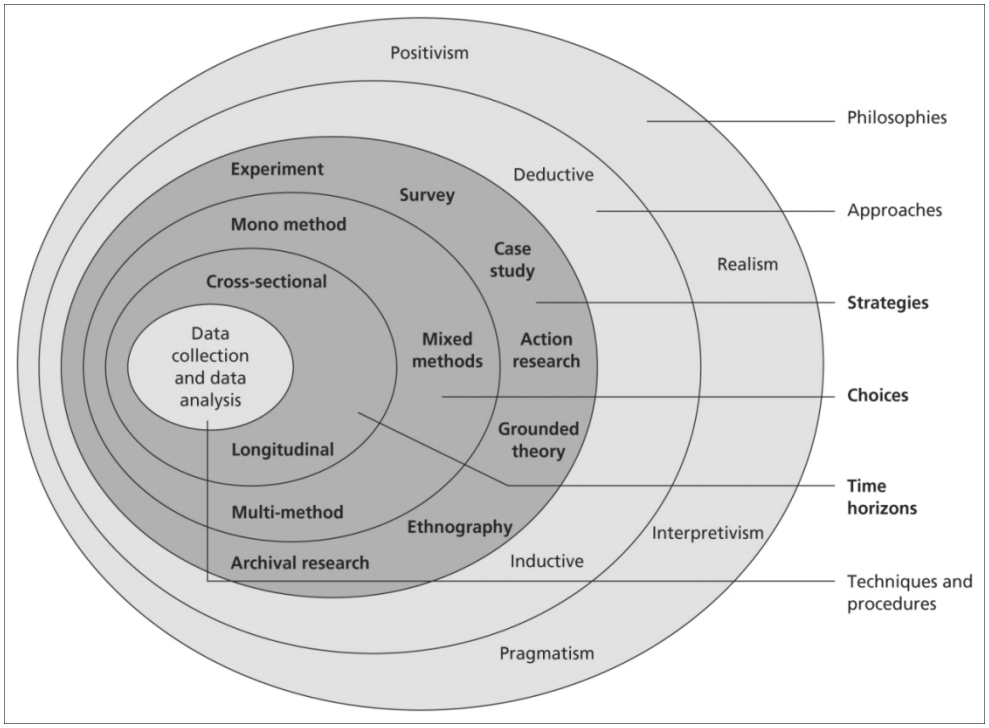


**Figure 3.1: Research Onion**

Source: Saunders, et al. (2012)

## 3.2 RESEARCH PHILOSOPHY

Research philosophy (Saunders, et al., 2012), worldview (Creswell, 2013) ontology, and epistemology (Crotty, 1998), even paradigm (Lincoln, Lynham and Guba, 2011), consist of beliefs and assumptions on how the world is perceived and also informs action. Research philosophy varies, depending on the goal of a study (Goddard and Melville, 2004). Research philosophy is fundamental to the formulation of knowledge by the researcher, the nature of the knowledge concerned (Saunders, et al., 2012), and consists of beliefs pertaining to the type of reality being examined (Bryman, 2015). Malhotra (2014) stipulates that research philosophy be used to guide the researcher in conducting the research strategy, procedures of research design, questionnaire design, and sampling.

Saunders, et al. (2012) indicate that the research philosophy allows researchers to examine assumptions about the world and whether such assumptions are relevant or not. Bryman (2015) highlights that the philosophical assumptions and beliefs affect the way the research is carried out, since they remain tacit throughout the research and because they inherently

dictate what should be studied, how research should be done, and how the results should be interpreted (Bryman, 2015). The assumptions defined during the research philosophy provides the foundation for the accomplishment of the research (Flick, 2015). Jonker and Pennink (2010) further highlight the importance of the research philosophy by indicating that the perception by the researcher towards the world provides a structure that informs the researcher's thinking and behaviour. It is therefore pivotal to understand the research philosophy adopted to explain the assumptions that intrinsically form part of the research process and how that subsequently aligns with the methodology being applied.

Creswell (2013) states, as illustrated in Figure 3.2, that during the formulation phase of research three factors are key interrelated considerations: philosophical worldview assumptions introduced to the study; the research design relevant to the worldview; and the particular methods that transform the approach into practice.

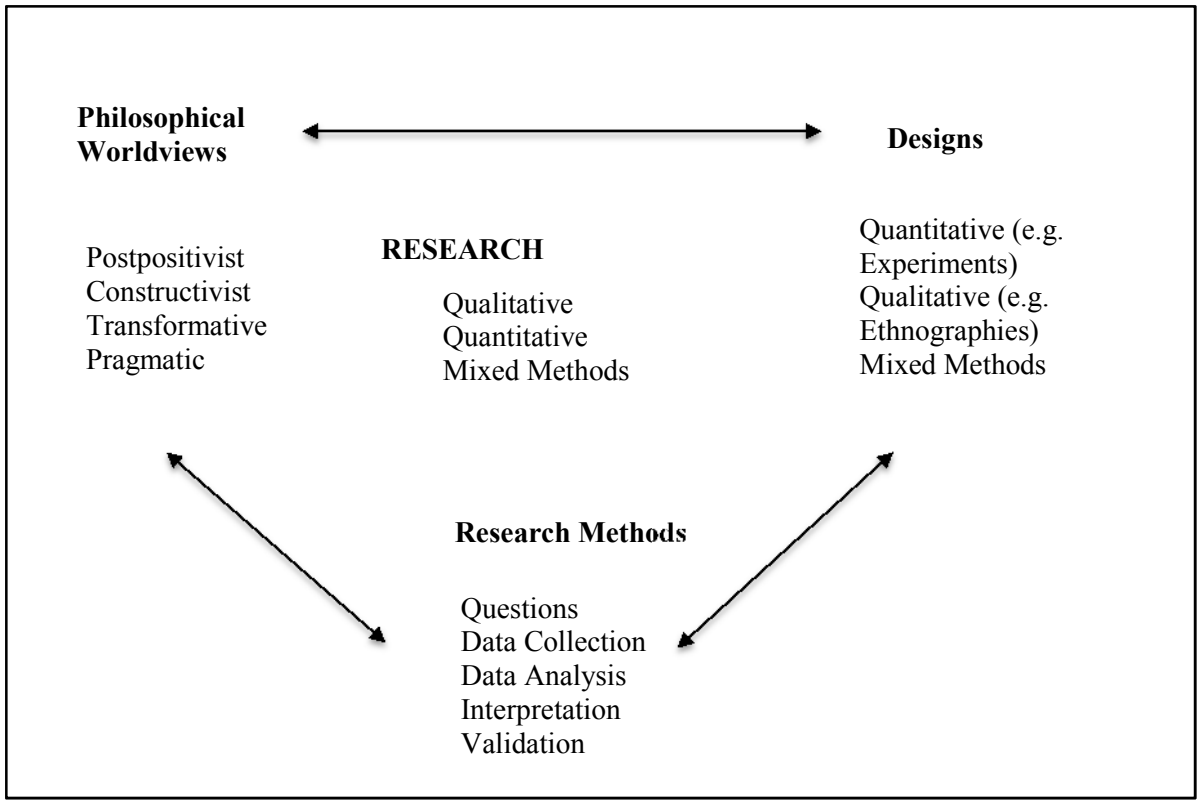


**Figure 3.2: Interconnection of worldviews, design, and research methods**

Source: Creswell (2013)

The **postpositivist** researcher worldview is concerned with the necessity to identify and evaluate the causes that determine outcomes, where knowledge is formulated through measurement and observation of the objective reality (Creswell, 2013).

The **constructivist or social constructivist** researcher worldview is that individuals strive to understand their area of existence and form mental meanings of their life experiences (Creswell, 2013). Others indicate that the constructivist researcher is concerned with shared understanding within contextual and cultural situations (Marshall and Rossman, 2014).

The **transformative** researcher worldview is linked closely to politics and the political agenda to confront oppression at whatever levels it happens (Mertens, 2014). The research is concerned with a plan that may resolve issues linked to people's lives and institutions they use (Morris, 2006). Other issues to resolve include oppression suppression, inequality, empowerment, domination, and alienation (Creswell, 2013).

The **pragmatic** worldview relates to situations, actions, and consequences instead of prior conditions (Creswell, 2013). The pragmatist researcher articulates the research problem, then uses all applicable approaches to know and comprehend the problem (Rossman and Wilson, 1985).

The philosophical worldview is the pivotal anchor of any study and has an influence on the selection of research design approach and the research methods for that approach. It is therefore important for the researcher to formulate philosophical worldview assumptions to assess the research design and unique research procedures in use (Table 3.1). Creswell (2013) provides a representation of how the philosophical worldviews, research design, methodology, as well as data collection and analysis techniques can be mapped together for an effective research approach.

**Table 3.1: Worldviews – relationship with three main research types**

| Category | Philosophical Worldview | Research Methodology | Data Collection and Analysis |
|---|---|---|---|
| Qualitative Approaches | Constructivism Transformative | Ethnography, Phenomenology, Narrative research, Case study, Grounded theory | Open-ended questions, text or image data qualitative analysis, emerging approaches |
| Quantitative | Postpositivism | Experiment Surveys | Closed-ended questions, numeric data quantitative analysis, predetermined approaches |
| Mixed Methods | Pragmatic | Transformative, concurrent, and sequential | Both closed-ended and open-ended questions, both predetermined and emerging approaches, and both qualitative and quantitative data and analysis |

Source: Creswell (2013)

The pragmatic and transformative research philosophies are viewed as compatible with the mixed methods research design. Postpositivism is generally associated with quantitative research, while constructivism is associated with qualitative. The study therefore adopts pragmatism because it is concerned with the collection, analysis, and interpretation of qualitative and quantitative data. The results of the research data are used to find solutions to the high failure rate among the introductory programming students at Unisa.

## 3.3 RESEARCH APPROACH

Deductive and inductive research approaches involve the research approach that could be used in either quantitative research (deductive) or qualitative research (inductive and limited deductive) (Trochim and Donnelly, 2007). Saunders, et al. (2012) highlight the importance of identifying whether the research is deductive or inductive in the study, and this should be explained clearly. Induction starts with observations, then aims to find themes within such observations, whereas deduction begins with the testing of patterns based on observations (Babbie, 2013).

In inductive research, the researcher builds theory by collecting data relevant to the topic, and once ample data has been collected, patterns are identified from the data to formulate a theory that could answer the research questions. Neuman (2006) states that inductive research starts with comprehensive observations of the world, then moves towards ideas and generalisations. The inductive approach allows for a broad and deeper explanation of the situation (Saunders, et al., 2012). Deductive research tests a theory and can be explained as making a transition from the particular to the general (Trochim and Donnelly, 2007). Deduction starts with the formulation of theory or hypotheses, followed by the design of a research strategy to test the developed theory (Trochim and Donnelly, 2007). The use of quantitative and qualitative research methods in this study allows for the adoption of an inductive research design to answer research questions subjectively and objectively.

## 3.4 RESEARCH STRATEGY

The research strategy focuses on how the work was carried out in providing answers to the research questions (Saunders, et al., 2012). The strategy defines the data collection sources, such as surveys, cases studies, interviews, systematic literature review, ethnography, action research, and experimental research. In addition, the research strategy specifies constraints and limitations associated with the research.

The study adopts a survey as the strategy to discover general patterns deductively and inductively in introductory programming students' behaviour, experiences, and opinions. Pinsonneault and Kraemer (1993, p.78) state that survey research is most suitable in cases where the key questions relating to the phenomena are "what is happening?" and "why and how is it happening?" The survey uses the cross-sectional time horizon to allow for a 'snapshot' of the situation, which is relationships between students and factors that result in a high failure rate in introductory programming at a specific time during the university semester. The snapshot data collected from the survey questionnaires that comprise both open- and closed-ended questions (Appendix A) will be subjected to both qualitative and quantitative research methods for the research questions to be answered. Surveys are an economical way of reaching out to a large number of research participants compared to methods such as interviews.

### 3.4.1 ETHICAL CONSIDERATIONS

The ethical aspects of the study were considered prior to the distribution of the survey questionnaire to the students. An ethical application request was made to Unisa's College Research and Ethics Committee (CREC) for permission to conduct research relating to Unisa students in line with the Unisa ethics code of conduct (Appendix B). The approval was granted by the CREC for the period of the research. The research's ethical considerations for the study were voluntary participation, confidentiality, consent, impartiality, and clear communication. The ethical considerations for the study are summarised as follows:

- Voluntary participation – where students were given an option of not participating or, if they did participate, they could choose to stop at any time during the questionnaire session. Participants could choose not to answer certain questions.
- Confidentiality – the identities of the students were kept anonymous in all parts of the research. Pseudonyms were adopted when a possibility of identification existed. All information relating to the participants was safely stored, was accessed only by the researcher for the purpose of the study, and was completely destroyed after the study.
- Impartiality – where throughout the study the researcher remained neutral to avoid any form of bias towards the participants of the study.
- Engagement – during the engagement students, there was no misrepresentation or distortion in any form.

### 3.4.2 LIMITATIONS OF THE STUDY

The limitations of the study are as follows:

- the time required for the design of the questionnaire;
- great reliance on the students to provide feedback on the questionnaire, resulting in iterative requests and prolonged data collection time;
- the number of responses from the students; and
- possible bias in the responses.

Data validity and reliability issues were possible. The dependency factor on time and students was managed by virtue of time management, while issues relating to bias in responses and data were kept in check through selection and random sampling bias techniques.

### 3.4.3 VALIDITY AND RELIABILITY

When a survey is adopted as a strategy for research, it is crucial to ascertain that the survey is valid and reliable so that credible information can be produced (Dochartaigh, 2002; Creswell, 2013; Williamson and Whittaker, 2014). Creswell (2013) states that the instrument must measure what it is supposed to measure (validity), and it should do so consistently (reliability). Validity ensures that the researcher indeed measures what is supposed be measured, while reliability focuses on how consistent a particular measure is (Williamson and Whittaker, 2014). The assessment of reliability and validity is linked to the assessment of reputation and confidence of the source (Dochartaigh, 2002).

Validity assesses if the findings are indeed what they appear to be. The study uses strategies adopted from both qualitative and quantitative approaches to evaluate the credibility of the finding, which is the hindrances to learning to program in an introductory programming module. The following factors affecting validity are considered (Creswell, 2013):

- diverging findings as a result of different concepts from both qualitative and quantitative approaches, and
- compromised outcome on either end due to unequal sample sizes.

Reliability is defined as the degree to which outcomes are consistent over time and consistently reflect an accurate representation of the population of the study (Joppe, 2000). Reliability is concerned with the stability and consistency of what is being measured in different conditions with the measurements yielding the same findings (Nunnally, 1978). The collection instrument and analysis methods for data are deemed reliable if consistent results can be recreated using the same research strategy (Nunnally, 1978).

Triangulation is the validity method that involves the use of different data or methods to study the same phenomenon (Leech and Onwuegbuzie, 2009). The use of triangulation

allows the researcher to factor in all relevant data sources to answer research questions (Creswell, 2013). It is often used to validate data and methods in a study to overcome the limitation of each method or data source. The effectiveness of triangulation is based on the premise that the limitation of one method will be counterbalanced by the advantages of another method (Jick, 1983).

The study acknowledges the limitations in proving validity and reliability of the study due to the exclusive use of a survey questionnaire as the source for data collection. The limitations are, however, countered by the use of various validity and reliability measures during triangulation as described in Chapter 4.

## 3.5 RESEARCH CHOICES

Research choices (methods) are procedures or techniques for collecting data related to some research question or hypothesis (Crotty, 1998). The choices include the mono method and the mixed methods, and a researcher may choose one (mono method) or a combination of two approaches (mixed methods) (Saunders, et al., 2012).

Over the past three decades, there have been several developments in research methods (Gelso, 1979; Howard, 1983; Greene, Caracelli and Graham, 1989; Newman and Benz, 1998; Tashakkori and Teddlie, 2003; Creswell, 2013). Consequently, the quantitative and qualitative methods were merged, resulting in the formation of mixed methods research (Figure 3.3). Mixed methods research has gained popularity in the field of research and may be considered a formal independent research method (Tashakkori and Teddlie, 2003; Onwuegbuzie and Teddlie, 2003; Creswell, 2013). The method is viewed as “the collection, analysis and interpretation of both qualitative and quantitative data in which the data are gathered simultaneously or sequentially, prioritised, merged at one or more stages in the research process” (Creswell, et al., 2003 p.212). The use of mixed methods research in a study allows for the enrichment of the study results in a manner that one single set of data does not permit (Brewer and Hunter, 1989; Tashakkori and Teddlie, 2003).

### 3.5.1 QUANTITATIVE RESEARCH METHOD

Quantitative research examines relationships among variables. It uses an empirical approach where the data consist of numbers and theory foreshadows observation. In this research, theories can be tested deductively. In quantitative research, the relationship among variables is explored in the form of some questions or hypotheses (Phillips and Burbules, 2000).

### 3.5.2 QUALITATIVE RESEARCH METHOD

Qualitative research is "an approach for exploring and understanding the connotation individuals or groups ascribe to a social or human problem" (Creswell, 2013, p.246). The process follows the collection of data in the setting under investigation, the analysis of data using themes and patterns, and the subsequent interpretations of data. The final written report comprises the outcomes with an adaptable structure.

### 3.5.3 QUANTITATIVE VERSUS QUALITATIVE RESEARCH METHOD

The drive to use a specific methodology should be based on its relevance to answering the research questions (Bryman, 2003). Denzin and Lincoln (2005, p.3) indicate that the qualitative researcher studies "things in their natural settings, attempting to make sense of or interpret phenomena in terms of the meanings people bring to them", whereas the quantitative researcher measures and analyses the causal relationships among variables (Creswell, 2013). Bruce and Berg (2001) differentiate between qualitative and quantitative research, maintaining that quantitative research is concerned with numbers and measurements, while qualitative research is concerned with the meanings, patterns, concepts, definitions, themes, characteristics, and symbols.

The differences between quantitative and qualitative research approaches depicted in Table 3.2 (Mack, et al., 2005) involve general methodology; types of questions; analytical objectives; the format of data under interrogation; and variance in the study design flexibility. Quantitative and qualitative research types are different, important, and valid; both can be applied in a study based on the research strategy. It is, however, possible for a

single study to use both strategies (Tashakkori and Teddlie, 2003; Onwuegbuzie and Teddlie, 2003; Bryman, 2003; Creswell, 2013).

**Table 3.2: Quantitative vs. qualitative research method**

| Description | Quantitative | Qualitative |
|---|---|---|
| **General framework** | Aims to confirm hypotheses about phenomena.<br><br>Instruments use a more stringent method of gathering and categorising responses to questions.<br><br>Use highly systematic methods such as surveys, questionnaires, and structured observation. | Aims to search for phenomena.<br><br>Instruments use a more flexible, repetitive method of gathering and categorising responses to questions.<br><br>Use semi-structured methods such as focus groups, in-depth interviews, and participant observation. |
| **Analytical objective** | To quantify variation, predict casual relationships, depict the nature of a population. | To describe variation, describe and explain relationships, describe individual experiences, describe group norms. |
| **Format of question** | Closed-ended. | Open-ended. |
| **Format of data** | Numerical – received by allocating numerical values to responses. | Textual – received from video tapes, audiotapes, and field notes. |
| **Study design flexibility** | Design of study is stable from beginning to end.<br><br>Participant responses do not determine or influence how and which questions researchers ask subsequently.<br><br>Design of study is subject to conditions and statistical assumptions. | Some aspects of the study are flexible, e.g. the exclusion, addition, or wording of specific questions.<br><br>Participant responses affect subsequent questions (which and how the researchers ask questions).<br><br>Study design is repetitive. Research questions and data collection are adjusted in line with what is learned. |

Source: Mack, et al. (2005)

### 3.5.4 MIXED METHODS

There are other methods of research that have properties of both the quantitative and qualitative research (Tashakkori and Teddlie, 1998). Although mixed methods have been adopted late in research (Tashakkori and Teddlie, 1998) compared to quantitative and qualitative research, there are various books and articles dedicated to the approach (Onwuegbuzie and Johnson, 2006; Greene, 2007; Johnson, Onwuegbuzie and Turner, 2007; Creswell and Clark, 2007; Tashakkori and Teddlie, 2010; Clark and Creswell, 2011). Conceding that quantitative or qualitative research has limitations, the view is that the use of mixed methods presents a unique advantage over the exclusive use of either the quantitative or qualitative method.

The application of mixed methods research is a spontaneous addition to both quantitative and qualitative research methods (Johnson, Onwuegbuzie and Turner, 2007). Creswell (2013, p.3) describes mixed methods as a balance between quantitative and qualitative approaches. Bryman (2003) supports the adoption of mixed methods and proposes the merger of quantitative and qualitative approaches to benefit from the advantages that come with each method. Creswell (2013, p.230) indicates that when adopting the mixed methods, the researcher needs to give consideration to five elements in the approach. The considerations are the expected outcome of the research, integration of data, timing on when the qualitative and quantitative data should be collected, balance given to the two data sets, and field being studied. It can be argued that the mixed methods approach allows for a better understanding of the research problem and balanced research, if used appropriately (Creswell, 2013). Table 3.3 (Creswell, 2013) highlights the benefits of the mixed methods approach through the application of various aspects of quantitative and qualitative research methods during the practical application of the method in the research.

**Table 3.3: Quantitative, mixed, and qualitative methods**

| Quantitative Methods | Mixed Methods | Qualitative Methods |
|---|---|---|
| Predetermined | Both emerging and predetermined methods | Emerging methods |
| Instrument-based questions | Both open-ended and closed-ended questions | Open-ended questions |
| Performance, census, observational, and attitude data | Multiple types of data focusing on all possibilities | Interview, document, audio-visual, and observation data |
| Statistical analysis | Statistical and textual analysis | Textual and image analysis |
| Statistical interpretation | Interpretation across multiple databases | Patterns, themes interpretation |

Source: Creswell (2013)

Qualitative research tends to be subjective and will primarily provide insights into the comprehension of students' behaviour and the reasons that influence such behaviour, while quantitative research will objectively focus on measurements and empirical analysis. The mixed methods strategy provides wide but deep qualitative investigation of students' opinion, behaviour, feelings, attitudes, inner experiences, as well as the quantitative degree and frequency of the challenges affecting the students. As a result, the limitations and individual benefits associated with each of the research approaches justify the use of the mixed methods approach to ensure a comprehensive and detailed assessment of individual situations and experiences, both subjectively and objectively.

Some authors caution against the use of mixed methods by indicating that all relevant characteristics of either quantitative or qualitative must be considered in order to effectively combine the method (Johnson, Onwuegbuzie and Turner, 2007; Creswell, 2013). Another concern is that the bias inherent in any single method could dilute or nullify the advantages of the other method (Johnson, Onwuegbuzie and Turner, 2007). When using the mixed methods, caution needs to be afforded to the attention required when gathering, analysing, and interpreting research data; the time-consuming and complex nature of the model due to the merging of two approaches; the knowledge required to do so; and the size of data associated with the approach (Creswell, 2013).

When mixed methods research is used, the importance is rarely on whether the research is quantitative or qualitative but on how research practices are situated between the two (Newman and Benz, 1998). The view indicates that studies can be either generally qualitative or quantitative, and one does not have to choose either one of them. The final written report comprises the structure with the introduction, theory, literature, methods, design, outcomes, discussion, and conclusion where bias should be avoided.

## 3.6 TIME HORIZONS

The time horizon is the amount of time required to complete the project (Saunders, et al., 2012). There are two types of time horizons: the cross-sectional and the longitudinal. The cross-sectional involves a snapshot of data at a specific time, while the longitudinal time horizon collects data several times over a prolonged period to evaluate the change over time (Goddard and Melville, 2004). The study adopts a cross-sectional time horizon.

The study adopts the cross-sectional approach to collect the data, since the plan is to investigate challenges associated with only the introductory programming module and specific issues only for the semester the students are registered for. The study did not seek to understand how the factors and challenges change over time.

## 3.7 TECHNIQUES AND PROCEDURES

The techniques and procedures are concerned with the collection and analysis of the primary and secondary research data and contribute to the overall validity and reliability of the study (Saunders, et al., 2012). Primary data is obtained from first-hand sources, while secondary data is deduced from other researchers' opinions or work (Newman and Benz, 1998). The study questionnaires form part of the primary sources, whereas the literature review serves as a secondary source of data.

### 3.7.1 DATA COLLECTION

Data collection involves the gathering and measuring of data on variables under investigation in an organised, systematic way that enables one to answer relevant research questions and assess the outcomes (Creswell, 2013). Questionnaires are the primary source of data collection in the study. The use of questionnaires allows for the profiling of participants and gathering of data relating to their opinion, attitudes, circumstances, and behaviour (Creswell, 2013). The questionnaire is made up of structured closed-ended questions for the quantitative research approach and unstructured, open-ended questions for the qualitative research approach.

The online electronic tool SurveyMonkey was used to facilitate the online questionnaire, as it provides comprehensive functions that include the unique tracking of respondents for every survey request sent that is essential to the study. The online questionnaire consists of 27 questions that can be summarised into three main types: open-ended questions, close-ended questions, and a combination of both comprising three categories described in Table 3.4. The use of open- and close-ended questions in the survey questionnaire allows for the quantitative and qualitative evaluation of the responses to answer the research questions.

**Table 3.4: Survey questions categorisation**

| Category | Type | Area of interest | Number of questions |
|---|---|---|---|
| 1 | Close-ended | Programming, curriculum, and personal | 17 |
| 2 | Open-ended | Programming, curriculum, and personal | 6 |
| 3 | Mix | Programming, curriculum, and personal | 4 |

#### 3.7.1.1 SAMPLE AND SAMPLING PROCEDURES

The research population consists of students studying an introductory programming module at Unisa. The sample was taken from a population of 791 students studying the Introduction to Interactive Programming module (ICT1512) at Unisa. Questionnaires were distributed to all students who studied the module in 2014 and 2015 at Unisa, and students were informed that the responses are optional, which made the sample self-selection (Oates, 2006). The

selection of the students was based on the observation that the students will provide a reasonable sample for introductory programming students. The number of responses for the study consisted of 205 students, 91 in 2014 and 114 in 2015, which represents 26% of the population size.

The sample process was therefore based on a non-probability sampling technique, which means there is possible bias towards the ICT1512 programming module students who were available and willing to respond to the study (Floyd and Fowler, 2009). The survey may be prone to error due to non-response from the participants, but the non-response is unlikely to result in bias that could influence the content of the survey (Floyd and Fowler, 2009). However, the high level of responses helps reduce the likelihood of potential error due to non-response (Floyd and Fowler, 2009).

The following non-probability self-selection sampling factors were considered in the study:

- Unknown representation of the population being studied
- Possible bias in the respondents
- Lower level of generalisation

### 3.7.2 DATA ANALYSIS

The research data analysis is based on the convergent parallel mixed methods approach illustrated in Figure 3.3 (Creswell, 2013). After the collection of both quantitative and qualitative data, the data analysis takes place as follows:

- data from quantitative and qualitative responses are analysed independently;
- results are compared; and
- findings from the two are analysed for similarities and differences.

In the study, the side-by-side data comparison is used during data analysis while keeping the two data sets independent (Creswell, 2013). The results from the quantitative data analysis are discussed first; thereafter, the findings from the qualitative data analysis are merged with those from the quantitative data analysis to confirm (converge with) or negate (diverge from) the quantitative results. The merged results are interpreted by assessing the diverging and

converging aspects to have a better view of the situation under evaluation in order to answer the research questions.

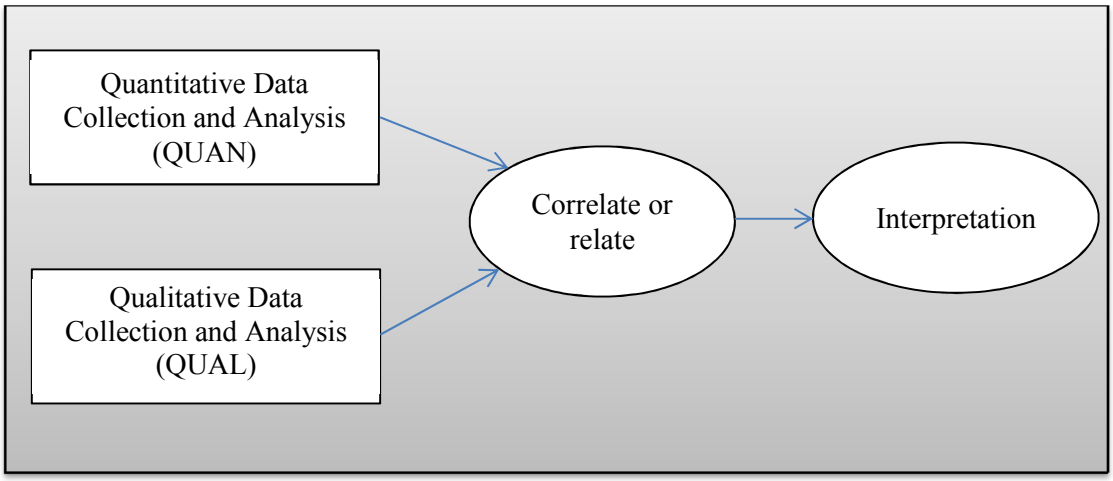


**Figure 3.3: Convergent parallel mixed methods**

Source: Creswell (2013)

Figure 3.4 shows the data analysis process used in the study to evaluate the responses. The data is triangulated using mixed methods to ensure a high level of accuracy and validity.

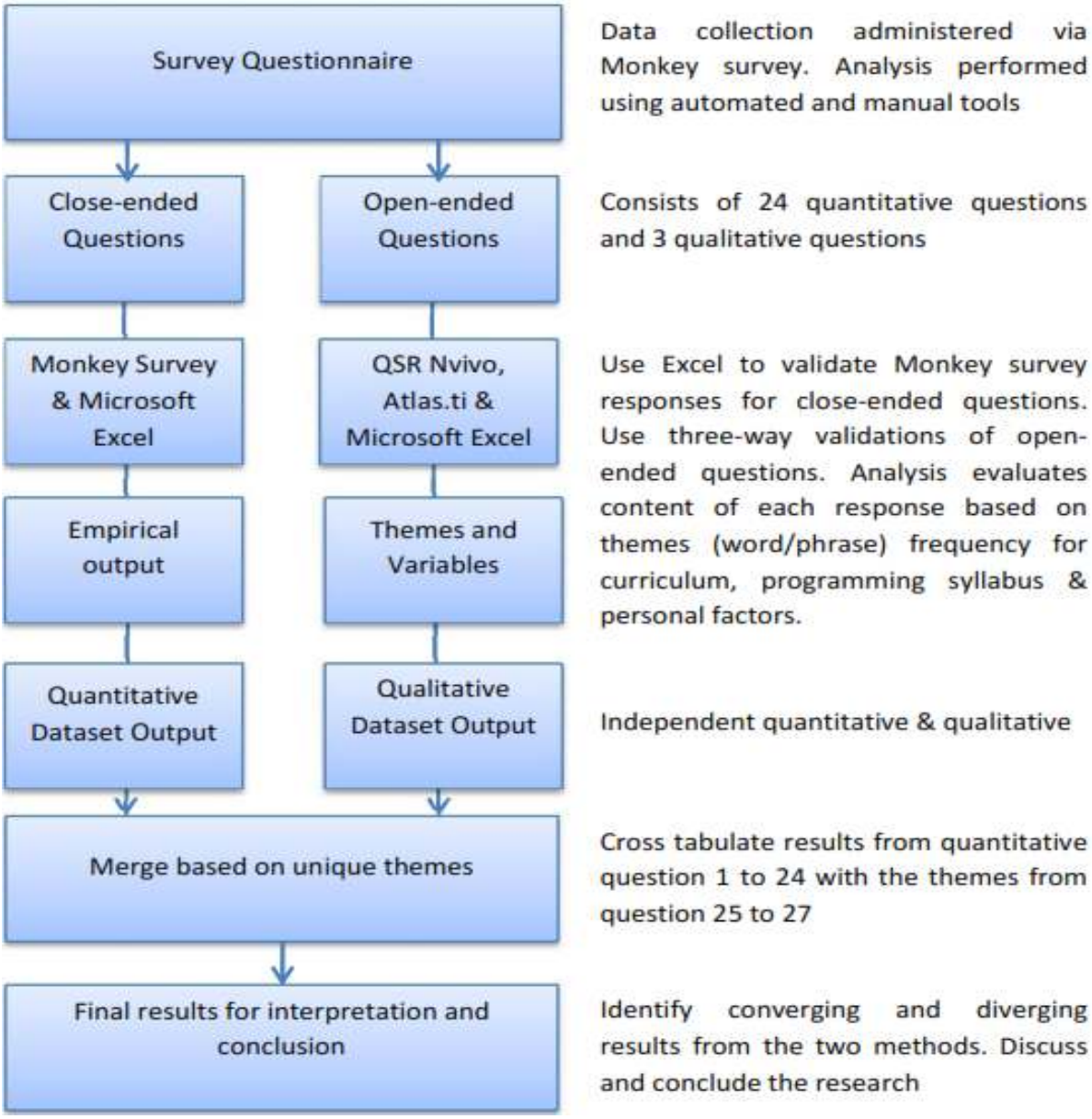


**Figure 3.4: The research data analysis**

## 3.8 SUMMARY

The research onion by Saunders, et al. (2012) provides the process model for the overall design and methodological approach of the study. The process model consists of stages that provide the research structure from the research paradigm to the methodology required to perform the research. As a result, this study is based on the acknowledgement that the study is founded on understanding reality, and the researcher's understanding is influenced by how he views reality, which, in turn, defines the methods he use to acquire

and interpret knowledge gathered and subsequent conclusions the researcher arrived at. The study adopts a pragmatic philosophy that holds the view that data can be collected, analysed, and interpreted using different methods to understand factors contributing to the high failure rates in introductory programming.

The study adopts both a deductive and an inductive research design to answer research questions from quantitative and qualitative research viewpoints. The research strategy in this chapter defined a survey as the best way to collect the research data based on the research questions. In addition, the flexibility of the survey provides access to a large number of students within a predefined time. The mixed methods approach is used in the study to ensure that questionnaire responses from the students can be collected, analysed, and presented with improved credibility from triangulation to merge both qualitative and quantitative results.

The next chapter deals with data analysis from a quantitative point of view.

# 4 DATA ANALYSIS: QUANTITATIVE

## 4.1 INTRODUCTION

The foregoing chapter discussed the research methodology employed in this study. This chapter focuses on the quantitative data analysis of the study based on the research questions in Chapter 1 on the curriculum, the programming syllabus, and personal factors that have an impact on learning to program. The literature review in Chapter 2 shows various research papers that studied the three areas with regard to introductory programming and also the gaps identified as a result of the review. The study planned to use knowledge derived from the literature reviewed. Additionally, the methodology outlined in Chapter 3 to analyse the data, in Section 4.2, Section 4.3 and Section 4.4. The survey questionnaires are based on the need to answer the main research question, "What are the factors that contribute to learning hindrances experienced by programming students at Unisa?"

## 4.2 DATA ANALYSIS: CURRICULUM FACTORS AFFECTING LEARNING TO PROGRAM

The data was collected from a total of 205 students out of 791, which translates to 26% of the students enrolled for Introduction to Interactive Programming (ICT1512) at the time of the survey (Appendix C). The outcome of the analysis is summarised at the end of this chapter to provide the final quantitative results of the research, which were used in conjunction with the qualitative results in Chapter 5 for interpretation and research finalisation.

The quantitative data was derived from 23 of 27 questionnaire questions. There were three types of quantitative questions, that is, the 5-point Likert scale (ranging from strongly disagree to strongly agree), dichotomous (Yes/No), and a list of items from which students could select multiple answers.

### 4.2.1 INSTITUTIONAL EDUCATION

The overall composition of participants studying Introduction to Interactive Programming (ICT1512) module at Unisa for the study is shown in Figure 4.1. The total number consisted largely of part-time registered students accounting for 81.5% of the students, while those that enrolled on a full-time basis represented only 18.5%.

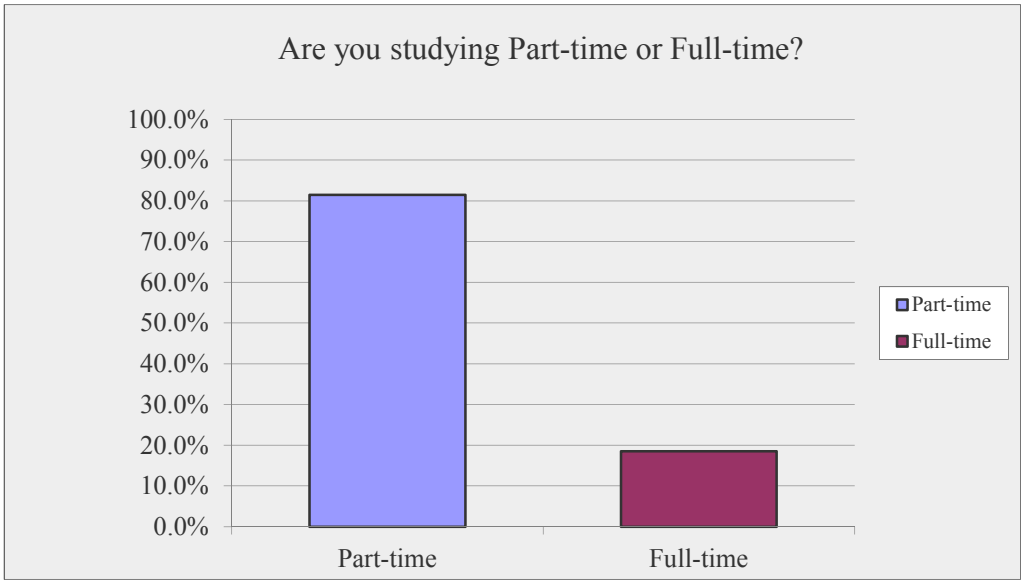


**Figure 4.1: Course enrolment schedule**

### 4.2.2 CURRICULUM PROGRAMME

The curriculum programme is analysed qualitatively in Section 5.2.1.

### 4.2.3 EDUCATIONAL MATERIALS

Because of the design of the research and the type of participants, the study does not contain appropriate data relevant for the analysis of the adequacy and relevance of educational materials for the module. The appropriateness of the materials used is, however, covered extensively under Section 4.3.2 where the structure and content of the materials used in the module are evaluated based on the perceptions provided by the students.

### 4.2.4 TEACHING AND LEARNING STRATEGY

When the students were asked to indicate the areas that were the most helpful in learning and understanding to program (as illustrated in Figure 4.2), most students indicated that prescribed books were the most effective, with a score of 75.1%, followed by practical exercises at 50.8% and learning units at 47.5%. Face-to-face tutorials were perceived to be the least helpful, with a score of 6.6%, while the analysis of responses for "other" (14.4%) showed that the students found the Internet useful in learning to program. The search engines and programming-related sites accounted to 44% of the Internet, while online video sites with programming material was 56%. The breakdown indicates that students relied on the videos the most compared to text-based information when visiting internet sites to learn to program. Figure 4.2 depicts the full statistical breakdown of various areas.

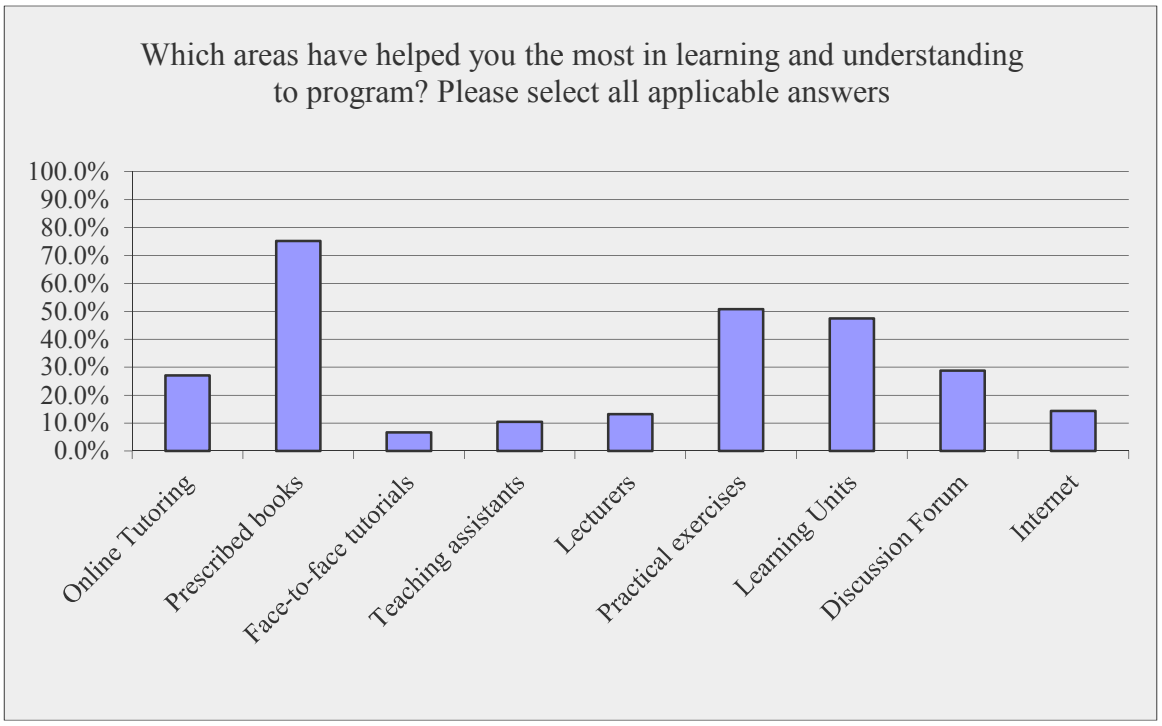


**Figure 4.2: Most helpful areas in learning to program**

## 4.3 DATA ANALYSIS: FACTORS RELATING TO THE PROGRAMMING SYLLABUS

Results in Figure 4.3 show that 29.9% of the students had studied the programming module before. In contrast, 70.1% of the students were studying the module for the first time.

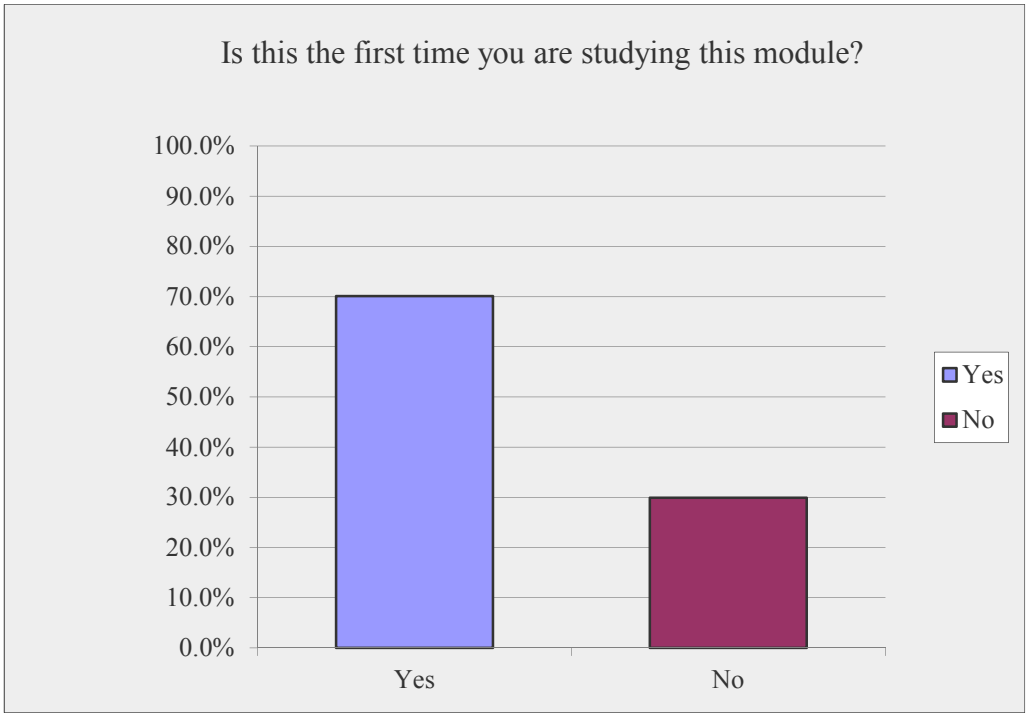


**Figure 4.3: New and repeating students**

### 4.3.1 TEACHING TO PROGRAM

The analysis of the data represented in Table 4.1 on whether students believed that the teaching staff explained the module outline and activities in a manner that was helpful to them showed that 39.5% of the students could not agree or disagree. There were 41.5% of the students who agreed (with 6% of this number strongly in agreement), while 19% indicated that the teaching staff did not explain things to them well, with 5.9% of that number putting very strong emphasis on how they feel.

**Table 4.1: Comprehensive teaching instructions**

| In general, the teaching staff for this module were good at explaining things. | | | | | |
|---|---|---|---|---|---|
| **Answer Options** | **Strongly Agree** | **Agree** | **Neutral** | **Disagree** | **Strongly Disagree** |
| | 12 | 73 | 81 | 28 | 11 |
| | 5.9% | 35.6% | 39.5% | 13.7% | 5.4% |
| | 41.5% | | 39.5% | 19.0% | |

### 4.3.2 LEARNING TO PROGRAM

The results of the analysis of the level of experience in programming among the students in Table 4.2 indicate that 10.4% of the students had a high level of experience, whereas 36.6% had a low level. Of the 36.6% that had a low level, 12.9% had a very low level of computer programming experience. The majority of the students, representing 53% of the total, indicated that they had neither a high nor a low level of experience in programming.

**Table 4.2: Level of experience in programming**

| What best describes your level of experience in programming? | | | | | |
|---|---|---|---|---|---|
| **Answer Options** | **Very High** | **High** | **Neutral** | **Low** | **Very Low** |
| | 2 | 19 | 107 | 48 | 26 |
| | 1.0% | 9.4% | 53.0% | 23.8% | 12.9% |
| | 10.4% | | 53.0% | 36.6% | |

Table 4.3 shows an analysis of the responses of the students as to whether they had spent enough time working on the programming exercises. A total of 37.7% of the students agreed, and of these, 5% strongly agreed. Moreover, 27.6% of them indicated that they had not dedicated enough time, with 5% of them indicating that they strongly agreed. Just fewer than 34.7% of the students remained neutral.

**Table 4.3: Time spent on programming exercises**

| **Do you feel you have spent enough time doing the actual programming exercises?** | | | | | |
|---|---|---|---|---|---|
| **Answer Options** | **Strongly Agree** | **Agree** | **Neutral** | **Disagree** | **Strongly Disagree** |
| | 10 | 65 | 69 | 45 | 10 |
| | 5.0% | 32.7% | 34.7% | 22.6% | 5.0% |
| | 37.7% | | 34.7% | 27.6% | |

The results of the analysis in Table 4.4 on whether the teaching materials had adequate or relevant content to help them learn to program as perceived by the students showed that 59.2% of the students agreed, followed by 24.4% of the students who did not indicate whether they agreed or disagreed. Just fewer than 17% of the students (16.4%) expressed disagreement on the adequacy and relevance of material content for the module.

**Table 4.4: Adequacy and relevance of the material content for module**

| **In your view, does the teaching material have enough and correct content for this level?** | | | | | |
|---|---|---|---|---|---|
| **Answer Options** | **Strongly Agree** | **Agree** | **Neutral** | **Disagree** | **Strongly Disagree** |
| | 27 | 92 | 49 | 24 | 9 |
| | 13.4% | 45.8% | 24.4% | 11.9% | 4.5% |
| | 59.2% | | 24.4% | 16.4% | |

When students were asked if they found it easy to follow and understand the structure and content of the module, 75.6% said “Yes”, 21% did not think so, and 3% did not provide an indication, as depicted in Table 4.5.

**Table 4.5: Module structure and content**

| Is the structure and content of the module easy to follow and understand? | | |
|---|---|---|
| **Answer Options** | **Response Percentage** | **Response Count** |
| Yes | 75.6% | 155 |
| No (please specify) | 21.0% | 43 |
| Skipped | 3% | 7 |

The students' responses in Table 4.6 on whether they felt that teaching instructions prepared them for the task (next reading chapter or assignment) that followed showed that 78.3% of the students agreed that what was given to them prepared them for the task that followed, 21.7% did not feel so, while 5% did not give an indication.

**Table 4.6: Transitional learning effectiveness for each chapter**

| Do you feel every chapter, assignment, or tutorial prepares you well enough for the next task? | | |
|---|---|---|
| **Answer Options** | **Response Percentage** | **Response Count** |
| Yes | 81.5% | 167 |
| No (please specify) | 13.2% | 27 |
| Skipped | 5% | 11 |

The results of the analysis of the level of understanding of the programming module content among the students (Table 4.7) indicated that the percentages of students with the perceived high level and low level of understanding were 22.9% and 18.5% respectively. The majority of students with neither a high level nor a low level represented 58.5% of the total.

**Table 4.7: Level of understanding of the programming module content**

| What is your level of understanding of the module content? | | | | | |
|---|---|---|---|---|---|
| **Answer Options** | **Very High** | **High** | **Neutral** | **Low** | **Very Low** |
| | 5 | 42 | 120 | 29 | 9 |
| | 2.4% | 20.5% | 58.5% | 14.1% | 4.4% |
| | 22.9% | | 58.5% | 18.5% | |

In the analysis of the responses from the students' responses (Table 4.8) on whether they found it confusing to learn the programming syntax, most students (39.2%) did not indicate whether they agreed or disagreed. A total of 31.2% agreed, with 8.5% of these students indicating that they strongly agreed. The remaining 29.6% of the total number of students indicated that it was not confusing to learn the programming syntax.

**Table 4.8: Learning programming syntax**

| Learning the programming language syntax is confusing. | | | | | |
|---|---|---|---|---|---|
| **Answer Options** | **Strongly Agree** | **Agree** | **Neutral** | **Disagree** | **Strongly Disagree** |
| | 17 | 45 | 78 | 48 | 11 |
| | 8.5% | 22.6% | 39.2% | 24.1% | 5.5% |
| | 31.2% | | 39.2% | 29.6% | |

Further analysis on the ability of the students to work with programs as represented in Table 4.9 shows that 26.6% of the students agreed that they wrote, compiled, ran, and debugged their own programs. The majority, represented by 41.1% of the students, did not indicate whether they had difficulties or not, whereas 32.2% of them indicated that they had challenges.

**Table 4.9: Program design, development, and execution**

| I can write, compile, run, and debug my own programs. | | | | | |
|---|---|---|---|---|---|
| **Answer Options** | **Strongly Agree** | **Agree** | **Neutral** | **Disagree** | **Strongly Disagree** |
| | 3 | 21 | 37 | 22 | 7 |
| | 3.3% | 23.3% | 41.1% | 24.4% | 7.8% |
| | 26.7% | | 41.1% | 32.2% | |

In Table 4.10, just fewer than 40% (39.3%) of the students agreed that they found it hard to understand errors from the programs they worked with. The analysis also shows that 26.8% of the students did not have any difficulties. The remaining 33.8% of the total neither agree nor disagree.

**Table 4.10: Program run-time error analysis**

| I have difficulties understanding errors from my own programs. | | | | | |
|---|---|---|---|---|---|
| **Answer Options** | **Strongly Agree** | **Agree** | **Neutral** | **Disagree** | **Strongly Disagree** |
| | 19 | 60 | 68 | 48 | 6 |
| | 9.5% | 29.9% | 33.8% | 23.9% | 3.0% |
| | 39.3% | | 33.8% | 26.9% | |

### 4.3.3 PROGRAMMING IN PRACTICE

When students were asked if they found it easy to design a program to solve a certain task (Table 4.11), 46.8% neither agreed nor disagreed, while 23.2% agreed. A total of 24.1% disagreed, with 5.9% of those students strongly disagreeing.

**Table 4.11: Practical application of programs developed**

| It is easy for me to design a program to solve a certain task. | | | | | |
|---|---|---|---|---|---|
| **Answer Options** | **Strongly Agree** | **Agree** | **Neutral** | **Disagree** | **Strongly Disagree** |
| | 4 | 43 | 95 | 49 | 12 |
| | 2.0% | 21.2% | 46.8% | 24.1% | 5.9% |
| | 23.2% | | 46.8% | 30.0% | |

## 4.4 DATA ANALYSIS: PERSONAL FACTORS AFFECTING LEARNING TO PROGRAM

### 4.4.1 PRIOR LEARNING

The analysis of the level of computer literacy among the students (Table 4.12) indicated that 80% of the students had a high level of computer literacy, while 18.8% of the students reported having neither a high nor a low level of computer literacy. The analysis further showed that a relatively low number of students had a low level of computer literacy compared to those with a high level.

**Table 4.12: Level of computer literacy**

| What best describes your level of computer literacy? | | | | | |
|---|---|---|---|---|---|
| **Answer Options** | **Very High** | **High** | **Neutral** | **Low** | **Very Low** |
| | 69 | 93 | 38 | 1 | 1 |
| | 34.2% | 46.0% | 18.8% | 0.5% | 0.5% |
| | 80.2% | | 18.8% | 1.0% | |

Analysis of students' responses as depicted in Figure 4.4 shows that the high number representing 72.1% of the students were taking formal programming classes for the first time.

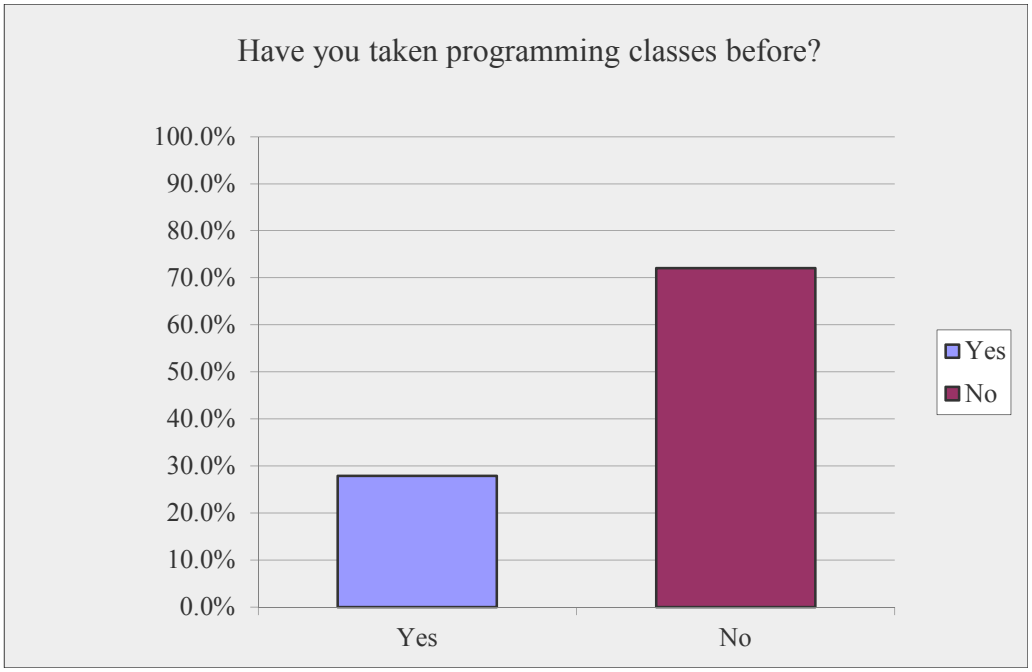


**Figure 4.4: Prior exposure to programming**

Figure 4.5 shows that 92.6% of the students had access to personal computers compared to 7.4% without access.

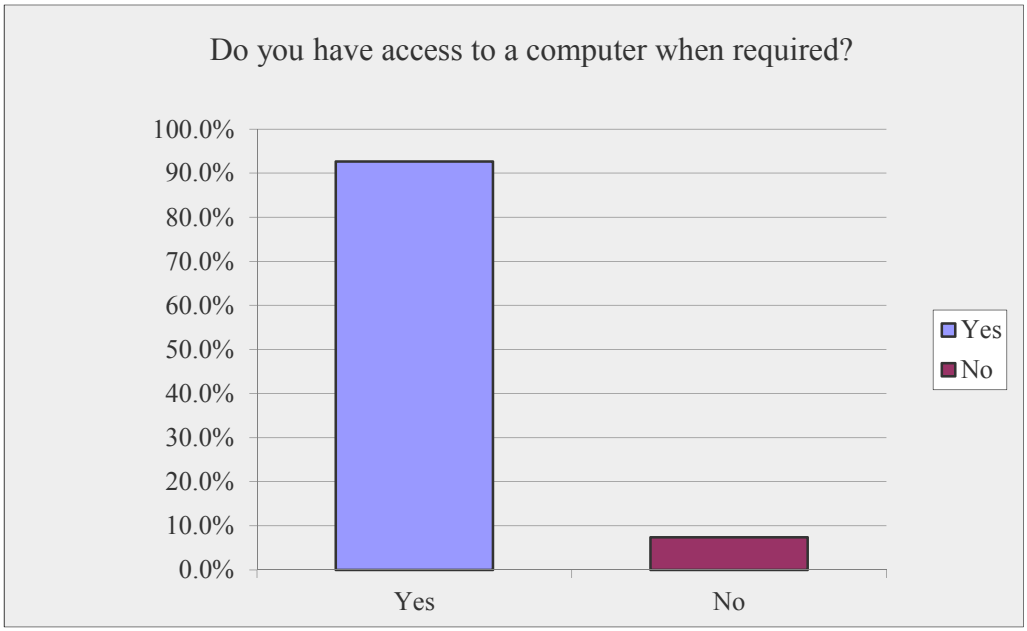


**Figure 4.5: Access to personal computer**

### 4.4.2 APTITUDE AND COGNITIVE FACTORS

Learning style is one of the cognitive factors that may impede learning to program (Jenkins, 2002; Eccles and Wigfield, 2002). Students who adopt the wrong learning style are likely to find learning to program difficult (Jenkins, 2002; Roddan, 2002; Kinnunen and Malmi, 2006).

In Table 4.13, 44% of the students reported that they had put consistent effort in their studies in learning to program, while 18.8% of them reported that they had not done so. Thirty-seven per cent of the students did not indicate whether they had worked consistently or not.

**Table 4.13: Consistent dedication throughout the term**

| **Do you feel you have worked consistently throughout the term on this module?** | | | | | |
|---|---|---|---|---|---|
| **Answer Options** | **Strongly Agree** | **Agree** | **Neutral** | **Disagree** | **Strongly Disagree** |
| | 17 | 70 | 73 | 32 | 5 |
| | 8.6% | 35.5% | 37.1% | 16.2% | 2.5% |
| | 44.2% | | 37.1% | 18.8% | |

Analysis of the responses from the students in Figure 4.6 shows the number of students who dedicated time as prescribed in the study guide for the module was 47.5%. Students that did not follow the guideline accounted for 52.5% of the students.

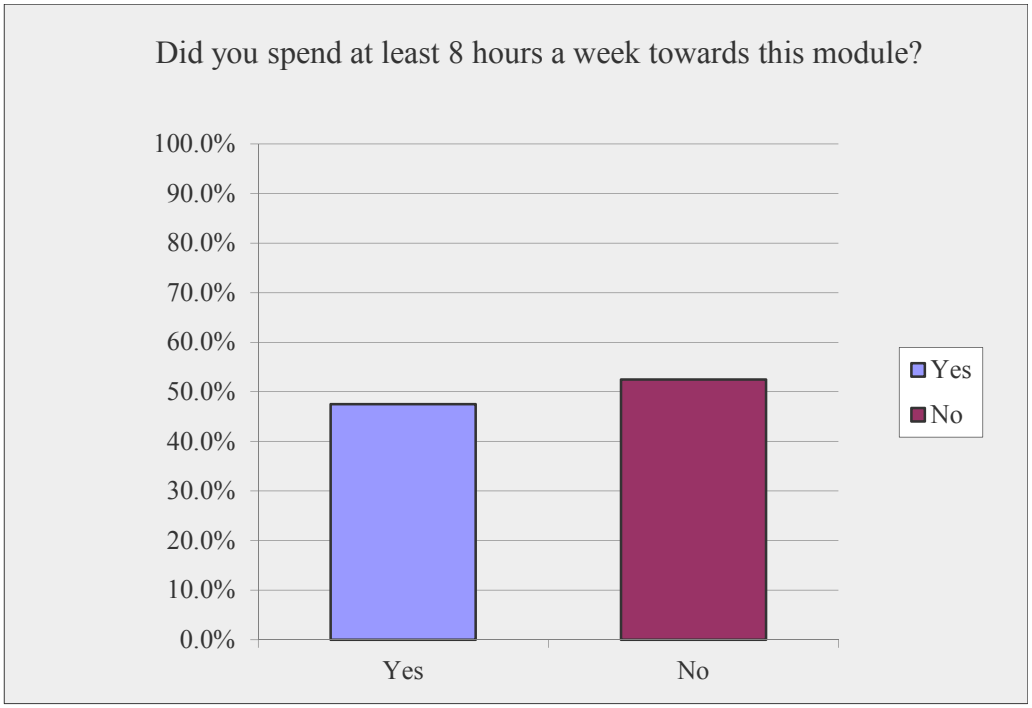


**Figure 4.6: Hours dedicated by students as prescribed for the module**

Analysis of the responses from the students in Figure 4.7 indicates that 70.1% of the students take part in or are at least aware of the online discussions relating to the programming module they have registered for. The students who do not participate or do not follow the discussions represent 29.9% of the total.

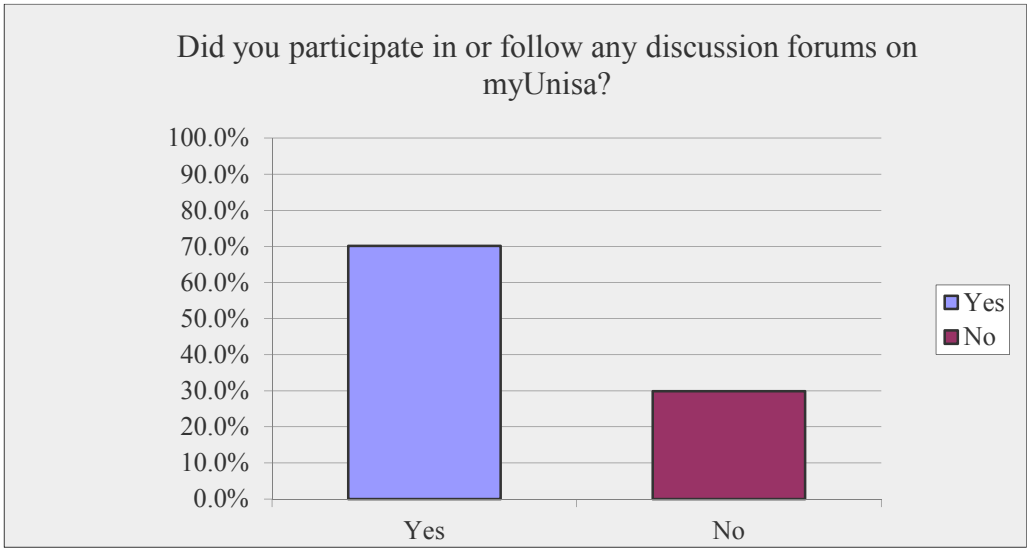


**Figure 4.7: General participation in the online discussion**

Results from the students' responses shown in Table 4.14 reveal that the majority of the students did not study with their peers, followed by 16.1% of the students who studied with their peers when in need of assistance. Further analysis shows that the percentage of students studying with peers weekly was 7.8%, whereas 6.3% did so a few times a week. Those who studied with peers around the examination period represented 6.3%. Only 1% studied with peers once a month.

**Table 4.14: Studies with fellow students**

| **On average, how often did you study with fellow students?** | | |
|---|---|---|
| **Answer Options** | **Response Percentage** | **Response Count** |
| Weekdays (daily) | 4.9% | 10 |
| Once a month | 1.0% | 2 |
| Only towards exams | 5.9% | 12 |
| Once a week | 6.3% | 13 |
| Only when in need of assistance | 15.6% | 32 |
| Never | 58.0% | 119 |
| Other (please specify) | 8.3% | 17 |

Results from the students' responses in Table 4.15 indicate that 53.5% of the students consulted with their tutor or attended tutor sessions only when they needed assistance. The second-largest percentage was 30.5%, and further analysis of the 30.5% shows that the students had indicated that they had never attended tutor sessions or sought help from any tutor.

**Table 4.15: Tutor and tutorial assistance**

| On average, how often did you attend tutor sessions or consult with a tutor? | | |
|---|---|---|
| **Answer Options** | **Response Percentage** | **Response Count** |
| Weekdays (daily) | 1.5% | 3 |
| Once a month | 5.5% | 11 |
| Only towards exams | 3.5% | 7 |
| Once a week | 5.0% | 10 |
| Only when in need of assistance | 52.5% | 105 |
| Other (please specify) – Never | 32.0% | 64 |

### 4.4.3 PERSONAL REASONS AND COMMITMENTS

Table 4.16 outlines the percentage breakdown of students' responses when asked to provide the reason that led to them considering withdrawing or caused them to drop out. The majority of the students (38.5%) indicated that time was the main factor, followed by 29.8% of the students who gave work or personal commitment as the reasons, while the course being a wrong choice was the most insignificant factor with less than 1%.

Motivation came as the fourth factor that leads to students considering withdrawing from their studies, with 12.7% of the students feeling that way, and relates to one of the cognitive factors that have been described in detail in Section 2.4.2. Motivation has been shown by studies to impede learning (Jenkins, 2002; Roddan, 2002; Kinnunen and Malmi, 2006).

**Table 4.16: Reasons for considering withdrawing from the module**

| **If so, any particular reason you have considered withdrawing from or did not finish this module? Please select all applicable answers.** | | |
|---|---|---|
| **Answer Options** | **Response Count** | **Response Percentage** |
| The module is too advanced beyond my capabilities | 15 | 7.3% |
| The module required more time than I could provide | 79 | 38.5% |
| I lacked motivation to study | 26 | 12.7% |
| The module content is confusing and difficult to follow | 23 | 11.2% |
| Tutors or lecturers offered inadequate support | 11 | 5.4% |
| I chose the wrong course | 1 | 0.5% |
| My work or personal-related commitments took time from the course | 61 | 29.8% |
| I had a personal reason(s) that compromised my performance in this module | 35 | 17.1% |

## 4.5 SUMMARY

The quantitative data analysis of the students' responses on the curriculum questions indicated that 82% of the participants registered for the programming module were enrolled for part-time distance learning education. The analysis of the data relating to the programming syllabus showed that 30% of the students had taken the module before. Just over 40% (41%) of the students felt that the teaching staff did not explain things well. The assessment of the results from the question asked on the level of experience in the programming module showed that 53% of the total number of students were unsure about whether they had a high or a low level of experience in programming. In fact, only 1% of the students indicated that they had a high level, yet the majority of the students (38%) believed that they had spent enough time practising the module.

The majority of the students felt that prescribed books were the most useful in learning and understanding to program and also indicated that the teaching instructions at hand prepared them for the next task. When it came to the module, 78% perceived the structure and content

comprehensive, and 59% felt that the module had appropriate and enough content. Fifty-nine per cent of the students were not sure if they understand the content of the programming module. The results from the students' responses relating to the practical application of programming brought out that many of the students (47%) were not sure if they could write their own programs for a specific purpose. Thirty-one per cent of the students found the programming syntax confusing, while 30% thought otherwise. Many of students (41%) had difficulties in writing programs that functioned, whereas 39% did not know how to fix the errors.

The final analysis of the quantitative data related to personal factors, and the results showed that 80% of the students were computer literate, while 72% had not been formally exposed to programming before. A significantly high number of students (over 93%) had access to a personal computer. Forty-four per cent of the students felt that they had worked consistently since registering for the module. The number of students who dedicated eight hours weekly, as prescribed by the institution, to the programming module was slightly lower (48%) than that of students who did not (53%).

When it came to seeking assistance or studying with fellow students, most students (62%) never studied with their peers, followed by 16% of the students who did so only when in need of assistance. An assessment of tutors and tutorial classes showed that 53.5% of the students – representing the majority – sought assistance from the tutors or took tutorial lessons, while 31% never do so. When students were asked to indicate factors that had adverse effects on their continuing with their studies, time and commitments to other personal matters were the top-most factors respectively.

The next chapter focuses on the qualitative analysis of the research data.

# 5 DATA ANALYSIS: QUALITATIVE

## 5.1 INTRODUCTION

The previous chapter was an analysis of data from a quantitative point of view. This chapter focuses on the qualitative data analysis of the research based on the research questions on curriculum, programming syllabus, and personal factors that have an impact on learning to program. In Chapter 4, the quantitative data analysis was performed based on the research questions in Chapter 1, the outcome of the review of literature in Chapter 2, and the research methodology set out in Chapter 3. The quantitative analysis of the data provides statistical results needed to understand the various factors affecting learning to program based on the data collected from the students. In this chapter, the survey data will be analysed qualitatively in Section 5.2 through the assessment of patterns to negate or confirm the quantitative results and ultimately answer the main research question, "What are the factors that contribute to learning hindrances experienced by programming students at Unisa?"

## 5.2 DATA ANALYSIS: FACTORS AFFECTING LEARNING TO PROGRAM

The data was collected from a total of 205 students out of 791, which translates to 26% of the students enrolled for Introduction to Interactive Programming (ICT1512) at the time of the survey. The outcome of the analysis is summarised at the end of this chapter to provide the final qualitative results of the research to be used in conjunction with the quantitative results in Chapter 4 for interpretation.

The qualitative data was derived from question 23 to 27 of the questionnaire. The questions were open-ended so that students could express their views unrestricted. The open-ended questions allowed the students to provide more information on the difficulties and experiences they had in learning to program. The results of the questions provided deeper insight into certain factors contributing to hindrances in learning to program that were not

considered during the planning of the research. It should be noted that some of the original responses from the students used in the study have been slightly modified, without a change in meaning, for readability purposes.

### 5.2.1 CURRICULUM PROGRAMME

This section presents a qualitative analysis of the curriculum programme based on the themes from qualitative Question 26 of the survey where students were asked, "How can this module be improved?" Table 5.1 shows the breakdown of the responses analysed versus those deemed inadmissible for consideration in this study. The analysed responses consist of two categories derived from the themes. The first category represents the responses that were analysed but isolated, since the students concerned had indicated that there was nothing to improve in the module. The second category consists of questions analysed in detail for the purpose of the study. The second category is further analysed based on Tables 5.2 and 5.3.

**Table 5.1: Question 26 – Responses analysed**

| | | |
|---|---|---|
| **Inadmissible** | No comment, skipped and invalid answer | 114 |
| **Analysed (Isolated)** | Students satisfied | 31 |
| **Analysed (detailed themes defined)** | Provided improvement comments | 60 |
| **Total** | | 205 |

The results in Table 5.2 relating to the curriculum improvement suggestions for the module are represented in two areas, that is, the institution-related- and programming module-related improvements. The institutional improvements are improvements deemed as far-reaching and beyond the outline of the specific module. The improvement suggestions include the view of changing the module schedule from a semester- to a year-based offering or changing the way the institution delivers books. The module-related improvement suggestions are still curriculum-related changes but could be changed within the confines of the module. Such changes could include the number of assignments, the amount of work given to the students, and how module content is explained or taught.

**Table 5.2: Question 26 – Summary of improvement suggestions**

| Suggested Improvements by Area | Count | % |
|---|---|---|
| Curriculum – related to institution | 60 | 57% |
| Curriculum – related to module | 47 | 43% |
| **Total** | **107** | **100%** |

Further analysis of the responses shown in Table 5.3 indicates that classes, programmes, and tutorials were the major improvement suggestions. Classes primarily involve the request by the students to have the university offer "face-to-face" classes. Some students suggested classes more regularly, with most suggesting daily. General administration involves many different suggestions, including improvements in the induction so that it explains to the students more clearly what the module is about; the reduction in blogging and changing the curriculum structure of the module; and improvements in tutorials, whether they be online- or class-based tutorials. Other improvements are summarised below.

- Classes: Face to face, extra, and more time-flexible
- General administration: delays, support, and communication
- Tutorial: additional and more comprehensive tutorial lessons
- Tutors: availability of and quicker response time from the tutors
- Too many assignments: reduction in number of assignments
- Programming: improvement in teaching, especially on concepts or foundation
- Practical application of programming: introduction of more practical exercises
- Reduced workload: generally, the reduction in the number of assignments
- Material content: changing of content in the prescribed books with emphasis on practical exercises
- Support from lecturers: the primary theme being "more support from lecturers"
- More timely educational materials: the time it takes for the study materials to reach the students upon registration

**Table 5.3: Question 26 – Details of improvement suggestions**

| Curriculum Improvements for the Module | Count | % | Related to |
|---|---|---|---|
| Classes (Face to face, extra, and more time-flexible) | 25 | 23% | Institution |
| General administration (delays, support, and communication) | 21 | 20% | Institution |
| Tutorials (extra and more comprehensive) | 14 | 13% | Module |
| Tutors (availability and more time) | 7 | 7% | Institution |
| Too many assignments | 7 | 7% | Module |
| Programming (write, run, test, and apply programs) | 7 | 7% | Module |
| Practical exercises | 7 | 7% | Module |
| Reduce workload | 6 | 6% | Module |
| Material content | 6 | 6% | Module |
| Support from lecturers (face to face and more frequent) | 4 | 4% | Institution |
| Delay in study material | 3 | 3% | Institution |
| **Total** | **107** | **100%** | |

### 5.2.2 PROGRAMMING SYLLABUS

This section shows a qualitative analysis of the programming syllabus based on the themes from Question 24, where students were asked, "Which programming parts of this module do you feel have mostly compromised your ability to succeed?" (Table 5.4).

**Table 5.4: Question 24 – Problematic coding areas in programming**

| Issue | Number of Students | % |
|---|---|---|
| Ability to understand concepts and write programs | 72 | 91% |
| Having issues with compiling of programs | 2 | 3% |
| Unable to successfully run programs | 1 | 1% |
| Issues with analysing and fixing errors during program debugging | 4 | 5% |
| **Total** | **79** | **100%** |

The question in Table 5.4 was informed by the need to find the most compromising factors that lead to specific hindrances in the programming module. The themes were built from the

responses to ascertain the in-depth issues associated with the different areas of programming. This form of breakdown of programming areas also ensured that the results of this analysis (Question 24) could be mapped with the results from the survey's quantitative Questions 16, 17 and 19 below of the quantitative research, where students were asked to indicate if they "… can write, compile, run, and debug my own programs". The survey questions asked are as follows:

- Question 16: "It is easy for me to design a program to solve a certain task".
- Question 17: "Learning the programming language syntax is confusing".
- Question 18: "I have difficulties understanding errors from my own programs".

The areas in Table 5.5 have been derived from different themes of the students' responses for qualitative Question 24 that asked the students to indicate the problematic programming areas. The results provide details of various areas and percentage breakdown based on the results presented in Table 5.4.

**Table 5.5: Question 24 – Detailed breakdown of problems in coding**

| **Programming Areas of the Module** | **Count** | **%** |
|---|---|---|
| Programming concepts | 36 | 46% |
| Loops, functions, conditions, arrays, string, variables | 17 | 22% |
| Building programs (coding) | 15 | 19% |
| IDE (run & compile) | 9 | 11% |
| Programming syntax | 2 | 3% |
| **Total** | **79** | **100%** |

The majority of the students had general issues with understanding the basic programming concepts on how to write a simple program. The majority of the students answered the question on what had compromised their ability to succeed with short answers such as "*everything*" or "*all of them*". Student 66 simply said, "*Covering all the necessary basics*" (Appendix H1).

There were a significant number of students who had issues working with different aspects of programming. These aspects of programming included functions, variables, control structures, and data structures in general. Student 154 had difficulties working with

programming concepts and had indicated that "*... it has been really difficult to figure out what to do*". The challenge with specific areas in programming is evident with Student 73 indicating that some difficulties in "*understanding and writing of functions, arrays*" (Appendix H2).

The third-largest group of respondents highlighted that they had difficulties with coding. Student 125 indicated, "*writing codes is very challenging to me, ...*" while Student 122 said, "*if the coding part was straightforward and understandable I think I would have mastered this module a long time ago*" (Appendix H3).

Some students had difficulties with syntax. Student 9 remarked that it was not easy to know "*where to place certain syntax Why do you place this there and that there?*" and Student 3 indicated that it was challenging "*to just get a simple programme to run*" (Appendix H4).

### 5.2.3 PERSONAL FACTORS

This section presents a qualitative analysis of the programming syllabus based on the themes from Question 25 of the survey, where students were asked, "If you were to study this module again, what would you do differently in order to do even better?" This question was developed on the view that students would share adverse personal experiences and how they had learned from such experiences. Table 5.6 depicts information on the number of responses considered for analysis, given the validity of the responses relative to the questions asked. Seventy-nine responses were not included in the analysis in Table 5.7 and Table 5.8 because students highlighted that they either had "no comment" or did not give a valid answer to the question asked or simply skipped the question. As a result, 126 responses formed part of the analysis for this question.

**Table 5.6: Question 25 – Responses considered**

| | | |
|---|---|---|
| **Inadmissible** | No comment, skipped and invalid answer | 79 |
| **Analysed** | Provided comments | 126 |
| **Total** | | **205** |

Table 5.7 shows a summary of the themes derived from the details in Table 5.5.

**Table 5.7: Question 25 – Summary of the responses analysed**

| Personal Improvement Area | Count | % |
|---|---|---|
| Related to curriculum | 9 | 7% |
| Related to programming syllabus | 20 | 16% |
| Related to self | 97 | 77% |

The summary helps align the results of this question to the overall design of the research that sought to answer the three secondary questions regarding the curriculum, programming module, and personal factors. This form of categorisation will help with proper interpretation of the data analysis results in the next chapter. The results presented in this section are primarily related to the students' personal issues that can be linked to the research question in Section 1.6 that asks, "What are the personal factors that have an impact on learning?"

The results shown in Table 5.7 represent the students who feel that the way they went about studying the module was not the best way. They also provided various ways on how they can do things differently to improve in the areas shown in Table 5.8.

**Table 5.8: Question 25 – Detailed themes**

| Frequency | Count | % |
|---|---|---|
| Time | 46 | 37% |
| Self-improvement | 21 | 17% |
| Better planning | 17 | 13% |
| Practical exercises | 14 | 11% |
| Assistance (from tutors, experts, and lecturers) | 9 | 7% |
| Dedication | 7 | 6% |
| Programming (coding, syntax, design, application) | 6 | 5% |
| Reduce subjects | 4 | 3% |
| Prior learning | 2 | 2% |

Factors related to self in Table 5.7 accounted for 77% of the total, which was the majority. The data analysis factors show that the students felt that they needed to change certain ways of conducting themselves if they wanted to do better. The changes relate to self-

improvement, better planning, prior learning, time management, dedication, and reductions in the number of course modules taken simultaneously (Table 5.8). Time management, which was the factor that contributed the most, saw the majority of students writing phrases such as "*put in more time*", "*time is never enough*", and "*allocate more time*" (Appendix H5). Self-improvement was linked to understanding the course outline and schedule; "working smarter"; having curriculum- and institutional requirements such as books; technology in place and time; adhering to the plan throughout the study period; and planning around job- and family-related commitments. Better planning was primarily linked to the need for the students to understand vague and difficult areas of the curriculum as soon as they enrolled for the module. "Dedication" was simply an indication by the students that they needed to dedicate more effort to their studies. Prior learning is also indicated as a factor where students felt that they should have taken a foundation programming course or learned other functions of programming prior to working with actual programming. Analysis of reduction in the number of subjects, as indicated by the students, shows that this factor was generally as a result of the demands associated with the module. Student 149 responded, "*this module need a lot of attention that means to take this module alone without any other modules ... especially what is expected from it*" (Appendix H6), while Student 44 said, "*I would take less other subjects at the same time*" (Appendix H6).

Factors related to the programming syllabus were represented by 17% of the total 126 students in Table 5.7. The students concerned saw the need to change certain elements of the programming module, which were those related to learning to program, writing programs, and doing practical exercises and assignments as shown in Table 5.8. The elements of programming included learning basic code, writing concepts, learning how to write programs, and learning how to apply the programs in practice. The practical exercises involved dedicating more time working on or experimenting with the actual program.

The final factor related to the curriculum, represented by 7% of the total 126 students in Table 5.7, links to "assistance" required by the students. Students primarily indicated that they would consider seeking assistance, as shown in Table 5.8, from the tutors or through the tutoring classes.

### 5.2.4 CURRICULUM, PROGRAMMING SYLLABUS, AND PERSONAL FACTORS

The qualitative data analysis of the themes from qualitative Question 27 of the survey encouraged the students to voice their opinions. “Please feel free to comment here on any aspect, positive or negative, of your learning experience on this module”. The question was developed on the view that the students would share personal experiences relating to the module. The experiences were expected to refer to personal challenges, the subject and curriculum, and lecturers, tutors, and university-related issues.

Table 5.9 shows the number of responses considered for analysis, given the validity of the responses relative to the questions asked. One hundred and fourteen responses were not included in the analysis in Table 5.10 and Table 5.11 because students indicated that they either had “no comment” or did not provide a valid answer to the questions or simply skipped the question. As a result, 60 responses formed part of the analysis for this question. The responses were analysed to see common and diverging themes, grouped into different categories as shown in Table 5.11, then summarised into three areas represented in Table 5.10.

**Table 5.9: Question 27 – Responses analysed**

| | | |
|---|---|---|
| **Inadmissible** | No comment, skipped and invalid answer | 114 |
| **Analysed (isolated)** | Students satisfied | 31 |
| **Analysed (detailed themes defined)** | Provided improvement comments | 60 |
| **Total** | | 205 |

The summary of the responses for Question 27 in Table 5.10 suggests that 43% of the students felt that the majority of the challenges that led to hindrances in learning to program were as a result of personal issues. The students feel that they need to improve on the management of personal issues they have in order to succeed in learning to program. The second contributing factor represented 40% of the total and involved improvements associated with the curriculum. The programming module-related improvements contributed to 17% of the total.

**Table 5.10: Question 27 – Summary of suggestions for improvement**

| Suggested Improvements by Area | Count | % |
|---|---|---|
| Curriculum | 24 | 40% |
| Programming syllabus | 10 | 17% |
| Personal (students) | 26 | 43% |
| **Total** | 60 | |

Table 5.11 provides details of the areas described in Table 5.10.

**Table 5.11: Question 27 – Details of suggestions for improvement**

| Detailed Suggested Improvements | Count | % |
|---|---|---|
| General administration | 9 | 15% |
| Time | 9 | 15% |
| Self-improvement | 8 | 13% |
| Cognitive | 5 | 8% |
| Module structure | 5 | 8% |
| Classes | 4 | 7% |
| Aptitude | 4 | 7% |
| Practical exercises | 3 | 5% |
| Delay in study material | 3 | 5% |
| Tutors | 2 | 3% |
| Material content | 2 | 3% |
| Tutorial | 2 | 3% |
| Support from lecturers | 2 | 3% |
| Assignments | 2 | 3% |

- Support from the lecturers, delay in study material, classes, material content, tutors, tutorial, and general administration form part of the curriculum factors.
- Programming, practical exercises, and assignments relate to the programming syllabus.
- Cognitive, time and self-improvement, and aptitude constitute personal factors.

Support from the lecturers relates to support and communication; delay in study material is primarily linked to the time it takes for the students to receive material; “tutors” is linked to the availability of tutors; and “tutorials” relates to the need to have contact in tutorial classes.

General administration involves the module schedule and the combination of different modules together at the same time, given the demands associated with the programming module and blogging requirements, and "material content" is generally linked to lack of practical exercises and also to solutions to programming found in the study materials. Student 177 highlighted the situation as follows, "*I found the textbook a bit lacking with regards to the exercises but no solutions, so you never really know if you are on the right track, because you need to post your exercise answers to the discussion board ... Usually by that time you have moved on to next chapter only to find out your understanding of previous chapter was wrong."* (Appendix H7).

"Module structure" relates to the outline of the module chapters and the approach to the way it is taught, with Student 1 describing it as "*... really, really long and tedious ...*" and Student 205 saying that "*Javascript is an embedded program. It embeds into html. You can't run it on its own. Why a study module can be structured to be learnt like this, without prior Html grounding ...*" (Appendix H8). "Practical exercises" involves the need for more practical exercises; "assignments" is the indication that the number of assignments is not adequate, with some suggestions to have an assignment for every chapter.

"Cognitive" is the challenge to find the best way to learn to program; "time" is primarily the challenge to dedicate adequate time for the activities in the module; "self-improvement" is linked to various improvements students feel that they need to make or are unable to make. "Self-improvements" includes balancing personal commitments and studies, as well as finding personal means to ensure that students' studies are not compromised; "aptitude" is concerned with the inability to understand programming in general as described by Student 59. This student stated, "*The module needs to be simplified since we do it on ODL and I think we are struggling a lot and I do not think it's me who is experiencing this kind of challenge as am repeating these module for several times*." (Appendix H9).

Another dimension considered in the study was the number of students who were content with the experience they had during their studies. By separating those satisfied, the number of students who felt that improvements were necessary can be uncovered. Table 5.12 shows that the majority of the students (66%) felt that certain improvements were required, while 34% of them were satisfied with how they had managed their studies for the module and with the general setup of the curriculum programme, including the programming syllabus.

**Table 5.12: Question 27 – Students' experience with the module**

| Description | Count | % |
|---|---|---|
| Unsatisfied | 60 | 66% |
| Satisfied | 31 | 34% |

## 5.3 SUMMARY

The qualitative data analysis of the students' responses from the question regarding the learning experiences shows that 66% of the students felt that improvements were necessary, while 34% of the students were satisfied. The breakdown of the results was the curriculum (40%), programming syllabus (17%), and personal factors (43%) of the total based on the responses from those who had indicated that improvements were necessary.

The data analysis relating to the curriculum programme shows that the majority of curriculum-related improvements (57%) were related to the university. The next highest number was factors relating to the programming module (43%).

The high-level view of the data analysis relating to the programming syllabus showed that the majority of the students (91%) had difficulties in writing basic programs, with the remainder of the 9% having issues with program compilation, execution, and error handling. The detailed analysis of the different factors contributing to the challenges in programming showed that the root causes were programming concepts (46%), working with building blocks, and overall building of programs (44%), and last, 11% related to the IDE, specifically program execution and error management.

The outcome of the data analysis on personal factors contributing to challenges in learning to program showed that the majority of the contributing factors related to the students (77%), followed by programming syllabus (16%), then the curriculum (7%). Personal factors were primarily linked to time (37%) and self-improvement (17%), better planning (13%), and inadequate practical exercises (11%).

The chapter that follows provides the analysis of data from both quantitative and qualitative data from the survey responses.

# 6 DATA ANALYSIS: QUANTITATIVE AND QUALITATIVE

## 6.1 INTRODUCTION

The preceding chapter was an analysis of data only from a qualitative point of view. This chapter focuses on both quantitative and qualitative data analysis of the research based on the research questions on curriculum, programming syllabus, and personal factors that have an impact on learning to program. In Chapter 4, the quantitative data analysis was performed based on the research questions in Chapter 1, the outcome of the review of literature in Chapter 2, and the research methodology set out in Chapter 3. The quantitative analysis of the data provides statistical results needed to understand the various factors affecting learning to program based on the data collected from the students. In this chapter, the survey data will be analysed based on mixed methods in Chapters 5 and 6 through the assessment of patterns in corresponding questions to negate or confirm the quantitative and qualitative results with the aim of ultimately answering the main research question, "What are the factors that contribute to learning hindrances experienced by programming students at Unisa?"

## 6.2 CURRICULUM

The data is analysed based on both quantitative and qualitative survey questions relating to curriculum factors that either compromise or help students succeed in programming as well as what students feel require improvement. The analysis of the data uses mixed methods through the cross-tabulation of data (Table 6.1) from quantitative Question 22 and qualitative Questions 24 and 26 to provide the common most and least influential factors in learning to program. The analysis assesses the areas of programming the students believed have helped them succeed and compare the results with the areas of programming the students feel have compromised their success in learning to program. The two data sets are

then compared to what students have highlighted as areas requiring improvement by the university.

The outcome of the data analysis indicates that students regarded face-to-face tutorials as the most helpful syllabus factor. The importance of the factor is validated by the highest score when students were asked how the areas of the module requiring improvement and the lowest score in areas that have compromised students' ability to succeed in learning to program.

Practical exercises is the second-highest factor, followed by the learning units. Discussion forums remained the least useful tool, and students did not see this factor as an area that requires improvement.

**Table 6.1: Questions 22, 24 and 26 – Analysis of responses**

| | **Q 22 - Which areas have helped you the most in learning and understanding to program?** | | **Q 24 - Which programming parts of this module do you feel have mostly compromised your ability to succeed?** | | **Q 26 - How can this module be improved?** | | | |
|---|---|---|---|---|---|---|---|---|
| **Answer Options** | Response Count | Response Percent | Response Count | Response Percent | Response Count | Response Percent | Response Count - Exluding Skipped and Invalid | Response Percent - Exluding Skipped and Invalid |
| Online Tutoring | 49 | 9.7% | 0 | 0.0% | 4 | 2.0% | 4 | 3.8% |
| Prescribed books | 136 | 27.0% | 1 | 0.5% | 4 | 2.0% | 4 | 3.8% |
| Face-to-face tutorials | 12 | 2.4% | 1 | 0.5% | 20 | 9.8% | 20 | 18.9% |
| Teaching assistants | 19 | 3.8% | 0 | 0.0% | 5 | 2.4% | 5 | 4.7% |
| Lecturers | 24 | 4.8% | 0 | 0.0% | 6 | 2.9% | 6 | 5.7% |
| Practical exercises | 92 | 18.3% | 77 | 37.6% | 8 | 3.9% | 8 | 7.5% |
| Learning Units | 86 | 17.1% | 3 | 1.5% | 6 | 2.9% | 6 | 5.7% |
| Discussion Forum | 52 | 10.3% | 0 | 0.0% | 1 | 0.5% | 1 | 0.9% |
| Internet | 26 | 5.2% | 0 | 0% | 1 | 0% | 1 | 0% |
| | | | | | | | | |
| **Curriculum** | **0** | **0.0%** | **7** | **3.4%** | **19** | **9.3%** | **19** | **17.9%** |
| ***Structure** | **0** | **0.0%** | **0** | **0.0%** | **15** | **7.3%** | **15** | **14.2%** |
| ***Administration** | **0** | **0.0%** | **0** | **0.0%** | **4** | **2.0%** | **4** | **3.8%** |
| **Module** | **0** | **0.0%** | **0** | **0.0%** | **8** | **3.9%** | **8** | **7.5%** |
| ***Length** | **0** | **0.0%** | **0** | **0.0%** | **3** | **1.5%** | **3** | **2.8%** |
| ***Assignments** | **0** | **0.0%** | **7** | **3.4%** | **5** | **2.4%** | **5** | **4.7%** |
| **Face-to-face Classes** | **0** | **0.0%** | **0** | **0.0%** | **24** | **11.7%** | **24** | **22.6%** |
| Invalid | 7 | 1.4% | 109 | 53.2% | 70 | 34.1% | 0 | 0.0% |
| Skipped | 0 | 0.0% | 0 | 0.0% | 29 | 14.1% | 0 | 0.0% |
| **Total** | **503** | **100.0%** | **205** | **100.0%** | **205** | **100.0%** | **106** | **100.0%** |

Table 6.2, similar to Table 5.3 from qualitative data analysis, supports the results of mixed methods analysis in Table 6.1. It indicates that face-to-face classes, university-related administration issues, and lack of tutorials are generally due to limited online tutorials.

**Table 6.2: Curriculum factors**

| Curriculum Improvements for the Module | Count | % | Related to |
|---|---|---|---|
| Classes (Face to face, extra, and more time-flexible) | 25 | 23% | Institution |
| General administration (delays, support, and communication) | 21 | 20% | Institution |
| Tutorials (extra and more comprehensive) | 14 | 13% | Module |
| Tutors (availability and more time) | 7 | 7% | Institution |
| Too many assignments | 7 | 7% | Module |
| Programming (write, run, test, and apply programs) | 7 | 7% | Module |
| Practical exercises | 7 | 7% | Module |
| Reduce workload | 6 | 6% | Module |
| Material content | 6 | 6% | Module |
| Support from lecturers (face to face and more frequent) | 4 | 4% | Institution |
| Delay in study material | 3 | 3% | Institution |
| **Total** | **107** | **100%** | |

## 6.3 SYLLABUS

The mixed methods data analysis is based on the comparison results from qualitative Question 24 in Table 6.3b relating to the question, "Which programming parts of this module do you feel have mostly compromised your ability to succeed?" and quantitative Questions 16, 17, 18 and 19 in Table 6.3a based on the following questions:

- Question 16: "It is easy for me to design a program to solve a certain task".
- Question 17: "Learning the programming language syntax is confusing".
- Question 18: "I have difficulties understanding errors from my own programs".
- Question 19: "I can write, compile, run, and debug my own programs".

The analysis of both quantitative and qualitative questions on syllabus-related issues indicated a similar outcome for top most factors with a negative impact on the performance of the students in learning to program. The quantitative data results in Table 6.3a showed most students disagreed that they found it easy working with programming design concepts (Question 16) and can run and manage programs as expected (Question 19). Most students agree that they have a challenge learning programming syntax (Question 17) and understanding program errors (Question 18).

The same factors from the quantitative data analysis remained a challenge and appeared in the qualitative results (Table 6.3b) as the top driving factors that influence learning to program. The factors from the qualitative data analysis are as follows in the order of impact:

- Inability to understand programming concepts
- Writing programming codes or programs
- Getting programs to run as desired
- Managing runtime errors

**Table 6.3a: Quantitative Questions 16, 17, 18 and 19 on problematic programming areas**

| | **Strongly Agree** | **Agree** | **Neutral** | **Disagree** | **Strongly Disagree** |
|---|---|---|---|---|---|
| **Question 16** | 23.2% | | 46.8% | 30.0% | |
| **Question 17** | 31.2% | | 39.2% | 29.6% | |
| **Question 18** | 39.3% | | 33.8% | 26.9% | |
| **Question 19** | 26.7% | | 41.1% | 32.2% | |

**Table 6.3b: Question 24 – Problematic coding areas in programming**

| Factor | Response Count | Response Percentage | Response Count (Excluding Invalid and Skipped) | Response Percentage (Excluding Invalid and Skipped) |
|---|---|---|---|---|
| Concepts | 37 | 18% | 37 | 38% |
| Coding | 20 | 10% | 20 | 21% |
| Errors | 9 | 4% | 9 | 9% |
| Run & Manage | 14 | 7% | 14 | 14% |
| Module Content | 5 | 2% | 5 | 5% |
| Assessments | 10 | 5% | 10 | 10% |
| Curriculum | 2 | 1% | 2 | 2% |
| Invalid | 30 | 15% | 0 | 0% |
| Skipped | 78 | 38% | 0 | 0% |
| **Total** | **205** | **100%** | **97** | **100%** |

## 6.4 PERSONAL

The analysis of personal factors from both quantitative and qualitative data (Table 6.4) indicated that time remained the most determinant personal factor affecting the ability of the students to succeed in learning to program. When students were asked to provide the reasons that have led them to consider withdrawing from the module and what they would do differently if there were to repeat the module, students indicated that time management is the factor requiring the most consideration.

Personal and work-related factors (Table 6.4), such as issues linked to work, family, life, and unexpected personal commitments are indicated as the second most factors that affected learning. The choice of the course and support from teaching staff are factors seen by the students as having little influence on their performance in learning to program (Table 6.4).

**Table 6.4: Questions 23 and 25 – Analysis of responses**

| | Q23 - If so, any particular reason you have considered withdrawing from or did not finish this module? Please select all applicable answers | | Q25 - If you were to study this module again what would you do differently in order to do even better? | | | |
|---|---|---|---|---|---|---|
| **Answer Options** | Response Count | Response Percent | Response Count | Response Percent | Response Count - Exluding Skipped and Invalid | Response Percent - Exluding Skipped and Invalid |
| The module is too advanced beyond my capabilities | 15 | 6.0% | 13 | 6.3% | 13 | 10% |
| The module required more time than I could provide | 79 | 31.5% | 57 | 27.8% | 57 | 44% |
| I lacked motivation to study | 26 | 10.4% | 0 | 0.0% | 0 | 0% |
| The module content is confusing and difficult to follow | 23 | 9.2% | 2 | 1.0% | 2 | 2% |
| Tutors or lecturers offered inadequate support | 11 | 4.4% | 0 | 0.0% | 0 | 0% |
| I chose wrong course | 1 | 0.4% | 0 | 0.0% | 0 | 0% |
| My work or personal related commitments took time from the course | 61 | 24.3% | 37 | 18.0% | 37 | 29% |
| I had personal reason(s) that compromised my performance in this module | 35 | 13.9% | 20 | 9.8% | 20 | 16% |
| Skipped | 0 | 0 | 60 | 29.3% | 0 | 0% |
| Invalid | 0 | 0 | 16 | 7.8% | 0 | 0% |
| **Total** | **251** | **100.0%** | **205** | **100.0%** | **129** | **100%** |

Further assessment personal factors comparing the quantitative data on the reason that may lead to students withdrawing and qualitative data on the general feedback of students (Table 6.5) confirmed the results from Table 6.4. The results (Table 6.5) showed time, personal and work-related factors, as well as personal reasons as the leading factors that have a negative impact on the performance of the students in learning to program. The difficulties arising from the perceived complexity of the module also remains one of the determinants of the learning performance. The results are supported by a low score (22.9% in Table 4.7) when students were asked, "What is your level of understanding of the module content?" another score (23.2% in Table 4.11) for the question, "it is easy for me to design a program to solve a certain task", and the final score (26.7% in Table 4.9) for the question "I can write, compile, run, and debug my own programs". Also, results of Question 24 in Table 6.3b for the analysis of syllabus-related factors support the students' view of perceived complexity in learning the module.

**Table 6.5: Questions 23 and 27 – Analysis of responses**

| | Q23 - If so, any particular reason you have considered withdrawing from or did not finish this module? Please select all applicable answers | | Q27 - Please feel free to comment here on any aspect, positive or negative, of your learning experience on this module | | | |
|---|---|---|---|---|---|---|
| **Answer Options** | Response Count | Response Percent | Response Count | Response Percent | Response Count - Excluding Satisfied, Skipped and Invalid | Response Percent - Excluding Satisfied, Skipped and Invalid |
| The module is too advanced beyond my capabilities | 10 | 4.0% | 3 | 1.5% | 3 | 4.8% |
| The module required more time than I could provide | 79 | 31.5% | 11 | 5.4% | 11 | 17.7% |
| I lacked motivation to study | 26 | 10.4% | 5 | 2.4% | 5 | 8.1% |
| The module content is confusing and difficult to follow | 23 | 9.2% | 5 | 2.4% | 5 | 8.1% |
| Tutors or lecturers offered inadequate support | 11 | 4.4% | 7 | 3.4% | 7 | 11.3% |
| I chose wrong course | 1 | 0.4% | 0 | 0.0% | 0 | 0.0% |
| My work or personal related commitments took time from the course | 66 | 26.3% | 24 | 11.7% | 24 | 38.7% |
| I had personal reason(s) that compromised my performance in this module | 35 | 13.9% | 7 | 3.4% | 7 | 11.3% |
| Satisfied | 0 | 0.0% | 27 | 13.2% | 0 | 0.0% |
| Skipped | 0 | 0.0% | 105 | 51.2% | 0 | 0.0% |
| Invalid | 0 | 0.0% | 11 | 5.4% | 0 | 0.0% |
| **Total** | **251** | **100.0%** | **205** | **100.0%** | **62** | **100.0%** |

## 6.5 SUMMARY

The results from the analysis of both quantitative and qualitative data for corresponding questions indicated the most and least driving factors in determining the success of students in learning to program. Curriculum factors indicated that face-to-face classes, university-related administration issues, and lack of tutorials are generally due to limited online tutorials. The discussion forums are seen as an area that is neither helpful nor requiring improvement. Students appear conceited with the support they receive from lecturers and do not feel much improvement is required in this area. Data analysis from the factors relating to the syllabus showed that a general understanding of programming concepts and the ability to build, run, debug, and manage errors to produce functional programs remain key challenges for students. The outcome of the data analysis on personal factors contributing to challenges in learning to program showed that the majority of the contributing factors related to the students' time management, personal and work-related issues, personal reasons, and aptitude factors.

The next chapter is an interpretation of the results of this study.

# 7 INTERPRETATION OF THE RESULTS

## 7.1 INTRODUCTION

The foregoing chapter focused on quantitative and qualitative data analysis of the research based on the research questions on curriculum, programming syllabus, and personal factors that have an impact on learning to program. In this chapter, the results from the quantitative and qualitative data analysis in Chapter 4, Chapter 5, and Chapter 6 respectively are interpreted. As indicated in Section 1.6, the aim of the study was to gain insight into the (1) challenges associated with the curriculum that contribute to learning barriers; (2) hindrances specific to learning to program that students experience; (3) personal learning challenges faced by students; and (4) learning and teaching strategies that may form part of the curriculum of the university in an introductory programming module. The interpretation, therefore, involves the synthesising of the quantitative and qualitative results categorised into three areas: the curriculum, programming syllabus, and personal factors in Section 7.2, Section 7.3, and Section 7.4. The similarities in and differences between Section 7.5 and Section 7.6 are covered. Finally, the summary in Section 7.7 presents the summary of the hindrances to learning to program in an introductory programming module.

## 7.2 CURRICULUM

The interpretation of the factors relating to the curriculum is informed by the quantitative and qualitative data analysis results from the previous chapters where the results of the qualitative data analysis were used to support the quantitative data analysis results in areas where the same or similar question was asked in both forms. The curriculum focused on the institutional education, the curriculum programme, the teaching and learning strategy, and education materials in line with the review of the literature and data analysis.

### 7.2.1 INSTITUTIONAL EDUCATION

The quantitative results for the curriculum established that the majority of the students (81.5%) enrolled for Unisa ODL studies on a part-time basis – something that is expected granted that the university is based on an ODL model. The high number of part-time students is confirmed by the report by the South African Department of Higher Learning and Training, which states that University of South Africa accounts for 87.9% of the distance learning education students in the field of science, engineering, and technology. The knowledge in the number of students enrolled for part-time study versus full-time study helps with the interpretation of the research results and conclusion thereof. The differentiation of part-time versus full-time registered students could help provide information on various challenges between the two groups regarding key factors such as time management, workload, and other personal commitments.

### 7.2.2 CURRICULUM PROGRAMME

The interpretation of the curriculum programme is based on the quantitative results. It is also based on the evaluation of the qualitative results from the suggestions on how the module can be improved.

The interpretation of the qualitative results relating to the curriculum-related question that asked about how the module could be improved revealed two areas requiring improvements. The first area comprised the curriculum changes relating to the institution, while the second represented the curriculum changes linked to the programming module. The issue with the curriculum was also evident in the results from a different qualitative question on the comments given regarding the learning experiences in the module. The indications are that the institutional curriculum-related challenges are the most common to deal with, followed by challenges relating to the programming module syllabus.

#### 7.2.2.1 CURRICULUM CHALLENGES RELATING TO THE INSTITUTION

The interpretation of the institutional curriculum results from the question on suggestions for the module improvement showed that students felt regular contact classes is the area that requires the most attention for improvement. The second area that students felt requires attention after regular contact classes is academic administration-related issues linked to the institution.

First, on contact classes, the results indicate that most of the students feel that extra contact classes would be helpful. The need for classes was also highlighted in the qualitative question asked to the students on their experiences of the module. The students felt that extra contact classes or contact classes with more flexible schedules would help them improve their performance in learning to program. The review of literature produced some agreement with the scarcity of contact classes by highlighting that in ODL, the teachers have limited or no ability to interact with students (Butler and Morgan, 2007; Sheard and Carbone, 2007). Although distance learning has benefits, as indicated in literature, often classes may be required. The literature further highlights that learning needs to take place with students in mind and that proper support needs to be afforded to the students and often one-on-one engagement with the students may be necessary. It is therefore apparent that an assessment of extra contact classes that are accessible to the students is an important consideration for helping students succeed in learning to program.

Second, the results from the "general administration" responses that represent the second part of the institutional curriculum changes showed several improvements that varied in nature. The detailed analysis of the responses did not establish any significant pattern. The responses did, however, provide insight into various areas such as communication preferences, year-based module, and booking of institution resources that may form part of future studies.

#### 7.2.2.2 CURRICULUM CHALLENGES RELATING TO PROGRAMMING SYLLABUS

The second aspect of the curriculum-related changes relates to the programming syllabus. The interpretation of the results showed that "tutorial" was the key factor requiring improvement. The main contributing factors were inadequate tutorial videos and classes.

It is observed from the quantitative results that a large number of students (53.5%) have never attended tutor sessions or contacted tutors. In addition, 35% have attended tutor sessions or contacted the tutors on an ad hoc basis. The high number of students not attending the tutor sessions or contacting the tutors or doing so on ad hoc basis can be largely linked to time or scheduling. This scheduling indicates that the tutorial sessions on offer need to be easily accessible – either through extra tutorial classes or flexible scheduling. Further analysis indicates that most of the students responded early in the semester. It was therefore expected that they would not have really formed study groups. It is observed that most of the 84% were active in the online forum, or virtual classroom, which would mean that they were in contact with other fellow students.

The number of tutorial videos came up regularly in the qualitative analysis of the results and showed up students as being inadequate for the module. The quantitative results from the question, "Which areas have helped you the most in learning and understanding to program?" showed online tutorials as one of the most useful tools for learning. The online tutorials are after books, learning units, practical exercises, and discussion forums. Literature shows that lecture videos are a critical component of the learning materials in learning to program (Matthíasdóttir and Geirsson, 2011). However, the qualitative results do not indicate online tutorials to be one of the most useful tools. The quantitative results on the online tutorials still present an important view that requires a look into how the strategy on the use of tutorial videos can be improved for learning in programming.

### 7.2.3 TEACHING AND LEARNING STRATEGY

The interpretation of teaching strategy was based on the evaluation of the quantitative results involving the question on how students felt about the support they received from the teachers and the way students approached learning. The qualitative results were based on the results from the perceptions on how the module could be improved and the quantitative results on students' level of participation in learning the module.

The review of literature on the importance of support by the educators to ensure that students succeed in learning highlights that factors such as a great focus on the students, face-to-face discussions, better use of technology, and effective communication to ensure continuous enrichment of information are key drivers (Threlkeld and Brzoska, 1994; Sanderson, Phua and Herda, 2000; Pinar, 2012). The qualitative results regarding the support from educators, lecturers, and tutors revealed that students required more support from the educators. This aspect is largely the result of the high number of students who indicated the limited availability or accessibility of tutors because of factors such as limited face-to-face discussions and communication – the message either not reaching the students or the message not being conveyed as intended.

The online discussion forums prove to be effective in areas of teaching and learning strategies, with quantitative results showing that 70% of the students participated in online discussions or were aware of the activities taking place in the forums. The results of the quantitative question on the most helpful areas in learning and understanding to program brought out that the discussion forums were ranked fourth out of nine selection options in the question. Since the participation in the online forums is a requirement for the module formative assessment, the percentage of the number of students participating in or aware of the forum is expected to be 100%. The fact that some students do not participate in the online forums may be a contributing factor towards the failure rate for the introductory module at Unisa. It is, however, inconclusive to know factually if this is the case, since the other 30% of the students could have indicated that they did not participate even when they had done so after the survey. No adequate evidence appears in either quantitative or qualitative results to suggest that the students had indeed not participated or were not aware.

Literature also points out the need for the students to be part of the programming community (Booth, 1992). In this case, the programming community would largely entail students being part of the discussion forums, but they could also meet in person to form study groups. The benefit of contact meeting is students can interact effectively and with an immediate response time. The use of the online discussion forums could have provided effective and efficient ways to receive assistance on questions posted to the online forums or e-tutor facilities. The quantitative results on whether the students contacted fellow students showed that 62.4% had never done so, while 16% did so only when they were in need of help.

The qualitative data results indicated that time and personal commitments prevented students from having physical meetings. This question was formulated to relate it to the one on discussion forums. Given that most students participated in the online discussion forums, the conclusion was that the virtual classroom along with the benefits already described and the accessibility of the forums resulted in most students finding it less necessary to meet in person. It is possible that the high level of participation in the online forum might have been mainly for compliance reasons, with regard to the number of mandatory hours required for participation in the forums.

The review of literature highlights that learning to program is hard and requires much time (Winslow, 1996; Kinnunen and Malmi, 2006; Kinnunen and Malmi, 2008; Mhashi and Alakeel, 2013). The university requires a minimum of eight hours per week to be allocated to the module for the purpose of practising (Appendix E). These results, however, could be compared to the quantitative results from the question, “Do you feel you have worked consistently throughout the term on this module?” to which many (44%) agreed. Only 18.7% of the students responded that they felt they did not work hard, while the rest of them were neutral. The results could have been different had the question been asked right at the end of the semester. Nonetheless, the results do establish that students did not equate consistent hard work to the number of hours dedicated to the studies, which means that the students had the perception that they were working hard. The number of hours allocated by the students to studies was far less than the required hours for the module, which may be a hindrance in learning to program.

### 7.2.4 EDUCATIONAL MATERIALS

The literature review on educational materials (Mock, 2003; Sheard and Carbone, 2007; Kinnunen and Malmi 2008; Vihavainen, Paksula and Luukkainen, 2011) showed the importance of learning materials as part of the curriculum and also the important role the materials played in both learning and teaching if developed in line with the curriculum needs of the students.

The literature further indicates that study materials are rated second after "assistance from others" as the strategy to resolve difficult issues when learning to program (Kinnunen and Malmi, 2006). However, the results from this study refuted some findings from literature and indicated that "assistance from others" was the least effective strategy for resolving learning-related issues. This study showed strategies such as face to face scoring the lowest at 6.6%, teaching assistants at 10.5%, and lecturers at 13% to be the least helpful relative to study materials such as prescribed books. The results based on responses to the quantitative question, "Which areas have helped you the most in learning and understanding to program?" showed the prescribed books achieving the highest score (74%) followed by the learning units (47.5%). The high percentages indicate that there should be a greater focus on the development strategy for the prescribed books and learning units.

Since the interpretation of the learning materials showed that prescribed books and learning units were important, the study further explored if any areas required improvements. Further investigation led to the structure and content of the learning materials because literature showed that the learning materials could only be effective if the structure and content are appropriate to ensure teaching instructions are disseminated effectively for learning. The quantitative results show the majority (78.3%) of the students agreed that the structure and content of the module was easy to follow and understand. For the study to gain a high level of reliability in the assessment of the module structure and content, deeper analysis was necessary. For this reason, further quantitative questions were asked and analysed. First, responses to the question, "Does the teaching material have enough and correct content for this level?" highlighted that the majority (59.2%) of the students agreed. Second, to the question, "Do you feel every chapter, assignment, or tutorial prepares you well enough for the next task?" most students (86%) indicated "Yes". It would therefore generally be

expected that it should be easy for students to understand the actual content of the study material if the majority of the students agreed the structure and content of the material was comprehensively structured, easy to follow, and also that the module syllabus was appropriate for the programming module.

However, further in-depth analysis of both quantitative and qualitative results provided a different view by highlighting that quantitatively very few students (22.9%) agreed that their level of understanding of the programming module content was high. The results of the qualitative analysis on the question of how students felt the module could be improved showed "the perceived change in material content" as one of the valuable suggestions for improvement. The results were therefore indicative that the curriculum setup of the module structure was adequate based on the outcome of the quantitative results on the study materials.

The challenge for students, however, was the students' comprehension when working with the actual content. This challenge can be attributed to various factors covered in the next sections on personal factors and programming syllabus. The factors showed issues about personal commitment and effort, and general comprehension of the programming concepts. Other factors involved personal reasons that affected the amount of attention given to the programming syllabus and inadequate focus on the content of the study materials.

## 7.3 PERSONAL FACTORS

The interpretation of the programming syllabus was based on both the quantitative and qualitative data analysis results from the previous chapters. The interpretation of the results consisted of the synthesis of prior learning, specific programming-related challenges, and experiences linked to learning to program. The synthesising of the experiences and challenges faced by the students related to aptitude- and cognitive-related factors, as well as personal reasons and commitments. The three areas for interpretation are in line with the review of literature and the chapters on data analysis.

### 7.3.1 PRIOR LEARNING

The quantitative results showed that a significantly high number of students (72.1%) had not been exposed to programming before, while the remaining 27.9% showed that the students had some level of programming experience. The review of the studies covered as part of prior learning in this study showed that a direct link exists between prior exposure to programming and the performance in learning to program. The comparison results (the refined results) between the quantitative question, "Have you taken programming classes before?" (Appendix D) from quantitative questions: "It is easy for me to design a program to solve a certain task"; "learning the programming language syntax is confusing"; and "I have difficulties understanding errors from my own programs" showed that students with prior experience or exposure to programming did well in program design, syntax, error management, and coding compared to those without prior exposure. This finding means efforts made in getting the students to experience programming would help students to improve their performance in learning to program.

### 7.3.2 APTITUDE AND COGNITIVE ABILITIES/FACTORS

Many studies reviewed in the literature linked aptitude and cognitive factors to learning to program. Aptitude involves the natural ability to perform certain things, whereas cognition includes judgement, reasoning, decision-making, and problem-solving. It is for these reasons that the study evaluated if the two resulted in hindrances to learning to program.

The quantitative results based on the question, "Any particular reason you have considered withdrawing from or did not finish this module?" indicated the module was seen as "confusing and difficult to follow". The difficulties in studying the module were also confirmed by the programming syllabus section of this chapter where the majority of students lacked a basic foundation in programming and had difficulties working with programs, including coding and execution. However, it is still not apparent that aptitude has an impact on the outcome of the programming module, since the reasons provided could be many factors – such as teaching or learning styles – as opposed to natural ability or capability to learn to program.

The quantitative results also assert that motivation was one of the main precursors to failure or dropout in programming module. The findings on motivation being one of the hindrances was informed by the results that showed that students indicated that after time, commitments, personal reasons, and motivation were keys determinant in this regard.

The interpretation of both quantitative and qualitative results did not indicate that there is a link between students' cognitive abilities and performance in learning to program.

### 7.3.3 PERSONAL REASONS AND COMMITMENTS

The quantitative results indicated that the majority of the students had access to the computer technology required to perform programming-related activities. The vast access to computer technology was informed by the fact that the results showed that over 92% of the students had access to personal computers. It was also noted that only 0.9% of the students had a low level of computer literacy, and 18% remained neutral on the question. These two factors – the level of computer literacy and access to the computer – do not appear to have a direct impact on the level of comprehension in programming, given that most students had difficulties with programming. The study was unable to establish the link between the level of computer literacy and the ability to program and also between access to a computer and the ability to program. It was, however, acknowledged that the module had computer access as a requirement for practising programming.

The quantitative results relating to the question on any reason that may have led the students to consider withdrawing or resulted in a student not completing the module revealed time as the factor that contributed the most. It was apparent that use of the word "more" was synonymous with the word "time" when students answered this question. The issue with time was confirmed by the qualitative results from the question, "If you were to study this module again, what would you do differently in order to do even better?" where time was provided as the factor that contributed the most. The results appear to show that time was a hindrance in the module. To determine conclusively the reason(s) behind time being a hindrance, the study performed deeper interpretation of the results from quantitative data analysis from the same question to establish the sources. The possible sources were the workload, students not making or dedicating enough time to the studies, or curriculum setup.

The results revealed that self-improvement was the factor that contributed the second most. The details of the reasons given for self-improvement showed that 67% of the students mentioned time management as something they would like to improve. There was no evidence from the curriculum programme or workload that suggested time was an issue. It could therefore be concluded that the reason why time was a factor that contributed to learning to program was lack of better time management by the students.

The second most contributing factor for students to consider withdrawing or that leads to students not completing the module was "work and personal commitments" that demanded time from the module. This result further provides more evidence regarding the issue on the "time" factors discussed. The purpose of this selection option, though, was to uncover if commitments apart from those related to the studies are part of personal factors resulting in learning hindrances in an introductory programming module. The results have proven that work and personal commitments have an impact on learning to program.

Literature indicates that personal reasons affect students' ability to program (Xenos, Pierrakeas and Pintelas, 2002; Simon, et al., 2006; Kinnunen and Malmi, 2008; Sarpong, Arthur and Amoako, 2013). The results from this study showed that personal factors have an influence on the outcome of the programming modules the students have undertaken. The study did not seek to uncover the specific reasons why personal reasons were a hindrance. The results indicated "personal reasons" to be among the main personal factors that led to hindrances in learning to program.

## 7.4 PROGRAMMING SYLLABUS

The interpretation of the programming syllabus involves both quantitative and qualitative results of data analysis from the previous chapters. The interpretation of the results consists of the synthesising of teaching methods, how the students go about learning programming and specific programming-related challenges linked to learning to program. The literature review and chapters on data analysis were used.

### 7.4.1 TEACHING TO PROGRAM

The quantitative results indicated that few students (19%) felt teachers were good at explaining aspects of the programming module compared to more than double (41.4%) of the students who did not feel so. However, 39% of the students were neutral, which might lead to an increase in the number of students who felt content with the way teachers explained the aspects of the module. The neutrality might have been for various reasons, including the early survey administration at the start of the semester for the second run, or the ODL-based programme. An ODL programme entails some of the students who did not have an opportunity to adequately interact with the teachers.

### 7.4.2 LEARNING TO PROGRAM

The review of literature suggests that programming concepts involve general orientation, the integrated development environment (IDE), and pragmatics, including the development, testing, debugging, and practical application of computer programs. It was for these reasons that the study looked into the specific programming factors contributing to the challenge of learning to program and in practical application in the next section.

The quantitative results from general orientation largely depend on the teaching methods as described in the "teaching to program" section in the chapter and the learning approaches taken by the students. The interpretation of the statistical results involving the learning approach revealed that few students (27%) spent enough time doing actual programming exercises. The "learning to program" section of the literature review indicates that many students find practical exercises an effective strategy to apply the knowledge learned when learning to program. The lack of adequate time dedicated to the practical exercises by the students would mean that the students did not meticulously practise the programming and failed to learn to program. This statement was supported by the findings whose interpretation showed that students did not spend the eight hours prescribed for the module (Appendix E) and the qualitative results that showed time to be the most powerful hindrance to learning for the module. It might also be important to know if lack of understanding of programming concepts by the students was linked to the inadequate time afforded to the practical exercises.

### 7.4.3 PROGRAMMING IN PRACTICE

The quantitative results on how students learn, write, and manage programs showed a lack of understanding of basic programming concepts for practical application. Thirty per cent of the students were unable to write a program to solve a certain problem or perform a task compared to 23% of the students who could. The "personal factors" section in this chapter based on Appendix D show that the 23% largely consisted of students with prior exposure to programming. It can be contended that the number of those that could develop a practical program would generally be even lower. The low number of students with prior exposure to programming entering first-year programming courses is a factor highlighted in literature. The review of literature indicates that most students enter university-level studies without prior learning in programming. Because 46.7% of the students remained neutral in response to the answer, the results from a different quantitative question were analysed to check for consistency. The consistency check was to further validate that most students were unable to write their own practical programs compared to those who could.

The results from the question, "I can write, compile, run, and debug my own programs" showed a similar trend of a high level of difficulties in basic coding. The results revealed that the ability to understand concepts and write programs (91%) rather than program execution or/and debugging was the factor that contributed the most. The most prevalent challenge was the ability to build simple programs that were easy to manage then build up to the more complex programs and lack of a systematic approach to solving the problem.

## 7.5 CONVERGING ASPECTS OF THE STUDY

The section outlines aspects of the study that indicate the convergence in the results during the interpretation of the quantitative and qualitative results, as well as the literature reviewed. All aspects described in this section form part of the findings of the research and are discussed next.

### 7.5.1 CURRICULUM

The majority of the students (81.5%) were enrolled for Unisa ODL studies on a part-time basis, something that was expected granted that the university is based on the ODL model. Most students also had access to technology and had a high level of computer literacy.

The results showed that personal factors contributed the most to the hindrances to learning to program, followed by the curriculum, then the programming syllabus. The literature review examined many studies that indicated the impact of various personal factors such as commitments and personal reasons (Jenkins, 2002; Winn, 2002; Bennett, 2003; Simon, et al., 2006; Kinnunen and Malmi, 2008; Mhashi and Alakeel, 2013). Programming-related challenges, working with coding building blocks and practical applications, curriculum-related factors, including aspects of the general administration component, resulted in hindrances to learning to program.

Most students had no prior exposure to or experience in programming, which is something that has been shown in literature and in this study to have an impact on the ability to succeed in learning to program. Literature further highlights the importance of teaching problem-solving skills to beginners (Ali, Kohun and Coraopolis, 2005; Goosen, Mentz and Nieuwoudt, 2007) and in the case of this study, it would be Unisa Introduction to Interactive Programming programming students. Efforts made in affording students some exposure to programming, including problem-solving skills, would help students to improve their performance in learning to program. Literature indicates that prior exposure to actual program practice and lessons in computers are deemed important in programming, with suggestions that students with enough exposure to at least one programming language perform better during assessments.

Extra contact classes or change in the scheduling of contact classes are important aspects in supporting the students to succeed in learning to program. This study showed that most students felt that contact classes would be one of the most important improvements in the module.

Additional time-flexible tutorial classes were rated as essential to the success of many students in learning to program. The study found that the tutorials meant for supporting the chapters of the study materials were not adequate. The students also had difficulties in understanding the examples, learning on their own, and even after posting their work in the online discussion forum for the module, the indication is that it takes longer for them to receive feedback. The forums are also seen as tedious and time-consuming.

Extra tutorial videos for the module would provide one of the most useful tools in helping the students improve in learning to program. Literature brings out the benefits of videos by highlighting that the videos generally contain systematic instruction on how the student can go about performing a given activity or function. In addition, the students can replay the videos repeatedly (Matthíasdóttir and Geirsson, 2011).

Prescribed books and learning units were the most useful areas in helping the students to learn to program. It was observed that these study materials had enough and relevant content for the module. Structures were easy to follow and comprised the content that prepares students well for subsequent activities. However, the challenge was the students had difficulties with comprehension when working with the actual contents.

### 7.5.2 PROGRAMMING MODULE

There is lack of compliance with regard to the required time to succeed in learning the module. This lack of compliance might have an impact on the amount of work the students cover, particularly the practical exercises. The review of literature indicated the importance of practising programming to ensure that one can relate theory to practical application of programs in order to solve problems. This study has shown that most students have difficulties in writing functional programs.

The majority of the students have no foundation in programming concepts or have an inadequate foundation in programming concepts, resulting in difficulties in dealing with basic coding skills such as syntax, data and control structures, functions, and error management. The literature review showed many studies that have indicated that learning to program is hard, takes time, and that most students have difficulties with understanding

design technique, the programming language syntax, and program results during runtime. The findings of this study confirm all aspects described in the review of literature.

The majority of the students were unable to write practical programs to solve problems or perform certain tasks. This is primarily due to inability to apply what is learned in practice. These findings confirm the studies from the literature reviewed that indicate that the majority of students lack the ability to apply basic programming concepts, problem-solving, and practical programming skills. Literature further indicates that the difficulties include in-depth programming knowledge, challenges dealing with very complex programming functions, and the amount of time required to learn to program. In addition, when the students learn to program, various elements are taught independently from a conceptual viewpoint.

### 7.5.3 PERSONAL FACTORS

Time is the most significant hindrance in learning to program. Students are unable to give enough time required for the module, primarily due to work and personal commitments. Time management by the students is therefore crucial to ensure success in learning to program.

Personal reasons were also indicated to be a factor. They are shown to be one of the main reasons that lead to the students considering withdrawing or dropping out from their studies.

Lack of motivation is one of the prominent factors that lead to failure or dropout in Introduction to Interactive Programming module at Unisa. The literature reviewed showed that students lacking motivation will have difficulties learning to program (Jenkins, 2002; Kinnunen and Malmi, 2006; Yacob and Saman, 2012).

The interpretation of the quantitative and qualitative results showed that the only cognitive factor that had an impact on learning to program was motivation. The review of literature suggests cognitive factors that may impede learning in programming are motivation to program (Jenkins, 2002; Yacob and Saman, 2012) and learning style (Jenkins, 2002;

Thomas, et al., 2002; Seyal, et al., 2015). The study did not evaluate if students' learning styles have an impact on learning to program.

## 7.6 DIVERGING ASPECTS OF THE STUDY

This section outlines parts of the study that indicate the divergence in the quantitative and qualitative results during their interpretation as well as in the literature reviewed. All materials described in this section form part of the limitations of the study or suggestions for future studies.

### 7.6.1 CURRICULUM

The review of literature showed that based on the perception of the students, the recordings of lectures provide the most useful form of learning because the students felt that the recordings showed each step and can be replayed (Matthíasdóttir and Geirsson, 2011). The use of lecture recordings might improve the learning experience but may not necessarily improve the pass rate. This study showed that face-to-face tutorials are the least helpful after the prescribed books, practical exercises, learning units, and online discussion forums respectively. The quantitative results of the study also established that there are improvements necessary for the online tutorials. It would be valuable to evaluate if the improvements in online tutorials setup may result in the online tutorials being the learning area preferred the most by the students.

The results from the general administration, which is the second aspect of the institutional curriculum improvements required, showed several improvements that varied in nature. General administration represents aspects of the curriculum that form part of the university administrative processes, policies, and qualification programmes. The detailed analysis of the responses did not establish any significant pattern, yet collectively these aspects led to a high percentage. The responses do, however, provide insight into various areas such as communication preferences, the change from a semester- to a year-based module, and booking of institution resources. These are "things" that the students wanted the university to change.

### 7.6.2 PROGRAMMING SYLLABUS

The students indicated that they believed they had consistently worked hard, but the results from another question suggest that the majority did not dedicate eight hours per week to the module as required. However, the results established that students did not equate consistent hard work with the number of hours dedicated to the modules.

### 7.6.3 PERSONAL FACTORS

The quantitative results showed that over 92% of the students had access to personal computers, and 80% had a high level of computer literacy. No conclusive evidence exists in the study from qualitative results or other quantitative data sets to indicate that the aspects had a direct impact on or had no direct impact on the level of comprehension in learning to program.

The results on whether a link existed between aptitude and the outcome of the programming module are inconclusive as a result of lack of evidence from curriculum- and personal-related results that could give more information on how teachers disseminate information and how students go about learning. The literature review showed that the link between aptitude and ability to program is one of the elements that most studies are unable to establish conclusively.

The results from the analysis of the factors linked to students considering withdrawing and areas requiring improvements highlighted that students feel motivation is one of the key factors for considering withdrawing, while the same factor is not seen by students as an area of improvement. When the students were asked about the areas they feel require improvement, they did not indicate motivation to be a factor they need to improve. The conflicting results from the two questions in the study require assessment in future studies to determine if motivation is the key determinant in learning to program.

## 7.7 SUMMARY

The interpretation of the quantitative and qualitative results indicated that there were indeed curriculum factors, programming syllabus factors, and personal factors that resulted in hindrances to learning to program in an introductory programming module. The outcomes are ultimately structured into the diverging and converging areas for proper synthesis. The indications are that personal factors are the factors that contributed the most followed by the curriculum, then the programming syllabus.

The converging outcomes that represented the hindrances that the study could establish were requirements for additional contact classes, online tutorials, and tutors; improvement in teaching methods; and improved ways to ensure students understand specific elements of the module material content. They also included development of teaching and learning strategies to help students understand basic programming concepts; an improvement in helping the students learn to write practical programs; time management by the students; and finally, preventative measures that help students not to consider withdrawing or dropping out for personal reasons.

The interpretation also showed diverging aspects of the study. These aspects could not be conclusively established and, as a result, formed part of future studies and the limitations of this study. The aspects include the link between computer literacy and learning to program; access to a computer and learning to program; understanding the impact other curriculum activities or requirements not covered in the study have on learning to program; and if improvement in the setup for online tutorials, particularly extra videos, could result in the tutorials being the most preferred tool in line with literature. Other diverging factors were students feeling that they had worked consistently hard yet not spending the required number of weekly hours prescribed for the module and, finally, the need to establish the link between aptitude and ability to program.

The next chapter provides the conclusion of the study based on all chapters of the research.

# 8 CONCLUSION

The penultimate chapter provided an interpretation of the results of the study. This chapter summarises the findings of the study in Section 8.1, gives recommendations in Section 8.2 based on the findings from Section 8.1. In Section 8.3, the implications of the study are shared. Section 8.4 covers the limitations of the study, including the challenges associated with research formulation, data collection and analysis, as well as diverging outcomes. The suggestions for future studies and concluding remarks are also covered in Section 8.5. The concluding remarks of the study are presented in Section 8.6.

## 8.1 SUMMARY OF KEY FINDINGS

The interpretation of the findings is set out below.

**Finding #1**

The high number of students without prior exposure or experience in programming is a factor that has been shown in literature and this study. Lack of prior exposure to programming has an impact on the performance of students in learning to program. Any efforts made in getting the students to have experience in or exposure to programming prior to the enrolment to the module would help students improve their performance in learning to program.

**Finding #2**

Extra contact classes or changes in the scheduling of contact classes could support the students to succeed in learning to program.

**Finding #3**

Additional time and flexible tutorial classes might be essential to the success of many students in learning to program.

**Finding #4**

Extra tutorial videos for the module would provide the most useful tool in helping students learn to program.

**Finding #5**

The students have difficulties with comprehension when working with the actual contents of the module. The prescribed books and learning units are the most useful areas for helping students to learn to program. These study materials have adequate and relevant content for the module, as their structure is easy to follow and their content allows for a smooth transition from one chapter or activity to the next. However, the difficulty is that students were unable to interpret the written content into something logical and practical for use. This finding aligns to that of previous studies and highlights the link between the ability to read and explain written programming syllabus content and improved performance in learning to program.

**Finding #6**

The majority of students were unable to write practical programs to solve problems or perform certain tasks primarily due to their inability to apply what they had learned in practice.

**Finding #7**

Time is the most powerful hindrance in learning to program. Students were unable to allocate the minimum amount of time as prescribed for the module because of work commitments and personal commitments. A lack of compliance exists with regard to the

recommended time for succeeding in learning the module. The non-compliance may have an impact on the amount of work the students cover, particularly the practical exercises.

Time management is therefore crucial to ensure success in learning to program.

**Finding #8**

Personal reasons were shown to be one of the main reasons that led to students considering withdrawing from or dropping out of their studies.

**Finding #9**

Lack of motivation is one of the primary factors that lead to failure in Introduction to Interactive Programming module at Unisa. The results showed that students reported motivation as the key determinant after personal commitments and personal reasons in this regard.

## 8.2 RECOMMENDATIONS

The recommendations of the study are important preventative measures and improvements that may be essential to the improvement of student performance in learning to program. The following recommendations are made:

- Include general problem-solving together with programming concepts prior to teaching students the skills required to build basic programs in the first chapter of the module. The reason for this recommendation is that many students have no prior background in programming. Lack of background in programming will entail logic being taught first, rather than syntax and programming, to get the beginners to relate programming to the real world.
- Develop a strategy to offer additional contact classes or change the scheduling of contact classes to accommodate the majority of students who are unable to participate in the classes generally because of their personal commitments. It would be helpful to conduct polls to assess the best suitable times and nature of the demands for additional contact classes.

- Develop additional time-flexible tutorial classes that may be paired with the contact classes or conducted through the e-tutor functionality on the university's online portal to address the perceived limited number of tutorials students need.
- Include additional tutorial videos for the module to help students in need of additional tutorial videos by having extra digital content that they could use whenever and however often they deem appropriate and sufficient. It may be useful to provide references to selected academic websites with videos in the way the list of referral websites have been provided in the first chapter of the module. This would help reduce the amount of time students spend searching the Internet for the most appropriate videos.
- Ensure that the actual content of introductory programming prescribed books is easy to read, understand, and interpret to support the suitable content found to be already in place. Measures that allow the students to feel less like the study material content represents complex coded programming information would be useful. It may be useful to help students find the content to be something they can generally relate to in real life. Furthermore, it may be meaningful to understand additional reasons the students find the actual content difficult to understand.
- Encourage the students through regular and effective communication means such as SMSs, online alerts, and social chat services to remind students of the importance of the minimum number of hours required for the module per week.
- Develop study content that focuses on assisting students to develop the ability to understand programming concepts and code writing. It may be useful to introduce certain elements of program debugging and error handling fairly early in the module to assist the students to find it easy to understand or even manage programs from the start.
- Introduce early detection measures to validate if the students have difficulties applying what they have learned in practice. Introduction of early detection measures could help reduce the high number of students who were unable to apply what they have learned in practice. The current setup has good measures that allow the students to apply programming in real-life scenarios. The students were asked to develop a website for a business, then hand the project in during the exam period. It would be helpful to introduce continuous monitoring of the progress they have made through

laboratory assessment or evaluation of the work completed during select times before the end of the semester.

- Formulating any interventions by the institution to help students manage time better would be invaluable. Time was the most significant hindrance in learning to program compared to other factors, such as the need for contact classes, extra tutorials, and ability to write programs. The main reasons the amount of time required for the module was something most students were unable to give were work commitments and personal commitments. The institution could introduce a student support programme to help the students with time management. It may be meaningful for the institution to develop preventative strategies to assess the collective amount and type of modules each student takes within one academic tenure to assist the student with their schedule, time, and workload management.
- Implement the administrative mid-term assessment to evaluate personal challenges the students experience and take necessary measures when possible. It would also be useful to send out regular communiques informing the students on the advice and support available to them from the counsellors at the Directorate for Counselling, Career and Academic Development (DCCAD), educators, student support offices, and academic administration centre. These measures are important in addressing potential hindrances related to personal issues and commitments, motivation, and time management, which are the leading causes for reasons behind the students considering withdrawing from their studies.

## 8.3 IMPLICATIONS OF THE STUDY

The outcomes of the study indicate several gaps, improvement areas, and benefits that could improve or have an adverse impact on an introductory programming module at Unisa. This section outlines the implications the study had on learning to program.

The study established that the hindrances to learning to program are because of issues emanating from the curriculum, programming syllabus, and personal factors. The findings indicate that learning to program is not only subject to issues pertaining to programming itself, but it is rather linked to a multitude of factors.

It could also be discovered that the most factors hindering learning are personal factors. Any success in addressing the issues in this area could result in a vast improvement in the success rate among introductory programming students at Unisa.

The findings of the study present the educators with information that could be used to enhance teaching in an introductory programming. This would ensure that the students succeed in learning to program.

The use of the mixed methods research approach has highlighted the importance of using both empirical and descriptive data to gain an in-depth understanding of the issues that affect the students. Value from the statistical data stems from its ability to allow assessment, degree, and frequency of the issues.

The findings of the study may have profound value in their contribution of knowledge to future studies. This knowledge could be used as a foundation to understanding challenges linked to learning to program, particularly at Unisa and possibly in distance education.

## 8.4 LIMITATIONS OF THE STUDY

The limitations of the study are informed by the gaps identified through the research. Some limitations were found to be a pivotal part of the suggestions for future studies in this research.

The study consists of more quantitative questions than qualitative questions, resulting in issues with proper triangulation, since not every quantitative question had the corresponding qualitative questions that could be used to confirm or negate the results of the quantitative data analysis. The application of the mixed methods approach on the unequal number of survey questions created an imbalance in the assessment of some of the results. In some cases, the results that appeared valid had to be discarded because they could not be triangulated for credibility.

During the data analysis and interpretation, it became apparent that the mixed methods approach using only survey questions resulted in limitations in the triangulation. The study findings would have been improved through the use of mixed methods based on a combination of survey questions and other forms of questioning, such as interviews.

The administration of the research survey started early in the semester is believed to have had an impact on the responses provided or answers selected by the students. The possible bias in the answers or responses provided is based on the view that some quantitative questions had high numbers of students who skipped the questions or remained neutral in their responses.

It was also assumed that the students understood the questionnaire, were inclined to provide feedback, and could do so in writing. Additionally, the possibility exists that certain students might not have been truthful in their responses. Some students provided inadequate information in the qualitative responses, while others did not answer certain questions at all. This fact proved a limitation in the mixed methods approach, since in some cases, the qualitative data could not be merged with the quantitative data to avoid challenges relating to validity and reliability.

The study did not consider other factors associated with the curriculum that may directly or indirectly have had an impact on students' performance results. Some of the key factors that had a direct impact were the number of subjects the students had taken, along with the programming module and the nature of the subjects that formed part of their qualification. For example, the difficulty level for students studying few science subjects together with programming may differ from that of the students studying engineering along with programming.

Owing to limited quantitative data, the study was unable to establish the impact, if any, that computer literacy had on learning to program. As described in the literature review, there appears to be a link between computer literacy and the performance shown by students when learning to program.

The study does not consider the input from the educators. It is believed that the input from the educators would have provided a different perspective on the research and, to a certain

extent, could have introduced balance and reliability in some of the study findings. Some of the information that could have been derived from the educators are the adequacy and relevancy of course materials, curriculum-related institutional challenges perceived by the students, including common elements such as administration issues, inadequate support by the teaching staff, and contact classes

Further, the research could have assessed what makes the students that succeed in learning to program different from those that do not. This study did not consider this factor because of time constraints.

The responses to the survey were self-reported. In this way, the students could have misrepresented themselves.

The study did not fully evaluate if students' learning styles have an impact on learning to program. Factors assessing how students conceptualise, reflect, experiment, and build concrete experience when learning to program were not explicitly explored.

In addition, "time" might be covering up other weaknesses by the students. The time factor as reported by the students might be a cover up for lack of aptitude or having a schooling background that would make it virtually impossible to pass the programming module.

## 8.5 SUGGESTIONS FOR FUTURE STUDIES

This section outlines factors associated with the research that have been identified to be crucial in the advancement of the research in the field of programming. The aim of the study was to understand the hindrances to learning to program in an introductory programming module. It would be useful for the study to be extended to other modules or universities after the limitations have been eradicated.

It would be important to investigate the best ways to help students understand the basic concepts in programming and even better and easier ways to apply the knowledge in practice given the time constraints. The students' challenges in understanding basic programming concepts and the application thereof is informed by the high number of students who were

unable to write basic programs to solve problems or perform a certain task. The main reason for the inability to write basic programs was lack of knowledge in working with basic building blocks of the programming language.

An opportunity exists to investigate the factors that contribute to the performance of students who succeed in completing the module. The findings could be useful in understanding important factors that could be adopted by other students in learning to program.

There is a need to explore preventative measures to address personal issues that result in the students considering withdrawing from an introductory programming module. The primary factors are time, personal and work commitments, motivation, and personal reasons.

The study showed that students' lack of time was the most common hindrance to learning to program. It would be beneficial to research the best strategies for time management in programming courses. The study reveals that the students need more time to study, do not spend the required time for the module, spend less time on the practical exercises, and feel that they can do better with fewer modules.

## 8.6 CONCLUDING REMARKS

This study identified the hindrances to learning to program at Unisa. Factors relating to the curriculum and programming syllabus were identified. The findings and recommendations were presented to help improve learning in the introductory course on programming at Unisa. The study findings could form part of the body of knowledge within the research community, for teaching and learning in programming, as well as for future research.

# REFERENCES


Alaoutinen, S. and Smolander, K., 2010. Student self-assessment in a programming course using Bloom's revised taxonomy. In: ITiCSE '10, Proceedings of the 15th Annual Conference on Innovation and Technology in Computer Science Education. Ankara, Turkey, 28-30 June 2010. New York: ACM.

Ali, A.I., Kohun, F. and Coraopolis, P., 2005. Suggested topics for an IS introductory course in Java. Informing Science: International Journal of an Emerging Transdiscipline, 2, pp.237-245.

Ala-Mutka, K., 2004. Problems in learning and teaching programming-a literature study for developing visualizations in the Codewitz-Minerva Project. Codewitz Needs Analysis [online] Finland: Tampere University of Technology . Available at: < http://www.cs.tut.fi/~edge/literature_study.pdf> [Accessed 13 November 2016].

Babbie, E.R., 2013. The basics of social research. 6th ed. Wadsworth: Cengage Learning.

Baldwin, P.L., 2016. Active learning in higher education. Sage Journals, 17 (2), pp. 91-97.

Ben-Ari, M.M., 2015. In defense of programming. In: ITiCSE '15, Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education. Vilnius, Lithuania, 04-08 July 2015. New York: ACM.

Bennedsen, J. and Caspersen, M. E., 2005. An investigation of potential success factors for an introductory model-driven programming course. In: ICER '05, Proceedings of the First International Workshop on Computing Education Research. Seattle, USA, 01-02 October 2005. New York: ACM.

Bennedsen, J. and Caspersen, M.E., 2007. Failure rates in introductory programming. ACM SIGCSE Bulletin, 39 (2), pp.32-36.

Bennett, R., 2003. Determinants of undergraduate student dropout rates in a university business studies department. Journal of Further and Higher Education, 27 (2), pp.123-141.

Bergin, S. and Reilly, R., 2005. The influence of motivation and comfort-level on learning to program. In: PPIG '05, Proceedings of the 17th Workshop on Psychology of Programming Interest Group. Brighton, England, 29 June – 01 July 2005. UK: Psychology of Programming Interest group.

Bers, T. and Younger, D., 2014. The First-Year Experience in Community Colleges. New Directions for Institutional Research, 2013(160), pp.77-93.

Biggs, J., 1999. Teaching for quality outcomes at university: What the student does. Buckingham: Society for Research into Higher Education and Open University Press.

Bloom, B.S., Englehart, M., Furst, E.J., Hill, W.H. and Krathwohl, D.R., 1956. Taxonomy of educational objectives: Handbook I. Cognitive domain. New York: David McKay.

Booth, S.A., 1992. Learning to program: A phenomenographic perspective. Göteborg: Acta Universitatis Gothoburgensis.

Bosman, W. and Frost, D., 1996. Policing and distance education in South Africa: a process in transformation. In: M.Pagon, Policing in Central and Eastern Europe: Comparing Firsthand Knowledge with Experience from the West. Ljubljana, Slovenia, 14-16 November 1996. Slovenia: College of Police and Security Studies.

Boyle, R., Carter, J. and Clark, M., 2002. What makes them succeed? entry, progression and graduation in computer science. Journal of Further and Higher Education, 26 (1), pp.3-18.

Brewer, J. and Hunter, A., 1989. Multimethod research: a synthesis of styles. Thousand Oaks: Sage Publications.

Bruce, B. and Berg, M., 2001. Qualitative research methods for the social sciences. 4th ed. Boston: Allyn and Bacon.

Bryman, A., 2003. Quantity and quality in social research. London: Routledge.

Bryman, A., 2015. Social research methods. 5th ed. Oxford: Oxford university press.

Buck, D. and Stucki, D.J., 2001. JKarelRobot: a case study in supporting levels of cognitive development in the computer science curriculum. ACM SIGCSE Bulletin, 33 (1), pp.16-20.

Butler, M. and Morgan, M., 2007. Learning challenges faced by novice programming students studying high level and low feedback concepts. In: ASCILITE 2007, *Proceedings of ASCILITE Singapore 2007. ICT: Providing Choices for Learners and Learning.* Nanyang Technological University, Singapore, 02-05 December 2007. Australia: Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education.

Byrne, P. and Lyons, G., 2001. The effect of student attributes on success in programming. ACM SIGCSE Bulletin, 33 (3), pp.49-52.

Caspersen, M.E. and Kolling, M., 2009. STREAM: A first programming process. ACM Transactions on Computing Education (TOCE), 9 (1), p.4.

Chipere, N., 2017. A framework for developing sustainable e-learning programmes. Open Learning: The Journal of Open, Distance and e-Learning, 32(1), pp.36-55.

Chmura, G.A., 1998. What abilities are necessary for success in computer science. ACM SIGCSE Bulletin, 30 (4), pp.55-58.

Clark, V.P. and Creswell, J.W., 2011. Designing and conducting mixed methods research. 4th ed. Thousand Oaks: Sage Publications.

Code, 2016. Leaders and trend-setters all agree on one thing. [online] Available at: < https://code.org/quotes> [Accessed 12 December 2016].

Coffield, F., Moseley, D., Hall, E. and Ecclestone, K. eds., 2004. Learning styles and pedagogy in post 16 learning: a systematic and critical review. Wiltshire: The Learning and Skills Research Centre.

Coles, C., 2003. The development of a curriculum for spinal surgeons': observations following the second spine course of the Spinal Society of Europe Barcelona.16-19 September 2003.

Conley, D.T., 2014. Getting ready for college, careers, and the Common Core: What every educator needs to know. San Fransisco: Jossey-Bass.

Creswell, J.W., 2013. Research design: Qualitative, quantitative, and mixed methods approaches. 4th ed. Thousand Oaks: Sage Publications.

Creswell, J.W. and Clark, V.L.P., 2007. Designing and conducting mixed methods research. 2nd ed. Thousand Oaks: Sage Publications.

Creswell, J.W., Plano Clark, V.L., Gutmann, M.L. and Hanson, W.E., 2003. Advanced mixed methods research designs. In: A. Tashakkori, and C. Teddlie, eds. 2003. Handbook of mixed methods in social and behavioral research.Thousand Oaks: Sage Publications. pp.209-240.

Crotty, M., 1998. The foundations of social research: meaning and perspective in the research process. London: Sage Publications.

Dann, W., Cooper, S. and Pausch, R., 2006. Learning to program with Alice. Upper Saddle River: Prentice-Hall.

Dasuki, S. and Quaye, A., 2016. Undergraduate Students' Failure in Programming Courses in Institutions of Higher Education in Developing Countries: A Nigerian Perspective. The Electronic Journal of Information Systems in Developing Countries, 76(8), pp.1-18.

Davy, J. and Jenkins, T., eds. 1999. Research-led innovation in teaching and learning computer programming. In: ITiCSE '99, Proceedings of the 4th Annual SIGCSE/SIGCUE ITiCSE Conference on Innovation and Technology in Computer Science Education. Cracow, Poland, 27-30 June 1999. New York: ACM.

Denzin, N. K. and Lincoln, Y. S., 2005. Introduction: The discipline and practice of qualitative research. In: N. K. Denzin, and Y. S. Lincoln, eds. The SAGE handbook of qualitative research. 2005. Thousand Oaks: Sage Publications. pp.1-32.

Derus, S.R.M. and Ali, A.Z.M., 2012. Difficulties in learning programming: views of students. In: ICCIE 2012, 1st International Conference on Current Issues in Education. Yogyakarta, Indonesia, 15-16 September 2012. Malaysia: National University of Malaysia.

Dochartaigh, N.Ó, 2002. The internet research handbook: A practical guide for students and researchers in the social sciences. London: Sage Publications.

Du Boulay, B., 1986. Some difficulties of learning to program. Journal of Educational Computing Research, 2 (1), pp.57-73.

Dunn, R. and Dunn, K.J., 1992. Teaching elementary students through their individual learning styles: Practical approaches for grades 3-6. Boston: Allyn and Bacon.

Dunn, R. and Griggs, S. eds., 2003. Synthesis of the Dunn and Dunn learning styles model research: who, what, when, where and so what–the Dunn and Dunn learning styles model and its theoretical cornerstone. New York: St.John's University.

Eccles, J.S. and Wigfield, A., 2002. Motivational beliefs, values, and goals. Annual Review of Psychology, 53 (1), pp.109-132.

Eckerdal, A., 2006. Novice students' learning of object-oriented programming. Doctoral dissertation. Uppsala Universitet. Available at: < https://uu.diva-portal.org/smash/get/diva2:117235/FULLTEXT01.pdf> [Accessed 24 April 2017].

Egan, M.W., Sebastian, J. and Welch, M., 1991. Effective television teaching: perceptions of those who count most… distance learners. In: Reaching Our Potential: Rural Education in the 90's, Proceedings of the Rural Education Symposium. Nashville, USA, 17-20 March 1991. USA: Educational Research Information Center (ERIC).

Essa, S.F., 2016. Using an e-learning tool to overcome difficulties in learning object-oriented programming. Masters dissertation. University of South Africa. Available at: <http://uir.unisa.ac.za/bitstream/handle/10500/21218/dissertation_essa_sf.pdf?sequence=1&isAllowed=y> [Accessed 07 November 2016].

Flick, U., 2015. Introducing research methodology: a beginner's guide to doing a research project. 2nd ed. London: Sage Publications.

Floyd, J. and Fowler, J., 2009. Survey research methods: Applied Social Research Methods Series, No. 1. 4th ed. Thousand Oaks: Sage Publications.

Friedman, M.I. and Fisher, S.P., 1998. Handbook on effective instructional strategies: Evidence for decision-making. Columbia: The Institute for Evidence-Based Decision-Making in Education.

Furnham, A., 1995. The relationship of personality and intelligence to cognitive learning style and achievement. 1995. In: D. Saklofske, and M. Zeidner, eds. International handbook of personality and intelligence. New York: Plenum Press. pp.397-413.

Gelso, C.J., 1979. Research in counseling: methodological and professional issues. The Counseling Psychologist, 8 (3), pp.7-36.

Giangrande, E., 2007. CS1 programming language options. Journal of Computing Sciences in Colleges, 22 (3), pp.153-160.

Glossop, C., 2002. Student nurse attrition: use of an exit-interview procedure to determine students' leaving reasons. Nurse education today, 22 (5), pp.375-386.

Goddard, W. and Melville, S., 2004. Research methodology: An introduction. 2nd ed. New Jersey: Blackwell Publishing.

Gomes, A. and Mendes, A.J., 2007. Learning to program-difficulties and solutions. In: ICEE 2007, International Conference on Engineering Education–ICEE. Coimbra, Portugal, 3-7 September 2007. Maryland: International Network on Engineering Education and Research. pp.283-287.

Goosen, L. and Breedt, M., 2012. Educating in the changing environment of computer applications technology. In: SACLA '12, Proceedings of the Annual Conference of the Southern African Computing Lecturers' Association (SACLA). University of Free State, South Africa, 1-3 July 2012. Bloemfontein: Southern African Computing Lecturers' Association.

Goosen, G., Mentz, E. and Nieuwoudt, E., 2007. Choosing the "best" programming language?!. In: CSITEd 2007, Proceedings of the Computer Science and IT Education Conference (CSITEd). Mauritius, 16-18 November 2007. Santa Rosa: Informing Science Press.

Goosen, L. and Van Heerden, D., 2013. Project-based assessment influencing pass rates of an ICT module at an ODL institution. In: ICEL 2013, Proceedings of the 8th International Conference on e-Learning. Cape Town, South Africa, 27-28 June 2013. Reading: Academic Conferences Limited.

Govender, I. and Grayson, D.J., 2008. Pre-service and in-service teachers' experiences of learning to program in an object-oriented language. Computers and Education, 51 (2), pp.874-885.

Greene, J.C., 2007. Mixed methods in social inquiry. San Francisco: Jossey-Bass.

Greene, J.C., Caracelli, V.J. and Graham, W.F., 1989. Toward a conceptual framework for mixed-method evaluation designs. Educational evaluation and policy analysis, 11 (3), pp.255-274.

Gross, P. and Powers, K., 2005. Evaluating assessments of novice programming environments. In: ICER '05, Proceedings of the First International Workshop on Computing Education Research. Seattle, USA, 01-02 October 2005. New York: ACM.

Guzdial, M. and Soloway, E., 2002. Teaching the Nintendo generation to program. Communications of the ACM, 45 (4), pp.17-21.

Hagan, D. and Markham, S., 2000. Does it help to have some programming experience before beginning a computing degree program?. ACM SIGCSE Bulletin, 32(3), pp.25-28.

Hagan, D., Sheard, J. and Macdonald, I., 1997. Monitoring and evaluating a redesigned first-year programming course. ACM SIGCSE Bulletin, 29(3), pp.37-39.

Hawi, N., 2010. Causal attributions of success and failure made by undergraduate students in an introductory-level computer programming course. Computers and Education, 54 (4), pp.1127-1136.

Hayes, J. and Allinson, C.W., 1996. The implications of learning styles for training and development: a discussion of the matching hypothesis. British Journal of Management, 7 (1), pp.63-73.

Holmberg, B., 1985. Communication in distance study: status and trends of distance education. 2nd ed. Lund: Lector Publishing.

Honey, P. and Mumford, A., 1982. The manual of learning styles. Berkshire: Peter Honey.

Honey, P. and Mumford, A., 1992. *The manual of learning styles*. 3rd ed. Maidenhead: Peter Honey Publications.

Howard, G.S., 1983. Toward methodological pluralism. Journal of Counseling Psychology, 30(1), pp.19-21.

Hsi, S. and Soloway, E., 1998. Learner-centered design: addressing, finally, the unique needs of learners. In: CHI '98, Proceedings of the CHI 98 Conference Summary on Human Factors in Computing Systems. Los Angeles, USA, 18-23 April 1998. New York: ACM.

Hwang, W., Shadiev, R., Wang, C. and Huang, Z., 2012. A pilot study of cooperative programming learning behavior and its relationship with students' learning performance. Computers and Education, 58 (4), pp.1267-1281.

Ismail, M., AzilahNgah, N. and Umar, I., 2010. Instructional strategy in the teaching of computer programming: a need assessment analysis. *Turkish Online Journal of Educational Technology*, 9 (2), pp.125-131.

Isroff, K. and Del Soldato, T., 1998. Students' motivation in higher education contexts. In: S. Brown, S. Armstrong and G. Thomson, eds. 1998. Motivating students. London: Kogan Page. pp.73-82.

Jenkins, T., 2002. On the difficulty of learning to program. In: LTSN-ICS 2002, Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences. Loughborough, UK, 27-29 August 2002. Loughborough: LTSN-ICS.

Jick, T., 1983. Mixing quantitave and qualitative methods: Triangulation in action. Qualitative methodology. London: Sage Publications.

Johnson, R.B., Onwuegbuzie, A.J. and Turner, L.A., 2007. Toward a definition of mixed methods research. Journal of mixed methods research, 1 (2), pp.112-133.

Jonker, J. and Pennink, B., 2010. The essence of research methodology: A concise guide for master and PhD students in management science. Berlin: Springer Science and Business Media.

Joppe, M., 2000. The Research Process. [online] Available at: < https://www.uoguelph.ca/hftm/research-process> [Accessed 06 November 2016].

Karagiorgi, Y. and Symeou, L., 2005. Translating constructivism into instructional design: Potential and limitations. Educational Technology and Society, 8 (1), pp.17-27.

Keegan, D. (1990). Foundations of distance education. 2nd ed. London: Routledge

Keith, H., 1999. Higher education through open and distance learning: world review of distance education and open learning. London: Routledge.

Kember, D., 2001. Beliefs about knowledge and the process of teaching and learning as a factor in adjusting to study in higher education. Studies in Higher Education, 26 (2), pp.205-221.

Kessler, C.M. and Anderson, J.R., 1987. Learning flow of control: recursive and lterative procedures. ACM SIGCHI Bulletin, 19 (1), pp.73-74.

Kinnunen, P. 2009. Challenges of teaching and studying programming at a university of technology-viewpoints of students, teachers and the university. Doctoral Dissertation. Helsinki University of Technology. Available at: <http://lib.tkk.fi/Diss/2009/isbn9789522481955/isbn9789522481955.pdf> [Accessed 07 November 2016].

Kinnunen, P. and Malmi, L., 2006. Why students drop out cs1 course?. In: ICER '06, Proceedings of the 2nd International Workshop on Computing Education Research. Canterbury, UK, 09-10 September 2006. New York: ACM.

Kinnunen, P. and Malmi, L., 2008. CS minors in a cs1 course. In: ICER '08, Proceedings of the 4th International Workshop on Computing Education Research, Sydney, Australia, 06-07 September 2008. New York: ACM.

Kinnunen, P., McCartney, R., Murphy, L. and Thomas, L., 2007. Through the eyes of instructors: a phenomenographic investigation of student success. In: ICER '07, Proceedings of the 3rd International Workshop on Computing Education Research. Atlanta, USA, 15-16 September 2007. New York: ACM.

Kitahara, R., Westfall, F. and Mankelwicz, J., 2011. New, multi-faceted hybrid approaches to ensuring academic integrity. Journal of Academic and Business Ethics, 3, p.1.

Kori, K., Pedaste, M., Leijen, Ä and Tõnisson, E., 2016. The role of programming experience in ICT students' learning motivation and academic achievement. International Journal of Information and Education Technology, 6 (5), p.331.

Kori, K., Pedaste, M., Tõnisson, E., Palts, T., Altin, H., Rantsus, R., Sell, R., Murtazin, K. and Rüütmann, T., 2015. First-year dropout in ICT studies. In: IEEE 2015, Proceedings of the 2015 IEEE Global Engineering Education Conference (EDUCON). Tallinn, Estonia, 18-20 March 2015. New York: IEEE.

Koulouri, T., Lauria, S. and Macredie, R.D., 2015. Teaching introductory programming: a quantitative evaluation of different approaches. ACM Transactions on Computing Education(TOCE), 14 (4), p.26.

Lahtinen, E., Ala-Mutka, K. and Jarvinen, H.M., 2005. A study of the difficulties of novice programmers. In: ITiCSE '05, Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education. Caparica, Portugal, 27-29 June 2005. New York: ACM.

Leech, N.L. and Onwuegbuzie, A.J., 2009. A typology of mixed methods research designs. Quality and quantity, 43 (2), pp.265-275.

Lenning, O.T., Beal, P.E. and Sauer, K., 1980. Attrition and retention: Evidence for action and research. Boulder: National Center for Higher Education Management Systems.

Li, Y., Chen, P. and Tsai, S., 2008. A comparison of the learning styles among different nursing programs in Taiwan: implications for nursing education. Nurse education today, 28 (1), pp.70-76.

Lincoln, Y.S., Lynham, S.A. and Guba, E.G., 2011. Paradigmatic controversies, contradictions, and emerging confluences. In: N. K. Denzin and Y. S. Lincoln. 2011. The Sage handbook of qualitative research. Los Angeles: Sage Publications. pp.97-128.

Lister R., Adams, E. S., Fitzgerald, S., Fone, W., Hamer, J., Lindholm, M., McCartney, R., Moström, E., Sanders, K., Seppälä, O., Simon, B., and Thomas, L., 2004. A Multi-National Study of Reading and Tracing Skills in Novice Programmers. ACM SIGCSE Bulletin, 36 (4), pp. 119-150.

Lister, R., Berglund, A., Clear, T., Bergin, J., Garvin-Doxas, K., Hanks, B., Hitchner, L., Luxton-Reilly, A., Sanders, K. and Schulte, C., 2006. Research perspectives on the objects-early debate. ACM SIGCSE Bulletin, 38 (4), pp.146-165.

Ma, L., Ferguson, J.D., Roper, M., Ross, I. and Wood, M., 2008. Using cognitive conflict and visualisation to improve mental models held by novice programmers. ACM SIGCSE Bulletin, 40 (1), pp. 342-346.

Mack, N., Woodsong, C., MacQueen, K.M., Guest, G. and Namey, E., 2005. Qualitative research methods: a data collectors field guide. North Carolina: Family Health International.

Mahmoud, Q.H., Dobosiewicz, W. and Swayne, D., 2004. Redesigning introductory computer programming with HTML, JavaScript, and Java. ACM SIGCSE Bulletin, 36 (1), pp.120-124.

Malhotra, N.K., 2014. Basic marketing research. 4th ed. Harlow: Pearson Education.

Marshall, C. and Rossman, G.B., 2014. Designing qualitative research. 5th ed. Newbury Park: Sage publications.

Martins, N., 2012. Student satisfaction with textbook usage at distance education institutions versus media at more traditional/residential universities. [online] Available at: <http://uir.unisa.ac.za/bitstream/handle/10500/9491/martins_n%20_ODL_059_2012.pdf?sequence=1&isAllowed=y> [Accessed 07 November 2016].

Kinnunen, P. 2009. Challenges of teaching and studying programming at a university of technology-viewpoints of students, teachers and the university. Doctoral Dissertation.

Helsinki University of Technology. Available at: <http://lib.tkk.fi/Diss/2009/isbn9789522481955/isbn9789522481955.pdf> [Accessed 07 November 2016].

Martzoukou, K. and Kemp, V., 2016. Nurturing supportive and engaging induction environments for distance-learning students. In: HEAd´16, 2nd International Conference on Higher Education Advances. València, Spain, 21-23 June 2016. Amsterdam: Elsevier. pp.535-540.

Matthews, R., 2014. Learning objects to improve cognitive understanding in learning introductory programming. Doctoral Dissertation. Multimedia University. Available at: <http://search.proquest.com/openview/6301262e4ab17ec18a9922e0f2d56e37/1?pq-origsite=gscholar&cbl=18750&diss=y> [Accessed 24 April 2017].

Matthíasdóttir, Á and Geirsson, H.J., 2011. The novice problem in computer science. In: CompSysTech '11, Proceedings of the 12th International Conference on Computer Systems and Technologies. Vienna, Austria, 16-17 June 2011. New York: ACM.

Mayes, J.T. and Fowler, C.J., 1999. Learning technology and usability: a framework for understanding courseware. Interacting with Computers, 11 (5), pp.485-497.

Mazlack, L.J., 1980. Identifying potential to acquire programming skill. Communications of the ACM, 23 (1), pp.14-17.

McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y.B., Laxer, C., Thomas, L., Utting, I. and Wilusz, T., 2001. A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. ACM SIGCSE Bulletin, 33 (4), pp.125-180.

Mead, J., Gray, S., Hamer, J., James, R., Sorva, J., Clair, C.S. and Thomas, L., 2006. A cognitive approach to identifying measurable milestones for programming skill acquisition. ACM SIGCSE Bulletin, 38 (4), pp.182-194.

Mertens, D.M., 2014. Research and evaluation in education and psychology: Integrating diversity with quantitative, qualitative, and mixed methods. 4th ed. Thousand Oaks: Sage Publications.

Mhashi, M. and Alakeel, A., 2013. Difficulties facing students in learning computer programming skills at Tabuk University. In: EDU '13, Proceedings of the 12th WSEAS International Conference on Education and Educational Technology. Iwate, Japan, 23-25 April 2013. Greece: WSEAS Press.

Mock, K., 2003. The development of a CS0 course for distance delivery. Journal of Computing Sciences in Colleges, 19 (2), pp.30-38.

Morris, T., 2006. Social work research methods: four alternative paradigms. Thousand Oaks: Sage Publications.

Morrow, W., 2007. Learning to teach in South Africa. Cape Town: HSRC Press.

Neuman, W.L., 2006. Social Research methods: qualitative and quantitative approaches. 6th ed. Boston: Allyn and Bacon.

Newman, I. and Benz, C.R., 1998. Qualitative-quantitative research methodology: exploring the interactive continuum. Carbondale: Southern Illinois University Press.

Nkomo, M., 2000. The national qualifications framework and curriculum development. [pdf] Pretoria: South African Qualifications Authority (SAQAA) . Available at: <http://www.saqa.org.za/docs/pol/2000/curriculum_dev.pdf> [Accessed 13 November 2016].

Nunnally, J., 1978. Psychometric theory. 2nd ed. New York: McGraw-Hill.

Oates, B. J., 2006. Researching Information Systems and Computing. London: Sage Publications.

Oblinger, D., 2003. Boomers gen-xers millennials. EDUCAUSE review, 500 (4), pp.37-47.

Oliva, P.F. and Gordon, W.R., 2012. Developing the curriculum. 8th ed. Boston: Allyn and Bacon.

Oliver, D., Dobele, T., Greber, M. and Roberts, T., 2004. This course has a Bloom rating of 3.9. In: ACE '04, Proceedings of the Sixth Australasian Conference on Computing Education. Dunedin, New Zealand, 18-22 January 2004. Darlinghurst: Australian Computer Society.

Onwuegbuzie, A.J. and Johnson, R.B., 2006. The validity issue in mixed research. Research in the Schools, 13 (1), pp.48-63.

Onwuegbuzie, A.J. and Teddlie, C., 2003. A framework for analyzing data in mixed methods research. Handbook of mixed methods in social and behavioral research, 2, pp.397-430.

Onwuegbuzie, A.J. and Teddlie, C., 2003. A framework for analyzing data in mixed methods research. In: A. Tashakkori, and C. Teddlie, eds. Handbook of mixed methods in social and behavioral research. Thousand Oaks: Sage Publications. pp.397-430.

Ornstein, A. and Hunkins, F., 2016. Curriculum: Foundations, Principles, and Theory. 7th ed. Harlow: Pearson Education.

Pediaa, 2016. Difference between curriculum and syllabus. [online] Available at: <http://keydifferences.com/difference-between-syllabus-and-curriculum.html> [Accessed 06 November 2016].

Pedroni, M., Oriol.M. and Bertrand Meyer B., 2009. *What do beginning cs majors know?*. Swiss Federal Institute of Technology, Zurich: Technical Report 63. Available at: <http://se.ethz.ch/~meyer/publications/teaching/background.pdf > [Accessed 13 November 2016].

Phillips, D.C. and Burbules, N.C., 2000. Postpositivism and educational research. Maryland: Rowman and Littlefield Publishers.

Pinar, W.F., 2012. What is curriculum theory? 2nd ed. New York: Routledge.

Pinsonneault, A. and Kraemer, K., 1993. Survey research methodology in management information systems: an assessment. Journal of Management Information Systems, 10 (2), pp.75-105.

Pityana, N. B., 2009. Open distance learning in the developing world: trends, progress and challenges. In: M-2009, 23rd ICDE World Conference on Open and Distance Education *including the 2009 EADTU Annual Conference*: Flexible Education for All: Open-Global-Innovative. Maastricht, Netherlands, 7-10 June 2009. Oslo: International Council for Open and Distance Education.

Porter, R. and Calder, P., 2004. Patterns in learning to program: an experiment. In: ACE '04, Proceedings of the Sixth Australasian Conference on Computing Education. Dunedin, New Zealand, 18-22 January 2004. Darlinghurst: Australian Computer Society.

Powers, K., Gross, P., Cooper, S., McNally, M., Goldman, K.J., Proulx, V. and Carlisle, M., 2006. Tools for teaching introductory programming: what works?. ACM SIGCSE Bulletin, 38 (1), pp.560-561.

Qahmash, A., Joy, M. and Boddison, A., 2015. To what extent mathematics correlates with programming: statistical analysis. In: CSEIT 2015, 6th Annual International Conference on Computer Science Education: Innovation and Technology (CSEIT 2015). Singapore , 5-6 Oct 2015, Singapore:Global Science and Technology Forum.

Ramist, L., 1981. College student attrition and retention. New York: College Entrance Examination Board.

Rakoma, M. and Ranko-Ramalli,M., 2013. Fact or fantasy? Implementing Supplemental Instruction in an ODL context, the case of University of South Africa (UNISA). Available through: <http://uir.unisa.ac.za/bitstream/handle/10500/8739/ranko-ramalli_m_ODL_045_2012.pdf?sequence=1&isAllowed=y> [Accessed 06 November 2016].

Renumol, V., Janakiram, D. and Jayaprakash, S., 2010. Identification of cognitive processes of effective and ineffective students during computer programming. ACM Transactions on Computing Education (TOCE), 10 (3), p.10.

Reges, S., 2006. Back to basics in cs1 and cs2. ACM SIGCSE Bulletin, 38 (1), pp.293-297.

Robins, A., Rountree, J. and Rountree, N., 2003. Learning and teaching programming: a review and discussion. Computer science education, 13 (2), pp.137-172.

Roddan, M., 2002. The determinants of student failure and attrition in first year computing science. Final year undergraduate project. Glasgow University. Available at: <http://www.psy.gla.ac.uk/~steve/localed/roddanpsy.pdf> [Accessed 07 November 2016].

Rogalski, J. and Samurçay, R., 1990. Acquisition of programming knowledge and skills. In: J.M.Hoc, T.R.G. Green, R. Samurçay, and D.J. Gillmore, eds. 1990. Psychology of Programming. London: Academic Press. pp.157-174.

Rossett, A., 2002. Waking in the night and thinking about e-learning. In: Rossett, ed. 2002. The ASTD e-learning handbook. New York: McGraw-Hill. pp.3-18.

Rossman, G.B. and Wilson, B.L., 1985. Numbers and words combining quantitative and qualitative methods in a single large-scale evaluation study. Evaluation Review, 9 (5), pp.627-643.

Rountree, N., Rountree, J. and Robins, A., 2002. Predictors of success and failure in a cs1 course. ACM SIGCSE Bulletin, 34 (4), pp.121-124.

Rountree, N., Rountree, J. and Robins, A., 2004. Interacting factors that predict success and failure in a cs1 course. ACM SIGCSE Bulletin, 36 (4), pp.101-104.

Rovai, A.P. and Downey, J.R., 2010. Why some distance education programs fail while others succeed in a global environment. The Internet and Higher Education, 13(3), pp.141-147.

Rumelhart, D. and Norman, D., 1978. Accretion, tuning and restructuring: three modes of learning. In: J.W. Cotton, and R. Klatzky, eds. 1978. Semantic Factors in Cognition. New Jersey: Lawrance Erlbaum Associates. pp.37-53.

Sanderson, A., Phua, V.C. and Herda, D., 2000. *The American faculty poll*. Chicago: National Opinion Research Center.

Sarpong, K.A., Arthur, J.K. and Amoako, P.Y.O., 2013. Causes of Failure of Students in Computer Programming Courses: The Teacher-Learner Perspective. *International Journal of Computer Applications,* [e-journal] 77 (12). Available through: <http://research.ijcaonline.org/volume77/number12/pxc3891311.pdf >[Accessed 06 November 2016]

Saunders, M., Lewis, P., Thornhill, A. and Wilson, J., 2012. *Research methods for business students*. 6th ed. Harlow: Pearson.

Schoeman, M.A., 2015. *Enhancing comprehension in open distance learning computer programming education with visualization.* [online] Available at: < http://uir.unisa.ac.za/bitstream/handle/10500/20715/thesis_schoeman_ma.pdf?sequence=1&isAllowed=y> [Accessed 06 November 2016].

Schulte, C. and Bennedsen, J., 2006. What do teachers teach in introductory programming?. In: ICER '06, *Proceedings of the 2nd International Workshop on Computing Education Research.* Canterbury, UK, 09-10 September 2006. New York: ACM.

Seyal, A.H., Mey, Y.S., Matusin, M.H., Siau, N.H. and Rahman, A.A., 2015. Understanding Students Learning Style and Their Performance in Computer Programming Course: Evidence from Bruneian Technical Institution of Higher Learning. *International Journal of Computer Theory and Engineering*, *7*(3), p.241.

Sheard, J., and Carbone, A., 2007. ICT teaching and learning in a new educational paradigm: lecturers' perceptions versus students' experiences. In: Koli Caling 2007, *Proceedings of the Seventh Baltic Sea Conference on Computing Education Research.* Koli National Park, Finland, 15-18 November 2007. Darlinghurst: Australian Computer Society.

Sheard, J., Morgan, M., Butler, M., Falkner, K., Weerasinghe, A. and Cook, B., 2014. Experiences of first-year students in ICT courses: good teaching practices. *Monash University (Lead)*.

Simon, Fincher, S.A., Robins, A., Baker, B., Box, I., Cutts, Q., de Raadt, M., Haden, P., Hamer, J., Hamilton, M., Lister, R., Petre, M., Sutton, K., Tolhurst, D. & Tutty, J., 2006. Predictors of Success in a First Programming Course. *Proceedings of the Eighth Australasian Conference on Computing Education.* Hobart, Australia, 16-19 January 2006. Darlinghurst: Australian Computer Society.

Smithson, D., 2012. *A curriculum fit for purpose*? *a discussion paper on the unconscious drive toward hidden curriculums and the effects thereof managing consultant, the inform group*. [online] Available at: <http://d1014147.u189.designpunk.co.uk/wp-content/uploads/2011/12/Curriculum-Paper.pdf> [Accessed 07 November 2016].

Steele, A., 2010. First year programming: using competition for motivation. In: NACCQ 2010 , *Proceedings of the 1st Annual Conference of Computing and Information Technology Education and Research in New Zealand.* Dunedin, New Zealand , 6-9th July 2010. Hamilton: National Advisory Committee on Computing Qualifications.

Stephenson, C., Gal-Ezer, J., Haberman, B. and Verno, A., 2005. The new educational imperative: improving high school computer science education. Final report of the CSTA Curriculum Improvement Task Force. [online] New York: Computer Science Teachers Association (CSTA). Available at: <http://csta.acm.org/Communications/sub/DocsPresentationFiles/TCEAPres07.pdf > [ Accessed 06 November 2016].

Surbhi, 2015. *Difference between syllabus and curriculum*. [online] Available at: < http://keydifferences.com/difference-between-syllabus-and-curriculum.html> [ Accessed 06 November 2016].

Tashakkori, A. and Teddlie, C., 1998. *Mixed methodology: combining qualitative and quantitative approaches.* Thousand Oaks: Sage Publications.

Tashakkori, A. and Teddlie, C., 2003. *Handbook of mixed methods in social and behavioral research*. Thousand Oaks: Sage Publications.

Tashakkori, A. and Teddlie, C., 2010. *Sage handbook of mixed methods in social and behavioral research.* 2nd ed. Los Angeles: Sage Publications.

Thomas, L., Ratcliffe, M., Woodbury, J. and Jarman, E., 2002, February. Learning styles and performance in the introductory programming sequence. *ACM SIGCSE Bulletin,* 34(1), pp. 33-37.

Threlkeld, R. and Brzoska, K., 1994. Research in distance education. In: B. Willis, ed. 2000. *Distance Education: Strategies and Tools*. Englewood Cliffs: Educational Technology Publications.

Tinto, V., 1975. Dropout from higher education: a theoretical synthesis of recent research. *Review of educational research,* 45 (1), pp.89-125.

Tinto, V., 1987. Leaving college: *Rethinking the causes and cures of student attrition*. Chicago: University of Chicago Press.

Tinto, V., 1997. Classrooms as communities: exploring the educational character of student persistence. *Journal of Higher Education,* 68 (6), pp.599-623.

Trochim, W. and Donnelly, J.P., 2007. *The research methods knowledge base*. 2nd ed. Mason: Thomson Publishing.

Tukiainen, M., and Mönkkönen, E., 2002. Programming aptitude testing as a prediction of learning to program. In: PPIG '14, *Proceedings of the 14th Annual Workshop of the Psychology of Programming Interest Group.* Brunel University, London, 18-21 June 2002. UK: Psychology of Programming Interest Group.

UNESCO, 2017. *Curriculum*. [online] Available at: < http://www.unesco.org/new/en/education/themes/strengthening-education-systems/quality-framework/core-resources/curriculum/> [Accessed 24 April 2017].

Unisa, 2016. *Open distance learning*. [online] Available at: <http://www.unisa.ac.za/Default.asp?Cmd=ViewContent&ContentID=17765> [ Accessed 06 November 2016].

Vihavainen, A., Paksula, M. and Luukkainen, M., 2011. Extreme apprenticeship method in teaching programming for beginners. In: SIGCSE '11, *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education.* Dallas, USA, 09-12 March 2011. New York: ACM.

Ventura Jr, P.R., 2005. Identifying predictors of success for an objects-first cs1. *Computer Science Education,* 15 (3), pp.223-243.

Vogts, D., Calitz, A.P. and Greyling, J.H., 2010. The effects of professional and pedagogical program development environments on novice programmer perceptions. *South African Computer Journal,* 45, pp.53-58.

Watson, C. and Li, F.W., 2014. Failure rates in introductory programming revisited. In: ITICSE '14, *Proceedings of the 2014 Conference on Innovation and Technology in Computer Science Education.* Uppsala, Sweden, 23-25 June 2014. New York: ACM.

Whittington, N., 1987. Is instructional television educationally effective? a research review. *American Journal of Distance Education,* 1 (1), pp.47-57.

Wiedenbeck, S. and Labelle, D. and Kain, V.N.R., 2004. Factors affecting course outcomes in introductory programming. In: PPIG '04, *Proceedings of the 16th Annual Workshop*

*of the Psychology of Programming Interest Group*. Carlow, Ireland, 5-7 April 2004. UK: Psychology of Programming Interest Group.

Williamson, G.R. and Whittaker, A., 2014. *Succeeding in literature reviews and research project plans for nursing students.* 2nd ed. London: Learning Matters.

Wilson, B.C. and Shrock, S., 2001. Contributing to success in an introductory computer science course: a study of twelve factors. *ACM SIGCSE Bulletin,* 33 (1), pp.184-188.

Winn, S., 2002. Student motivation: A socio-economic perspective. *Studies in Higher Education*, 27 (4), pp.445-457.

Winslow, L.E., 1996. Programming pedagogy—a psychological overview. *ACM SIGCSE Bulletin,* 28 (3), pp.17-22.

Xenos, M., Pierrakeas, C. and Pintelas, P., 2002. A survey on student dropout rates and dropout causes concerning the students in the Course of Informatics of the Hellenic Open University. *Computers and Education,* 39 (4), pp.361-377.

Yacob, A. and Saman, M.Y.M., 2012. Assessing level of motivation in learning programming among engineering students. In ICIA 2012, *The International Conference on Informatics and Applications (ICIA2012)* Kuala Terengganu, Malaysia 3-5 June 2012. USA: The Society of Digital Information and Wireless Communication.

Yeh, Y., Chen, M., Hung, P. and Hwang, G., 2010. Optimal self-explanation prompt design in dynamic multi-representational learning environments. *Computers and Education,* 54 (4), pp.1089-1100.

York, C.M., Bollar, S. and Schoob, C., 1993. Causes of college retention: a systems perspective. In: APA 1993, *Convention of the American Psychological Association*. Toronto, Canada, 20-24 August 1993. Washington: American Psychological Association.

Zander, C., Thomas, L., Simon, B., Murphy, L., McCauley, R., Hanks, B. and Fitzgerald, S., 2009, July. Learning styles: novices decide. *ACM SIGCSE Bulletin*, 41(3), pp. 223-227.

# APPENDICES

## APPENDIX A: QUESTIONNAIRE FOR THE STUDENTS

The Hindrances to Learn to Program

**Thank you for your time in completing the survey.The survey will take approximately 5 to 8 minutes to complete. Please select the option that best represents your answer to each question, or fill in your response in the space provided.**

1. Are you studying Part-time or Full-time?

- ◯ Part-time
- ◯ Full-time

2. Is this the first time you are studying this module?

- ◯ Yes
- ◯ No

3. Do you have access to a computer when required?

- ◯ Yes
- ◯ No

4. Have you taken programming classes before?

- ◯ Yes
- ◯ No

5. Did you participate in or follow any discussion forums on myUnisa?

- ◯ Yes
- ◯ No

6. Did you spend at least 8 hours a week towards this module?

- ◯ Yes
- ◯ No

7. On average, how often did you study with fellow students?

- ◯ Weekdays(daily)
- ◯ Once a month
- ◯ Only towards exams
- ◯ Once a week
- ◯ Only when in need of assistance
- ◯ Never
- ◯ Other (please specify)

8. On average, how often did you attend tutor sessions or consult with a tutor?

- ◯ Weekdays(daily)
- ◯ Once a month
- ◯ Only towards exams
- ◯ Once a week
- ◯ Only when in need of assistance
- ◯ Other (please specify)

9. What best describes your level of computer literacy?

| Very High | High | Neutral | Low | Very Low |
|---|---|---|---|---|
| ◯ | ◯ | ◯ | ◯ | ◯ |

10. What best describes your level of experience in programming?

| Very High | High | Neutral | Low | Very Low |
|---|---|---|---|---|
| ◯ | ◯ | ◯ | ◯ | ◯ |

11. What is your level of understanding of the module content?

| Very High | High | Neutral | Low | Very Low |
|---|---|---|---|---|
| ○ | ○ | ○ | ○ | ○ |

12. In general the teaching staff for this module were good at explaining things.

| Strongly Agree | Agree | Neutral | Disagree | Strongly Disagree |
|---|---|---|---|---|
| ○ | ○ | ○ | ○ | ○ |

13. In your view, does the teaching material have enough and correct content for this level?

| Strongly Agree | Agree | Neutral | Disagree | Strongly Disagree |
|---|---|---|---|---|
| ○ | ○ | ○ | ○ | ○ |

14. Do you feel you have worked consistently throughout the term on this module?

| Strongly Agree | Agree | Neutral | Disagree | Strongly Disagree |
|---|---|---|---|---|
| ○ | ○ | ○ | ○ | ○ |

15. Do you feel you have spent enough time doing the actual programming exercises?

| Strongly Agree | Agree | Neutral | Disagree | Strongly Disagree |
|---|---|---|---|---|
| ○ | ○ | ○ | ○ | ○ |

16. It is easy for me to design a program to solve a certain task.

| Strongly Agree | Agree | Neutral | Disagree | Strongly Disagree |
|---|---|---|---|---|
| ○ | ○ | ○ | ○ | ○ |

17. Learning the programming language syntax is confusing.

| Strongly Agree | Agree | Neutral | Disagree | Strongly Disagree |
|---|---|---|---|---|
| ○ | ○ | ○ | ○ | ○ |

18. I have difficulties understanding errors from my own programs.

| Strongly Agree | Agree | Neutral | Disagree | Strongly Disagree |
|---|---|---|---|---|
| ○ | ○ | ○ | ○ | ○ |

19. I can write, compile, run and debug my own programs

| Strongly Agree | Agree | Neutral | Disagree | Strongly Disagree |
|---|---|---|---|---|
| ○ | ○ | ○ | ○ | ○ |

20. Is the structure and content of the module easy to follow and understand?

- ◯ Yes

No (please specify)

21. Do you feel every chapter, assignment or tutorial prepares you well enough for the next task?

- ◯ Yes

No (please specify)

22. Which areas have helped you the most in learning and understanding to program? Please select all applicable answers

- ☐ Online Tutoring
- ☐ Prescribed books
- ☐ Face-to-face tutorials
- ☐ Teaching assistants
- ☐ Lecturers
- ☐ Practical exercises
- ☐ Learning Units
- ☐ Discussion Forum

Other (please specify)

23. If so, any particular reason you have considered withdrawing from or did not finish this module? Please select all applicable answers

- [ ] The module is too advanced beyond my capabilities
- [ ] The module required more time than I could provide
- [ ] I lacked motivation to study
- [ ] The module content is confusing and difficult to follow
- [ ] Tutors or lecturers offered inadequate support
- [ ] I chose wrong course
- [ ] My work or personal related commitments took time from the course
- [ ] I had personal reason(s) that compromised my performance in this module

Other (please specify)

24. Which programming parts of this module do you feel have mostly compromised your ability to succeed?

25. If you were to study this module again what would you do differently in order to do even better?

26. How can this module be improved?

27. Please feel free to comment here on any aspect, positive or negative, of your learning experience on this module

## APPENDIX B: ETHICAL CLEARANCE FROM UNISA

UNISA | college of science, engineering and technology

Thomas Marokane (32092415)
School of Computing
UNISA
Pretoria

2013-10-14

**Permission to conduct research project**

**Ref: 092/TM/2013**

The request for ethical approval for your MTech in Information Technology research project entitled "The hindrances to learning to program in the Introduction to Programming module at Unisa" refers.

The College of Science, Engineering and Technology's (CSET) Research and Ethics Committee (CREC) has considered the relevant parts of the studies relating to the abovementioned research project and research methodology and is pleased to inform you that ethical clearance is granted for your study as set out in your proposal and application for ethical clearance.

Therefore, involved parties may also consider ethics approval as granted. However, the permission granted must not be misconstrued as constituting an instruction from the CSET Executive or the CSET CREC that sampled interviewees (if applicable) are compelled to take part in the research project. All interviewees retain their individual right to decide whether to participate or not.

We trust that the research will be undertaken in a manner that is respectful of the rights and integrity of those who volunteer to participate, as stipulated in the UNISA Research Ethics policy. The policy can be found at the following URL:
http://cm.unisa.ac.za/contents/departments/res_policies/docs/ResearchEthicsPolicy_apprvCounc_21Sept07.pdf

Please note that if you subsequently do a follow-up study that requires the use of a different research instrument, you will have to submit an addendum to this application, explaining the purpose of the follow-up study and attach the new instrument along with a comprehensive information document and consent form.

Yours sincerely

Chair: School of Computing Ethics Sub-Committee

University of South Africa
College of Science, Engineering and Technology
Preller Street, Muckleneuk Ridge, City of Tshwane
PO Box 392 UNISA 0003 South Africa
Telephone + 27 12 429 6122 Facsimile + 27 12 429 6848
**www.unisa.ac.za/cset**

## APPENDIX C: SURVEY ANNOUNCEMENTS

**From:** ICT1512-14-S1 <no-reply@my.unisa.ac.za>
**Sent:** 15 September 2014 10:51 AM
**To:** ICT1512-14-S1
**Subject:** [ ICT1512-14-S1 - Announcement ] Survey

An announcement has been added in the "ICT1512-14-S1" site at myUnisa (https://my.unisa.ac.za/portal/site/ICT1512-14-S1)

Subject: Survey

Group: Site

Message:

Good day,

Please assist the following student with his studies by completing the survey in the link below.

My name is Thomas Marokane and I am a Magister Technologiae: Information Technology student in the College of Science, Engineering and Technology at University of South Africa (Unisa).<o:p></o:p>

I am writing to request for your and your students' participation in a study I am conducting on "the hindrances to learning to program in the Introduction to Programming module at Unisa" under the supervision of Professor Ian Sanders.

<o:p></o:p>http://www.surveymonkey.com/s/unisa_survey_introduction_programming_hindrances

The student does have ethical clearance to perform the study and it will be totally anonymous .

Regards,

Ms D van Heerden
ICT1512@unisa.ac.za
011 670 9185
060 746 7156

This automatic notification message was sent by myUnisa (https://my.unisa.ac.za/portal) from the ICT1512-14-S1 site.
You can modify how you receive notifications at My Workspace > Preferences.

**From:** ICT1512-14-S2 <no-reply@my.unisa.ac.za>
**Sent:** 15 September 2014 10:51 AM
**To:** ICT1512-14-S2
**Subject:** [ ICT1512-14-S2 - Announcement ] Survey

An announcement has been added in the "ICT1512-14-S2" site at myUnisa (https://my.unisa.ac.za/portal/site/ICT1512-14-S2)

Subject: Survey

Group: Site

Message:

Good day,

Please assist the following student with his studies by completing the survey in the link below.

My name is Thomas Marokane and I am a Magister Technologiae: Information Technology student in the College of Science, Engineering and Technology at University of South Africa (Unisa).<o:p></o:p>

I am writing to request for your and your students' participation in a study I am conducting on "the hindrances to learning to program in the Introduction to Programming module at Unisa" under the supervision of Professor Ian Sanders.

<o:p></o:p>http://www.surveymonkey.com/s/unisa_survey_introduction_programming_hindrances

The student does have ethical clearance to perform the study and it will be totally anonymous .

Regards,

Ms D van Heerden
ICT1512@unisa.ac.za
011 670 9185
060 746 7156

This automatic notification message was sent by myUnisa (https://my.unisa.ac.za/portal) from the ICT1512-14-S2 site.
You can modify how you receive notifications at My Workspace > Preferences.

## APPENDIX D: PRIOR PROGRAMMING CLASS VS PERFORMANCE

| | Programming Classes Before (Answer) | Programming Classes Before (Count) | Neutral | Agree | Disagree | Strongly Disagree | Strongly Agree | Total | No.match btwn Prior vs Likert score | Score (Higher % Better) |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | |
| Program Design | Yes | 57 | 22 | 18 | 9 | 0 | 4 | 53 | 22 | 42% |
| Syntax | Yes | 57 | 17 | 10 | 21 | 0 | 0 | 48 | 21 | 44% |
| Error Handling | Yes | 57 | 19 | 15 | 17 | 0 | 2 | 53 | 17 | 32% |
| Write, Run, Compile and Debug | Yes | 57 | 10 | 25 | 7 | 0 | 9 | 51 | 34 | 67% |
| | | | | | | | | | | |
| Program Design | No | 146 | 72 | 24 | 40 | 0 | 0 | 136 | 24 | 18% |
| Syntax | No | 146 | 60 | 34 | 27 | 0 | 17 | 138 | 27 | 20% |
| Error Handling | No | 146 | 48 | 45 | 30 | 0 | 17 | 140 | 27 | 19% |
| Write, Run, Compile and Debug | No | 146 | 72 | 25 | 31 | 0 | 2 | 130 | 27 | 21% |

## APPENDIX E: MINIMUM NUMBER OF HOURS FOR THE MODULE

**Logistical environment assumed to be in place**

As with many of the other modules in this diploma, programming is a practical skill, and the best way of learning a practical skill is by practicing it.

Regular access to both a computer and the internet is a requirement for the National Diploma in Information Technology. Without access to these two tools, you are not going to be able to complete this module. The minimum hours required are as follows:

- computer access – 6 hours per week
- internet access – 2 hours per week

You must have access to these tools from the very first week of the semester to ensure that you keep up to date with your studies.

If you do not have access to these tools at home or at work, you can arrange to use the computers provided by Unisa at their computer labs.

The module is run over a semester of 14 weeks, and 2 hours is deducted from the notional hours for the exam. The time you will need to spend on this module will thus be as follows:

118 (hours) ÷ 14 (weeks) = 8,25 hours per week per module

8,25 (hours) ÷ 7 (days) = 1,18 hours per day per module

If you are registered for 4 modules, for example, you will need to spend a minimum of 5½ hours per day on your studies.

## APPENDIX F: FULL-TIME EQUIVALENT STUDENT ENROLMENT IN PUBLIC HEIS BY ATTENDANCE MODE, MAJOR FIELD OF STUDY AND INSTITUTION, IN 2014 (DHET, 2015)

| Institution | Contact | | | | | Distance | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Science, Engineering and Technology | Business and Management | Education | All Other Humanities and Social Sciences | Total Contact | Science, Engineering and Technology | Business and Management | Education | All Other Humanities and Social Sciences | Total Distance |
| Cape Peninsula University of Technology | 10 466 | 7 447 | 2 154 | 4 159 | **24 226** | 17 | 251 | 0 | 0 | **268** |
| University of Cape Town | 8 435 | 4 713 | 462 | 6 445 | **20 055** | 0 | 0 | 0 | 0 | **0** |
| Central University of Technology, Free State | 4 480 | 2 119 | 2 240 | 1 873 | **10 712** | 8 | 47 | 0 | 58 | **113** |
| Durban University of Technology | 8 579 | 5 556 | 825 | 4 411 | **19 371** | 0 | 0 | 0 | 0 | **0** |
| University of Fort Hare | 2 756 | 1 444 | 1 603 | 4 799 | **10 602** | 0 | 0 | 0 | 0 | **0** |
| University of the Free State | 7 325 | 3 252 | 2 463 | 6 972 | **20 012** | 15 | 720 | 1 266 | 884 | **2 885** |
| University of Johannesburg | 10 920 | 14 025 | 3 055 | 10 263 | **38 263** | 0 | 0 | 0 | 0 | **0** |
| University of KwaZulu-Natal | 12 717 | 5 786 | 4 571 | 10 146 | **33 220** | 355 | 0 | 652 | 0 | **1 007** |
| University of Limpopo | 9 533 | 2 398 | 2 551 | 5 264 | **19 746** | 0 | 0 | 0 | 0 | **0** |
| Mangosuthu University of Technology | 4 378 | 2 045 | 0 | 1 248 | **7 671** | 0 | 0 | 0 | 0 | **0** |
| University of Mpumalanga | 42 | 16 | 77 | 1 | **136** | 0 | 0 | 0 | 0 | **0** |
| Nelson Mandela Metropolitan University | 7 212 | 6 384 | 1 128 | 5 010 | **19 734** | 15 | 0 | 54 | 0 | **69** |
| North West University | 9 553 | 6 700 | 2 916 | 11 035 | **30 204** | 1 568 | 118 | 8 519 | 1 974 | **12 179** |
| University of Pretoria | 18 607 | 6 996 | 3 199 | 9 386 | **38 188** | 2 | 0 | 2 788 | 0 | **2 790** |
| Rhodes University | 1 859 | 1 116 | 487 | 2 674 | **6 136** | 0 | 0 | 0 | 0 | **0** |
| University of South Africa | 0 | 0 | 0 | 0 | **0** | 25 417 | 57 413 | 34 781 | 69 431 | **187 042** |
| University of Stellenbosch | 10 867 | 4 646 | 1 276 | 5 688 | **22 477** | 0 | 0 | 0 | 0 | **0** |
| Sol Plaatje University, Northern Cape | 79 | 0 | 46 | 0 | **125** | 0 | 0 | 0 | 0 | **0** |
| Tshwane University of Technology | 14 540 | 11 988 | 2 893 | 10 350 | **39 771** | 0 | 0 | 0 | 973 | **973** |
| University of Venda | 4 221 | 1 194 | 1 342 | 4 404 | **11 161** | 0 | 0 | 0 | 0 | **0** |
| Vaal University of Technology | 5 932 | 4 333 | 88 | 2 305 | **12 658** | 24 | 143 | 0 | 27 | **194** |
| Walter Sisulu University | 6 382 | 2 873 | 5 087 | 5 266 | **19 608** | 0 | 0 | 0 | 0 | **0** |
| University of Western Cape | 5 749 | 1 889 | 1 318 | 6 753 | **15 709** | 0 | 0 | 0 | 0 | **0** |
| University of Witwatersrand | 11 441 | 4 226 | 2 111 | 7 002 | **24 780** | 0 | 0 | 0 | 0 | **0** |
| University of Zululand | 2 649 | 2 422 | 4 725 | 5 067 | **14 863** | 0 | 0 | 0 | 0 | **0** |
| **Total** | **178 722** | **103 568** | **46 617** | **130 521** | **459 428** | **27 421** | **58 692** | **48 060** | **73 347** | **207 520** |
| **Percentage** | **39%** | **23%** | **10%** | **28%** | **100%** | **13%** | **28%** | **23%** | **35%** | **100%** |

***Source: 2014 HEMIS database, extracted in August 2015.***

Note 1: FTE student enrolments are calculated (a) by assigning to each course a fraction representing the weighting it has in the curriculum of a qualification and (b) by multiplying the headcount enrolment of that course with this fraction.

Note 2: FTE contact students are those who are registered mainly for courses offered in contact mode.

Note 3: FTE distance students are those who are registered mainly for courses offered in distance mode.

Note 4: Definitions for fields of study employed here are the same as those employed in Table 4.

Note 5: The totals above include Undergraduate and Postgraduate courses.

Note 6: As a result of rounding off, numbers and percentages may not necessarily add up.

Note 7: Audited data was amended to ensure that totals balance.

## APPENDIX G: REGISTRATION PASS AND DROPOUT RATES, DISTINCTIONS FOR COS1511 MODULE (SCHOEMAN, 2015)

| Exam Year Sitting | Registration Module Count*** | Examination Sitting Admitted | Normal Wrote* | Normal Pass* | Normal* Pass Rate Percentage (Passed /written %) | Number of Distinctions (Percentage of written in brackets)** | Dropout *** | Dropout rate (Dropout / Enrollment %) |
|---|---|---|---|---|---|---|---|---|
| 2011 Jun | 2381 | 2183 | 1923 | 549 | 28.5% | 198 (10%) | 458 | 19.2% |
| 2011 Nov | 1457 | 1197 | 1191 | 346 | 29.1% | 111 (9%) | 266 | 18.3% |
| 2012 Jun | 2213 | 2061 | 1827 | 582 | 31.9% | 197 (11%) | 386 | 17.4% |
| 2012 Nov | 1528 | 1417 | 1267 | 352 | 27.8% | 106 (8%) | 261 | 17.1% |
| 2013 Jun | 1180 | 1083 | 988 | 337 | 34.1% | 138 (14%) | 192 | 16.2% |
| 2013 Nov | 1145 | 1020 | 920 | 276 | 30.0% | 83 (9%) | 225 | 19.7% |
| 2014 Jun | 976 | 889 | 809 | 258 | 31.9% | 101 (13%) | 167 | 17.1% |
| 2014 Nov | 1054 | 1017 | 894 | 263 | 29.4% | 78 (9%) | 160 | 15.1% |

**(Legend: * – The term 'Normal' refers to students who were registered for the specific semester.**
**** – The percentage of students who wrote the examination and obtained distinctions appears in brackets after the number of distinctions.**
***** – 'Dropout' refers to students who were registered at the end of the semester but did not write the examination. It does not include cancellations, which will raise the dropout rate considerably when taken into account.) Similarly, 'Registration Module Count' indicates the number of students still registered by the end of the semester and therefore excludes cancellations.**

## APPENDIX H: OPEN-ENDED QUESTIONNAIRE RESPONSES



### Q24 Which programming parts of this module do you feel have mostly compromised your ability to succeed?

Answered: 127 Skipped: 78

| # | Responses | Date |
|---|---|---|
| 1 | Open book assessments | 7/10/2015 8:41 PM |
| 2 | To just get a simple programme to run | 5/1/2015 8:48 PM |
| 3 | Loops | 4/30/2015 2:31 PM |
| 4 | Manupualating date with string and arrays | 4/29/2015 12:40 PM |
| 5 | Chapter 3 | 4/29/2015 10:56 AM |
| 6 | I score 34% for my assignement to design a website that affected me to the core and highly discouraged me | 4/29/2015 9:59 AM |
| 7 | Where to place certain syntax Why do you place this-there and that-there | 4/29/2015 9:50 AM |
| 8 | Pseudocode, flow charts and trace tables | 4/28/2015 9:40 PM |
| 9 | None | 4/28/2015 8:09 PM |
| 10 | none | 4/28/2015 8:08 PM |
| 11 | The practical assignment | 4/28/2015 3:33 PM |
| 12 | error handling | 4/28/2015 1:49 PM |
| 13 | The textbook requires a lot of reading and I really learn a lot from practising. | 4/28/2015 1:36 PM |
| 14 | n/a | 3/22/2015 8:54 PM |
| 15 | . | 3/20/2015 5:40 PM |
| 16 | runing and compling | 3/4/2015 5:14 PM |
| 17 | im only on chapter 1. I can answer that yet | 3/4/2015 9:55 AM |
| 18 | jumping between java script and styling script is a little confusing | 2/26/2015 4:40 PM |
| 19 | practical side | 2/23/2015 4:45 PM |
| 20 | Typing out the code. | 2/23/2015 10:50 AM |
| 21 | Its not this module in particular, I am having difficulties doing this many modules at one time, it is mostly the time required. | 2/20/2015 10:30 AM |
| 22 | Technical terms that take time to become familiar with | 2/20/2015 7:56 AM |
| 23 | The foundation | 2/19/2015 11:08 PM |
| 24 | dom and bom | 2/19/2015 11:25 AM |
| 25 | none | 2/19/2015 7:02 AM |
| 26 | Browser | 2/18/2015 9:38 PM |
| 27 | vatiables | 2/18/2015 10:32 AM |
| 28 | understanding the outline of what javascript is and what does it do. | 2/17/2015 7:43 PM |
| 29 | Conditional execution, functions,and data structure. | 2/17/2015 7:02 PM |
| 30 | For now I would say Looping, the prescibe book for ICT1511 didn't cover the section enoungh, so for now I feel it's the most challenging part of the module I have enocuntered | 2/17/2015 10:37 AM |
| 31 | Programming in general. I prefer the practical part (coding) rather than the theory. | 2/17/2015 10:29 AM |

| | | |
|---|---|---|
| 32 | I've done some C++, C# and Python. The program flow in javascript seems weird...i've seen this with the work assignments when trying to get animations to work on the canvas. That is the most frustrating thing at present...in the other languages, i can see what's being called...here, it seems confusing (for now at least, i hope). Also, i have work assignments that are more complex than the work (at that time) for whichever chapter...having to jump between something complex to something simple is annoying. | 2/17/2015 10:07 AM |
| 33 | The syntax and the support when needed | 2/16/2015 9:40 PM |
| 34 | Covering all the necessary basics. | 2/16/2015 7:54 PM |
| 35 | javascript | 2/16/2015 1:44 PM |
| 36 | Introduction to world wide web | 2/16/2015 8:45 AM |
| 37 | Don't know yet | 2/15/2015 7:08 PM |
| 38 | understanding and writing of functions , arrays | 2/15/2015 6:56 PM |
| 39 | Programing language | 2/14/2015 12:37 PM |
| 40 | XHTML AND HTML5...CHARTER 1 AND 2 | 2/14/2015 9:39 AM |
| 41 | the rules of JavaScripiting | 2/13/2015 9:51 PM |
| 42 | Creating programs | 2/13/2015 2:44 PM |
| 43 | It is too early to say | 2/13/2015 2:16 PM |
| 44 | the coding part of things, but i'm getting a hang of it now | 2/13/2015 1:29 PM |
| 45 | None | 2/13/2015 12:47 PM |
| 46 | codes need more time | 2/13/2015 12:28 PM |
| 47 | I would have to say that I'm not really struggling with any programming parts in the module. | 2/13/2015 11:50 AM |
| 48 | none | 2/13/2015 11:34 AM |
| 49 | project | 2/13/2015 11:22 AM |
| 50 | Project assignment takes too much of my time and my own work load too. | 2/13/2015 9:35 AM |
| 51 | Designs the website. Testing the website in different machines | 2/13/2015 9:23 AM |
| 52 | The specific programming languages are very complex | 2/13/2015 8:47 AM |
| 53 | validating | 2/12/2015 9:50 PM |
| 54 | exam assignment | 2/12/2015 8:39 PM |
| 55 | Negating the situation regarding efforts needed to find textbooks due to availability and cost I would say that a more sufficient manner should be structured for prescribed textbooks(Part of cost/ Created text books by local lecturers). | 2/12/2015 8:05 PM |
| 56 | The assignments are very basic, it makes it hard when you consider the "real worlds" opinion of a programmer. I expected to gain more knowledge in the module. I entered knowing nothing, and left knowing psuedocode, which is helpful, but not actually programming. | 2/12/2015 7:58 PM |
| 57 | presently none | 2/12/2015 5:11 PM |
| 58 | debugging | 2/12/2015 5:09 PM |
| 59 | Coding | 2/12/2015 4:51 PM |
| 60 | none | 2/12/2015 4:03 PM |
| 61 | Notepad++ really isn't a great tool for me, I've spend more time troubleshooting my code than actually successfully writing any workable code. I've there was a better Dev Env it would be great | 2/12/2015 3:59 PM |
| 62 | html | 2/12/2015 3:46 PM |
| 63 | cookies I couldn't understand | 10/24/2014 11:00 PM |
| 64 | object oriented javascript andd classes was quiet difficult | 10/17/2014 5:49 PM |
| 65 | Coding | 10/15/2014 5:28 PM |
| 66 | Preject | 10/8/2014 12:38 PM |
| 67 | if the coding part was straight forward and understandable I think I would have mastered this module a long time ago. | 10/3/2014 12:08 PM |

| | | |
|---|---|---|
| 68 | The Array but I went and bought books to understand arrays much better and tried to get more examples around that from these books. | 10/2/2014 4:31 PM |
| 69 | writing codes is very challenging to me, but through the help that i got here, i am positive that i will make it. | 9/30/2014 10:45 PM |
| 70 | The loops | 9/30/2014 6:16 PM |
| 71 | javascript | 9/30/2014 4:23 PM |
| 72 | The part were i have to solve a given problem. | 9/30/2014 11:09 AM |
| 73 | adding certain things to codes | 9/29/2014 4:45 PM |
| 74 | Understanding the basic concepts: No programming background + Doing the module with no study partners + Part time studying make it more of a challenge. | 9/28/2014 9:40 PM |
| 75 | the begining because I got incorrect value from the book most of the time | 9/27/2014 10:01 AM |
| 76 | Chapter 3-8 are very confusing | 9/25/2014 4:11 PM |
| 77 | practicals | 9/25/2014 12:24 AM |
| 78 | everything | 9/24/2014 5:57 PM |
| 79 | inavailability of tutorial classes | 9/24/2014 1:18 PM |
| 80 | further modules, advancing in later modules. | 9/23/2014 11:51 PM |
| 81 | everything | 9/23/2014 9:59 PM |
| 82 | html What you see Is what you get | 9/23/2014 4:04 PM |
| 83 | None, just the lack of time due to 6 day 45 hours shifts | 9/23/2014 2:28 PM |
| 84 | All of them I guess | 9/23/2014 12:29 PM |
| 85 | THE ARRAYS PART AND LOOP | 9/23/2014 11:02 AM |
| 86 | Nothing from my side the programming part make sense to me | 9/23/2014 9:34 AM |
| 87 | coding | 9/23/2014 9:23 AM |
| 88 | Javascript | 9/23/2014 8:17 AM |
| 89 | none , i enjoy programmimg | 9/22/2014 11:27 PM |
| 90 | creating forms | 9/22/2014 9:38 PM |
| 91 | I have yet to fully understand regular expressions, and will have to spend additional time on it. Time is hard to come by for me. | 9/22/2014 9:08 PM |
| 92 | Having to do Do-While, it has been really difficult to figure out what to do. | 9/22/2014 7:24 PM |
| 93 | I think i need to get the basics of programming right then everything will go smoothly | 9/22/2014 7:13 PM |
| 94 | NONE | 9/22/2014 5:54 PM |
| 95 | The confusion on how html code works with JavaScript | 9/22/2014 5:03 PM |
| 96 | at first i was confused on how an image can be linked to a site. the book does not explain it very well nor does the exercises on the learning units. but now i managed to find out how to add image on my own. | 9/22/2014 4:21 PM |
| 97 | Practising from the prescribed textbooks is hard, not explained well enough why certain things are done | 9/22/2014 4:20 PM |
| 98 | Not so much on programming parts but time allocation. I need to give more time to the module | 9/22/2014 3:42 PM |
| 99 | Interpreting a case study to put it in the programming language | 9/22/2014 3:11 PM |
| 100 | Designing a form | 9/22/2014 2:48 PM |
| 101 | FUNCTIONS | 9/22/2014 2:40 PM |
| 102 | Security, graphics | 9/22/2014 2:37 PM |
| 103 | NONE | 9/22/2014 1:54 PM |
| 104 | None. Time is what I lack mostly to excel in my studies. | 9/19/2014 3:45 PM |
| 105 | None | 9/19/2014 11:48 AM |

| | | |
|---|---|---|
| 106 | Coding | 9/18/2014 9:27 AM |
| 107 | Arrays and nested if statements | 9/17/2014 11:03 PM |
| 108 | Algorithms | 9/17/2014 9:54 PM |
| 109 | i kinda struggled when using the document.write() but it was not difficult | 9/17/2014 5:04 PM |
| 110 | The confusion of creating a java script but do not show us how to use in and understand it in html | 9/17/2014 3:01 PM |
| 111 | Understanding Javascript | 9/17/2014 1:51 PM |
| 112 | Chapter 8..Debugging and error handling | 9/17/2014 12:44 AM |
| 113 | None | 9/16/2014 11:23 PM |
| 114 | None | 9/16/2014 1:56 PM |
| 115 | None | 9/16/2014 1:56 PM |
| 116 | The module could place more emphasis on the fact that it is teaching theoretical thinking, not actual coding practices. | 9/16/2014 1:42 PM |
| 117 | the whole module | 9/16/2014 1:08 PM |
| 118 | functions | 9/16/2014 1:00 PM |
| 119 | none | 9/16/2014 12:21 PM |
| 120 | Object Oriented javascript | 9/16/2014 11:03 AM |
| 121 | all | 9/16/2014 8:38 AM |
| 122 | coding and just reading a normal code.. | 9/16/2014 2:57 AM |
| 123 | algorithm | 9/15/2014 9:32 PM |
| 124 | Functions. Loops. Arrays. Error Handling. | 9/15/2014 8:33 PM |
| 125 | none. it just needs more time | 9/15/2014 7:16 PM |
| 126 | Trace tables | 9/15/2014 3:37 PM |
| 127 | html,css and functions | 9/15/2014 3:36 PM |

## Q25 If you were to study this module again what would you do differently in order to do even better?

Answered: 146 Skipped: 59

| # | Responses | Date |
|---|---|---|
| 1 | allocate more time for this course | 7/10/2015 8:41 PM |
| 2 | I have no idea but I'd put more effort find someone to assist me on a full time basis | 5/1/2015 8:48 PM |
| 3 | Save for my studies so I can purchase all the prescribed books in time | 4/30/2015 2:31 PM |
| 4 | Concentrate on assignment more thats where foundaton is | 4/29/2015 12:40 PM |
| 5 | yes | 4/29/2015 10:56 AM |
| 6 | Attend tutorials more | 4/29/2015 9:59 AM |
| 7 | Reference Google and Youtube even more | 4/29/2015 9:50 AM |
| 8 | watch the tutorial on Pseudocode and then try to read all the book | 4/28/2015 9:40 PM |
| 9 | read the schedule | 4/28/2015 8:39 PM |
| 10 | Spend more time on the lessons in the text book | 4/28/2015 8:09 PM |
| 11 | Try to make more time to revise all the material. | 4/28/2015 8:08 PM |
| 12 | Keep on going immaterial personal challenges, work smarter and be more focused on time management. | 4/28/2015 3:33 PM |
| 13 | Allocate some more time for the module | 4/28/2015 2:10 PM |
| 14 | I would study more. | 4/28/2015 1:49 PM |
| 15 | 1. Stick to the schedule provided by the lecturers. 2. Study and Practise consistently. 3. Finish all the exercises. 4. Listen to the lecturers. | 4/28/2015 1:36 PM |
| 16 | Access my self with all the study materials | 4/2/2015 12:05 PM |
| 17 | n/a | 3/22/2015 8:54 PM |
| 18 | I'm studying it for the first time. | 3/20/2015 5:40 PM |
| 19 | keep on practicing code | 3/4/2015 5:14 PM |
| 20 | nothing | 3/4/2015 9:55 AM |
| 21 | make a lot of time | 3/3/2015 10:04 AM |
| 22 | i would have completed wed page design first then started on java script | 2/26/2015 4:40 PM |
| 23 | spending more time with my studies | 2/23/2015 4:45 PM |
| 24 | Focus more from the beginning. | 2/23/2015 10:50 AM |
| 25 | Spend more time studying it | 2/21/2015 2:59 AM |
| 26 | I would want to give it time, but my work on the other hand is taking my time and energy, so that I gave mi self 2 hours a day from 1-3 am in the morning and I sometime fail to wake up due to the fact that I will be super tired that particular day, | 2/20/2015 10:55 AM |
| 27 | Register less | 2/20/2015 10:30 AM |
| 28 | I would take less other subjects at the same time. | 2/20/2015 7:56 AM |
| 29 | Yes | 2/19/2015 11:08 PM |
| 30 | increase study and programming time | 2/19/2015 11:25 AM |
| 31 | read the instruction and announcement from time to time | 2/19/2015 7:02 AM |
| 32 | Have more time to study | 2/18/2015 9:38 PM |

| | | |
|---|---|---|
| 33 | Consistently study as per schedule I created myself. | 2/18/2015 5:13 PM |
| 34 | put more time | 2/18/2015 10:32 AM |
| 35 | know my syntax better and understand my coding better | 2/17/2015 7:43 PM |
| 36 | practice the module and work very hard!!! | 2/17/2015 7:02 PM |
| 37 | Ask my mother to save money for textbooks so I'd would start studying the same time s everybody else instead of chasing them, because that doesn't give me enough time to learn a chapter since I'm trying to catch up...and I would prioritise more | 2/17/2015 10:37 AM |
| 38 | I'm in a auditing environment where I'm constantly involved in projects and do a lot of travelling. I can't afford to quit my job as I pay for my studies. less travelling will allow me more time to study. | 2/17/2015 10:29 AM |
| 39 | Take on fewer modules and spend more time with video tuts. | 2/17/2015 10:07 AM |
| 40 | Spend more time on the module and acquire the prescribed book very early in the beginning of the semester | 2/16/2015 9:40 PM |
| 41 | Start earlier | 2/16/2015 7:54 PM |
| 42 | improve my technique when working with java | 2/16/2015 1:44 PM |
| 43 | Take time to study hard | 2/16/2015 8:45 AM |
| 44 | Try understanding it more better. Find more time to study | 2/15/2015 7:08 PM |
| 45 | try to spend as much time as possible with it | 2/15/2015 6:56 PM |
| 46 | Do activities many times | 2/14/2015 12:37 PM |
| 47 | CREATE MORE TIME FOR THE STUDYS | 2/14/2015 9:39 AM |
| 48 | take more time to understand the rules given to me. | 2/13/2015 9:51 PM |
| 49 | Spend more time on it. | 2/13/2015 3:24 PM |
| 50 | still doing the subject | 2/13/2015 2:44 PM |
| 51 | Get the prescribed book earlier to start studying in order to get ahead of time. | 2/13/2015 2:16 PM |
| 52 | things seem okay the way they are | 2/13/2015 1:29 PM |
| 53 | Focus, focus and focus on the module | 2/13/2015 12:47 PM |
| 54 | Put in more effort and use my time more efficiently towards studying. | 2/13/2015 12:28 PM |
| 55 | I would devote more time to practicing programming in my own time. | 2/13/2015 11:50 AM |
| 56 | study more | 2/13/2015 11:34 AM |
| 57 | do all the self ass and execises on the prisc book | 2/13/2015 11:22 AM |
| 58 | Spend more time studying and practising | 2/13/2015 9:35 AM |
| 59 | Put extra time in this module, go beyond. Do extra research about to gain more knowledge | 2/13/2015 9:23 AM |
| 60 | I would do it full time | 2/13/2015 8:47 AM |
| 61 | study more | 2/13/2015 8:07 AM |
| 62 | spend more time on coding | 2/12/2015 9:50 PM |
| 63 | put in more time | 2/12/2015 8:39 PM |
| 64 | start earlier. 14 weeks can be compact for part time learning on such an interactive unit. | 2/12/2015 8:05 PM |
| 65 | I would study from day one! | 2/12/2015 7:58 PM |
| 66 | Study CSS first before JavaScript | 2/12/2015 6:12 PM |
| 67 | try to build more programme and run it and do more activities. Engage your tutor, discussion forum and lecturer for new development toward the subject | 2/12/2015 5:11 PM |
| 68 | time time time | 2/12/2015 5:09 PM |
| 69 | More Exercises | 2/12/2015 4:51 PM |
| 70 | Go to a full time university | 2/12/2015 4:03 PM |

| | | |
|---|---|---|
| 71 | Try to complete the portfolio | 2/12/2015 3:59 PM |
| 72 | study more, and try to focus on hard topic | 2/12/2015 3:46 PM |
| 73 | START THE EXAM PROJECT ON TIME | 10/25/2014 10:04 AM |
| 74 | take the module in the first semester | 10/24/2014 11:00 PM |
| 75 | try to allocat more time to doing all the exercises | 10/17/2014 5:49 PM |
| 76 | coding | 10/15/2014 5:28 PM |
| 77 | Practise alot | 10/8/2014 12:38 PM |
| 78 | I would really make time and effort to do well | 10/3/2014 12:08 PM |
| 79 | I will do more exercises to learn the code syntax much faster then now, basically spend more time on the module. | 10/2/2014 4:31 PM |
| 80 | obvious, i was going to learn more how write codes | 9/30/2014 10:45 PM |
| 81 | Yes ofcourse | 9/30/2014 6:16 PM |
| 82 | time management | 9/30/2014 4:23 PM |
| 83 | Reduce the number of the modules maybe to two in order for me to get a proper time frame for this module. I would also try by all means to engage with other students and attend tutorial classes. | 9/30/2014 11:09 AM |
| 84 | I would reading the work with understanding, give this module a lot if time | 9/29/2014 4:45 PM |
| 85 | Quit my job :) impossible though. Will just have to be tough enough to take every punch of this module. | 9/28/2014 9:40 PM |
| 86 | Concentrate more on the assignment than blogging | 9/27/2014 10:01 AM |
| 87 | Time is never enough when one is working and studying part time. | 9/26/2014 4:39 PM |
| 88 | I would say maybe if they offer classes this would really help. | 9/25/2014 4:11 PM |
| 89 | concetrating more on practicals | 9/25/2014 12:24 AM |
| 90 | i would even resign from my current job just to stay focus on this module and get a self tutor of 3 hours per day | 9/24/2014 5:57 PM |
| 91 | atend extra classes. | 9/24/2014 1:18 PM |
| 92 | stick to my daily schedule | 9/23/2014 11:51 PM |
| 93 | study with understanding participate on group discussion | 9/23/2014 4:04 PM |
| 94 | - | 9/23/2014 2:28 PM |
| 95 | Try to have some more focus when I go through the prescribed textbook | 9/23/2014 12:29 PM |
| 96 | Spend at least 8 hours a week on it | 9/23/2014 11:23 AM |
| 97 | DO PROJECTS AT THE END OF EACH CHAPTER | 9/23/2014 11:07 AM |
| 98 | YES | 9/23/2014 11:02 AM |
| 99 | This module need a lot of attention, that means to take this module alone without any onther modules... especially what are expected from it. | 9/23/2014 9:34 AM |
| 100 | put more strength | 9/23/2014 9:23 AM |
| 101 | Maybe, but there is so much to get through in so little time. | 9/23/2014 8:17 AM |
| 102 | I would just take this module alone for the semester and nothing else with it | 9/23/2014 7:30 AM |
| 103 | practise more | 9/22/2014 11:27 PM |
| 104 | looking for a tutor | 9/22/2014 9:38 PM |
| 105 | Practice more of the examples in the prescribed book. Spend more time on the module overall, if I was able to. | 9/22/2014 9:08 PM |
| 106 | Attend tutorial classes and lectures. | 9/22/2014 7:24 PM |
| 107 | master programming basics and manage my time properly | 9/22/2014 7:13 PM |
| 108 | Would register at the earliest time and start with studying before its late | 9/22/2014 6:29 PM |
| 109 | PUT MORE TIME | 9/22/2014 5:54 PM |

| | | |
|---|---|---|
| 110 | give more time to each learning unit and also do more exercises. being a week for each unit is not enough | 9/22/2014 4:21 PM |
| 111 | Get more practice programming | 9/22/2014 4:20 PM |
| 112 | Spend more time on exercises and practicals provided by the module. Try to spend more hours weekly on this module as well as discussion forums | 9/22/2014 3:42 PM |
| 113 | Get my own computer with internet access so that I can be able to participate in e-learning and stay up-to-date with what's happening on the module | 9/22/2014 3:11 PM |
| 114 | I will join the tutorial classes | 9/22/2014 2:48 PM |
| 115 | STUDY AND FINISH THE PRESCRIBED BOOK IN TIME | 9/22/2014 2:40 PM |
| 116 | Redo html basics again to design interface | 9/22/2014 2:37 PM |
| 117 | GET MORE TIME | 9/22/2014 1:54 PM |
| 118 | I dont want to repeat, im going to work very hard to pass | 9/19/2014 5:01 PM |
| 119 | Give myself more time to read the text book. | 9/19/2014 3:45 PM |
| 120 | I would spend much more time doing practical exercises and making simple(ish) programs | 9/19/2014 11:48 AM |
| 121 | time | 9/18/2014 9:20 PM |
| 122 | dedicate more time to it, and seek for assistance more | 9/18/2014 9:27 AM |
| 123 | Spend more time on each chapter | 9/17/2014 11:03 PM |
| 124 | Learn more through videos on the internet. Subscribe to sites that give tutorials and exercises. | 9/17/2014 9:54 PM |
| 125 | i would learn and study javascript hard so that i can know where to use it in the webpage | 9/17/2014 5:04 PM |
| 126 | get some help from programmers that already have experience in java script | 9/17/2014 3:01 PM |
| 127 | Yes I probably would but I am working hard to pass this | 9/17/2014 1:51 PM |
| 128 | have access to enough internet time | 9/17/2014 12:44 AM |
| 129 | Time management | 9/16/2014 11:23 PM |
| 130 | I would take fewer modules so as to have enough time to concentrate on this module. | 9/16/2014 5:25 PM |
| 131 | Nothing | 9/16/2014 1:56 PM |
| 132 | Nothing | 9/16/2014 1:56 PM |
| 133 | Concentrate on presenting a clearer understanding of my logic flow and not waste time designing actual functioning algorithms. | 9/16/2014 1:42 PM |
| 134 | Trying to understand better and have tutor for assistance | 9/16/2014 1:08 PM |
| 135 | I would take less modules so that I had more time to focus on this module as it requires a lot more attention. I would then be able to do every example and exercise in the textbook | 9/16/2014 1:00 PM |
| 136 | spend more time on reading the prescribed textbook and doing practicals | 9/16/2014 12:21 PM |
| 137 | improve my time management to these module | 9/16/2014 11:03 AM |
| 138 | no | 9/16/2014 8:38 AM |
| 139 | i guess study like crazy as it blocks other modules,just to get paa through it and over and done with | 9/16/2014 2:57 AM |
| 140 | none | 9/15/2014 9:32 PM |
| 141 | Have more time in the schedule to understand the practicals. | 9/15/2014 8:33 PM |
| 142 | hands on practical work | 9/15/2014 7:16 PM |
| 143 | Have more time for this module | 9/15/2014 5:24 PM |
| 144 | dedicate more time towards the subject | 9/15/2014 5:21 PM |
| 145 | Seek the guidance of a tutor | 9/15/2014 3:37 PM |
| 146 | Nothing.The best would be to take on an Html course before this one | 9/15/2014 3:36 PM |

## Q26 How can this module be improved?

Answered: 135 Skipped: 70

| # | Responses | Date |
|---|---|---|
| 1 | Tutors can pay more attention to comments / queries students post online. | 7/10/2015 8:41 PM |
| 2 | The assignment is a bit too difficult the one where we have to create a program | 5/1/2015 8:48 PM |
| 3 | More assignments can be included as part of the module | 4/30/2015 2:31 PM |
| 4 | I think offer face to face classed maybe for 5 day especially for those that are not familiar with a PC AND Computer skills | 4/29/2015 12:40 PM |
| 5 | more tutorial videos be provided not only videos for exercises. | 4/29/2015 10:56 AM |
| 6 | Unisa centres to make sure that tutorial classes starts early | 4/29/2015 9:59 AM |
| 7 | Explain why you place this-there and that-there | 4/29/2015 9:50 AM |
| 8 | is there any full time classes? | 4/28/2015 9:40 PM |
| 9 | The online disscussions students and myself hadf no audio | 4/28/2015 8:39 PM |
| 10 | By adding more code driven tasks. | 4/28/2015 8:08 PM |
| 11 | Provide face to face classes. | 4/28/2015 3:33 PM |
| 12 | I don't think there is no purpose of improving this module | 4/28/2015 2:10 PM |
| 13 | make it a year module | 4/28/2015 1:49 PM |
| 14 | The prescribed textbook is heavy on text, I would like to see more hands on textbooks. Practising makes learning a way easier. | 4/28/2015 1:36 PM |
| 15 | None | 4/2/2015 12:05 PM |
| 16 | Standardize myUnisa elements and make it user-friendlier. | 3/22/2015 8:54 PM |
| 17 | . | 3/20/2015 5:40 PM |
| 18 | by doing practicalls than theorical | 3/4/2015 5:14 PM |
| 19 | Adding in more exercises to do. | 3/4/2015 9:55 AM |
| 20 | a few more practical exercises. for example, here is your page. make changes so that is can do abc... with out giving us the step by step on how to do it. so we can try work it out first, before getting the answer | 2/26/2015 4:40 PM |
| 21 | attending classes on a daily basis | 2/23/2015 4:45 PM |
| 22 | I don't think the module is the problem, I think am the one who is not giving it enough time | 2/20/2015 10:55 AM |
| 23 | More practical examples on videos, the prescribed book only provides 1 example for each part you study, end of the chapter you think you understood where as you where doing the wrong thing all along. | 2/20/2015 10:30 AM |
| 24 | I am happy with this module - I think training materials are fantastic. | 2/20/2015 7:56 AM |
| 25 | Na | 2/19/2015 11:08 PM |
| 26 | exposing students to professional web developers | 2/19/2015 11:25 AM |
| 27 | I feel this module is in a good state | 2/19/2015 7:02 AM |
| 28 | Is perfect | 2/18/2015 9:38 PM |
| 29 | meat a lecture one or twice a month | 2/18/2015 10:32 AM |
| 30 | break every coding into small parts and explain what that part(of coding)does | 2/17/2015 7:43 PM |
| 31 | Modify the module in such a way that can be comprehended by every student whose enrolling the very module | 2/17/2015 7:02 PM |

| | | |
|---|---|---|
| 32 | I think this module needs "face to face" lactures who will teach students, because there are things one would understand better when they are in a lecture hall, espcially for first year students who are use to seeing their lecture face to face...I know students adapt to the online learning enviroment eventually but it would definitley improve the student's learning pace. | 2/17/2015 10:37 AM |
| 33 | No comment on my side. | 2/17/2015 10:29 AM |
| 34 | Video tuts allow me to playback content faster meaning less time spent. As a part-time student, this time saving is really important...so any access to video tuts would be great..or having all the vodcasts available at the outset. I prefer to go through as much as possible when i do have time which might allow to breathe easier...being restrained by content not being available messes that up. | 2/17/2015 10:07 AM |
| 35 | Quicker responses from the lecturer and eTutor | 2/16/2015 9:40 PM |
| 36 | I don't feel it can | 2/16/2015 7:54 PM |
| 37 | If we could get the prescribed book on time | 2/16/2015 1:44 PM |
| 38 | provide us with teaching assistances | 2/16/2015 8:45 AM |
| 39 | They must send cd/dvd that has a lecture explaining everything about programming | 2/15/2015 7:08 PM |
| 40 | face to face tutorials | 2/15/2015 6:56 PM |
| 41 | Detailed information | 2/15/2015 2:50 PM |
| 42 | Providing tutorial classes timeously | 2/14/2015 12:37 PM |
| 43 | STOP USING DEFFICULT TERM | 2/14/2015 9:39 AM |
| 44 | a study guide might useful especially when it comes to explain the rules of writing code. | 2/13/2015 9:51 PM |
| 45 | Classes | 2/13/2015 2:44 PM |
| 46 | It is too early to say | 2/13/2015 2:16 PM |
| 47 | time management, in regards of the tutors spending more time with students | 2/13/2015 1:29 PM |
| 48 | to me it is a wonderful book with practical excirses | 2/13/2015 12:47 PM |
| 49 | By increasing the amount of exercises | 2/13/2015 12:28 PM |
| 50 | With giving out third-party websites where programming can be practiced in the relevant languages. | 2/13/2015 11:50 AM |
| 51 | video lessons or tutorials | 2/13/2015 11:34 AM |
| 52 | its better now | 2/13/2015 11:22 AM |
| 53 | Overall the module is very good we and students need to try and spend time with our books and do additional research, but due to life's unforeseen circumstances, at times we unable to even get time off from work or etc, to study, my opinion is that we just need to manage our time better. | 2/13/2015 9:35 AM |
| 54 | Put extra exercises on the modules so learners can clear understanding | 2/13/2015 9:23 AM |
| 55 | I think the students can try to get together more | 2/13/2015 8:47 AM |
| 56 | printed materials for students who don't have access to a computer. | 2/13/2015 8:07 AM |
| 57 | more face to face interaction | 2/12/2015 9:50 PM |
| 58 | once a week face to face class | 2/12/2015 8:39 PM |
| 59 | Mostly good so no need on improvement expect for above mentioned. | 2/12/2015 8:05 PM |
| 60 | provide more self assessments, so we can test our knowledge without giving ourselves the benefit of the doubt as we would when studying alone and getting an answer incorrect. self assessments sort of hits you in the face and says you dont know this stuff! | 2/12/2015 7:58 PM |
| 61 | by introducing the Saturdays tutors as your have already started, it help I experience it | 2/12/2015 5:11 PM |
| 62 | requirements of assignments too intensive | 2/12/2015 5:09 PM |
| 63 | By more support from the lectures | 2/12/2015 4:51 PM |
| 64 | by making sure all the students have their study material before giving assignments | 2/12/2015 4:03 PM |
| 65 | by making sure that the book are on time . specially the prescribed textbook | 2/12/2015 3:46 PM |

| | | |
|---|---|---|
| 66 | IT IS FINE THE WAY IY IS | 10/25/2014 10:04 AM |
| 67 | by having everyday classes | 10/24/2014 11:00 PM |
| 68 | more online tutorial videos | 10/17/2014 5:49 PM |
| 69 | Create a start pooint and focuse on the foundation and what is most important and needed buy students to pass module and show understanding. | 10/8/2014 12:38 PM |
| 70 | by not giving us a lot do so we can do better, e.g after doing your assignment or even in between them you have to do exam project how will I get time to prepare for the actual examination. | 10/3/2014 12:08 PM |
| 71 | i think for now it is still good, even though there is always a room for improvement in everything, but for now its fine. | 9/30/2014 10:45 PM |
| 72 | I can say that the module is perfect easy to understand compared to c++ | 9/30/2014 6:16 PM |
| 73 | less chapters | 9/30/2014 4:23 PM |
| 74 | By providing additional study material like pre-recorded videos for students who are unable to attend tutorial classes. | 9/30/2014 11:09 AM |
| 75 | by provide face-to-face tutorials | 9/29/2014 4:45 PM |
| 76 | The module is well organized and structured, has a very motivational and caring in instructor. I would love having a group of students that I could meet with often. Realizing how far behind you are from the discussion forums just takes its toll on you. | 9/28/2014 9:40 PM |
| 77 | Remove blogging | 9/27/2014 10:01 AM |
| 78 | If they put in classes for this module. | 9/25/2014 4:11 PM |
| 79 | i think it is fine for now | 9/25/2014 12:24 AM |
| 80 | by it being compulsory to attend classes maybe during the week or weekends | 9/24/2014 5:57 PM |
| 81 | by offering tutorial classes.Tha | 9/24/2014 1:18 PM |
| 82 | uploading more videos for this module | 9/23/2014 11:51 PM |
| 83 | by employing the lecturers who can help us to understand | 9/23/2014 9:59 PM |
| 84 | having a tutor everyday and have study guides as well | 9/23/2014 4:04 PM |
| 85 | Provide to the point type of Prescribed Books.. Like W3schools | 9/23/2014 2:28 PM |
| 86 | For me I think its good and well organized, and easy to pass | 9/23/2014 12:29 PM |
| 87 | classes | 9/23/2014 11:23 AM |
| 88 | NEED CLASSES | 9/23/2014 11:02 AM |
| 89 | To make it a year module where more attention could be given, | 9/23/2014 9:34 AM |
| 90 | by attending tutorial classes | 9/23/2014 9:23 AM |
| 91 | No Blogging as i think its a total waste of time. Lectures should try and make the subject more appealing and find different ways of getting the content across. Better textbook selection should be made. | 9/23/2014 8:17 AM |
| 92 | It need perhaps online tutoring or webnars to explain most part of the course | 9/23/2014 7:30 AM |
| 93 | for me is fine | 9/22/2014 11:27 PM |
| 94 | classes | 9/22/2014 9:38 PM |
| 95 | Instead of just sending messages to everyone, telling them what they should be doing in which week (which can easily be dismissed and then forgotten), I feel that some form of dialogue with each and every student to confirm that they are aware what is expected from them will improve the likelihood that students will make a point to visit myUNISA and find out what they need to know and do before it is too late. Alternatively, maybe create a mandatory quiz that students have to complete where answers consist of details about the module and can be found in the learning units. | 9/22/2014 9:08 PM |
| 96 | More tutorial classes. | 9/22/2014 7:24 PM |
| 97 | giving enough information on how to start on programming | 9/22/2014 7:13 PM |
| 98 | Giving us example of examination project, exam guide lines | 9/22/2014 6:29 PM |
| 99 | VIDEO TUTORS | 9/22/2014 5:54 PM |

| | | |
|---|---|---|
| 100 | by providing some extra support to students eg class sessions like all other students who are doing social work or teaching,we all same as student we deserve equal treatment and support | 9/22/2014 5:25 PM |
| 101 | have an interactive class. for example have a lecture can have a video link with the student explaining the unit for that week and also having a weekly test like the self assessment included in the year mark | 9/22/2014 4:21 PM |
| 102 | Applying the theory learnt can be explained or motivated better(instead of just being told what to do like in the textbooks) | 9/22/2014 4:20 PM |
| 103 | By providing students with study guide consisting of many different examples as possible | 9/22/2014 3:11 PM |
| 104 | Doing a video for da prescribed book | 9/22/2014 2:48 PM |
| 105 | BY PROVIDING TUTORS WHO KNOW THEIR STORY | 9/22/2014 2:40 PM |
| 106 | More support from lectures | 9/22/2014 2:37 PM |
| 107 | By being given tutors after hours and weekends | 9/22/2014 1:54 PM |
| 108 | By adding more learning units | 9/19/2014 5:01 PM |
| 109 | The tutors are there when needed, the lecturer is also participant (though a bit harsh at times) but she is there. You are sure if you've asked a question, it will definitely be answered. We only have to dedicate our time to our studies. | 9/19/2014 3:45 PM |
| 110 | I had no problems with the academic demands of this module as a whole, I simply couldn't find enough time due to personal factors | 9/19/2014 11:48 AM |
| 111 | build up a library of usefull modules that can be added. reusable components (widgets). display dash boards. | 9/18/2014 9:20 PM |
| 112 | offer classes | 9/18/2014 9:27 AM |
| 113 | A different textbook | 9/17/2014 11:03 PM |
| 114 | By giving less complicated assignments so early in the semester. By the time people have gotten their prescribed books and material, you are forced to submit difficult assignments that you are not ready for. And the you lose marks because you submit a half-arsed assignment just to comply with the rules. | 9/17/2014 9:54 PM |
| 115 | it is right the way it is | 9/17/2014 5:04 PM |
| 116 | by Creating tutor classes because java scripting is mostly practical | 9/17/2014 3:01 PM |
| 117 | More time is needed to work through a 900 page textbook, there is just not enough time. | 9/17/2014 1:51 PM |
| 118 | It is fair as it is but decrease the blogging requirements and discussions class. The workload is too much. | 9/17/2014 12:44 AM |
| 119 | More support from the lecturers. | 9/16/2014 11:23 PM |
| 120 | All the resources are available really but nothing beats face to face tutoring. | 9/16/2014 5:25 PM |
| 121 | Provide classes where we will be able to meet the tutor and discuss the module face to face | 9/16/2014 1:56 PM |
| 122 | Provide classes where we will be able to meet the tutor and discuss the module face to face | 9/16/2014 1:56 PM |
| 123 | More practical exercises could be offered and more full examples of theoretical logic should be provided. Students should be taught earlier WHY and HOW pseudo-code is necessary and implemented, to prevent learners from designing algorithms but rather explain their required functionality. | 9/16/2014 1:42 PM |
| 124 | To have tutor atleast twice a week | 9/16/2014 1:08 PM |
| 125 | More tutorial classes with different times could be made available. | 9/16/2014 1:00 PM |
| 126 | I think they should videos demonstrating the basics of programming | 9/16/2014 12:21 PM |
| 127 | It should be base more into practical than theory | 9/16/2014 11:03 AM |
| 128 | n/a | 9/16/2014 8:38 AM |
| 129 | by introducing classes,not tutorial classes or atleast video conferences once a week,it wouldnt hurt anyone hey.... | 9/16/2014 2:57 AM |
| 130 | By getting a face to face tutor | 9/15/2014 9:32 PM |
| 131 | By scheduling more time to complete chapters. | 9/15/2014 8:33 PM |
| 132 | have some classes for practical's | 9/15/2014 7:16 PM |
| 133 | By removing many assignments you have to submit. | 9/15/2014 5:24 PM |
| 134 | By offering classes at least 3-4 times a month | 9/15/2014 3:37 PM |

| | | |
|---|---|---|
| 135 | Students need to have prior knowledge of html,css and basic programming.Otherwise the liw pass rate continues | 9/15/2014 3:36 PM |

## Q27 Please feel free to comment here on any aspect, positive or negative, of your learning experience on this module

Answered: 101 Skipped: 104

| # | Responses | Date |
|---|---|---|
| 1 | Some of the chapters are really, really long and tedious... It takes 2 days to complete the actual self assessment test (after all the self assessment exercised you've done) The chapters are far too long and some of the activities you have to do in the prescribed book is also way too long. | 7/10/2015 8:41 PM |
| 2 | Not everyone has a laptop or PC and internet if these can be offered with a module at least you buy or be put in bold that you will need to buy a 3g and laptop from school it can help a lot | 4/29/2015 12:40 PM |
| 3 | the module has taught me many things i did not know and the way it is structured, also the learning units do help alot. | 4/29/2015 10:56 AM |
| 4 | I think if I were to attend tutorial my understanding of this module will be better | 4/29/2015 9:59 AM |
| 5 | The module is not difficult, it requires a lot of time and effort. If we(full time employed students) have access to a tutor after hours to contact(eg. a chat room 1-on-1 or group chat), we would be able to ask the relevant questions and possibly get the answers we need promptly. This chat room can operate from 17h00 to around 22h00 daily and provide the student with personnel who could assist. I for one, understand concepts better if i make examples of it and the lecturer/tutor guides me if I misunderstand something. The greatest challenge I have is with time. When I get stuck with a concept, it is difficult to move on without clearly understanding it, then having to wait x-number of days for a reply is just a no-no for me. | 4/29/2015 9:50 AM |
| 6 | I really think some IT modules need us to have full time classes. if you come from a previously disadvantage background, studying on your own it quite a challenge!! and i don't get how we can be require to make a website for a business when we haven't even finished reading the book!!! | 4/28/2015 9:40 PM |
| 7 | Too much content limited time | 4/28/2015 4:35 PM |
| 8 | Provide maybe a lower level of this module's outcome to Electrical Students | 4/28/2015 3:33 PM |
| 9 | The lecturers for this module were excellent, especially that Ms ME VAN HEERDEN, she is excellent and I hope all lecturers from UNISA take note of how passionate she is. | 4/28/2015 1:36 PM |
| 10 | So far i haven't had any problems | 4/2/2015 12:05 PM |
| 11 | Positive: The syllabus looks decently structured Negative: The syllabus does not have Mobile technology programming such as Android and iOs | 3/22/2015 8:54 PM |
| 12 | . | 3/20/2015 5:40 PM |
| 13 | i love proramming and am doing well in programming | 3/4/2015 5:14 PM |
| 14 | Im not complaining i can't say much because i haven't bought books yet due to family problems but as soon as i get the books i'm going to work hard than no one else | 3/3/2015 10:04 AM |
| 15 | i really dislike the blogging, i feel it has no value to my studies, i would rather have small practical exercises/assignments for each chapter or 3 chapters then have to blog | 2/26/2015 4:40 PM |
| 16 | I love the module but haven't yet experienced the whole thing | 2/24/2015 11:51 PM |
| 17 | Only time is needed for this module to become successful. | 2/23/2015 4:45 PM |
| 18 | I am really enjoying this module because it challenges me. I wish I had taken one less subject this semester so I could spend more time doing practical exercises. | 2/20/2015 7:56 AM |
| 19 | develop a unisa study manual as prescribed books are expensive | 2/19/2015 11:25 AM |
| 20 | the module has put lot of sense In mind and it reminded me why I choose to study IT this module revives my drive for programming | 2/19/2015 7:02 AM |
| 21 | The program uses to much data and I can't afford to spend hours on the web. | 2/18/2015 9:38 PM |
| 22 | The module need to be simplified since we do it on ODL and i think we struggling alot and i do not think its me whose experiencing this kind of challenge as am repeating htese module for several times. | 2/17/2015 7:02 PM |

| | | |
|---|---|---|
| 23 | Online learning enviroment is the only negtive comment I have about this module, because not everyone has computer access or internet access , what if one is going through a chapter and they come across something they don't understand they have to go to the discussion forum to see if anyone has came across the same problem before posting or when it's the weekend, they have to wait until Monday, book a computer at the Lab (which is an hour duration) to ask a question, which means their weekend study have to come to a hault, hence learning pace is slow. The positive aspect is that you learn to be independant, as a programmer you have to learn how to work independantly and with a group..so the independant part teaching you to be different, to think differently and that is when one is at their most innovative stage or "in their zone" that's what our generation calls it. | 2/17/2015 10:37 AM |
| 24 | I thing studying and practising on a regular basis will prepare you for exam if you have enough time on your side. Working and studying at a same time it's not a child's play | 2/17/2015 10:29 AM |
| 25 | The module is good the bad thing about it is the prescribed book is not available at the moment and its taking time | 2/16/2015 1:44 PM |
| 26 | Positive | 2/16/2015 8:45 AM |
| 27 | I still have a lot to learn about programming, it's my first time doing this module and it scares me. I did I.T. Networking now programming is a whole new thing to me | 2/15/2015 7:08 PM |
| 28 | this is a very interesting module to students with no programming experience but it get very confusing coding does not work which sometimes causes you to loose interest | 2/15/2015 6:56 PM |
| 29 | Please provide tutorial classes | 2/14/2015 12:37 PM |
| 30 | I REALY LIKE TO STUDY FUTHER AND FINISH BY UNDERSTANDING THE MODULE AND USE IT ON MY OWN | 2/14/2015 9:39 AM |
| 31 | I don't any comment. | 2/13/2015 9:51 PM |
| 32 | Some of these questions are being asked to early for first year students to answer. | 2/13/2015 6:35 PM |
| 33 | So far, I have enjoyed learing this module and know that I will learn a lot more as I go on with the module. | 2/13/2015 2:16 PM |
| 34 | things are good thys far and i'm appreciating the disscussions we have with our tutors the most. | 2/13/2015 1:29 PM |
| 35 | I am still at the beginning of the module; it will be unethical to comment on it right now. | 2/13/2015 12:47 PM |
| 36 | Well it is interesting and comes with new challenges so looking forward to it | 2/13/2015 12:28 PM |
| 37 | none | 2/13/2015 11:34 AM |
| 38 | This module is well structured to produce competent developers. | 2/13/2015 9:35 AM |
| 39 | Time frames this modules need time to study, working and studying reduce my lifestyle is hard | 2/13/2015 9:23 AM |
| 40 | I enjoy the new concepts | 2/13/2015 8:47 AM |
| 41 | All in all I enjoyed the module because it was fairly easy. | 2/12/2015 7:58 PM |
| 42 | through the Saturday tutor you can love programming, no matter how your background in programming is poor, you can make it easily. | 2/12/2015 5:11 PM |
| 43 | This module need more support from the lecture | 2/12/2015 4:51 PM |
| 44 | Most of us didn't receive our study material so please help us with despatch since its not answering our emails we don't want to get 0 for our first assignments | 2/12/2015 4:03 PM |
| 45 | Good module, but I just can't figure out why I can master PHP and get a distinction in ICT2613 but I can't pass this module. I guess JavaScript doesn't interest me as much as PHP does. | 2/12/2015 3:59 PM |
| 46 | the problem is the text book we order and we don't get it on time , that makes not be able to the assignment on time, and also preparing the exam. | 2/12/2015 3:46 PM |
| 47 | its a difficult module to understand especially when there is no time in the second semester | 10/24/2014 11:00 PM |
| 48 | due to studying part time, there is not alot of time to spend on all modules. some online assistance or online class once a week to help with difficult exercises. There was a quiet a few spelling errors and incorrect references in Prescribed textbook. PLease review getting a new textbook or correcting errors | 10/17/2014 5:49 PM |
| 49 | I really wish to master this module but I feel I am losing hope since I have been doing bad on it. I hope the coming examination I will do great and finish this module. | 10/3/2014 12:08 PM |
| 50 | It was tough because of the blogging of each chapter, I do recommend to post an exercise from the book to the blog rather then writing stuff. I never did it in my previous development courses as well. This is a correspondent course so why keep tabs on students as to whether they read the book. No blogging for me | 10/2/2014 4:31 PM |

| | | |
|---|---|---|
| 51 | i think if lectures can, keep on informing students about any new development using SMS's than, my life or myunisa, because some students does not even know how to go about in my life side. | 9/30/2014 10:45 PM |
| 52 | Well to me this is the most joyful module to ever deal with, and I encourage fellow students to take time as I did in the module. | 9/30/2014 6:16 PM |
| 53 | its a bit challenging | 9/30/2014 4:23 PM |
| 54 | I came across of a lot of new and difficult things in this module, but I am willing to learn and get better in programming | 9/29/2014 4:45 PM |
| 55 | So far I blame it on me!!! :) | 9/28/2014 9:40 PM |
| 56 | two assignments for this module is not enough. The second assignment is too big in a short space of time | 9/27/2014 10:01 AM |
| 57 | This module is interesting, but the problem is that there is a lot of work to do here. They should try to decrease the workload here. | 9/25/2014 4:11 PM |
| 58 | if the university can post more of videos so that we can see some example to go forward | 9/25/2014 12:24 AM |
| 59 | somehow i learnt a lot from this module and if i failed it i would not blame anyone but re-register it and focus because i once had a dream pf being a programmer. | 9/24/2014 5:57 PM |
| 60 | Thanks for the effort you are putting to better our understanding. | 9/24/2014 1:18 PM |
| 61 | Tutors must make more time to respond to student's questions | 9/23/2014 11:51 PM |
| 62 | this modules make it hard espcily when using Java language it needs fully dedicates | 9/23/2014 4:04 PM |
| 63 | My experience with this module was good | 9/23/2014 12:29 PM |
| 64 | ITS A CHALLENGING MODULE THAT NEED TIME ANS CLASSES BUT I LOVE IT CAUSE IT MAKES ONE THINK AND BRE CHALLENGED | 9/23/2014 11:02 AM |
| 65 | I am learining a lot in this module expecially when I can create my own web page and to do changes accordingly. However 6 months is to short for this course and what are expected in a shaort time frame, that the other modules are not even considered. I think that is the reason most of the students dont get time to grasp all the work and this happen every semester. | 9/23/2014 9:34 AM |
| 66 | hard work and dedication is the key | 9/23/2014 9:23 AM |
| 67 | I have learnt a lot through youtube to try and get the result that i want to get in programming. I think the lectures expect a lot from us and not taking into consideration that we all have other commitments as well. There are too many assignments which requires a lot of time. You are unable to concentrate on the other subjects as programming takes up a lot of your time. | 9/23/2014 8:17 AM |
| 68 | i wish to pass as this will help me | 9/22/2014 11:27 PM |
| 69 | minimum | 9/22/2014 9:38 PM |
| 70 | The lecturer for ICT1512 and ICT1513 is the best lecturer I have encountered during my short time at UNISA. She is always present on the discussion forums, ready to help, and it shows that she cares about students passing the module. This kind of presence motivates students MUCH more to do better than other modules I have done where lecturers were completely absent and unresponsive to any form of contact. | 9/22/2014 9:08 PM |
| 71 | I felt reLly great when i mannaged to do some of the applications to solve problems on my own. | 9/22/2014 7:24 PM |
| 72 | I having problems with the practical part of programming if I can get help on this part I all will be well with my studies. | 9/22/2014 7:13 PM |
| 73 | the course structure is well designed and i feel inlove with this course. i might feel that other things are not well explained, we have the resources to help us get were we need to. i have not issues with the course,other than the pace as some people might take a while to understand just one unit. moving to next chapter quickly for other might not work me included. | 9/22/2014 4:21 PM |
| 74 | I enjoy the content but struggled with certain aspects of the practical side | 9/22/2014 4:20 PM |
| 75 | More time is required in study of the module | 9/22/2014 3:11 PM |
| 76 | So far I try my best to complete it cos I repeat the module | 9/22/2014 2:48 PM |
| 77 | GOOD | 9/22/2014 2:40 PM |
| 78 | Please provide a more comprehensive study schedule | 9/22/2014 2:37 PM |
| 79 | This module needs more time and classes twice a month from lecturer.it would be better to get guidance from lecturer face to face | 9/19/2014 5:01 PM |

| | | |
|---|---|---|
| 80 | The learning material and prescribed textbook are very well laid out, where each section or chapter is always only one step forward from the previous one, making the work easy to follow and understand. | 9/19/2014 11:48 AM |
| 81 | i am studying electronics for micro-controllers. desk top programming is not interesting for me. partnering with Atmel and learning to program there micro-contollers in c would be more usefull. | 9/18/2014 9:20 PM |
| 82 | Positive | 9/18/2014 9:27 AM |
| 83 | I found the textbook a bit lacking with regards to the exercises but no solutions , so you never really know if your on the right track, because you need to post your exercise answers to discussion board and hopefully receive a reply within two days if you lucky. Usually by that time you moved on to next chapter only to find out your understanding of previous chapter was wrong. | 9/17/2014 11:03 PM |
| 84 | The one thing that I am really disappointed with, is that when people requested an extension on the second assignment, they were told that this cannot be done. Most students were not ready for this assignment and the lecturer refused to extend the deadline. However, we later found out that students who did not submit on the deadline where given 10 extra days to completed and submit!!! The students who did submit on time, were cheated of this extra time and had to forfeit extra marks yet they complied with the rules. I think that this was unjust and unfair and when students found out and questioned this, they were ignored. We all paid our hard earned money for this course but are not treated with respect and faimess. Some of us who work full time and have to still come home to responsibilities of family and household duties, and we still try to complete our studies. This is very de-motivating and disappointing. | 9/17/2014 9:54 PM |
| 85 | its a great module for me besides i love coding so it just need time to learn the basics | 9/17/2014 5:04 PM |
| 86 | This module is great you learn a lot but without a tutor class to show you the practical side of programming you can not pass this module 100% | 9/17/2014 3:01 PM |
| 87 | I enjoy this module but would prefer if this was rather a year module not a semester module. | 9/17/2014 1:51 PM |
| 88 | I can not complain much because I ended up loving this module the self assessments were okay, the assignments were exciting but the only thing I despised was the weekly blogging. | 9/17/2014 12:44 AM |
| 89 | It was challenging but worth it. | 9/16/2014 11:23 PM |
| 90 | If there was a tutor available on every province available to assist students on a face-toface basis, it would go a long way into helping students cope and understand. | 9/16/2014 5:25 PM |
| 91 | I enjoy this module,though I strongly feel we must have classes,and study material must be delivered in time | 9/16/2014 1:56 PM |
| 92 | I enjoy this module,though I strongly feel we must have classes,and study material must be delivered in time | 9/16/2014 1:56 PM |
| 93 | I think this module need full time student | 9/16/2014 1:08 PM |
| 94 | My experience with this module is good | 9/16/2014 12:21 PM |
| 95 | no comment | 9/16/2014 11:03 AM |
| 96 | honestly i dont have any positivity about this module i hate with all my being.i am more of a business informatics person lol..... | 9/16/2014 2:57 AM |
| 97 | none | 9/15/2014 9:32 PM |
| 98 | nothing negative thus far. only the points mentioned on the above two questions. | 9/15/2014 8:33 PM |
| 99 | this module is good, fun and interesting. it just needs more time, otherwise its not that difficult | 9/15/2014 7:16 PM |
| 100 | more sms would be greatly appreciated to complement the emails that we receive notifying us of what to study next. | 9/15/2014 5:21 PM |
| 101 | Javascript is an embedded program.It embeds into html.you cant run it on its own.Why a study module can be structured to be learnt like this,without prior Html grounding is beyond me.I raised it with the lecturer the first week and she is aware of the problem | 9/15/2014 3:36 PM |

## APPENDIX I: PROFESSIONAL EDITING DECLARATION


# DECLARATION BY LANGUAGE EDITOR

21 July 2017

TO WHOM IT MAY CONCERN

**DECLARATION: Language Editing of Dissertation**

I hereby declare that I have edited the Magister Technologiae (in Information Technology) dissertation of THOMAS SELAKANE MAROKANE entitled *"HINDRANCES TO LEARNING TO PROGRAM IN AN INTRODUCTORY PROGRAMMING MODULE"*. All changes made by me were made up to 21 July 2017. The editing scope included the research summary and from the glossary of terms to the end of chapter 8. During the editing process, some 6 116 corrections were made. It is the responsibility of the student to address any comments from the editor or supervisor. Additionally, it is the final responsibility of the student to make sure of the correctness of the dissertation.

**Khomotso Bopape**

*Full Member of the Professional Editors' Guild*

Professional EDITORS Guild

*Let's Edit is a **Level 1 EME B-BBEE Contributor** (Procurement Recognition Level = **135%**)*

Address: **P.O. Box 40208, Arcadia, Pretoria, 0007**
Tel No.: **012 753 3670,** Fax No.: **086 267 2164** and Email Address: **khomotso@letsedit.co.za**