PROGRAMMING CONCEPTS, TEACHING AND LEARNING STYLES AND COGNITIVE FACTORS

Don Kannangara[1], Darrell Fisher[2]

[1]**Waiariki Institute of Technology, New Zealand**

[2]**Curtin University, Australia**

ABSTRACT

In the last decades, a great deal of educational research has been carried out considering demographic, cognitive and social factors to improve the teaching of computer programming. Nonetheless, settling on an effective teaching method is still a contentious issue among programming tutors. There is a range of varying but essential concepts to be understood in learning a programming language. Thus, the teaching styles to be used to teach each of these different concepts can vary due to the complexity and nature of each. This study is aimed at identifying such concepts and the preferred teaching style for each in JAVA language. The results of a survey of students who recently completed introductory level JAVA programming language highlights difficulties with these concepts among certain learners, and also suggests the preferred teaching style for each one. Furthermore, this study investigates the preferred learning styles for learners with either artistic or logical intelligences. It reveals a strong correlation between such cognitive factors and preferred teaching styles in learning programming concepts, and, despite some individual variations, supports the utilisation of visual and kinaesthetic rather than auditory channels.

Keywords: introductory computer programming, cognitive factors, teaching styles, learning styles

INTRODUCTION

Tutors spend a significant amount of class time on explaining the fundamental computer language concepts of classes and the relevant algorithms to students of computer programming (Carlisle, 2004). Some novice students learn their first computer language without any difficulty, while others struggle and require a huge amount of support and assistance from tutors (Garner, Haden & Robins, 2005). According to Robins, Rountree and Rountree (2003) such differences between novices could be due to their past knowledge, the strategies they use and the mental models of the programmes. Renumol, Jayaprakash and Janakiram (2009) say that the reason for many students finding it difficult to learn programming is the high cognitive requirements. Innate cognitive ability is but one factor of many that affect students' grades in introductory programming (de Raadt et al, 2005), but nonetheless a crucial one. According to a study carried out by Dillashaw and Bell (1985), either the student has the cognitive ability or not, as the hypothesis that logical thinking skills are enhanced by teaching them computer programming or vice versa may not be true . Nonetheless, logical cognition is not a single concept, but can be further subdivided into various cognitive styles. Even allowing for the requirement of a high level of cognitive ability, there is still scope for variation of teaching methodology. For example, as a result of an extensive literature survey, Prasad and Fielden (2002) suggested a balanced approach to teaching introductory programming which spans multiple cognitive styles.

In this present research project, an exploration of cognitive styles reveals that there are numerous, and at times diverse, definitions. The free on-line dictionary defines the word *cognitive style* as "a term used in cognitive psychology to describe the way individuals think, perceive and remember information, or their preferred approach to using such information to solve problems" (TheFreeDictionary, 2009, p.1). According to Carland and Carland (1990), cognitive style refers to the cluster of thought processes and patterns employed by individuals. A more detailed study done on cognitive styles by Liu and Ginther (2005) expands this definition from a range of different perspectives. Aspects of cognition include breadth of categorizing, cognitive complexity versus cognitive simplicity, deep-elaborative versus shallow-reiterative, divergent versus convergent, global versus analytical, objective versus non-objective, organiser versus nonorganiser, right brained versus left brained, risk taking versus cautiousness, scanning versus focusing, sensitisers versus repressors and simultaneous versus successive, as well as a wide range of sensory modality preferences. These include visual versus haptic perceptual type, verbaliser versus imager, verbaliser versus visualiser, visualiser versus haptic, holist versus analytic, holist analytic versus verbal imagery and holist versus serialist. Other models include Kolb's learning style and the MBTI learning style (Liu & Ginther, 2005). The difficulty of translating this high number of varying and complex cognitive models into actual classroom tutorial practice need not be elucidated. Thus, the transparency and general acceptance of the right versus left-brained perspective of human cognition led to this model being applied in the present research project.

According to the brain-dominance theory introduced by Roger Sperry, the left half of the brain uses rational thinking for logical conclusions whereas the right half of the brain synthesises the whole picture (Dew, 1996). Most people accept that the left side of the brain is more functional in a logical person, whereas an artistic person has a more functionally developed right side (Pessis,2007). Micheal Saling, an Australian neuropsychologist, describes right or left brain as a metaphor for an overall cognitive style, and notes that both work in concert with each other (as cited in ABC Science, 2008). In this research output, the authors contend that people with strong logical, analytical thinking and mathematical abilities have left brain dominance and people with high artistic ability, such as intelligences used in singing, painting, and writing poetry, have right brain dominance.

This research also included a literature search on learning styles that have been experimented with in the teaching of computer programming. Zander et al. (2009) categorised learners as active, reflective, sensing, intuitive, visual, verbal, sequential, global, inductive, and deductive. A survey carried out by a group of researchers on suitable learning styles to teach mathematics and programming concluded that programming students prefer sequential, visual, and active learning styles whereas mathematics students prefer sequential, inductive, and deductive learning styles (Zander et al. 2009). According to Carlisle (2004), students who prefer visual representations are more successful at learning programming concepts. Other aspects of learning styles include "seeing and hearing; reflecting and acting; reasoning logically and intuitively; memorizing and visualizing and drawing analogies and building mathematical models; steadily and in fits and starts" (Felder & Silverman, 1988, p.674). Thomas (2000) suggested the ordering of learning styles from the time-honoured perspectives of visual, auditory and kinaesthetic in order to improve teaching Unified Modelling Language (UML) modelling of object-oriented systems, and this seems to the present authors to encapsulate neatly many of the aspects mentioned in greater detail by other theorists.

Thus, this research has been focussed on cognitive factors with regard to left/right brain-dominance theory, examining the relevance to different concepts of the JAVA programming language. Cognitive factors therefore include the logical and artistic tendencies of students. We employ three learning styles, namely visual, auditory and kinaesthetic. The primary aim of this research is to explore the difficulty level of each JAVA concept for students and then to the focus on the relationship between the difficulty level of each concept and the cognitive factors of students. We also examine suitable teaching styles for each concept, taking the students' cognitive factors into consideration.

METHODOLOGY

Twenty seven students who had attended the introductory programming course using JAVA programming language at the Waiariki Institute of Technology, Rotorua, participated in a web-based survey. The questionnaire had three main parts. The first part had multiple choice questions for collecting participants' demographic information such as age group, gender, work experience, and prior academic qualifications. Each question had a number of options listed using radio buttons and the participants of the survey had to select one option for each question. The last two questions in the part were for the participants to make a judgement on their level of artistic and logical abilities. Measures of artistic ability included skill in singing, painting, and writing poetry. Logical ability determinants included logical, analytical and mathematical ability. Each of these two questions had options Poor, Average, Good, Very Good, and Excellent. Options were listed using radio buttons. The participants in the survey had to select one option in each question.

The second part of the questionnaire was designed to collect information about the participants' opinion of the difficulty level of the most important concepts in JAVA programming. The following sixteen concepts were listed:

- Variable types (int, char, & double etc.)
- Variable categories (local, parameter & instance)
- Using conditional statements (if ..then ..else)
- Using repetitive statements (loops)
- Understanding of the concept of classes and objects
- Creation of an object using a class
- Creation of a template for a class

- Use of parameter variables (arguments) in a method
- Returning a value from a method
- Testing and debugging
- Preparation of test data for a programme
- Using arrays for primitive data types
- Using arrays for objects of a class
- String manipulation in JAVA (using methods in String class)
- Using text files for input and output
- Logic depiction methods (structure diagrams etc.)

Each concept had five options with the labels “too difficult”, “very difficult”, “difficult”, “not difficult” and “very easy” with a radio button for each option. The participants in the survey had to select one radio button option for each concept. This part of the questionnaire also had blank spaces for the participants to add any other concepts which are not listed and could be important in JAVA language on the questionnaire and also to indicate the difficulty level.

The aim of the third part of the questionnaire was to find out each student’s preferred teaching style(s) for each of the JAVA programming concepts. This part of the questionnaire included textual descriptions of the visual, kinaesthetic and auditory learners. It also had self-explanatory pictures of the three types of learners, aiming to educate students about them. The first question in this part was for the student to make a self judgement of his/her type of learning style(s). The question had three check boxes with the labels for the three types: visual, kinaesthetic, and auditory. The student could tick one or more check boxes in this question. The last set of questions in this part of the questionnaire was to gather information about the most suitable teaching style to teach each concept from the students’ points of view. As in part two, sixteen concepts were listed. Each concept had three options with the labels “visual”, “kinaesthetic” and “auditory” with a check box for each option. The students had to select one or more check box options for each concept.

While it may appear that all computer work is intrinsically visual and kinaesthetic, the same can not be said of all tutorial methodologies. Some aspects of the theory behind programming language seem, at least to tutors, to suit a “lecture” or “set reading” approach. Other commonly favoured methodologies may be described as “read and do” or “listen and copy”, where the student follows a list of instructions to perform example tasks.

This questionnaire also provided details related to ethical issues, where the participant had to give consent to participate in the survey by ticking a check box. Twenty seven students participated in the survey. The Statistix 7 statistical analysis package was used to analyze the data and Excel spreadsheets were used for graphs.

RESULTS AND DISCUSSION

Table 1 contains percentage figures for respondents concerning difficulty levels of each JAVA concept as identified by students. According to these figures, the most difficult concepts to understand were the use of arrays for storing primitive data and objects. Many students had no difficulty in understanding the concept of primitive data types.

Table 1: Difficulty Levels of Java Concepts

JAVA Concept	Too difficult	very difficult	difficult	not difficult	very easy
Variable types (int, char, & double etc.)	0%	0%	12%	44%	44%
Variable categories (local, parameter & instance)	0%	4%	27%	50%	19%
Using conditional statements (if ..then ..else)	4%	4%	26%	44%	22%
Using repetitive statements (loops)	4%	4%	30%	40%	22%
Understanding of the concept of classes and objects	0%	19%	15%	47%	19%
Creation of an object using a Class	0%	4%	8%	52%	16%
Creation of a template for a class	0%	12%	35%	41%	12%
Use of parameter variables (arguments)in a method	0%	8%	31%	42%	19%
Returning a value from a method	0%	4%	23%	42%	31%
Testing and debugging	4%	12%	23%	53%	8%
Preparation of test data for a program	4%	8%	26%	50%	12%
Using arrays for primitive data types	8%	15%	42%	31%	4%
Using arrays for objects of a class	4%	19%	42%	27%	8%
String manipulation in JAVA (using methods in String class)	0%	8%	31%	42%	19%
Using text files for input and output	12%	15%	27%	42%	4%
Logic depiction methods (structure diagrams etc.)	0%	17%	29%	42%	12%

The five options for levels of difficulty given for each concept were quantified with the values ranging from 1 to 5. The difficulty levels were calculated as the totals of values chosen for each concept for students who rated themselves high to excellent either for artistic abilities or for logical and analytical thinking abilities, and also for those who claimed both abilities. Figure 1 shows the total difficulty levels of each concept for students with excellent or very good artistic, logical, and both abilities. Figure 1 clearly indicates the fact that many students with higher artistic abilities found it difficult to understand almost all the JAVA concepts than the students with higher logical and both abilities.

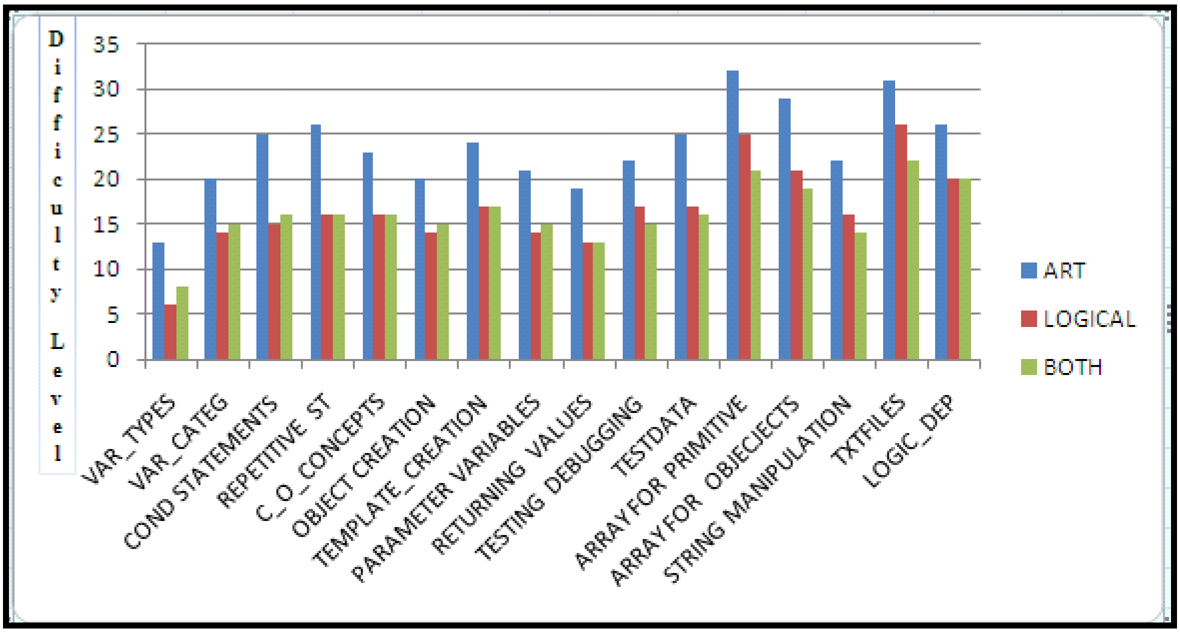

Figure 1 : Difficulty level of Java Concepts

In the second part of the questionnaire, each student indicated his/her learning type(s) by ticking one or more check boxes labelled visual, kinaesthetic, and auditory. According to Figure 2, there are many learners who are either kinaesthetic, kinaesthetic combined with visual or kinaesthetic combined with auditory. None of the learners had the combination of visual and auditory learning types.

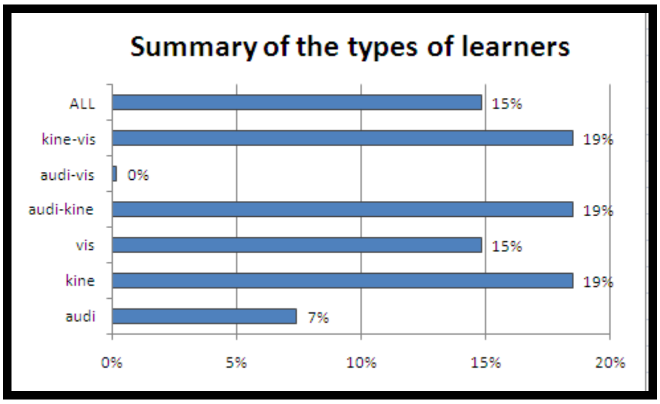

Figure 2: Types of learners

Each student indicated a preferred learning style for each of the concepts listed on the questionnaire. Figure 3 shows the total number of visual learners who indicated the preferred learning style as visual, kinaesthetic, or auditory for each concept.

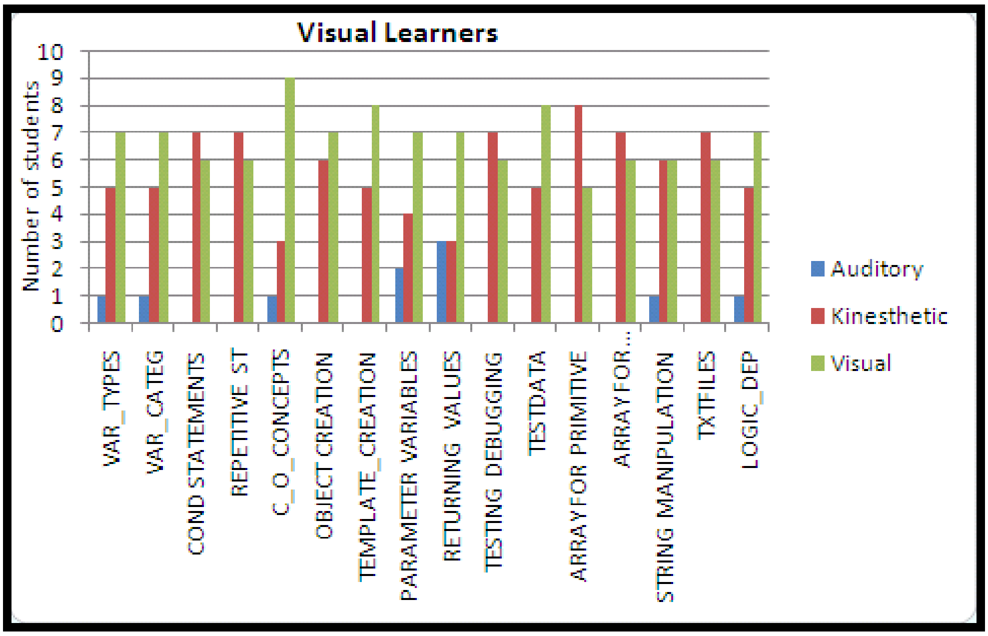

Figure 3: Visual learners' preferred teaching styles

According to Figure-3, visual learners also preferred a visual and kinaesthetic learning style for almost all the concepts. There was limited support for the auditory medium for most concepts.

Figure 4 shows the total number of kinaesthetic learners who indicated their preferred learning style as visual, kinaesthetic, or auditory for each concept.

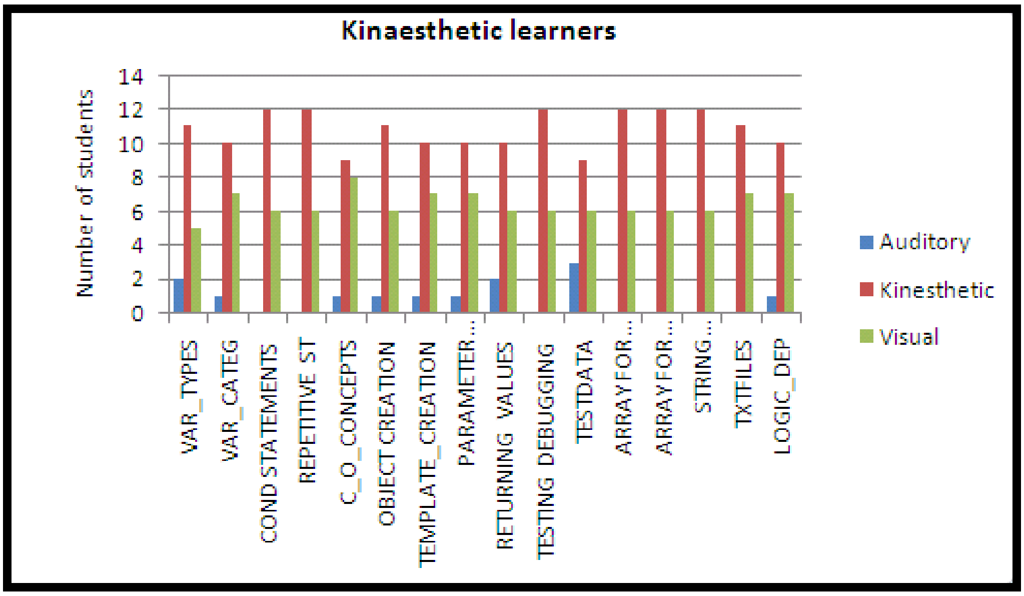

Figure 4: Kinaesthetic learners' preferred teaching styles

According to Figure 4, kinaesthetic learners preferred kinaesthetic and visual ways of learning for almost all the concepts. Again, there was negligible support for auditory input for most of the concepts.

Figure 5 shows the total number of auditory learners who indicated their preferred learning styles as visual, kinaesthetic, or auditory for each concept.

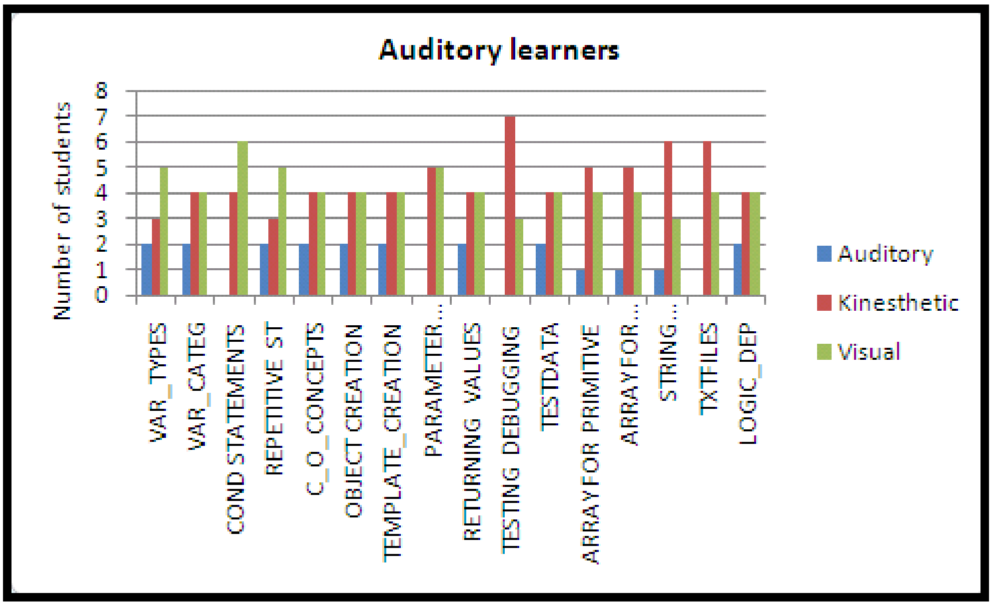

Figure 5: Auditory learners' preferred teaching styles

Although we expected auditory learners to favour auditory methods, Figure 5 shows that many auditory learners believe that kinaesthetic and visual learning styles should be used, with limited recourse to auditory learning styles, to teach most JAVA concepts.

The questionnaire also included an item on preferred logic depiction method that could be used to show the logic flow of the methods used in JAVA language. The students had to choose one of three options given with the labels "flow chart", "structured English", and "structure diagram". The percentages of the students' preferences are shown in Figure 6. It reveals that most students preferred the structure diagram as a logic depiction method for JAVA language.

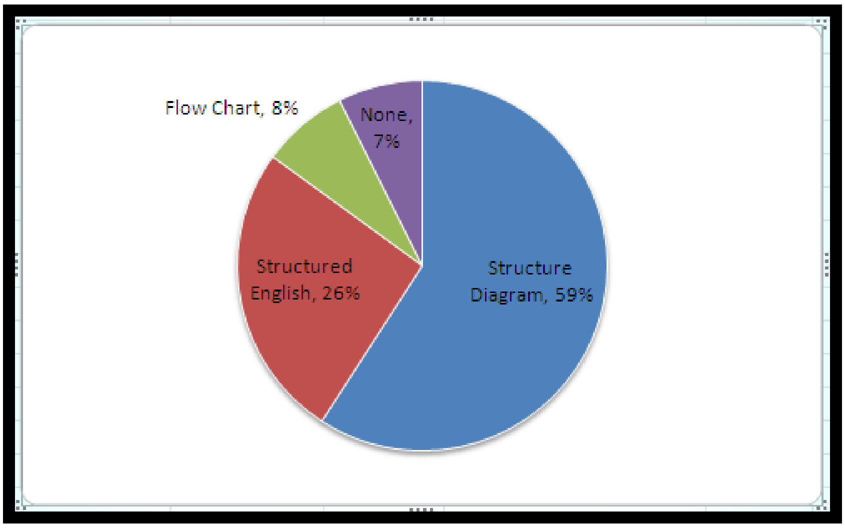

Figure 6: Preferred logic depiction methods

CONCLUSION

The outcomes of this research are twofold, on the one hand providing a salutary reminder of the intrinsic logical difficulty of learning programming, but on the other indicating a way forward for teaching methodology. The first outcome tallies with the expectations of brain-dominance theory that learning programming requires high capacity for logical thought and a preference for this mode of thinking. Students with artistic abilities, in contrast, find it significantly more difficult to understand many concepts in JAVA language than students with logical abilities and those who have both artistic and logical abilities. The second outcome, based on the results of this survey, is that kinaesthetic and visual ways of teaching are likely to help students achieve better understanding of JAVA programming at introductory level, and this may impact on the amount of actual guided practice tasks utilised, as opposed to lecture and reading material. It also revealed that the use of the structure diagram as a logic depiction method could help students understand JAVA programming logic, and this may obviously have a strong bearing on the manner of presenting information visually. These findings would ideally lead to further studies in designing kinaesthetic and visual teaching tools to improve teaching JAVA concepts at an introductory level.

REFERENCES

ABC Science. (2008). *Left brain, right brain.* Retrieved October 25, 2009, from http://www.abc.net.au/science/features/brain/

Carland, J. A., & Carland, J. W. (1990). Cognitive styles and the education of computer information systems students. *Journal of Research on Computing in Education, 23*(1), 114–127.

Carlisle, M. C. (2004) . RAPTOR: A visual programming environment for teaching object-oriented programming. *Journal of Computing Sciences in Colleges, 19*(4), 275-281.

de Raadt, M., Hamilton, M., Lister, R., Tutty, J., Baker,B., Box, I., et al. (2005). Approaches to learning in computer programming students and their effect on success. *Higher Education in a Changing World: Research and Development in Higher Education*, *2*, 407-414.

Dew, J. R. (1996, April) Are you a right-brain or left-brain thinker? *Quality Progress Magazine*, 91-93.

Dillashaw, F., & Bell, S. (1985). Learning outcomes of computer programming instruction for middle-grades students: A pilot study. *Proceedings of the 58th Annual Meeting of the National Association for Research in Science Technology*. Indiana, IN.

Felder, R. M., & Silverman, L. K. (1988). Learning and teaching styles in engineering education. *Engr. Education, 78*(7), 674-681.

Garner, S., Haden, P., & Robins, A. (2005). *My program is correct but it doesn't run: A preliminary investigation of novice programmers' problems*. Paper presented at the Australasian Computing Education Conference.

Gorman, L. S. (1986). Cognitive style, personality type, and learning ability as factors in predicting the success of the beginning programming student. *ACM SIGCSE Bulletin, 18*(4), 80-89.

Liu, Y. & Ginther, D. (2005). Cognitive styles and distance education. *Online Journal of Distance Learning Administration*. Retrieved September 1, 2009, from http://www.westga.edu/~distance/liu23.html

Pessis, R. (2007). *Are you artistic or logic – maybe both.* Retrieved October 25, 2009, from http://www.allmyfaves.com/blog/weekly-faves/are-you-artistic-or-logic-maybe-both/

Prasad, C. & Fielden, K. (2002). Introducing programming: A balanced approach. *proceedings of the 15th Annual Conference of the National Advisory Committee on Computing Qualifications*. Hamilton, New Zealand, 101--107.

Robins, A., Rountree, J. & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education 13*(2), 137- 172.

Renumol, V. G., Jayaprakash, S., & Janakiram, D. (2009). *Classification of cognitive difficulties of students to learn computer programming.* Retrieved September 1, 2009, from http://dos.iitm.ac.in/publications/LabPapers/techRep2009-01

TheFreeDictionary. (2009). *Cognitive style*. Retrieved November 15, 2009, from http://encyclopedia.thefreedictionary.com/Cognitive+styles.

Thomas, L. A. (2000). The implication of different learning styles on the modelling of object-oriented systems. *Proceedings of the 22nd International Conference on Software Engineering* (ICSE'00). Limerick, Ireland.

Zander, C.,Thomas, L., Simon, B., Murphy, L., McCauley, R., Hanks, B. et al. (2009). Learning styles: Novices decide. *ACM SIGCSE Bulletin, 41*(3), 223-227.