Asian Social Science; Vol. 14, No. 11; 2018
ISSN 1911-2017 E-ISSN 1911-2025
Published by Canadian Center of Science and Education

Programming Skills and the Relation in Fostering Students' Higher Order Thinking

Ibrahim Abdullah Alshaye[1], Nurul Farhana Jumaat[1] & Zaidatun Tasir[1]

[1] Department of Educational Sciences, Mathematics and Creative Multimedia, School of Education, Faculty of Social Sciences and Humanities, Universiti Teknologi Malaysia, Skudai, Johor, Malaysia

Correspondence: Ibrahim Abdullah Alshaye, Department of Educational Sciences, Mathematics and Creative Multimedia, School of Education, Faculty of Social Sciences and Humanities, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia. Tel: 966-55-322-6635. E-mail: ib.shaye@gmail.com

Received: August 1, 2018 Accepted: September 1, 2018 Online Published: October 22, 2018

doi:10.5539/ass.v14n11p76 URL: https://doi.org/10.5539/ass.v14n11p76

Abstract

Programming skills (PS) refer to coding and debugging that are required for those who write a program in any programming language. Coding can be described as the implementation aspect of programming, whereas debugging can broadly be defined as fixing any incorrect code that is found after running the programming test. Higher-order thinking skills (HOTs) refer to the top three levels of Bloom's taxonomy which are Analysis, Synthesis, and Evaluation. This study aims to determine the relationship between PS and HOTs among secondary students. Many studies indicate that students who attend programming courses for first time have low levels of performance in PS. Coding and debugging skills reflect higher-order thinking levels. Therefore, an objective of this study was to investigate the effect of coding and debugging skills on their HOTs. The benefits of having PS among learners are that they are able to achieve HOTs. Indeed, these relationships may be explained by programmers need to apply all these HOTs throughout the three phases of the programming process. Students who have low levels of PS are able to achieve the analysis level, while students who have moderate levels of PS able to achieve the synthesis level, and finally, students with high levels of PS are able to achieve the evaluation level.

Keywords: programming skills, coding, debugging, problem-solving, higher-order thinking

1. Introduction

Learning programming skills is not an easy task for those who are attempting to learn programming for first time, as supported by many research studies (Mhashi & Alakeel, 2013). Novice students lack the knowledge and skills needed for programming, such as coding and debugging; these skills are essential to computer-related fields of study (Jawdah, 2015; McCracken et al., 2001; Mhashi & Alakeel, 2013; Taha, 2015). Most freshmen students taking programming courses face difficulties due to a lack of ability to analyse problems (Gondim, Ambrósio, & Costa, 2011; Ismail, 2005). The benefits of having programming skills among learners are that they are able to program and develop computer apps, as well as to achieve higher levels of thinking (Fessakis, Gouli, & Mavroudi, 2013).

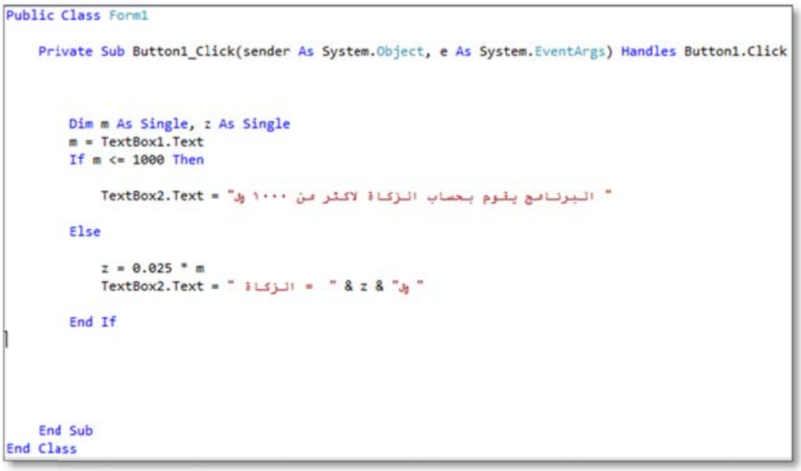
Many researches have been conducted for the purpose of investigating why novice students have difficulty when learning to program, especially in understanding programming mechanics, writing and debugging code, recognising the types of variable, and applying of programming techniques to different situations and problems (Alastal, 2009; Aljaberi, 2005; Alturki, 2016; Carbone, Hurst, Mitchell, & Gunstone, 2001; Eid, 2009; Ismail, 2005; Jawdah, 2015; McCracken et al., 2001; Mhashi & Alakeel, 2013; Mohammed, 2015; Soloway, Ehrlich, Bonar, & Greenspan, 1983; Taha, 2015; Thomas, Ratcliffe, Woodbury, & Jarman, 2002). For example, an early study by McCracken et al. (2001) showed that programming skill is a predictable outcome of a student's teaching in computer science, yet and many computer teachers report that novice students lack such skills. A universal anticipation is that novice students have to understand solving problems process in the field of computer science. From a set of related program-writing tasks, each student was expected to write a particular program. Due to the weak problem-solving capacity of most of the students, their performance in executing the task was poor and many were not able to even get close to the completion of the assigned task (Lopez, Whalley, Robbins, & Lister, 2008; McCracken et al., 2001).

Fessakis et al. (2013) stated that programming requires higher order thinking skills (HOTs). Programming environments assist learners to organize, analyse, and evaluate their ideas in a clear way during the process of problem-solving. The present study seeks to determine the relationship between PS and HOTs among secondary students.

2. Literature Review

2.1 Coding and Debugging Skills

Every person has thoughts, but a little can make them take place. The ability to write code divides persons who have thoughts from those who can make their thoughts into a truth. We should encourage our student to learn programming, because we want them to be a creator and thinker who can bring thought to life. Coding is usually used as a synonym for programming, when the skill to write code is needed. It can be described as the application of programming. Furthermore, Coding gives students confidence that they can be builders and designers. Duncan, Bell, and Tanimoto (2014) stated that the term ‘coding’ is widespread, thus it would be useful to use it instead of ‘programming’. Figure 1 shows a sample of source code for the Zakat Al Mal Calculator project written in Visual Basic (VB).



```
Public Class Form1

    Private Sub Button1_Click(sender As System.Object, e As System.EventArgs) Handles Button1.Click


        Dim m As Single, z As Single
        m = TextBox1.Text
        If m <= 1000 Then

            TextBox2.Text = " البرنامج يقوم بحساب الزكاة لاكثر من ١٠٠٠ رل"

        Else

            z = 0.025 * m
            TextBox2.Text = " الزكاة =  " & z & "رل "

        End If



    End Sub
End Class
```

Figure 1. Source code of Zakat Al Mal Project

Source: Computer and Information Technology 1 curricular (MOE, 2017).

In turn, debugging is fixing any incorrect code that is found after running the programming test. Thereby, it is necessary to debug the code in any application to verify its runtime behaviour and to fix any issues that occur (Satheesh & Subashni, 2013).

Hiltunen (2016) has shown that teaching programming is beneficial and possible to novice students. According to Duncan, Bell, and Tanimoto (2014), there are challenges of teaching programming to novice students, for instance, in the context, culture, teacher training, learning tools, confidence, skills and engagement level of learning environment. Programming can help students to analyse and assess their ideas during the time of solving a problem (Fessakis et al., 2013; Hiltunen, 2016). Learning to program (coding) help novice students make sense of explore ideas and make things, for work and play. Furthermore, it helps them to work collaboratively and creativity (European Comission, 2015).

2.2 Students’ Performance in Coding and Debugging Skills

Nowadays, learning how to program will be beneficial for students due to a large number of software and devices around us in the world. It will be help them to use these devices and software flexibly. In future, it appear that the need for programmers will increase because of the fast growth of computing in community (Boyer, Langevin, & Gaspar, 2008; Duncan et al., 2014; Hiltunen, 2016). Novice students spend little time in testing code, then attempt to settle their code with small fixes; students also have difficulties with various issues related to program construction (Dijkstra, 1989; Mhashi & Alakeel, 2013). Whereas, students can learn programming with the correct way and obtain the ability to write their own programs, when the skills in logical thinking are at a high level. This will possibly lead to boosted potentiality in employment market. (Hiltunen, 2016).

In Saudi Arabia, many studies conducted to investigate the performance of programming among secondary school students, the results of these studies revealed that generally students suffer low levels of programming skills, while the performance is dissimilar between theoretical and practical parts. Students have the ability to follow and understand the teacher while he or she is explaining programming lessons in theory. However, they face difficulties while performing practical exercises, such as being unable to write program code according to problem solving steps (Alamri, 2014; Alferm, 2011; Alghamdi, 2014; Alhabashi, 2015; Alsheikhi, 2012). Figure 2 illustrate the students' gap between performance in theoretical and practical parts. As example, Alsheikhi (2012) conducted a study on the effectiveness of an educational software program proposal to provide Saudi secondary school students with programming skills in VB. He observed that during the course of his studies, it was very difficult for the students to understand programming instructions and write computer programs, no matter how easy they were. The students at this stage had the ability to follow the teacher while explaining the computer lessons and to understand the lessons, but when they were doing programming exercises in laboratory, these students were unable to write or even debug code, they suffered low levels of problem-solving ability. Therefore, the aim of this paper was to investigate the effect of coding and debugging skills on higher-order thinking skills (HOTs) among novice students.

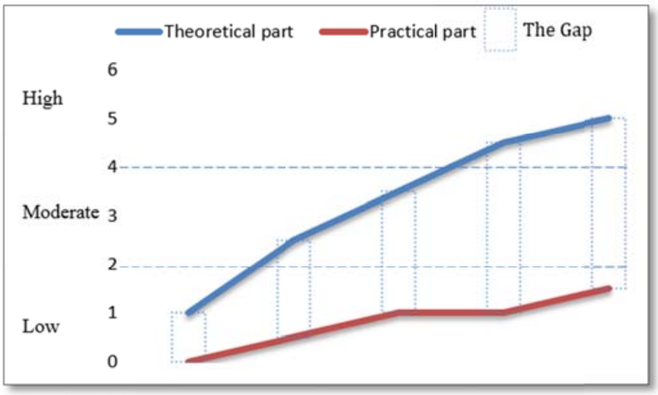


Figure 2. The students' gap between performance in the theoretical part and practical part in Computer and Information Technology 1 (CIT1) curricular in Saudi Arabia

As a computer teacher for the Ministry of Education in Saudi Arabia, the researcher has found that the Computer and Information Technology 1 (CIT1) curriculum is based on several basic skills. Such as, problem solving programming by (VB) is one of the most important basic skills in stage one of secondary school in Saudi Arabia. Problem solving contains several steps: Analysis of problem elements, write an algorithm, draw flowcharts, writing a program by use a programming language (coding), interpreting the program into assembly language, and testing the program and fixing its errors (debugging) (MOE, 2017). These steps require skill enhancement in both theory and practice. Table 1 illustrate the syllabus of CIT1 curricular in Saudi Arabia.

Table 1. The syllabus of CIT1 curricular

Module 1	Introduction to Programming	The concept of programming The importance of programming
Module 2	Solving Programming Problem	The goal of learning Solving Programming Problem Steps to solve programming problem Problem identification Writing an algorithm Draw a flowchart
Module 3	Visual Basic	The stages of writing the program by VB Data types in VB Calculations and Logic VB Programming tools Procedures and Functions

Source: (MOE, 2017)

The Ministry of Education (MOE) in Saudi Arabia reported that Computer and Information Technology 1 (CIT1) is a compulsory course for all majors in secondary school that is taught in the first year. CIT1 involves both theoretical and practical skills. The aim of CIT1 is to enable students to master the basic methods and design of programming and cultivate their ability to analyse and work through problem solving steps by computer (MOE, 2017). Table 2 illustrates solving programming problem steps; these steps must be followed one by one to accomplish the end goal.

Table 2. Solving Programming Problem Steps that stated by Ministry of Education (2017).

Steps No.	Solving Programming Problem Steps
1st	Problem identification by finding out the type of input data, and the processing and output requirement.
2nd	Writing an algorithm: a set of steps which can be used to handle and solve the problem.
3rd	Drawing a flowchart.
4th	Writing a computer program (Coding) by writing down the steps from the algorithm in computer programming form.
5th	Translating the computer program code to machine language using a compiler, as this is the only language that is directly understood by the computer.
6th	Testing and Debugging. Locating and correcting any errors (bugs) and ensuring that all data are processed properly.

2.3 The Relationship between Students' Higher-Order Thinking Skills and Their Performance in Coding and Debugging Skills

Programming constitutes an important competence for learning and for the development of algorithmic problem solving skills, as well as HOTs (Fessakis et al., 2013). HOTs allow students to analyse and create new bodies of knowledge from current knowledge that used today, and then to evaluate this lately knowledge. (Prayaga & Coffey, 2008). Fessakis et al. (2013) highlight that learning programming skills can be beneficial to the enhancement of HOTs generally. Programming language instruction is currently used as an instructional environment to promote higher-order thinking skills (Palumbo, 1990). From a pedagogical view, it is considered valuable in the enhancement of HOTs over the development of algorithmic problem-solving skills (DiSessa, 2000; Papert, 1991).

Bloom's taxonomy identifies six separate levels of knowledge within the cognitive domain, namely knowledge, comprehension, application, analysis, synthesis, and evaluation (Refer to Figure 3). This paper focuses on the higher-order thinking skills (HOTs), which include analysis, synthesis, and evaluation.

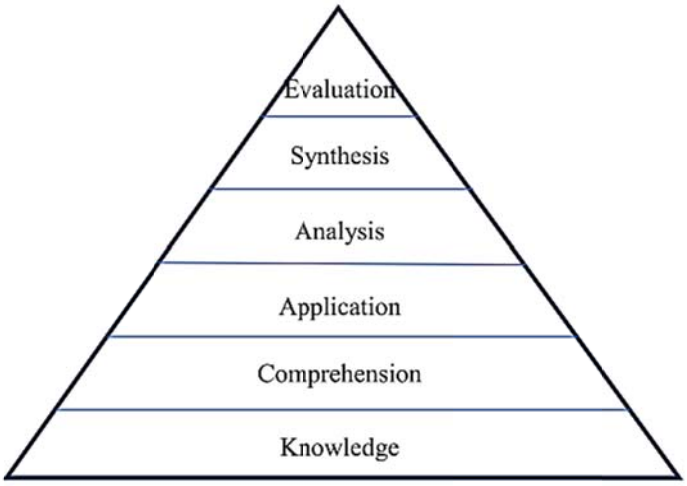


Figure 3. Bloom's Taxonomy (Bloom, Engelhart, Furst, Hill, & Krathwohl, 1956)

Anderson and Krathwohl (2001) revised Bloom's Taxonomy as follows: remembering, understanding, applying, Analysing, evaluating, and creating. The difference between the original and revised versions of the taxonomy is that evaluation is no longer the highest level. A new category called 'creating' is at the top. The taxonomy is also reworded from nouns to verbs. However, the researcher in this paper was adopted the original version of Bloom's Taxonomy (1956), as students on programming courses need to synthesize and implement code before they can evaluate it.

3. Recommendation

Programming skills are vital for anyone studying programming courses. In this paper, programming skills refer to coding and debugging. Learners are required to have these skills to write a program (coding) in any programming language, which will subsequently be processed by a computer compiler. The compiler is a program which translates a program written in one programming language to an equivalent program in a machine language that a computer's processor understands. Thereby, the debugger is used to test program code. The debugger is a computer program that is used to find bugs/errors in coding. It also allows the execution of the program to be stopped, and can help the programmer to ensure that the program is behaving correctly and to help in identifying bugs in the code. In turn, the programmer must fix any wrong code that is found after running the programming test (debugging).

Problem-solving requires the use of HOTs such as analysis, synthesis and evaluation. The role of the programmer is to carry out the coding process and perform testing and maintenance procedures. During this process, the programmer needs to apply all HOTs among the three phases. In the first phase, when we give students a programming problem, they need to understand and analyse the problem: that is, to determine the input data, processing, and outputs. Then, in the second phase, students have to synthesize the solution (coding) and implement the code based on the flowchart. In the last phase, they need to evaluate the code. The code must be verified and validated (debugging). After the debugger has tested the program code, then he or she needs to correct any errors (Prayaga & Coffey, 2008). The researcher recommended that to take into account improve HOTs among secondary school students in order to enhance programming skills which includes coding and debugging skills.

4. Conclusion

HOTs are an important way to improve students' learning ability (Wang, 2011). Benjamin Bloom created a taxonomy that includes six different levels, namely Recall, Comprehension, Application, Analysis, Synthesis, and Evaluation (Bloom et al., 1956). These levels are observable actions that indicate that something is happening in the brain. In this paper, the programming skills (coding and debugging) reflect higher-order thinking levels, which are the top three levels of Bloom's taxonomy (analysis, synthesis, evaluation).

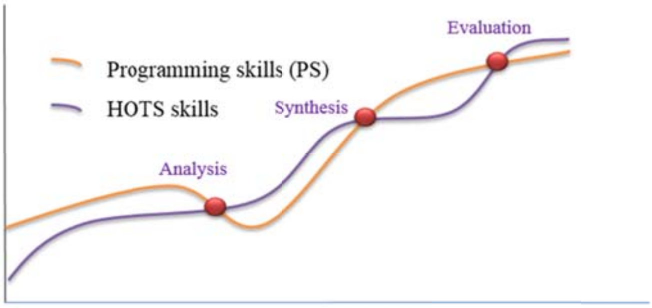


Figure 4. The relationship between PS and HOTs

In this sense, students who have low levels of programming skills are able to achieve the analysis level, while students who have moderate levels of programming skills able to achieve the synthesis level, and finally, students with high levels of programming skills are able to achieve the evaluation level (Refer to Figure 4).

Acknowledgments

The authors would like to thank Universiti Teknologi Malaysia (UTM) for the financial support under Research University Grant (RUG Tier 2) vot no. Q.J130000.2631.13J96.

References

Alamri, S. M. (2014). *The Influential Role of Electronic Support Techniques into Developing skills of problem-solving of Object-Oriented programming for Secondary Stage Female-Students.* Arab East Graduate Studies colleges.

Alastal, M. Z. (2009). *Enrich the programming unit in the information technology curriculum in the light of performance standards of programming, and then measure the impact of the proposed enrichment material on the level of the programming skills of the 11th secondary grades.* The Islamic University of Gaza.

Alferm, H. B. (2011). *Designing an Intelligent Tutor Software and Measuring It's Effect on Developing Basic Concepts in Major Programming and Attitude Towards It's Study for Third year Secondary Students in Riyadh City.* King Saud University.

Alghamdi, A. M. A. (2014). *The Effect of Interaction Between Instructional Control in Multimedia Software and Cognitive Style on Programming Performance for Secondary Stage Students.* Albaha University.

Alhabashi, N. F. (2015). *The effectiveness of the electronic unit based on learning strategy projects in the Acquisition of skills of programming by language Visual Basic for students of third secondary grade.* Arab East Graduate Studies colleges.

Aljaberi, N. M. R. (2005). *The impact of learning programming language on enhancing the ability in math modelling and problem solving of university students in Jordan.* Amman Arab University.

Alsheikhi, M. M. H. (2012). *The Effectiveness of an educational software program proposal to provide high school students with programming skills in Visual Basic. Net.* Umm Alqura.

Alturki, R. A. (2016). Measuring and Improving Student Performance in an Introductory Programming Course, *15*(2), 183-204. https://doi.org/10.15388/infedu.2016.10

Anderson, L., & Krathwohl, D. A. (2001). *Taxonomy for Learning, Teaching and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives.* New York: Longman.

Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H., & Krathwohl, D. R. (1956). *Axonomy of educational objectives: The classification of educational goals*. New York: David McKay Company.

Boyer, N., Langevin, S., & Gaspar, A. (2008). Self direction & constructivism in programming education. In *In Proceedings of the 9th ACM SIGITE conference on Information technology education* (pp. 89-94). ACM.

Carbone, A., Hurst, J., Mitchell, I., & Gunstone, D. (2001). Characteristics of programming exercises that lead to poor learning tendencies: Part II. *ACM SIGCSE Bulletin*, 93-96. https://doi.org/10.1145/377435.377494

Dijkstra, E. W. (1989). On the Cruelty of Really Teaching Computing Science. *ACM*, *32*, 1398-1404.

DiSessa, A. (2000). *Changing minds: Computers, learning, and literacy.* MIT Press.

Duncan, C., Bell, T., & Tanimoto, S. (2014). Should your 8-year-old learn coding? In *Proceedings of the 9th Workshop in Primary and Secondary Computing Education. ACM* (pp. 60-69).

Eid, K. S. (2009). *The effectiveness of developing Visual Basic tools for improving assessment design skills among students of applied science and education technology at the Islamic University.* The Islamic University of Gaza.

European Comission. (2015). *European Comission.* Retrieved June 10, 2017, from www.codeweek.eu/

Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5-6 years old kindergarten children in a computer programming environment: A case study. *Computers and Education*, *63*, 87-97. https://doi.org/10.1016/j.compedu.2012.11.016

Gondim, H. W. A. S., Ambrósio, A. P. L., & Costa, F. M. (2011). TaskBoard - Using XP to implement problem-based learning in an introductory programming course. *Lecture Notes in Business Information Processing*, *77 LNBIP*, 162-175. https://doi.org/10.1007/978-3-642-20677-1_12

Hiltunen, T. (2016). *Learning and Teaching Programming Skills in Finnish Primary Schools – The Potential of Games.*

Ismail, Z. M. A. (2005). *The effectiveness of a computer program in teaching Introduction of Programming by using a problem-solving method.* Ain Shams University.

Jawdah, A. M. Y. (2015). *Effect of using educational Fourm on developing some programming skills of the third year preparatory student in Experimental schools.*

Lopez, M., Whalley, J., Robbins, P., & Lister, R. (2008). Relationships between reading, tracing and writing skills in introductory programming. *Proceeding of the Fourth International Workshop on Computing Education Research - ICER, 8*, 101-112. https://doi.org/10.1145/1404520.1404531

McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y. B.-D., … Wilusz, T. (2001). A

Multi-national, Multi-institutional Study of Assessment of Programming Skills of First-year CS Students. *SIGCSE Bull*, *33*(4), 125-180. https://doi.org/10.1145/572139.572181

Mhashi, M. M., & Alakeel, A. L. I. M. (2013). Difficulties Facing Students in Learning Computer Programming Skills at Tabuk University. *Recent Advances in Modern Educational Technologies*, 15-24.

MOE. (2017). *Computer and Information Technology 2*. Riyadh: Obeikan.

Mohammed, M. M. S. (2015). The Effectiveness of Web Quests in the Development of Programming Skills among Students in the Third Grade Prep. *Journal of Faculty of Education, Benha University*.

Palumbo, D. B. (1990). Programming language/problem solving research: A review of relevant issues. *Review of Educational Research*, *60*(1), 65-89.

Papert, S. (1991). Mindstorms: Children, computers and powerful ideas. *Athens: Odysseas Publications (in Greek).*

Prayaga, L., & Coffey, J. W. (2008). Computer Game Development: An Instructional Strategy to Promote Higher Order Thinking Skills. *Journal of Educational Technology*.

Satheesh, N., & Subashni, S. (2013). *Software Testing using Visual Studio 2012*. MUMBAI: Packt Publishing Ltd.

Soloway, E., Ehrlich, K., Bonar, J. G., & Greenspan, J. (1983). What do novices know about programming? In *Directions in Human--Computer Interactions* (Vol. 6, pp. 27-54). Ablex Publishing, Norwood, NJ.

Taha, A. M. A. M. (2015). the effectiveness of the use of blended learning in the development of the skills of production projects Visual Basic NET Visual Basic. *Journal of College of Education, Banha University*, *26*(102).

Thomas, L., Ratcliffe, M., Woodbury, J., & Jarman, E. (2002). Learning styles and perfomance in the introductory programming sequence. *Acm*, 33-36. Retrieved from http://0-dl.acm.org.wam.seals.ac.za/citation.cfm?id=563352

Wang, S. (2011). The empirical study of the effect of information teachnology in improving pupil's higher order thinking skills--the case of teaching declarative knowledge. South China Normal University.

Copyrights