INTERNATIONAL JOURNAL OF ACADEMIC RESEARCH IN BUSINESS & SOCIAL SCIENCES
Vol 12, Issue 9, (2022) E-ISSN: 2222-6990

# Revalidation on Computer Programming Self-Efficacy Scale Using Fuzzy-Delphi Method Approach

Nor Zalina Binti Ismail, Mohd Rizal Bin Razak
Universiti Teknologi MARA Pahang, Malaysia
Corresponding Authors Email: nza1601@uitm.edu.my

**To Link this Article:** http://dx.doi.org/10.6007/IJARBSS/v12-i9/14881 DOI:10.6007/IJARBSS/v12-i9/14881

***Published Date:*** *11 September* 2022

**Abstract**
Computer programming self-efficacy is the tool to measure the students' competencies in programming either success or fail. It is important for the students in all over the world and in Malaysia because if the score is low, the students must work harder and find the solution or ways to learn programming effectively. Then, for the instructor, it can be the guideline to plan the teaching material and mechanism to deliver the subject. This research re-validates the computer programming self-efficacy that can be tailored to Malaysian students. This research was used the Fuzzy-Delphi Method that adopt the questionnaire in the literature. This questionnaire has five main item which are Logical Thinking, Cooperation, Algorithm, Control and Debug to measure the student's self-efficacy. The item in the questionnaire were re-validated by five programming experts in order to tailor the questionnaire in the Malaysia context. The findings of the research were the expert agreement on the acceptance of the item used in the questionnaire. The result shows all the experts agreed almost 100% of the item used.
**Keywords:** Computer Programming, Self-Efficacy, Programming Expert

## Introduction

Computer Programming is the subject that may develop critical thinking, analytical thinking and problem-solving skill. As mention by Kukul et al (2017) that the development of other advanced abilities, such as problem-solving, logical inference, and creative thinking, is facilitated by knowledge of computer programming. Due to this advantage, many higher institutions offer programming subject to the non-computer sciences students and even the secondary and primary school is also requiring their students to learn programming,

However, programming is the challenging subject to teach because have a high failure rate and dropout (Silva et al., 2019). The subject need the correct comprehension of abstract concepts(Lahtinen et al., 2005). Students cannot connect the theories learn in the programming subject in order to complete the programming task. In addition, learner came to view computer programming as a challenging endeavour due to the challenges they encountered when doing programming on a program-compiler and involve in the uninteresting programming activities during teaching and learning process (Kukul et al., 2017).

Self-efficacy theory has become a significant tool for comprehending and forecasting an individual's performance (Askar & Davenport, 2009) and may reduce the number of failure rates and dropout in programming. In education, self-efficacy is important and useful(Askar & Davenport, 2009). Self-efficacy is defined as a person's belied of her ability to achieve in a given domain (Gorson & O'Rourke, 2020). In addition, according to *Bandura (1986)*, perceived self-efficacy refers to people's assessments of their capacities to plan and carry out the courses of action necessary to achieve specified types of performances.

### Computer Programming Self-Efficacy

Computer programming self-efficacy can be the important tools to measure the capability and ability of the students to carry out the programming task in teaching and learning process. If the result of the computer programming self-efficacy is low, it is an alarm to the instructor to develop and execute the prevention plan to reduce the failure rates. Students may have different perception on programming either positive or negative and either difficult or easy (Erumit et al., 2019). This competency level perception in programming is define as self-efficacy (Erumit et al., 2019). This belief can be the driven either students might success or fail in the programming subject. Research done by Gorson & O'Rourke(2020), stated that "self-efficacy – an individual's belief in their ability to complete a task – can influence whether students decide to persist in Computer Science". Computer programming self-efficacy can cause the rejection of the task and activities during learning programming (Gorson & O'Rourke, 2020).

### Computer Programming Self-Efficacy in Malaysia

There are many research done in Malaysia that focus on computer self-efficacy(Roslani Embi, 2007) and the usage of web application(Ramayah & Aafaqi, 2004). There is few research done that focus on computer programming self-efficacy. Research done by Kanaparan et al (2019) that use the sample from Malaysia and New Zealand student stated that "students' programming self-efficacy beliefs had a strong positive effect on enjoyment, while gratification and interest had a negative effect on programming performance." Based on these findings, will help the course instructor to plan the programming course detail. Based on the benefit of the computer programming self-efficacy to the education area it is an important to have the validated computer programming self-efficacy scale that focus and can be apply in Malaysia situation and respondent.

### The Research Aims

This study aims to re-validate the self-efficacy computer programming scale using Fuzzy Delphi Method. It is important to re-validate the scale in order to tailored for Malaysian usage and relevant with this education era.

### Methodology

This research uses the Fuzzy Delphi Method (FDM). This method is used because it can analyse the expert agreement in order to determine the concrete decision. There are two main activities in this research which are finding the literature and design the expert questionnaire.

Firstly, in the activity 1 literature review is used to find the recent questionnaire design. Based on this, this research use the questionnaire design by Tsai et al (2019) that have five items related to the programming competency; logical thinking, cooperation, algorithm, control and debug. These factors have their own construct questions.
Secondly, in the activity 2 the expert questionnaire was design and develop based on the items and construct questions or sub item identify in the activity 1. The 7-point scale questionnaire are distributed to five programming experts that have an experience in teaching and learning programming more than five years. Lastly, all the consent from the experts was analysed using Fuzzy Delphi Method (FDM).

**Sampling Procedure**
In the analysis, the strategy used is purposeful sampling. The researcher needs to obtain the agreement of pre-determined item which is in this study is the well-develop questionnaire. Based on study done by Hasson et al (2000), purposeful sampling strategy in Fuzzy Delphi Method suit the criteria of this type of research. The sampling used in this research is five experts have been chosen and agreed to participate in the research. These experts have an experience in various programming language such as C, C++, Java and web programming. The list of experts details showed in Table 1.

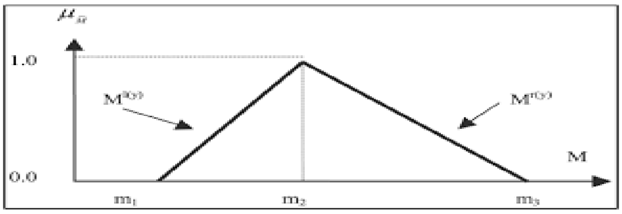
Table 1
*List of Experts*

| Expert | Field of expertise | Institution |
|---|---|---|
| 1 Doctorate | Introductory and High Level of Computer Programming in C, C++, Java language and Web Programming | Public university |
| 2 Senior Lecturer | | |
| 2 Lecturer | | |

**Expert Criteria**
Expert means the person that earned qualification, training, experience, professional membership and peer recognition (Booker & McNamara, 2004). The experts chosen in this research have a minimum of Master and one expert have a PhD. Two of them have about more than five years' experience in industry. The selection of expert in Fuzzy Delphi research is the critical issue. The researcher selects an expert that learn, have been working in industry and also an instructor for programming subject that involve more than one programming language.

**Fuzzy Delphi Step**
There are six steps involve in Fuzzy Delphi Method. Table 2 list and evaluate the steps involve in the Fuzzy Delphi Method.

Table 2
*Fuzzy Delphi Step*

| Step | Formulation |
|---|---|
| 1. Expert selection | • A total of 5 experts were included in this report. A panel of experts was assembled to assess the significance of the assessment parameters on the factors to be evaluated using linguistic variables. and definitions of potential problems with the piece, and so on. |
| 2. Determining linguistic scale | • This procedure entails translating all linguistic variables into the counting of fuzzy triangles (triangular fuzzy numbers). This move also includes the addition of fuzzy numbers to the translation of linguistic variables (Hsieh et al., 2004). The Triangular Fuzzy Number represents the values m1, m2, and m3 and is written as follows (m1, m2, m3). The value of m1 represents the smallest possible value, the value of m2 represents a rational value, and the value of m3 represents the highest possible value. While Triangular Fuzzy Number is used to generate Fuzzy Scale for the purpose of converting linguistic variables into fuzzy numbers.   Figure 1: Triangular fuzzy number |
| 3. The Determination of Linguistic Variables and Average Responses | • Once the researcher gain input from the specified expert, the researcher must convert all measurement findings to Fuzzy scales. This is often recognized as the acknowledgment of each answer (Benitez et al., 2007). |
| 4. The determination of threshold value "d" | • The threshold value is crucial in determining the degree of agreement among experts (Thomaidis et al., 2006). The distances for each fuzzy integer m = (m1, m2, m3) and n = (m1, m2, m3) are determined using the formula: $d(\tilde{m},\tilde{n}) = \sqrt{\frac{1}{3}[(m1-n1)^2 + (m2-n2)^2 + (m3-n3)^2]}$ |
| 5. Identify the alpha cut aggregate level of fuzzy assessment | • If an expert consensus is reached, a fuzzy number is assigned to each piece (Mustapha & Darussalam, 2017). The below is the approach for calculating and measuring fuzzy values: (1) 4 (m1 + 2m2 + m3) Amax |

| | |
|---|---|
| 6. Defuzzification process | • This process uses the formula Amax = (1) ⁄4 (a1 + 2am + a3). If the researcher uses Average Fuzzy Numbers or average response, the resulting score number is a number that is in the range 0 to 1 (Ridhuan et al., 2014). In this process, there are three formulas namely: i. A = 1/3 * (m1 + m2 + m3), or; ii. A = 1/4 * (m1 + 2m2 + m3), or; iii. A = 1/6 * (m1 + 4m2 + m3). A-cut value = median value for '0' and '1', where α-cut = (0 + 1) / 2 = 0.5. If the resulting A value is less than the α-cut value = 0.5, the item will be rejected because it does not indicate an expert agreement. According to Bojdanova (2006) the alpha cut value should exceed 0.5. It is supported by Tang & Wu (2010) who stated that the α-cut value should be more than 0.5. |
| 7. Ranking process | • The positioning process is carried out by means of defining elements based upon values of defuzzification based on expert agreement that the element with highest importance is the most important place for decision (Fortemps & Roubens, 1996) |

**Instrumentation**

The instrument in this Fuzzy Delphi research is based on the existing related material found in the literature. Skulmoski et al (2007) mention that the researcher can redesign the questionnaire items based on the literature, pilot study and experience. In addition, in Fuzzy Dephi Method, researcher used research literature material such as expert interviews and focus group approach (Mustapha & Darusalam, 2018).

So, the researchers list the item and sub-item that will be used in the expert questionnaire based on literature. This questionaire is based on five main item that used a 7-point scale. The more scales used, the more accurate and precise result will gain (Chang et al., 2011). In order to ease the experts to answer the questionnaire, the researchers change the fuzzy scale into questionnaire scale.  Table 3 shows the detail of the questionnaire scale and its equivalent fuzzy scale.

Table 3

*Questionnaire Scale versus Fuzzy scale*

| Item | Questionnaire Scale | Fuzzy number |
|---|---|---|
| Strongly disagree | 1 | 0.0 |
| Disagree | 2 | 0.1 |
| Somewhat Disagree | 3 | 0.3 |
| Neutral | 4 | 0.5 |
| Somewhat agree | 5 | 0.7 |
| Agree | 6 | 0.9 |
| Strongly agree | 7 | 1.0 |

**The Major Required Items and Sub-item of Self-Efficacy Computer Programming Scale for Students**

There are five major items and sub-items of required scale related to computer programming self-efficacy. Table 4 shows the list of major items and sub-items based on related literature and the early item rank. Then, the researcher used the Fuzzy Delphi Method to re-validate the item and sub-item whether it is appropriate or not in the scale.

Table 4
*The List of Major Item and Sub-Item of Self-Efficacy Computer Programming Skill*

| Early Item Rank | Item | Sub-Item |
|---|---|---|
| I1S1 | Logical Thinking | I can understand the basic logical structure of a program. |
| I1S2 | | I can understand a conditional expression such as ‘‘if...else...’’ |
| I1S3 | | I can predict the final result of a program with logical conditions. |
| I1S4 | | I can predict the result of a program when given its input values. |
| | | |
| I2S1 | Cooperation | I know programming work can be divided into sub-tasks for people. |
| I2S2 | | I can work with others while writing a program. |
| I2S3 | | I can make use of divisions to enhance programming efficiency. |
| | | |
| I3S1 | Algorithm | I can figure out program procedures without a sample. |
| I3S2 | | I don’t need others’ help to construct a program. |
| I3S3 | | I can make use of programming to solve a problem. |
| | | |
| I4S1 | Control | I can open and save a program in a program editor. |
| I4S2 | | I can edit and revise a program in a program editor. |
| I4S3 | | I can run and test a program in a program editor. |
| | | |
| I5S1 | Debug | I can find the origin of an error while testing a program. |
| I5S2 | | I can fix an error while testing a program. |
| I5S3 | | I can learn more about programming via the debugging process. |

**Finding**

The finding of this research gives an expert agreement on the self-efficacy computer programming scale. The questionnaire distributed to five programming experts and the data collected based on their responses. The questionnaire is based on five main items: logical thinking, cooperation, algorithm, control and debug. Table 5 until Table 6 represent the analysis result based on five main items mention earlier.

Table 5
*Analysis Result for Item 1(Logical Thinking)*

| Results | Sub-Item1 | Sub-Item2 | Sub-Item3 | Sub-Item4 |
|---|---|---|---|---|
| Expert1 | 0.01155 | 0.01155 | 0.01155 | 0.01155 |
| Expert2 | 0.04619 | 0.01155 | 0.04619 | 0.04619 |
| Expert3 | 0.12702 | 0.04619 | 0.04619 | 0.04619 |
| Expert4 | 0.04619 | 0.01155 | 0.04619 | 0.04619 |
| Expert5 | 0.04619 | 0.01155 | 0.12702 | 0.12702 |

| Statistics | Sub-Item1 | Sub-Item2 | Sub-Item3 | Sub-Item4 |
|---|---|---|---|---|
| Value of the item | 0.05543 | 0.01848 | 0.05543 | 0.05543 |
| Value of the construct | | | | 0.04619 |
| Item < 0.2 | 5 | 5 | 5 | 5 |
| % of item < 0.2 | 100% | 100% | 100% | 100% |
| Average of % consensus | | | | 100 |
| Defuzzification | 0.92 | 0.98 | 0.92 | 0.92 |
| Ranking | 2 | 1 | 2 | 2 |
| Status | Accept | Accept | Accept | Accept |

Table 6
*Analysis Result for Item 2(Cooperation)*

| Results | Sub-Item1 | Sub-Item2 | Sub-Item3 |
|---|---|---|---|
| Expert1 | 0.03464 | 0.11547 | 0.06928 |
| Expert2 | 0.02309 | 0.05774 | 0.06928 |
| Expert3 | 0.03464 | 0 | 0.06928 |
| Expert4 | 0.02309 | 0 | 0.10392 |
| Expert5 | 0.02309 | 0.05774 | 0.10392 |

| Statistics | Sub-Item1 | Sub-Item2 | Sub-Item3 |
|---|---|---|---|
| Value of the item | 0.02771 | 0.04619 | 0.08314 |
| Value of the construct | | | 0.05235 |
| Item < 0.2 | 5 | 5 | 5 |
| % of item < 0.2 | 100% | 100% | 100% |
| Average of % consensus | | | 100 |
| Defuzzification | 0.96 | 0.9 | 0.82 |
| Ranking | 1 | 2 | 3 |
| Status | Accept | Accept | Accept |

Table 7
*Analysis Result for Item 3(Algorithm)*

| Results | Sub-Item1 | Sub-Item2 | Sub-Item3 |
|---|---|---|---|
| Expert1 | 0.02309 | 0.02309 | 0.01155 |
| Expert2 | 0.03464 | 0.02309 | 0.04619 |
| Expert3 | 0.02309 | 0.02309 | 0.01155 |
| Expert4 | 0.02309 | 0.03464 | 0.01155 |
| Expert5 | 0.03464 | 0.03464 | 0.01155 |

| Statistics | Sub-Item1 | Sub-Item2 | Sub-Item3 |
|---|---|---|---|
| Value of the item | 0.02771 | 0.02771 | 0.01848 |
| Value of the construct | | | 0.02463 |
| Item < 0.2 | 5 | 5 | 5 |
| % of item < 0.2 | 100% | 100% | 100% |
| Average of % consensus | | | 100 |
| Defuzzification | 0.94 | 0.94 | 0.98 |
| Ranking | 2 | 2 | 1 |
| Status | Accept | Accept | Accept |

Table 8
*Analysis Result for Item 4(Control)*

| Results | Sub-Item1 | Sub-Item2 | Sub-Item3 |
|---|---|---|---|
| Expert1 | 0.03464 | 0.03464 | 0.03464 |
| Expert2 | 0.13856 | 0.13856 | 0.13856 |
| Expert3 | 0.03464 | 0.03464 | 0.03464 |
| Expert4 | 0.03464 | 0.03464 | 0.03464 |
| Expert5 | 0.03464 | 0.03464 | 0.03464 |

| Statistics | Sub-Item1 | Sub-Item2 | Sub-Item3 |
|---|---|---|---|
| Value of the item | 0.05542 | 0.05542 | 0.05542 |
| Value of the construct | | | 0.05542 |
| Item < 0.2 | 5 | 5 | 5 |
| % of item < 0.2 | 100% | 100% | 100% |
| Average of % consensus | | | 100 |
| Defuzzification | 0.94 | 0.94 | 0.94 |
| Ranking | 1 | 1 | 1 |
| Status | Accept | Accept | Accept |

Table 9
*Analysis Result for Item 5(Debug)*

| Results | Sub-Item1 | Sub-Item2 | Sub-Item3 |
|---|---|---|---|
| Expert1 | 0.04619 | 0.01155 | 0 |
| Expert2 | 0.01155 | 0.01155 | 0.11547 |
| Expert3 | 0.01155 | 0.04619 | 0 |
| Expert4 | 0.01155 | 0.01155 | 0.05774 |
| Expert5 | 0.01155 | 0.01155 | 0.05774 |

| Statistics | Sub-Item1 | Sub-Item2 | Sub-Item3 |
|---|---|---|---|
| Value of the item | 0.01848 | 0.01848 | 0.04619 |
| Value of the construct | | | 0.02772 |
| Item < 0.2 | 5 | 5 | 5 |
| % of item < 0.2 | 100% | 100% | 100% |
| Average of % consensus | | | 100 |
| Defuzzification | 0.98 | 0.98 | 0.9 |
| Ranking | 1 | 1 | 2 |
| Status | Accept | Accept | Accept |

Based on the Table 5 until Table 9, the threshold values of sub-item for all main item (logical thinking, cooperation, algorithm, control and debug) are less than 0.2. It shows that expert agree with the sub-item for all the main items listed. The average threshold value is less than 0.2 or 0.04619 for Item1(Logical Thinking, 0.05235 for Item2(Cooperation), 0.02463 for Item3(Control), 0.05542 for Item4(Control) and 0.02772 for Item5(Debug). As mention by Chang et al (2011), if the average of value of item <0.2 the item get the high agreement from the experts. And also, the average of % consensus of experts is 100% for all the main items that indicate the expert agreement were met. As a main contribution of the study, the researcher lists the sub-item based on the new rank that agree by the experts. Table 10 shows the detail of the new rank.

Table 10
*The List of Sub-items Based on Experts Agreement*

| Item | Early Sub-Item Rank | New Sub-Item Rank | Sub-Item |
|---|---|---|---|
| Logical-Thinking | I1S1 | I1S2 | I can understand the basic logical structure of a program. |
| | I1S2 | I1S1 | I can understand a conditional expression such as ‘‘if...else...’’ |
| | I1S3 | I1S3 | I can predict the final result of a program with logical conditions. |
| | I1S4 | I1S4 | I can predict the result of a program when given its input values. |
| | | | |
| Cooperation | I2S1 | I2S1 | I know programming work can be divided into sub-tasks for people. |
| | I2S2 | I2S2 | I can work with others while writing a program. |
| | I2S3 | I2S3 | I can make use of divisions to enhance programming efficiency. |
| | | | |
| Algorithm | I3S1 | I3S2 | I can figure out program procedures without a sample. |
| | I3S2 | I3S3 | I don’t need others’ help to construct a program. |
| | I3S3 | I3S1 | I can make use of programming to solve a problem. |
| | | | |

| | | | |
|---|---|---|---|
| Control | I4S1 | I4S1 | I can open and save a program in a program editor. |
| | I4S2 | I4S2 | I can edit and revise a program in a program editor. |
| | I4S3 | I4S3 | I can run and test a program in a program editor. |
| | | | |
| Debug | I5S1 | I5S1 | I can find the origin of an error while testing a program. |
| | I5S2 | I5S2 | I can fix an error while testing a program. |
| | I5S3 | I5S3 | I can learn more about programming via the debugging process. |

**Conclusion and Suggestion**

Computer Programming should be learned by many levels of educations either young age like pre-school, primary school, secondary school and the most essential is higher institution. This is due to the advantage that learners can be gained from study programming such as problem-solving skill, critical thinking skill and many more. In Malaysia, it is also moving towards fostering for all levels of education to learn the subject of programming. But based on the literature teaching and learning programming is not easy. In the higher institution there are many failures and drop out for this subject. Self-efficacy is the medium that may reduce the failure rate because it can predict the commitment of the students either to strive hard or not in the subject or drop the subject. The output of the computer programming self-efficacy can be used to help instructor to boost the student's motivation and study skill in order to get the better grades and avoid the drop-out. The instructor also can use the result to strategize their teaching material and delivery. It is important to have the valid self-efficacy scale that can be used in Malaysia environment.

In the findings there are five items that are evaluated to determine the student's self-efficacy level in computer programming. The first item is logical thinking which is the ability to understand the basic logical structure of the program and can predict the result of the program/coding based on the given input values. Second item is cooperation, which is the students aware and can cooperate and divide the one big task into sub-task with other team members to produce expected output in the expected due date. Third item is algorithm which is the student can do the step-by-step process to solve the programming problem. Fourth item is control which is the students can use the programming editor successfully to write, run and test the programming code. Fifth item is debug which is the ability to identify and fix either the logical or syntax error. And in the debug item, students should be able to learn more on programming during debugging process. The items evaluated in the findings as mentioned above is based on the research literature and in this research during the re-validation process all the item is agreed by the experts can be used to evaluate the self-efficacy of computer programming in Malaysia.

This computer programming scale can be used in Malaysia environment to identify the ability of the students to pursue and success in programming subject. For the future works, the researcher suggest:

a) The computer programming self-efficacy scale is design based on target audience which is early age (4-5 years), pre-school, primary school, secondary school, higher institution and elderly.

b) Suggestion for programming self-efficacy scale that focus for higher institution add the item on understanding and applying the programming structure like looping, array and function.
c) For all categories of students, consider how the student overcomes the stuck to solve the programming problem faster in order to meet the due date.
d) The computer programming self-efficacy scale is design based on type of delivery type which is face-to-face, blended learning, hybrid learning or online learning.

**References**

Askar, P., & Davenport, D. (2009). An investigation of factors related to self-efficacy for java programming among engineering students. *Turkish Online Journal of Educational Technology*, *8*(1), 26–32.

Booker, J. M., & McNamara, L. A. (2004). Solving black box computation problems using expert knowledge theory and methods. *Reliability Engineering & System Safety*, *85*(1), 331–340.

Chang, P. L., Hsu, C. W., & Chang, P. C. (2011). Fuzzy Delphi method for evaluating hydrogen production technologies. *International Journal of Hydrogen Energy*, *36*(21), 14172–14179. https://doi.org/10.1016/j.ijhydene.2011.05.045

Erumit, K. A., Karal, H., Sahin, G., Aksoy, D. A., Gencan, A. A., & Benzer, A. I. (2019). A model suggested for programming teaching: Programming in seven steps. *Egitim ve Bilim*, *44*(197), 155–183. https://doi.org/10.15390/EB.2018.7678

Gorson, J., & O'Rourke, E. (2020). Why do CS1 Students Think They're Bad at Programming?: Investigating Self-efficacy and Self-assessments at Three Universities. *ICER 2020 - Proceedings of the 2020 ACM Conference on International Computing Education Research*, 170–181. https://doi.org/10.1145/3372782.3406273

Gregory, J., Skulmoski, H. F. T., & Krahn, J. (2007). The Delphi Method for Graduate Research. *Journal of Information Technology Education*, *6*, 93–105. https://doi.org/10.1007/3-540-47847-7_10

Hasson, F., Keeney, S., & McKenna, H. (2000). Research guidelines for the Delphi survey technique. *Journal of Advanced Nursing*, *32*(4), 1008–1015. https://doi.org/10.1046/j.1365-2648.2000.t01-1-01567.x

Kanaparan, G., Cullen, R., & Mason, D. (2019). *Effect of Self-efficacy and Emotional Engagement on Introductory Programming Students*. *23*, 1–24.

Lahtinen, E., Ala-Mutka, K., & Järvinen, H. M. (2005). A study of the difficulties of novice programmers. *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, 14–18. https://doi.org/10.1145/1067445.1067453

Mustapha, R., & Darusalam, G. (2018). *Aplikasi Kaedah Fuzzy Delphi dalam Penyelidikan Sains Sosial*.

Embi, R. (2007). *Computer Anxiety and Computer Self-Efficacy among Accounting Educators at Universiti Teknologi MARA (UiTM), MALAYSIA*.

Silva, D. B., Aguiar, R. D. L., Dvconlo, D. S., & Silla, C. N. (2019). Recent Studies about Teaching Algorithms (CS1) and Data Structures (CS2) for Computer Science Students. *Proceedings - Frontiers in Education Conference, FIE*, *2019-Octob*, 0–7. https://doi.org/10.1109/FIE43999.2019.9028702

Ramayah, T., Aafaqi, B. (2004). Role of Self-Efficacy in E-Library Usage Among Students of a Public University in Malaysia. *Malaysian Journal of Library & Information Science*, *9*(1), 39–57.

Tsai, M. J., Wang, C. Y., & Hsu, P. F. (2019). Developing the Computer Programming Self-Efficacy Scale for Computer Literacy Education. *Journal of Educational Computing Research*, *56*(8), 1345–1360. https://doi.org/10.1177/0735633117746747