DIFFICULTIES IN LEARNING PROGRAMMING: VIEWS OF STUDENTS

SitiRosminah MD Derus[1], Ahmad ZamzuriMohamad Ali[2]

Faculty of Art, Computing and Creative Industry
UniversitiPendidikan Sultan Idris
35900, TanjongMalim, Perak
Malaysia

[1]ctrosminah@gmail.com, [2]zamzuri@fskik.upsi.edu.my

Abstract

Programming courses areamong the important components of the curriculum to be studied, not only in the field of Information Technology, but also in most of the field including Science, Mathematics, and Engineering at tertiary levels. However, the subject is considered difficult, complex and categorized as part of the seven grand challenges in computing education. In this research, a study was conducted to investigate and explore the views of students and the difficulties they faced in learning fundamental programming courses. The study involved 105 Polytechnic students inthe Electrical Engineering Department, who undergo the Fundamental Programming course. The results from statistical analysis shows that the level of students‘ understanding of the topic is moderate and the most difficult topic for them is understanding the abstract concepts involving the role of variable position in computer memory (such as multi-dimensional array, looping statement and function) when the program was executed. Also, students faced difficulties in understanding the basic concept of programming structure and designing a program to solve certain tasks. In addition, majority of students agreed that practical situation such as laboratory activities was very helpful for them to learn programming. However, it was identified that lecturers not providing enough examples was one of the major factorinfluencing students‘ understanding of programming. Finally, the authors proposed visualization tool as alternative to learn programming in practical manner, and most of the students agreed the function of this tool in facilitating them learning programming.

Keywords: *programming, programming difficulties*

1. Introduction

With the widespread development of computer technology in the era of open world and globalization, the needs for the use of computers are increasing. This requires the need of experts in the field of Information and Communications Technology (ICT) such as software programming, database, software engineering, computer networking and creative multimedia [1]. In fact, the research findings shows that the industrys‘ needs for software engineers is high and this indicates that the programming personnel are a critical requirement for the industries[2]. Therefore to meet industrys‘needs, programming course is an essential component of the curriculum to be studied, not only in the field of Information Technology, but also required in most program areas such as Science, Mathematics, and Engineering at tertiary levels [3]. As a leader in the field of ‗Technical and Vocational Education and Training‘ (TVET), polytechnic was also involved in this challenge. Thus, the engineering programs offered are not only engineering-based curriculum alone, in fact, the subject of ICT also inserted, for instance the subject of Fundamental Programming (EC201) need to be studied under the programs offered in the field of Electrical Engineering.

However, the question arises, were the students were really being prepared to meet the critical needs of industry expertise in programming? Basically, the subject is said to be difficult, complex and categorized as one of the seven grand challenges in computing education [4]. Students especially novices, have problems in reading, tracking, writing and designing a simple code fragment. This causes students to have the attitude of depending on others to complete the task given and consequently become indolent and do not have the courage to learn and just expect marks of sympathy from lecturer [5]. As a result, the level of programming skills development is not satisfactory

2. Research Background

Many studies have been reported regarding the phenomenon of students struggling with the first steps of learning to program. A multi-national, multi-institutional study was conducted to explore and investigate students‘ programming performance and skill (see [3], [6], [7]). The study revealed that students have problems in writing [3], reading [6] as well as designing [7] a simple programming code. Therefore, it is important to investigate the factors that lead to students learning difficulty in programming. The difficulties need to be identified in order to be able to aid them learning in an effective manner.

A survey has been conducted by [8] and [9] to identify the factors that lead to students learning difficulty in programming with a various dimensions such as: computing background, computing experience, difficulties while learning programming and factors that lead to poor performance in programming course. The main finding revealed from the survey is that the most difficult topics faced by students were memory-related concepts which students were not capable in creating a clear mental model of memory movement during program execution. Moreover, the problem worsens when static media were employed in teaching dynamic concepts of programming [10]. Meanwhile, [11] had identified four major sources of problem faced by students in learning programming. The problems are: (i) students lacked skill in the techniques of analyzing problems, (ii) ineffective use of representation techniques for problem solving (iii) ineffective use of teaching strategies for problem solving and coding and (iv) do not understand and master the programming syntax and constructs.

According to [12], an effective way in learning to program need students take a real world problem and then translated it to program code to solve it. Although such approach is considered hard by students because the process of learning programming is to be said a multi-layers skill which is boring, intimidating and unrelated to day-to-day experience where students only learned in single context [13]. This happened to a novice student with no programming experience where at the initial stage they need to learn particular programming language syntax before applying their skills in structuring and programming style. Hence, it will be the resultof negative programming habits that may affect the flexibility of learning another language in a different context [8]. In fact, when the students limited only to basic programming syntax, it caused them applying programming approach from "line to line" rather than with a more meaningful or structured approach [14]. These factors caused by a relatively novice student‘s is limited at surface knowledge that lead them cannot capture more detail mental model of problem solving involving the programming process. Visualization tools have potential to create a suitable learning environment to help novices construct a viable mental model of programming concepts [15]. Since students cannot ‗see‘ what is happening ‗in‘ a computer when a program is executed, a teaching aid to assist them in visualizing the programming process dynamically should be developed to enhance student's conceptual understanding in programming environment.

3. Research Objectives

This study aims to explore the views of students and the difficulties they experience while learning Fundamental Programming courses. In particular, the objectives of the study are as follows:

1. To know the level of understanding of different topics of the Fundamental Programming course.
2. To investigate the nature of difficulties while learning Fundamental Programming course.
3. To get an overview the situation that would help to learn Fundamental Programming course.
4. To investigate the factors that lead to perform poorly in Fundamental Programming course.
5. To know the level of agreement if uses Visualization tools to visualize the program execution will help in learning Fundamental Programming course.

4. Methodology

The study employs a survey questionnaire which was adapted from existing instruments of the [8] and [9]. The items consist of two main parts covering the general background information from a respondent, including programming experience and respondent experience while learning basic programming courses (Table 1).

Table 1. Reference of items in questionnaire

Item	Description	References
Part A:		
1-3	Respondent‘s general information	-
4-5	Programming experience	-
Part B :		
6	Level of understanding of different topics of the programming course	Fundamental Programming Course Synopsis
7	The nature of difficulties while learning programming courses	[8]&[9]
8	The situations that would help to learn programming courses effectively:	[8]&[9]
9	The factors that lead the students perform poorly in basic programming courses.	[8]
10	Respondents' views on the use of visualization tools while learning basic programming courses	[8]

Respondents were about 105 second semester students from three programs in Electrical Engineering Department at Polytechnic (Table 2).

Table 2. Number and percent distribution of respondent according to program

Code	Program	Num	%
DEU	Diploma in Electronic Engineering (Medical)	31	29.5
DJK	Diploma in Electronic Engineering (Control)	17	16.2
DEP	Diploma in Electronic Engineering (Communication)	57	54.3
Total		**105**	**100.0**

5. Results and Discussion

Of the 105 respondents who answered the questionnaires, 49.5% (n = 52) are male and 50.5% (n = 53) are females. While the majority of respondents are Malay students is 88.6% (n = 93), followed by 5.7% (n = 6) students are Indian, 4.6% (n = 5) students are Chinese and 1% (n = 1) is indigenous student from Sabah/Sarawak. 97.1% (n =92)ofrespondentshad no experiencewithcomputer programming. This shows the Fundamental Programming courses (EC201) is their first experience in the programming development environment.

Table 3 summarizes the level of students‘ understanding of the topic of Fundamental Programming courses. The item shown in Table 3 is designed with 5-point Likert scale (1, not understand at all; 2, not understand; 3, sometimes understand; 4, understand, and 5, very understand). The sequence of topics in the table is listed from the lowest mean value. Overall, the level of students‘ understanding of basic programming topic is at a moderate level. Multidimensional Array, Looping Statements, Function and Array Data Structure are the difficult topics to be studied by students (3.10 – 3.30). These topics which require considerable skill of students to understand the abstract concept involving the position of the variable values. According to [9], the reason‘s students having problems in learning programming is they are less familiar with the role of a variable position in the computer memory when the program code executed such as how the variables are stored and the relationship between the variables in the computer memory. Therefore, using the visualization tools to depict the position of the variables that occur when the program code executed may help the student understand the role of variables in computer memory [16]

Table 3. Level of students‘ understanding on different topic on programming course

Topics	Mean	SD
Multidimensional Array	3.10	0.894
Looping Statements (e.g : while , do-while, goto)	3.19	0.833
Function	3.28	0.925
Array Data Structure	3.30	0.856
Variables, constants and data types	3.50	0.822
Selection statements (e.g : if-else, switch)	3.51	0.735
Input/Output statements (e.g : printf , scanf)	3.71	0.743

Onward, students‘ needs to respond to the difficulties faced by them while learning fundamental programming course where they need to evaluate the level of agreement with the questionnaire item given. 5-point Likert scale (1, Strongly Disagree; 2, Disagree; 3, Neutral; 4, Agree; 5, Strongly Disagree) is used for the items shown in Table 4. The sequence of topics in the table is listed from the highest mean value

Table 4. Difficulty while learning programming

Nature of Difficulties	Mean	SD
Understanding basic concepts of programming structure	3.40	0.884
Designing a program to solve certain tasks	3.36	0.952
Learning the programming language syntax	3.35	0.980
Gaining access to a computer	3.33	0.916
Using program development environment	3.22	0.930
Finding bugs from my own program	3.19	1.066

The findings show that there are three interrelated types of students‘ nature of difficulties while learning programming: i) difficulty in understanding the basic concepts of programming structure (3.40); ii) a program designed to complete a particular task (3:36) and iii) learn the syntax of programming languages (3:35). This problem often faced by novices whom they are not problematic to the basic understanding of programming concepts, but rather to apply the abstract concepts of programming with the programming structure [17], as well as additional skills such as management training, editing, compiling and debugging program code [18]. This is because the programming skills involving more than a different process in which at an early stage, the problems faced by the programmer needs to be translated into a form of algorithms, which are then translated on to the program code [19].

Table 5 provides a result of a situation knowing that could help students in programming courses. The questionnaire item show using 5-point Likert scale similar as in Table 4. The majority of students agreed that the practical or laboratory activities (3.99) could help them to learn fundamental programming effectively as well as to holding discussions together with lecturers or friends (3.82). Learning programming need to involve practical activities and intensive training because of its dynamic concept [20]. To implement programming in the lab will help the student address the understanding of difficult terminology in programming as well as stimulate students' interest in the field of programming through the activities carried out [21]. However, laboratory activities should be implemented through active learning strategies that encourage students to actively engage in the activity by applying the knowledge gained in an effort to improve knowledge of programming [22], [23].

Table 5. Situations that would help to learn programming

Situations	Mean	SD
In practical or lab sessions	3.99	0.893
Consultation or discussion with lectures, seniors or friend	3.82	0.918
In small group exercise sessions	3.74	0.844
In lectures	3.73	0.891
While working alone on programming coursework	3.39	0.995

Furthermore, Table 6 shows the student level of agreement of the factors that made them perform poorly while learning basic programming course. The questionnaire item show using 5-point Likert scale also similar as in Table 4. The findings indicate that the student is difficult to understand the process of learning programming is due to the lecturers were not provided enough example to the students (3.38), as well as computer equipment in the lab do not work well (3.32) and the method of teaching by the lecturers to be less effective (3.30). Instructional technique and strategies are vital in the delivery of educational information. In order to help students to master the basic problem solving skill, the teaching method used should be applicable to the content of programming with different paradigms and to make sure the active involvement of students in programming practical session [11].

Table 6. Factors that lead to poor performance in programming

Factors	Mean	SD
Less examples of practical use are shown	3.38	1.032
Computers provided in labs are not functioning well	3.32	1.157
Teaching methodology is less effective	3.30	1.099
Presentation of instructors and their attention on students	3.25	1.099
Students' lack of interest to learn	3.22	0.961
Syllabus focuses too much on theory	3.13	1.048
Syllabus coverage per semester is too wide	3.11	0.934
Learning environment that is not conducive	3.04	0.990

About 72% (n=76) students agreed on the use of visualization tools for visual display in the program will help students learn basic programming courses. One of the reason students learns to program is difficult because of their inability to visualize the execution process of the program code [2], [9]. As a result, they cannot figure out the problem occurred when things do not work. Therefore, effective learner support mechanisms should be developed to provide the students to learn program with the optimal learning environment that they need. A visualization tool can be used as learning aids to help students learn programming and understand the important elements related to programming behavior during program execution [24]. In addition, the visualization tool can be used as a learning model to help students linking new information with old knowledge [25].

6. CONCLUSION

Being successful in programming is being able to transform abstract problems into program code as a working solution on the computer. For novices, to succeed in programming, they need a set of learning strategies to help them to cope with the process of programming as well as to influence the way the novice perceives the programming process. In fact, lacking to visualize the program state during code execution is one of the factors of students' difficulty in learning programming. Therefore, visualization tools as learning aids is appropriate to be developed to help students understand the implementation of the program code, such as changes occur at the position of the variable in the computer memory. However, students should be actively involved with the visualization tool to ensure that it can have a positive impact on students' understanding of information to be delivered. In addition, the use of this visualization tool can be used during practical activities in the laboratory through group activity so that students can discuss and explore more actively the activities undertaken.

REFERENCES

[1] A. Ungku HarunAl'Rashid, ‒Meeting the demands of global firms: Survey Finding,‖ 2004. [Online]. Available: http://www.epu.gov.my/html/themes/epu/images/common/pdf/seminars/slide-ungku harun.pdf.

[2] Ahmad Rizal Madar, Nurliana Musa, and YahyaBuntat, ‒Kesan model pembelajaranberasaskankaedahpenyelesaianmasalah keataspelajarberbezagayakognitifdankemahiranlogik,‖(*The effect of using learning model based on problem solving method on students with different cognitive style and logic ability)* in *2nd International Malaysian Educational Technology Convention*, 2007.

[3] M. McCracken et. al, ‒A multi-national , multi-institutional study of assessment of programming skills of first-year CS students,‖ *ACM SIGCSE Bulletin*, vol. 33, no. 4, pp. 125–180, 2001.

[4] A. McGetrick, R. Borle, R. Ibbett, J. Llyod, G. Lovegrove, and K. Mander, ‒Grand challenges in computing: Education-A Summary,‖ *The Computer Journal*, vol. 48, no. 1, pp. 42–48, Jan. 2005.

[5] Aziz Deraman, ‒Kilangperisianpengajaran : Satupendekatanuntukpeningkatankemahiranpengaturcaraan,‖*(Teaching software factory: An approach to increasing programming skills)* in *Proceeding of Science Programming Workshop: Teaching and Learning in Malaysia* (*ATUR'03)*, 2003, pp. 1–7.

[6] R. Lister et. al, ‒A multi-national study of reading and tracing skills in novice programmers,‖ in *ITiCSE-WGR '04 Working group reports from ITiCSE on innovation and technology in Computer Science education*, 2004, pp. 119 – 150.

[7] J. Tenenberg, S. Fincher, K. Blaha, D. Bouvier, T. Chen, D. Chinn, S. Cooper, A. Eckerdal, H. Johnson, R. Mccartney, A. Monge, J. E. Moström, M. Petre, K. Powers, M. Ratcliffe, A. Robins, D. Sanders, L. Schwartzman, B. Simon, C. Stoker, A. E. Tew, and T. Vandegrift, ‒Students Designing Software : a Multi-National , Multi-Institutional Study,‖ *Informatics in Education*, vol. 4, no. 1, pp. 143–162, 2005.

[8] T. Phit-Huan, T. Choo-Yee, and L. Siew-Woei, ‒Learning difficulties in programming courses: Undergraduates' perspective and perception,‖ *2009 International Conference on Computer Technology and Development*, vol. 2, pp. 42–46, 2009.

[9] I. Milne and G. Rowe, ‒Difficulties in learning and teaching programming — Views of students and tutors,‖ *Education and Information Technologies*, vol. 7, no. 1, pp. 55–66, 2002.

[10] T. Linden and R. Lederman, ‒Creating visualizations from multimedia building blocks : A simple approach to teaching programming concepts,‖ in *Information Systems Educators Conference - ISECON 2011*, 2011, pp. 1–10.

[11] MohdNasir Ismail, Nor AzilahNgah, and IrfanNaufal Umar, ‒Instructional strategy in the teaching of computer programming : A need assessment analyses,‖ *TOJET*, vol. 9, no. 2, pp. 125–131, 2010.

[12] A. Rudder, M. Bernard, and ; Shareeda Mohammed,, ‒Teaching programming using visualization,‖ in *WBED'07 Proceedings of the sixth conference on IASTED International Conference Web-Based Education - Volume 2,* 2007, pp. 487–492.

[13] R. Moser, ‒A fantasy adventure game as a learning environment: Why learning to program is so difficult and what can be done about it .,‖ in *ITiCSE '97 Proceedings of the 2nd conference on Integrating technology into computer science education*, 1997, pp. 114–116.

[14] L. E. Winslow, ‒Programming pedagogy- A psychological overview,‖ *ACM SIGCSE Bulletin*, vol. 28, no. 3, pp. 17–22, Sep. 1996.

[15] M. Ben-Ari, ‒Program visualization in theory and practice,‖ *Informatique*, no. 2, 2001.

[16] J. Sajaniemi and M. Kuittinen, ‒Program animation based on the roles of variables,‖ *Proceedings of the 2003 ACM symposium on Software visualization - SoftVis '03*, p. 7, 2003.

[17] E. Lahtinen, K. Ala-Mutka, and H.-M. Järvinen, ‒A study of the difficulties of novice programmers,‖ *ACM SIGCSE Bulletin*, vol. 37, no. 3, p. 14, Sep. 2005.

[18] I. T. C. Mow, ‒Issues and difficulties in teaching novice computer programming,‖ in *Innovative Techniques Technology, E-learning, E-assessment and Education*, 2008, pp. 199–204.

[19] T. Jenkins, ‒On the difficulty of learning to program,‖ in *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences*, 2002, vol. 4, pp. 53–58.

[20] A. Gomes and A. J. Mendes, ‒Learning to program-difficulties and solutions,‖ in *International Conference on Engineering*, 2007.

[21] J. R. Parham, ‒An assessment and evaluation of Computer Science education,‖ *Journal of Computing Sciences in Colleges*, vol. 19, no. 12, pp. 115–127, 2003.

[22] M. Fetaji, S. Loskovska, B. Fetaji, and M. Ebibi, ‒Combining virtual learning environment and integrated development environment to enhance e-learning,‖ in *Proceedings of the ITI 2007 29th Int. Conf. on Information Technology Interfaces*, 2007, pp. 319–324.

[23] T. L. Naps et al., ‒Exploring the role of visualization and engagement in computer science education,‖ in *ITiCSE-WGR '02 Working group reports from ITiCSE on Innovation and technology in computer science education*, 2002, pp. 131–152.

[24] T. Rajala, M.-J. Laaksi, E. Kaila, and T. Salakoski, ‒Effectiveness of Program Visualization: A case study with the ViLLE tool,‖ *Journal of Information Technology Education: Innovatins in Practice*, vol. 7, pp. 15–32, 2008.

[25] A. Hyrskykari, ‒Development of program visualization systems,‖ in *2nd Czech British Symposium of Visual of Man-Machine Systems*, 1993, pp. 1–21.