INTERNATIONAL JOURNAL OF ACADEMIC RESEARCH IN PROGRESSIVE EDUCATION & DEVELOPMENT

OPEN ACCESS JOURNAL

Vol 12, Issue 2, (2023) E-ISSN: 2226-6348

# Factors Affecting Engineering Students in Learning the Programming Subject at UiTM Pulau Pinang: A Study on Educators' Ability

Jamal Othman[1], Siti Balqis Mahlan[2], Rozita Kadar[3], Maisurah Shamsuddin[4] and Naemah Abdul Wahab[5]

[1,3,5]Computing Sciences Studies, College of Computing, Informatics and Media, Universiti Teknologi MARA Cawangan Pulau Pinang, Malaysia, [2,4]Mathematical Sciences Studies, College of Computing, Informatics and Media, Universiti Teknologi MARA Cawangan Pulau Pinang, Malaysia

**To Link this Article:** http://dx.doi.org/10.6007/IJARPED/v12-i2/17359 DOI:10.6007/IJARPED/v12-i2/17359

***Published Online:*** 19 May 2023

**Abstract**

Teaching the subject of computer programming, especially to engineering students at higher education institutions requires a comprehensive teaching approach, modifiable pedagogy and hybrid creativity to generate an attractive and realistic learning environment. This research was conducted demonstrating that the students are not attentive to learning the programming subject owing to several factors. This paper concentrates on the factors related to educators as the contributors to the students' attraction to learn the programming subject. This study was conducted at Universiti Teknologi MARA, Pulau Pinang Branch with a total of 241 students from the engineering school responding to the online survey. Through the mean and the standard deviation analysis, it was found that the educators' roles affected the students' attraction and understanding ability in learning the programming subject. This finding can help the educators to improvise and upgrade their teaching approach to make the computer programming class fascinating and enhance the students' learning curve.

**Keywords:** Educators, Teaching, Computer Programming, Engineering Student, Programming Problem.

## Introduction

The subject of computer programming is one of the core papers offered to all engineering students at any higher education institution in Malaysia. Failure to complete this subject will lead the students to be unable to graduate and unqualified to obtain their degree or certificate from the university. The role of educators is important to make the computer programming class effective and exciting.

According to Cheah (2020), although many programming tools are available in teaching the programming subject, the students' performance continues to drop. One of the critical reasons is due to the lack of students' ability in problem-solving. Furthermore, the level of critical thinking skills is very low among students at the tertiary education level, which contributes to the decline in programming subject performance. Based on past research

(Ismail et al., 2010), a higher level of knowledge of 'when' and 'why' from the metacognitive skills is needed during the first stage of programming education. Besides, static teaching materials such as printed book references with unattractive presentations or very textual explanations without infographics are ineffective as teaching material for learning the dynamic nature of computer programming subject (Bennedsen & Caspersen, 2005).

Hence, this paper concentrates on the educator factors that influence the performance in the programming subject among engineering students at Universiti Teknologi MARA (UiTM), Penang branch. This research was intentionally carried out to assist and guide computer science educators to improvise their teaching methodology for computer programming, as well as improve students' interest and performance.

**Roles of Educators in Teaching Programming**

The roles of educators are extremely important for making the programming class interesting and fun. The traditional teaching methods using conventional static materials such as the textbook, marker pen and slide do not raise the effectiveness of learners' understanding (Bennedsen & Caspersen, 2005). Conventional approaches are appropriate if the class is a combination of several groups of learners and handled as a large crowd for lectures, which involves more than 50 to 100 students. Teaching the programming subject requires live interaction between the educator and learners besides the creation of dynamic communication and understanding of programming concepts. The educators can give immediate feedback to the learners when the class is divided into small groups and this approach looks ideal as detailed explanations can be provided whenever needed by the learners (Zhang et al., 2013). Interactivity in the class that inject the elements of spatial and visualisation are much more effective than conventional static programming materials such as hardcopy and softcopy notes. Thus, using the contemporary approach will degrade the learning curve among learners and decrease the learners' interest in the programming subject.

Some educators are very concerned about the syntax of the programming language rather than understanding the problem-solving methods. This is because the curriculum was designed to focus on the popularity of the programming language with the current demands of the industrial revolution (IR) 4.0. Gomes in his article elaborated that the suitability of pedagogy for teaching the programming subject has been put aside by the curriculum designers while the popularity of programming language is prioritised (Gomes & Mendes, 2007). Selecting inappropriate programming language and the teaching pedagogy will affect the effectiveness of learning programming and create a negative impact on learners to understand the subject (Brown & Wilson, 2018). As a result, the learners will become unable to apply the programming concepts in real-life or problem-solving. This is agreed by Byrne and Lyons (2001) stating that the learners who face difficulties in mastering the concepts of numbers theory, calculus, geometric and trigonometric will fail to transform the abstract or problem statements into mathematical formulas.

The instructors should be competent in the demands of high-level abstraction and analytical thinking to produce comprehensive solutions for any problem statements (Robins, 2019). Hence, the selection of programming language should be less complex, easy to remember and improve the learning curve of learners during self-explanatory or self-study without formal guidance by the educators. Thus, the selection based on the popularity of the programming language with current industrial demands is not an excellent choice because

the curriculum was designed according to education policy standards and not for professional purposes (Gomes & Mendes, 2007). Understanding the programming semantics is fundamental as the learners can apply the same semantics to any programming language, which only concentrates on learning the programming syntax.

In the research conducted by Ismail et al (2010), his team identified that the main problem in teaching computer programming is the ineffective use of presentation techniques for problem-solving. Most educators still sustain with the pseudocode and the flowchart to explain to the students on the problem-solving steps. These tools are only applicable for structured programming. Nevertheless, the conventional tools are not appropriate for the object-oriented programming language, unless the educators know how to apply the Unified Modelling Language (UML) class-diagram tools for an object-oriented approach. Approaches that provide more visualisation in explanation are needed to allow the students to have a mental representation of the problem (Robins, 2019).

Table 1
*Summary of Problems Faced by the Educators in Teaching the Programming Subject*

| Problem | Author(s) | Descriptions |
|---|---|---|
| Using the conventional static materials | Bennedsen & Caspersen(2005) | • The educators are using the textbook, marker pen and slide for teaching purposes.<br>• The teaching delivery is not interesting, creative and effective enough to create students' interest and awareness. |
| The class should be divided into small groups | Zhang et al (2013) | • Detailed explanations can be done effectively for the students.<br>• Creates interactivity in the class that injects the elements of spatial and visualization to the learners. |
| Educators are concerned with the programming syntax instead of the problem-solving | Gomes & Mendes (2007) | • For fulfilling the current demands of the industrial revolution (IR) 4.0<br>• Selecting popular programming languages based on the current market popularity. |
| Selecting the wrong pedagogy and programming approaches | Brown & Wilson (2018); Ismail et al (2010), Robins (2019) | • Selecting inappropriate programming language and teaching pedagogy, for example, using the structured approach such as pseudocode and flowchart for Object-Oriented Programming (OOP). Supposedly, the educators should apply the Unified Modelling Language (UML) for OOP. |
| The programming language selected is difficult to teach | Gomes & Mendes (2007); Robins(2019) | • Instructors should be competent in the demands of high-level abstraction and analytical thinking.<br>• Programming language should be less complex with easy to remember syntaxes.<br>• Understanding the programming semantics is fundamental for any programming language. |

## Methodology

This study involved 241 students who took the programming subject at the UiTM Pulau Pinang branch. They consisted of diploma and degree students from the Faculty of Mechanical Engineering (FKM) and the Faculty of Civil Engineering (FKA). Table 2 below displays the number of students who took programming language by semester. For diploma level, students will take this subject in semester 2 while for degree level, students will take programming language in semester 2 or semester 4. The remaining are students who repeat the subject.

Table 2
*Number of Students by Semester*

| Semester | Programming Code CSC128 (Diploma) | CSC425 (Degree) |
|---|---|---|
| 2 | 142 | 12 |
| 3 | 1 | 3 |
| 4 | 0 | 76 |
| 5 | 1 | 1 |
| 6 | 4 | 0 |
| 8 | 1 | 0 |

This study's questionnaire was divided into two sections. The first section is about the course or subject information that was taken, and the second section is about the educator's factors associated with the students. All questions contained 14 items and were divided into three sections as indicated in the table below (Table 3 and Table 4).

Table 3
*Construct Questions in Course or Subject Information*

| Construct | Options |
|---|---|
| Program Code | 1. *Mechanical Engineering*<br>2. *Civil Engineering* |
| Study Level | 1. *Diploma*<br>2. *Degree* |
| Semester | *1/2/3/4/5/6/7/8* |
| Programming Code Taken | 1. *CSC128*<br>2. *CSC425* |
| Status Taken | 1. *First Timer*<br>2. *Not First Timer* |

Table 4
*Construct Questions in Educator-Related Factors*

| Construct | Statements |
|---|---|
| *A.* Personality Traits of My Programming Lecturer … | 1. *Has a good relationship with the students.* |
| | 2. *Shows smartness, confidence and firmness in making decisions.* |
| | 3. *Enforces proper discipline and is strict in following the prescribed rules.* |
| | 4. *Has an interesting personality with a good sense of humor.* |
| | 5. *Is open to suggestions and opinions and is worthy of praise.* |
| *B.* Teaching Skills of My Programming Lecturer… | 1. *Explains the objective of the lesson clearly at the start of each class.* |
| | 2. *Has mastery of the subject matter.* |
| | 3. *Is organized in presenting subject matters by systematically following the course or subject outline.* |
| | 4. *Is updated with present trends, relevant to the subject matter.* |
| | 5. *Uses various strategies, teaching aids/devices and techniques in presenting the lessons.* |
| *C.* Instructional Materials of My Programming Lecturer… | 1. *'Chalk and blackboard' in explaining the lesson.* |
| | 2. *Workbooks/textbook.* |
| | 3. *Visual aids (e.g., PowerPoint).* |
| | 4. *Articles/material/notes/hand-outs for additional references.* |

A reliability Test or Cronbach's Alpha was performed first before analysing the questionnaire. Reliability describes how reliable and consistent a research instrument's measurement of a variable is. The better the instrument's reliability, the fewer errors it generates (Kumar, 2018). Cronbach's Alpha values are based on (Choi et al., 2001).

Cronbach's Alpha was used in this analysis to measure the internal consistency of the items tested. According to Table 5, the Cronbach's alpha value for all 14 questionnaires tested was 0.889. This value was greater than 0.8, which is considered reliable.

Table 5
*Reliability Test*

| Cronbach's Alpha | N of items |
|---|---|
| .889 | 14 |

These questionnaires used the five-point Likert scale. The educators' related factors used the range from 5-always, 4-often, 3-sometimes, 2-rarely, and 1-never; values greater than 3 are positive and values less than 3 are negative statements. Figure 1 shows the students' responses to Construct A: Personality Traits of My Programming Lecturer. It was found that the number of students who strongly agreed with the statement was very high.

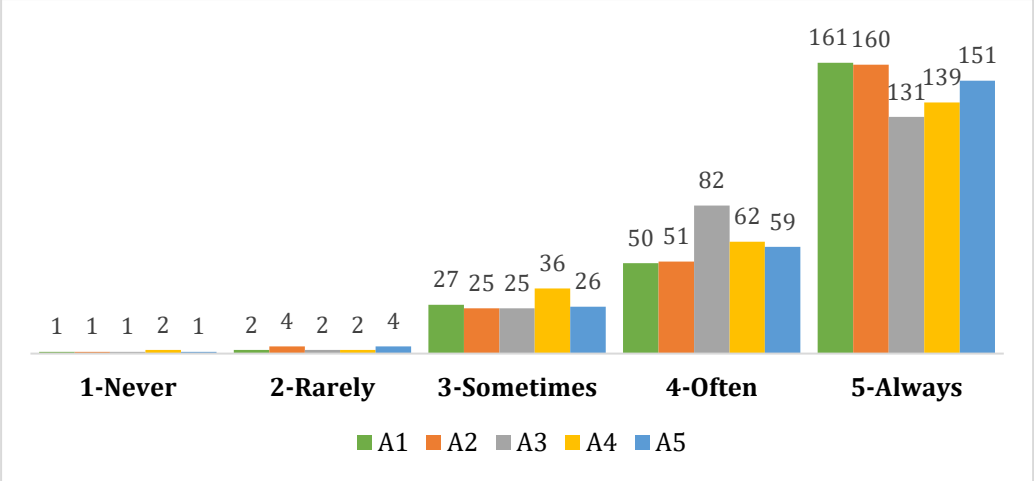


Figure 1. Students' Responses for Construct A (Personality Traits of My Programming Lecturer)

Figure 2 below shows the students' responses to Construct B: Teaching Skills of My Programming Lecturer. It was also found that the response of the students was also very high for those who agreed with the statement. Only a few students disagreed with the statement.

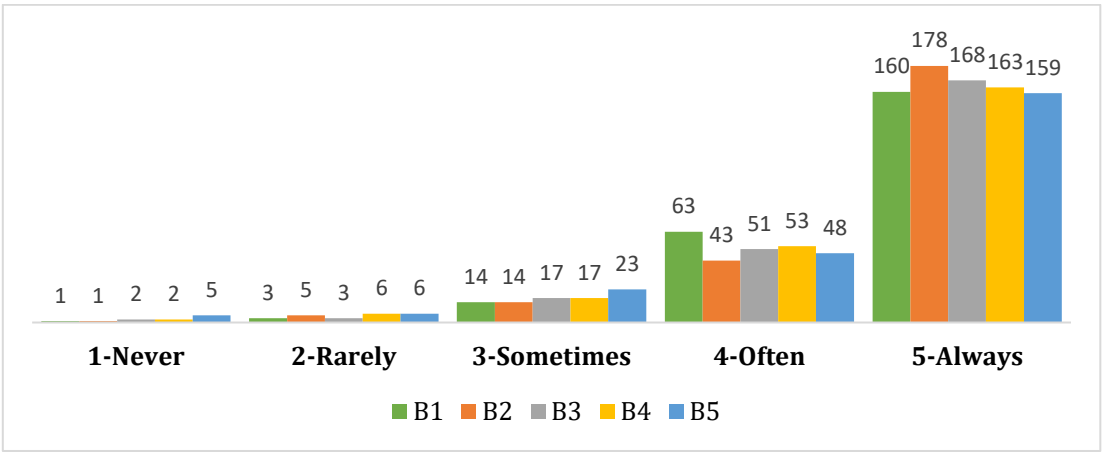


Figure 2. Students' Responses for Construct B (Teaching Skills of My Programming Lecturer)

It can be seen in Figure 3 below presents that majority of students prefer visual aids (e.g. PowerPoint) and Articles/Materials/Notes/Hand-outs for additional references. However, some were seen to disagree. Furthermore, there was a relatively similar response in each Likert scale for the selection of the Chalk and Whiteboard method.

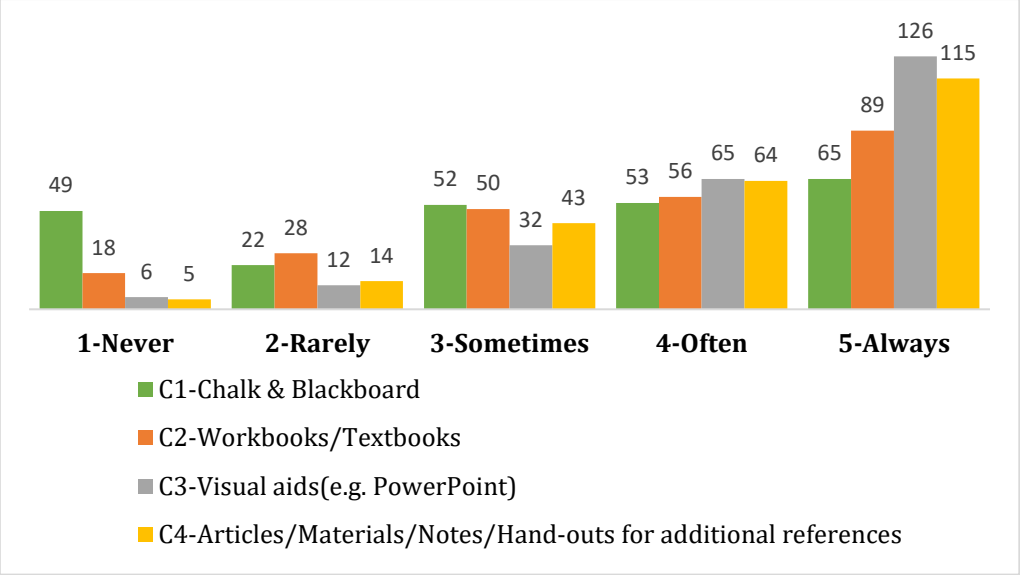


Figure 3. Student' Responses for Construct C (Instructional Materials of My Programming Lecturer)

Analysis using mean and standard deviation for each construct is discussed further to know more relevant details in the Result and Discussions section.

**Results and Discussion**

In this paper, descriptive statistics was applied. The analysis using mean and standard deviation values can be used to identify in general educator-related factors that influence students in learning a programming language. According to Table 6, the personality traits and teaching skills of the programming lecturers (items A1 to B5) showed a mean greater than 4. All the students were satisfied with their lecturers. Lecturers often show good personality such as intelligence, confidence and assertiveness in decision making, deliver things in an orderly manner by following course or subject guidelines systematically, besides using a variety of strategies, aids, or tools and techniques in delivering lessons. The standard deviation was also not too high. This implies that the ratings on the questionnaire were consistent. In terms of teaching materials (items C1 to C4), students stated that lecturers rarely used chalk and blackboard to explain courses or subjects (mean = 3.26), as well as workbooks or textbooks (mean = 3.71) compared to visual aids and materials for additional references (mean more than 4). The standard deviation for these four items was also relatively high, indicating that students' responses were inconsistent.

Table 6
*Educator -Related Factors*

| Item No. | Statement | Mean | SD |
|---|---|---|---|
| A1 | Has a good relationship with the students. | 4.53 | 0.758 |
| A2 | Shows smartness, confidence, and firmness in making decisions. | 4.51 | 0.78 |
| A3 | Enforces proper discipline and is strict in following the prescribed rules. | 4.41 | 0.743 |
| A4 | Has an interesting personality with a good sense of humor. | 4.39 | 0.829 |
| A5 | Is open to suggestions and opinions and is worthy of praise. | 4.47 | 0.785 |
| B1 | Explains the objective of the lesson clearly at the start of each class. | 4.57 | 0.699 |
| B2 | Has mastery of the subject matter. | 4.63 | 0.726 |
| B3 | Is organized in presenting subject matters by systematically following the course or subject outline. | 4.58 | 0.75 |
| B4 | Is updated with present trends, relevant to the subject matter. | 4.53 | 0.801 |
| B5 | Uses various strategies, teaching aids/devices and techniques in presenting the lessons. | 4.45 | 0.917 |
| C1 | Chalk and blackboard' in explaining the lesson. | 3.26 | 1.464 |
| C2 | Workbooks/textbook. | 3.71 | 1.278 |
| C3 | Visual aids (e.g., PowerPoint). | 4.22 | 1.018 |
| C4 | Articles/material/notes/hand-outs for additional references. | 4.12 | 1.032 |

Overall, it was found that educator-related factors are very helpful in programming learning. This was proven by the mean of 3 and above, which means that students agreed with all statements. There were only several items in teaching materials (C1 – C4) that have a high standard deviation indicating a lack of uniformity of teaching materials choices among students. This may be due to online learning where each lecturer uses different types of learning materials and students are still in the process of adjusting to the new norms.

**Conclusion**

Educators should dynamically improvise their teaching materials and teaching delivery to make the class more attractive, effective and able to enrich the programming knowledge. If the students can remember at least 50 per cent of the lectures, so it is considered an efficacious teaching method. Educators should focus on the root causes such as lack of problem-solving skills and critical thinking ability, which contribute to the student's performance in the programming subject. The educators should imply and execute dynamic problem-solving related to the students' daily life or environment and put themselves in their students' shoes and slowly work together with them in solving the real problems, step by step until everybody can continue solving the problem individually without any help from the

educators. The role of an educator is very important here as guidance, but not spoon-feeding them until they are unable to be independent anymore.

Further research should focus on the best practices among educators that contribute to inclining programming subject performance. The elements that should be focused on are problem-solving techniques, dynamic critical thinking, avoidance of confusion in programming, effective pedagogy, and the educators' behaviour.

**Acknowledgement**

First and foremost, praises and thanks to the God, the Almighty, for His showers of blessings throughout our research work to complete the research successfully. We would like to express our deep and sincere gratitude to University of Technology MARA for giving us the opportunity to do research and providing invaluable guidance throughout this research. We are extremely grateful to our fellow friends for the keen interest shown and their continuing support to complete this research work successfully.

**Corresponding Author**

Jamal Bin Othman

Computing Sciences Studies, College of Computing, Informatics and Media Universiti Teknologi MARA Cawangan Pulau Pinang, Malaysia.

Email: jamalothman@uitm.edu.my

**References**

Bennedsen, J., & Caspersen, M. E. (2005). Revealing the programming process. *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education*, 186–190.

Brown, N. C. C., & Wilson, G. (2018). Ten quick tips for teaching programming. *PLoS Computational Biology*, *14*(4), e1006023.

Byrne, P., & Lyons, G. (2001). The effect of student attributes on success in programming. *Proceedings of the 6th Annual Conference on Innovation and Technology in Computer Science Education*, 49–52.

Cheah, C. S. (2020). Factors contributing to the difficulties in teaching and learning of computer programming: A literature review. *Contemporary Educational Technology*, *12*(2), 1–14. https://doi.org/10.30935/cedtech/8247

Choi, N., Fuqua, D. R., & Griffin, B. W. (2001). Exploratory analysis of the structure of scores from the multidimensional scales of perceived self-efficacy. *Educational and Psychological Measurement*, *61*(3), 475–489.

Gomes, A., & Mendes, A. J. (2007). Learning to program-difficulties and solutions. *International Conference on Engineering Education–ICEE*, *7*.

Ismail, M. N., Ngah, N. A., & Umar, I. N. (2010). Instructional strategy in the teaching of computer programming: a need assessment analyses. *TOJET: The Turkish Online Journal of Educational Technology*, *9*(2).

Kumar, R. (2018). *Research methodology: A step-by-step guide for beginners*. Sage.

Robins, A. V. (2019). Novice Programmers and Introductory Programming. *The Cambridge Handbook of Computing Education Research*, 327.

Zhang, X., Zhang, C., Stafford, T. F., & Zhang, P. (2013). Teaching introductory programming to IS students: The impact of teaching approaches on learning performance. *Journal of Information Systems Education*, *24*(2), 147–155.