

PAPER • OPEN ACCESS

Evaluating the Effectiveness of Course Assessments Method in Determining Student's Competency Level in Programming

To cite this article: Faridah Hani Mohamed Salleh *et al* 2021 *J. Phys.: Conf. Ser.* **1878** 012067

View the [article online](#) for updates and enhancements.

You may also like

- [Triangular fuzzy number in probabilistic fuzzy goal programming with pareto distribution](#)
Eka Susanti, Oki Dwipurwani, Robinson Sitepu et al.
- [A Mathematical Study on "Additive Technique" Versus "Branch and Bound Technique" for Solving Binary Programming Problem](#)
Suliadi Sufahani, M. Ghazali Kamardan, Mohd Saifullah Rusiman et al.
- [Compiler-assisted performance tuning](#)
Chun Chen, Jacqueline Chame, Yoonju Lee Nelson et al.



ECS
The
Electrochemical
Society
Advancing solid state &
electrochemical science & technology

DISCOVER
how sustainability
intersects with
electrochemistry & solid
state science research

Evaluating the Effectiveness of Course Assessments Method in Determining Student's Competency Level in Programming

Faridah Hani Mohamed Salleh^{1*}, Sulfeeza Md Drus², Saraswathy Shamini Gunasekaran³, Deshinta Arrova Dewi^{4*}

^{1,2,3} Department of Computing College of Computing & Informatics University of Tenaga Nasional Jalan IKRAM-UNITEN, Kajang, Malaysia 43000

⁴Centre of Emerging Technologies INTI International University Jalan Persiaran Perdana Bandar Baru Nilai, Negeri Sembilan

*Corresponding Author: faridahh@uniten.edu.my, deshinta.ad@newinti.edu.my

Abstract. Students who choose to study in Computer Science in Malaysia at the Degree level, basically have a diverse background of academic achievement at the final level of their secondary school. Although there are minimum requirements set for admission to Computer Science courses, students tend to demonstrate different achievements after being at university. It is possible that the method of evaluation of educational institutions is not in line with the actual skills mastered by the students. Therefore, this study takes all the test data that has been collected before and assesses which type of test is actually most effective to determine the level of student's competency in programming. This study proves that compared to other type of assessments, students' achievement in the final examination, additional lab assessment with challenging questions and test correlated positively with the student's programming competency. Our experiments also prove that if the final exam score is less than 60%, it is very likely that the student should be classified as incompetent. This suggests the final exam marks should carry significant weightage so that students do not just rely on carried marks to get a high grade.

Keywords- *Programming Assessments, Education*

1. Introduction

Programming is one of the most important technical skills in computer science. Since the idea of the development of Multimedia Super Corridor in the year 1988, IT technology development in Malaysia has continued to grow to this day. High demand for system development by local companies has prompted the industry to race to find highly skilled workers in programming. However, when graduates apply for a job, the student's academic performance does not reflect the student's actual technical skills. Often, system development companies are dealing with graduates who appear to be academically sound, but their actual technical skills are poor. Therefore, this study aims to identify the most efficient method to evaluate students in the subject of programming to ensure that the grades obtained reflect the actual programming skills. We use data from two types of studies conducted on the students who took the subject of Object-Oriented Programming (OOP) which is a compulsory subject for Software Engineering and Cyber Security students at Universiti Tenaga Nasional (UNITEN), Malaysia. OOP is



Content from this work may be used under the terms of the [Creative Commons Attribution 3.0 licence](https://creativecommons.org/licenses/by/3.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

taught using Java and students have taken 2 programming subjects before, which are C and C++. The topics taught in the OOP subject are:

- Getting Started with JAVA and Numerical Data
 - Differentiate classes and objects
 - The differences between OOP and procedural language, examples of language, benefits and weaknesses of OOP
 - Basic components of Java programs (declare a name, create and use object), import statements and package
 - Types of comments in JAVA, including Javadoc comment, types of numerical data, constant
 - Built-in classes and random number generation
 - Arithmetic expressions in Java, type casting, promotion and demotion
- Programmer-defined class
 - Passing and receiving object from method
 - toString() method and 'this' keywords
 - Visibility modifiers and static keywords
 - Constructor and method overloading
- Selection and repetition statements
 - loop-and-a-half control, compare objects
- Exceptions handling and assertions
 - Unchecked and checked exceptions, assertions
- String manipulation
- Arrays
 - Array of objects, 2-D array
- Inheritance and polymorphism
 - Abstraction, polymorphism, interface
- File Input/Output and all related classes

2. Object-Oriented Programming Assessments

Not many studies have been found related to undergraduate programming assessments. Among those found were [1] and [2]. Most studies are about peer code review [3], assessing the students' competency level in programming using Bloom's Taxonomy[4] and analyzing student error patterns [5]. The majority of studies are related to introducing automated tools [6], [7] and improvement of teaching methods. Programming assessments are usually determined by the lecturer or through discussions among faculty members. The following (Table 1) are examples of types of assessments conducted by other institutions for Object-Oriented Programming subjects that use either C ++ or Java:

Table 1. Respondent's CGPA and Achievements in Additional Mathematics and Physics

Institutions and reference sources	Assessment Task and Weighting
The University of Adelaide [8]	4 practical 4 % Major practical 5% 4 practical exams 22% Online ethics exercise 4% Workshops 5% Final written exam 60%
Swinburne University of Technology [9]	Portfolio (for Pass and Credit) 100% Semester test Pass/Fail
University of Kent [10]	Lab exercises (Pass/Fail) In-class test (Pass/Fail)
University of New Castle [11]	Tutorial / Laboratory Exercises: Laboratory Exercises

	Written Assignment: Programming Assignment 1 Programming Assignment 2 Mid Term Exam Final Exam <i>**Weighting not mentioned</i>
University of Maryland [12]	Module Exercises/Quizzes (11) 22 points Homework (6) 30 points Midterms (2) 20 points Final Project 15 points Reflections (3) 9 points Participation 4 points TOTAL 100 points, frequency in bracket

The existence of a variety of testing methods indicates the need for a basic guide in designing a course outline for OOP subjects. There is no denying that most institutions are aware of the importance of lab sessions for OOP subjects. However, it is better if we know what is the best way to make lab sessions more interesting for students because without realizing, there are students who feel stressed with lab sessions due factors such as do not understand what is learned in class before attending the lab and taking too long to complete a task that eventually leads to frustration and no longer interested in programming.

3. Experiments Design: Quantitative Analysis

Experiments have been divided into quantitative and qualitative studies. For quantitative studies, students complete several types of assessments divided into practice and written. Data from all these assessments were then processed using RapidMiner. The total number of respondents surveyed was 103 people and they were students taking Object-Oriented Programming. Student achievement in the subject of Additional Mathematics and Physics at the SPM level (Malaysian Certificate of Education) is taken to study whether there is a connection between the mastery of the subject and the mastery of programming. The two subjects were chosen because Additional Mathematics and Physics involved analytical thinking, logic, deductive reasoning and an ability to deal with complexity [13]. SPM is a national examination for all form 5 students who follow the education pathway of Malaysian government schools. Taken after 11 years of schooling, SPM is one of the pathways to tertiary education. In Malaysia, Additional Mathematics is offered as an elective to upper secondary students within the public education system. Science stream students are required to take Additional Mathematics as one of the subjects in the SPM examination, while Additional Mathematics is an optional subject for students who are from arts or commerce streams. It covers various topics including functions, quadratic equations, quadratic functions, simultaneous equations, indices, logarithms, coordinate geometry, statistics and circular measure [14]. Physics will only be taken by students who are in the Science stream. Student's CGPA is also recorded to study the relationship between CGPA and student's programming competency level. Table 2 shows the achievement of respondents. Table 3 shows the percentage for each assessment.

Challenging question which is one of the types of tests shown in Table 3 is an opportunity to get additional marks given to students who have completed assignments during the laboratory session. Laboratory sessions are allocated for 2 hours per week. Over 90% of assignments require students to complete them within the 2-hour time range. In each semester, there are usually several students who can complete the assignment in less than 2 hours. These students are given the choice of either leaving the lab early or making a challenging question. Scores from this additional test will be added to the total carried marks. The questions in this category are more difficult with students having to produce programs for real scenarios and concepts that need to be used sometimes never learned in class and they need to get the solution quickly using the remaining time available. The faster they complete the compulsory lab tasks on the day, the longer the time they get for this difficult question.

Lab sessions are conducted for 2 hours per week with the presence of a lab instructor. Students use IDE to run the program. In most weeks, students are required to demonstrate two or three questions out

of an average of 8 questions divided into 3 levels of difficulty as shown below. Students are informed that other questions that have not been demonstrated by them in class will be re-asked in the lab test.

Easy category:

- Completing fragment of codes/ program (missing JAVA keywords/ identifiers/ operators/punctuation,)
- Correct the program errors (logic errors, incompatible data types, mistakes related to over passing by value, and passing by reference, accessing non-static member variables from static methods, catch unspecific exceptions, class not found)

Medium category

- Correct the program errors but this time more than one classes may be involved
- Write a program

Challenging category

- Write a program for the real-scenario based questions or
- Topics that have not been taught in class

For the written category, year-end exams, quizzes and tests are assessments that have a mix of theoretical and programming questions and are done in writing. However, tests in this category do not involve any practical demonstrations but rely entirely on exam results.

Year-end examination of this subject according to Bloom Taxonomy level with 20% questions category remembering and understanding, 60% category applying and analyzing and another 20% from evaluating and creating category. All lab exercises at the time of this study was done manually marked without the use of any automated tool.

Table 2. Respondents' CGPA and Achievements in Additional Mathematics and Physics

CGPA	Number of Respondents	Additional Mathematics	Number of Respondents	Physics	Number of Respondents
3.5-4.0	38	A+,A,A-	21	A+,A,A-	18
3.0-3.5	36	B+,B	18	B+,B	25
2.5-3.0	11	C+,C	28	C+,C	19
2.0-2.5	2	Below C	17	Below C	16

Table 3. Assessments Categories

Category of Assessments	Assessments	Percentage
Practical	Lab test	10%
	Lab exercise	10%
	Additional lab exercises (challenging questions)	5%
	Total	25%
Written	Final examination	40%
	Quiz	10%
	Test	15%
	Total	65%

Each percentage of tests shown in Table 3 is classified into 4 categories namely Advanced, Proficient, Developing and Does Not Demonstrate based on the range of marks shown in Table 5. Competencies rating as described by [15] as shown in Table 6. The final target is to determine whether

students fall into the competent category or not. Table 5 shows the level to be achieved for both types of written and practical tests to be classified as competent. This study stipulates that students need to at least have to achieve advanced level for one type of test category to achieve competent level.

Table 4. UNITEN Grade Scale and Its Mapping to the Competency Level

UNITEN Grade Scale		
0	E	
40	D	
43	D+	Does Not Demonstrate
45	C-	
50	C	
55	C+	
60	B-	Developing
65	B	
70	B+	Proficient
75	A-	
80	A	Advanced
90	A+	

Table 5. Definition of Target

Practical	Written	Target
Advanced	Advanced	Competent
Advanced	Proficient	Competent
Proficient	Advanced	Competent

Table 6. Competencies Rating [15]

Competencies	Competencies
4 – Advanced	Has broad and deep understanding and skills, with substantial expertise and experience in this area. Can apply this competency regularly and independently and display this competency in complex, varied situations. Role model for this competency.
3 – Proficient	Has sufficient understanding and experience to operate at a full professional level with this broad range of moderately complex situations. Can generalize basic principles to effectively function in both predictable and new situations.
2 – Developing	Newly developing in this area; has a general understanding of key principles but limited or no applied experience with this competency. Is capable of using this competency with coaching and support, in simple situations.
1 – Does not demonstrate	Does not demonstrate this competency at the expected level, even with available assistance or direction from others.

4. Experiments Design: Descriptive Analysis

The experiment was divided into two parts consisting of calculations using RapidMiner and surveys. The number of respondents involved in the survey is 87 people (56 males, 31 females), slightly less than the actual number of students whose data is used for quantitative analysis. This is because not all students in the class answered this survey. The purpose of descriptive analysis is to obtain additional information that cannot be obtained from quantitative study for example, from this descriptive analysis we found that educational institutions have the potential to polish students' programming skills because most of them said still interested in attending lab, although at the same time they less interested in programming in general.

This also proves there is the potential to improve programming subjects by incorporating practical-oriented elements. Interest in practical learning is in line with the results of descriptive analysis shown on the learning style that is most interested in respondents, namely kinesthetics. Kinesthetic learners love to touch objects and are physically involved with whatever they learn [16]. Students are asked if they are interested in programming. Students are allowed to write additional reasons if they are not interested in programming. A specific question related to the interest of attending the lab was also raised. Students are also asked about the after-graduation plan by being given 3 main options along with sub-options as presented below:

- IT Technical
 - Work in IT-related companies and do works related to IT
 - Start my own IT-related business
- Non-IT Related
 - Work in the area that is not related to IT
 - Start my own business that not relate to IT
- Further study

Finally, students are also asked about the learning style which consists of several options such as kinesthetic, auditory and visual. The questions used to assess the student learning style are from [17].

5. Results of Quantitative Analysis

Fig.1 shows Decision Tree Gain Ratio [18] which shows information from different sides. From this decision tree, it was found that students who obtained the final exam result of 31.4% and above from 40%, obtained the probability to achieve a competent level. It can also be seen that the final exam score is less than 26.2% (out of 40%) is a less competent student. Additional lab assessments which is not a compulsory test but an option for students to do it to increase marks, only requires students to get 3.5% out of 5% to achieve a competent level. Apart from the additional lab exercises that are completely technical, it is also found from the decision tree that there is a high probability of achieving a competent level if the lab test achieves an almost perfect score. Referring to Fig. 2, through the experiments conducted, among all the variables, 4 variables namely final examination, additional lab assessment, test and CGPA most impact. This means that if the 4 variables are taken into account in a programming competency assessment, there is a high probability that the competency assessment results will produce accurate and reliable results. The top variables that have the highest positive correlation with the target (target) in sequence are the final examination, additional lab assessment and test. While the highest variable that is negatively correlated with the target is CGPA. The target (target) in this context is students with high competence in programming. It can be seen here that, not all students with a high CGPA (over 3.5) will also achieve a high level of competence and student achievement in the subjects

Additional Mathematics and Physics, two subjects that are said to have high relevance to logic thinking, problem solving and programming, does not significantly affect students' level of competence. If we look at the results of the study in detail, it is also found that out of 103 students, 20 people are classified as competent with 18 of them getting high marks in additional lab assessments (Table 3), which is additional training done in practical time and should be completed in a limited time.

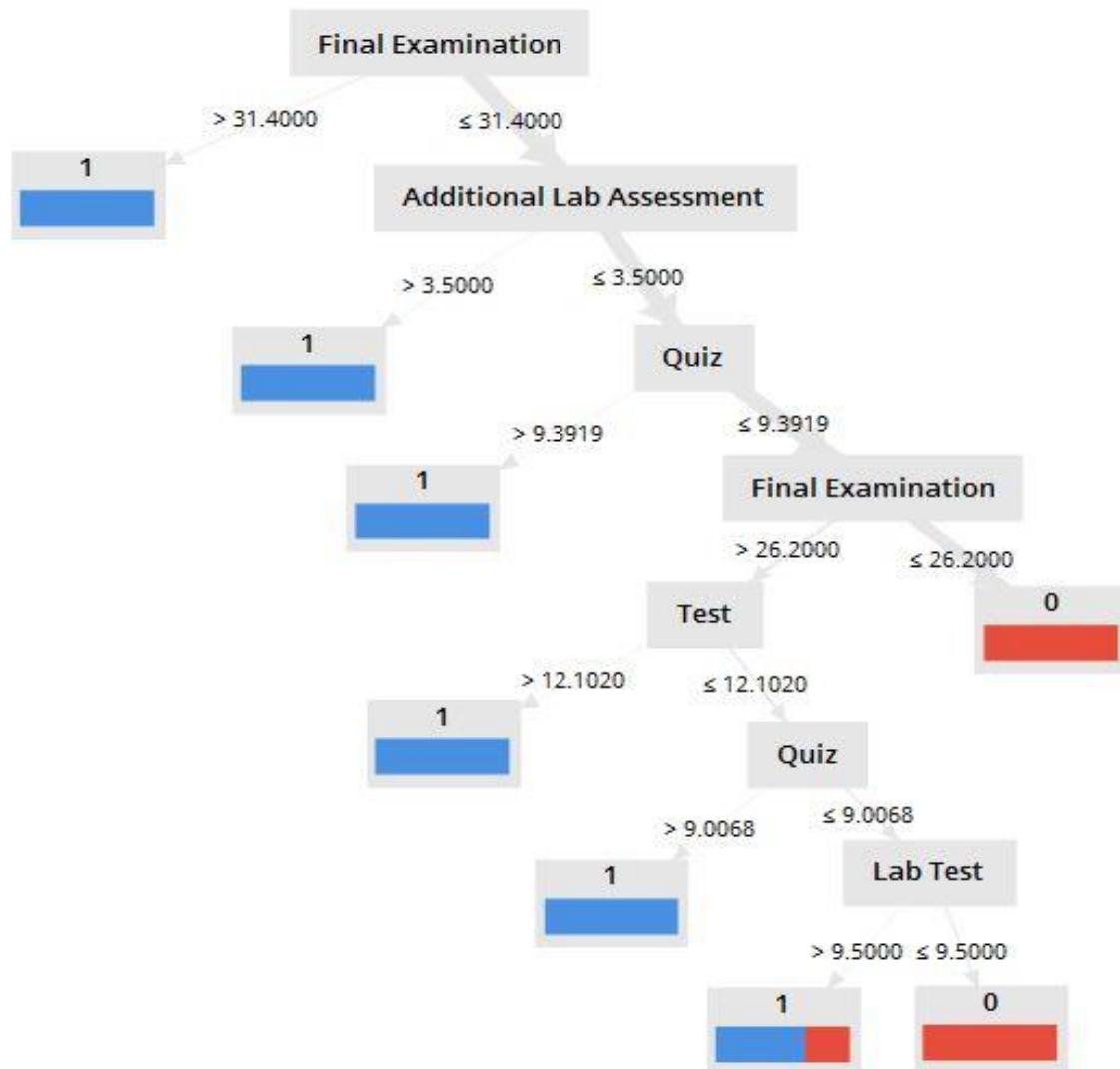


Figure 1. Decision Tree Gain Ratio

Attributes	Gender	CGPA	Physics	Add Math	Lab Test	Lab Exercie	Additional Lab Assessment	Quiz	Test	Final Examination	Target
Gender	1	0.5047	0.0611	0.1324	-0.2126	-0.3688	-0.1173	-0.0195	-0.1126	-0.3030	-0.1308
CGPA	0.5047	1	0.1142	0.0530	-0.1728	-0.3527	-0.2212	-0.0905	-0.2768	-0.3688	-0.2600
Physics	0.0611	0.1142	1	0.4299	-0.0343	-0.0963	-0.0220	0.0023	0.0974	-0.1270	-0.0626
Add Math	0.1324	0.0530	0.4299	1	0.0794	-0.0964	0.0634	0.0207	0.0813	-0.1662	0.0647
Lab Test	-0.2126	-0.1728	-0.0343	0.0794	1	0.5905	0.5174	0.4940	0.5229	0.5859	0.4074
Lab Exercie	-0.3688	-0.3527	-0.0963	-0.0964	0.5905	1	0.3286	0.4688	0.3093	0.5373	0.3038
Additional Lab Assessment	-0.1173	-0.2212	-0.0220	0.0634	0.5174	0.3286	1	0.4303	0.4898	0.5605	0.5877
Quiz	-0.0195	-0.0905	0.0023	0.0207	0.4940	0.4688	0.4303	1	0.5278	0.5149	0.4362
Test	-0.1126	-0.2768	0.0974	0.0813	0.5229	0.3093	0.4898	0.5278	1	0.5404	0.5748
Final Examination	-0.3030	-0.3688	-0.1270	-0.1662	0.5859	0.5373	0.5605	0.5149	0.5404	1	0.5952
Target	-0.1308	-0.2600	-0.0626	0.0647	0.4074	0.3038	0.5877	0.4362	0.5748	0.5952	1

Figure 2. Correlation Matrix

6. Results of Descriptive Analysis

The results of the descriptive analysis survey are shown in Table 7. The highest percentage of students are less interested in programming because they often take too long to complete a programming assignment. Overall, more than half of the students were not interested in programming. However, unexpected results are shown in the question of whether they are interested in learning in the laboratory or not. It was found that almost half of the students were actually interested in attending the lab, although at the same time they also stated that they were not interested in programming. Students' interest in practical learning is in line with most of their minutes on kinesthetic learning methods. Students' interest in practice shows that there is potential for students to enjoy programming if the right steps are taken. Similarly, the high percentage recorded by students who are interested in venturing into the field of IT technical-related work, indirectly indicates the potential of their skills can be polished. However, at the same time, it is stated that they are not interested in programming. This indicates that it is very appropriate if attention is given to strengthen students' programming skills because they are actually interested in the technical field of IT.

Table 7. Results of Descriptive Analysis

<i>Category of Assessments</i>	<i>Assessments</i>	<i>Percentage</i>
Interest in Programming	Less interest due to negative perception	14.94%
	Less interest due to long time needed to complete programming-related exercises	54.02%
	Less interest, others. Among other reasons are:	1.15%
	<i>"Unable to write program starting from scratch"</i> <i>"Did not spend time to study programming"</i> <i>"Feel that programming involves memorizing, which she does not like"</i> <i>"Easily give up"</i> <i>"Not enough time to practice"</i>	
	Have interest	39.08%
Interest in Attending Lab	Less interest due to inconvenient lab setting	3.45%
	Less interest due to bound by a set timeframe	24.14%
	Less interest due to peer pressure	21.84%
	Less interest due to lack of understanding when not doing enough preparation	14.94%
	Less interest, others	5.75%
	<i>"The lab setting is inconvenient".</i> <i>"Works have to be completed within the timeframe set by the instructor. "</i> <i>"Peer pressure (friends have completed their works but you are not)."</i> <i>"I did not understand many topics as I did not study or tried to understand the theory before attending lab. "</i> <i>"Worried and nervous"</i> <i>"Lab session too early 8 am"</i> <i>"Tend to mix Java and C++"</i> <i>"Has no friends"</i>	
	Enjoy lab session	49.43%
Plan After	IT technical	73.56%

Graduation	Non-IT technical	21.84%
	Further study	27.59%
Learning Style	Kinesthetic	58.62%
	Auditory	13.79%
	Visual	32.18%

*** The percentage is calculated by dividing total responses with total number of respondents and multiply by 100.*

7. Overall Analysis and Conclusion

In Malaysia, students who choose to pursue Computer Science consist of students with different academic achievement backgrounds. The final grade obtained by students for the subject of programming sometimes does not reflect the actual level of competence of the students due to factors such as training completed by students outside of class time, the level of difficulty of inappropriate questions, and marking methods that do not assess certain important skills. From the experimental results, it is concluded that the training should be practical oriented which can be done by not only asking students to write the program only but can also be in the form of completing the program or correcting program errors. If an automated tool is not used, checking the student program one by one will take a long time, especially if the number of students is high. CGPA should not be the only measure of the level of programming competence since it has been proven through experiments that show a negative correlation between CGPA and programming competency level. This experiment also proves that if the final exam score is less than 60%, it is very likely that the student should be classified as incompetent. This information can actually tell us that despite having a carried marks system, the final exam marks should be set to reach the minimum level so that students do not just rely on carried marks to get a high grade.

Apart from academic-related tests, the attitude of students who strive to complete additional exercises with difficult questions deserves attention because this attitude is proven to have a high positive correlation with programming competencies. This has been proven from experimental results showing that Additional Lab Exercises (Table 2) is the most influential variable. The attitude of students who make additional exercises consisting of difficult questions reflects resilience. This is in line with the opinion [19] that states “Resilience is about moving forward and adapting but grades can trap individuals into a sense of entitlement defined by a perception of themselves according to their grade average”. Resilience greatly determines the attitude of students in learning new things in the future.

Despite realizing that the test should be fully conducted in the laboratory with most questions of a practical nature, it could not be implemented due to the issue of marking. The large number of students and the assessment that do not use special software to enable marking automatically become among the barrier factors. For learning style kinesthetic or practical, learners learn best by carrying out physical activities [16]. As applied by the ScEd-AdaptiveLearning System (ScEd-ASL), which is a computer-based science learning media that accommodates students’ learning style variations [20], students are requested to straight to practice, being assertive, direct evidence, compiling material related facts and working with selecting real examples. All these features they provide for the type of kinesthetic learning.

This study also shows that there is a need for the country to make its students interested in programming. If they express interest in venturing into the field of IT after graduation, but are not interested in programming, this situation is quite worrying because it is feared that the prospective IT workforce who will one day lead the country will not have the proper skills.

What is discussed in this paper is an initial step towards preparing students before venturing into the industry. Certainly there are some other more stringent methods used by the industry to evaluate their prospective employees such as evaluating whether their candidates passionate about technology, programs as a hobby, interested in talking about technology when the topic is raised, has some side projects, learns new technologies on his or her own, opinionated about which technologies are better for various usages, can have great conversations on a variety of

topics, started programming long before university or work, and knowledge of a large variety of unrelated technologies [21].

Not all universities provide tutors who can help with marking. Therefore, testing using automated tools may be one of the best solutions at the moment. However, not all automated tools conform to the testing concept implemented by an institution. Focus should not be given only to assessments but should also be on teaching methods in order to support preparing students in completing the assessments. It can be seen here that learning that shows the relevance of the concepts learned to the actual scenario, is more efficient than a long lecture session in class.

The percentage allocated to the final exam conducted in writing before should be reduced by including a practical assessment whose questions require the creation of an algorithm and write a program based on that algorithm. Students should be allowed to obtain information from any online source. Students need a basic program to gain confidence to generate better ideas. Through the descriptive analysis of this study, it has been shown that students are less interested in programming because asking students to write programs from scratch is difficult for some students.

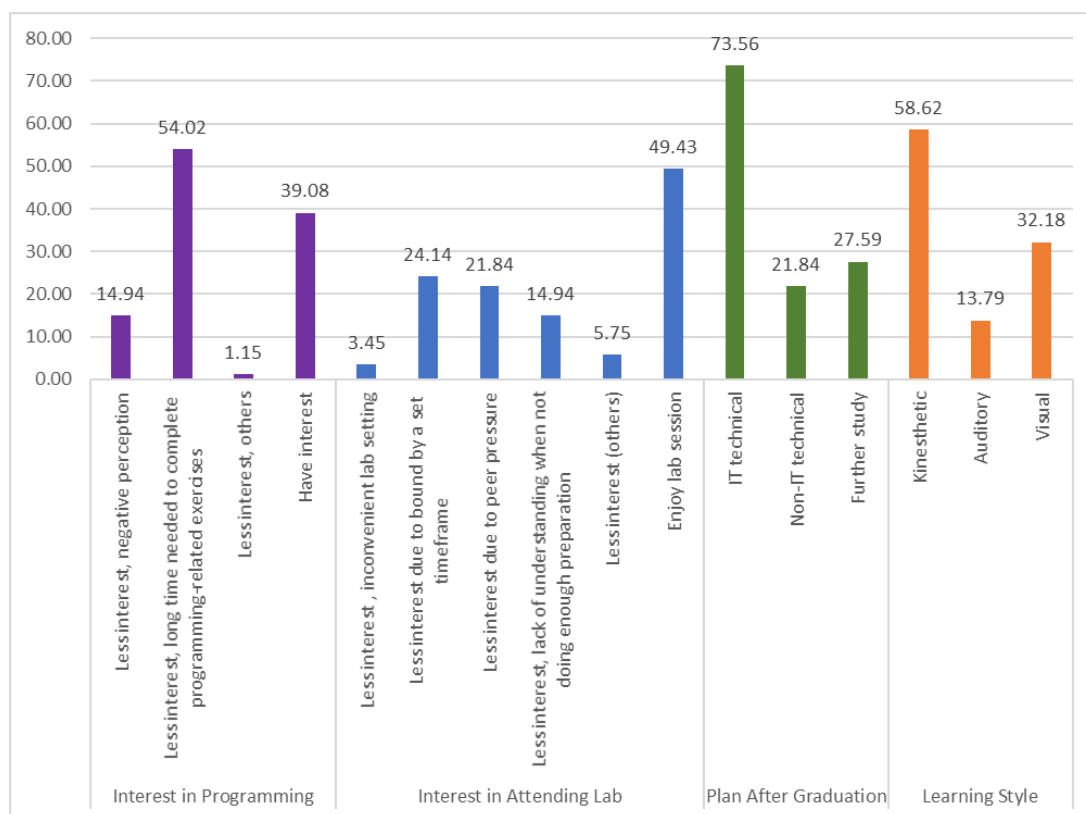


Figure 3. Summary of Survey Results

Acknowledgment

This work was supported by the Universiti Tenaga Nasional BOLD Grant RJ010517844/006.

REFERENCES

- [1] R. A. & N. I. Norazlina Khamis, Sufian Idris, "Assessing Object-oriented Programming Skills in the Core Education of Computer Science and Information Technology: Introducing New Possible Approach," *WSEAS Trans. Comput.*, vol. 7, no. 9, pp. 1427–1436, 2008, [Online]. Available: <http://eprints.um.edu.my/5732/>.
- [2] B. A. B. Roger McDermott, Andrew Luxton-Reilly, Jacqueline L. Whalley, "Developing Assessments to Determine Mastery of Programming Fundamentals Title," *Proceedings of the 2017 ITiCSE Conference on Working Group Reports*, 2017. <https://dl.acm.org/doi/10.1145/3174781.3174784>.
- [3] Q. S. W. R. Liu, "Formative assessment of programming language learning based on peer code review: Implementation and experience report," *TSINGHUA Sci. Technol.*, vol. 24, no. 4, pp. 423–434, 2019, [Online]. Available: <https://ieeexplore-ieee-org.ezproxy.uniten.edu.my/stamp/stamp.jsp?tp=&arnumber=8660528>.
- [4] Z. U. A. L. M. J. A. H. A. J. S. F. Saleem, "A Rule-Based Method for Cognitive Competency Assessment in Computer Programming Using Bloom's Taxonomy," *IEEE Access*, vol. 7, 2019.
- [5] E. L. N. Kiran, I. Indian Institute of Technology Bombay, Mumbai, Maharashtra, and ; Kannan M. Moudgalya, "Evaluation of Programming Competency Using Student Error Patterns," in *2015 International Conference on Learning and Teaching in Computing and Engineering*, 2015.
- [6] J. A. C. Daniel Galan, Ruben Heradio, Hector Varga, Ismael Abad, "Automated Assessment of Computer Programming Practices: The 8-Years UNED Experience," *IEEE Access*, vol. 7, 2019.
- [7] M. B. Igor Mekterović , Ljiljana Brkić , Boris Milašinović, "Building a Comprehensive Automated Programming Assessment System," *IEEE Access*, vol. 8, 2020.
- [8] "The Univeristy of Adelaide, COMP SCI 1102 - Object Oriented Programming," 2020. <https://www.adelaide.edu.au/course-outlines/105877/1/sem-2/> (accessed Nov. 08, 2020).
- [9] "Swinburne University of Technology, Object-Oriented Programming," 2020. <https://www.swinburne.edu.au/study/courses/units/Object-Oriented-Programming-COS20007/local> (accessed Nov. 08, 2020).
- [10] "University of Kent, Object-Oriented Programming - CO881," 2020. <https://www.kent.ac.uk/courses/modules/module/CO881> (accessed Nov. 08, 2020).
- [11] "University of New Castle , SENG1110 Object-Oriented Programming," 2020. <https://www.newcastle.edu.au/course/SENG1110%0A%0A> (accessed Nov. 09, 2020).
- [12] J. A. Westgard, "University of Maryland, INST326: Object-Oriented Programming," https://ischool.umd.edu/sites/default/files/syllabi/inst326_0103_fall_2018_westgard.pdf (accessed Nov. 10, 2020).
- [13] R. Landau, "Computational Physics: A Better Model for Physics Education?," *Comput. Sci. Eng.*, vol. 8, no. 5, pp. 22–30, 2006.
- [14] "Additional Mathematics - Wikipedia." https://en.wikipedia.org/wiki/Additional_Mathematics (accessed Sep. 26, 2020).
- [15] "ePerformance Rating Scales OVERALL PERFORMANCE ratings." Accessed: Sep. 26, 2020. [Online]. Available: <https://hls.harvard.edu/content/uploads/2015/04/ePerformance-Ratings-QR.pdf>.

- [16] M. G. Zahra Nasiri, Samane Gharekhani, "Relationship between Learning Style and Academic Status of Babol Dental Students," *Electron Physician*, vol. 8(5), 2016, [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4930252/>.
- [17] "Determine My Learning Style." <https://www.how-to-study.com/learning-style-assessment/> (accessed Sep. 26, 2020). [18] R. H. A Alsagheer, A. F. H Alharan, and A. S. A Al-Haboobi, "International Journal of Computer Science and Mobile Computing Popular Decision Tree Algorithms of Data Mining Techniques: A Review," 2017. Accessed: Nov. 02, 2020. [Online]. Available: www.ijcsmc.com.
- [19] "Focus on resilience, not grades | The Star." <https://www.thestar.com.my/news/education/2020/11/01/focus-on-resilience-not-grades> (accessed Nov. 02, 2020).
- [20] Z. Zulfiani, I. P. Suwarna, and S. Miranto, "Science Education Adaptive Learning System As A Computer-Based Science Learning With Learning Style Variations," *J. Balt. Sci. Educ.*, vol. 17, no. 4, pp. 711–727, 2018, [Online]. Available: http://www.scientiasocialis.lt/jbse/files/pdf/vol17/711-727.Zulfiani_JBSE_Vol.17_No.4.pdf.
- [21] Katherine Noyes, "What Makes a Good Programmer?," *LinuxInsider*, 2008. <https://linuxinsider.com/story/what-makes-a-good-programmer-61588.html> (accessed Nov. 07, 2020).