



An Assessment Of Students' Academic Achievement In Learning HTML, PHP, JAVASCRIPT, JQUERY, AJAX Programming Languages

¹Iriakuma T. Christopher & ²Zibs, Dowell Feremo Woripere

^{1,2}Department of Computer Science
Federal Polytechnic of Oil and Gas Bonny, Rivers State, Nigeria

¹Kristex14@gmail.com/ +2347010860132

²dowelwore@gmail.com/ +2348136938817

ABSTRACT

Programming is essential for students' development in computer science. The study assessed students' academic achievement in learning HTML, PHP, JAVASCRIPT, JQUERY, AJAX programming languages. Two research questions and one hypothesis was used in the study. The Quasi experimental research design was adopted for the study. The researcher used a population of ten (10) year two undergraduate students from different polytechnics. The students had a pre-knowledge on the use of programming languages. The population was adopted as sample for the study. After two weeks intensive training on basis of HTML, PHP, JAVASCRIPT, JQUERY, AJAX programming languages, a post test was administered to students to ascertain their level of comprehension on the various programming languages. The scores obtained was analyzed using simple mean and the hypothesis was analyzed using Pearson Product Moment Correlation Coefficient. Findings obtained from research question 1, table 1 showed that the mean scores of students taught using HTML and PHP programming language are 55.6 and 52.0 respectively. Findings also showed that the mean scores of students taught using JAVASCRIPT, JQUEERRY and AJAX programming language are 66.8, 60.7 and 56.8 respectively. Further findings revealed that the calculated value of -0.001 is less than r-tabulated value of 0.432858. This implies that there is no significant correlation between scores of students basic JAVAscript and AJAX programming language. Finally, it was recommended that tertiary institutions should create the integrated cloud platform to accommodate various programming language to enable students interact and educate themselves as future programmers.

Keywords: Academic Achievement, HTML, PHP, JAVASCRIPT, JQUERY, AJAX, Programming Languages

INTRODUCTION

With the widespread development of computer technology in the era of open world and globalization, the needs for the use of computers are increasing. This requires the need of experts in the field of Information and Communications Technology (ICT) such as software programming, database, software engineering, computer networking and creative multimedia (Ungku Harun and Rashid, 2004). In fact, the research findings shows that the industry's' needs for software engineers is high and this indicates that the programming personnel are a critical requirement for the industries (Ahmad Rizal Madar, Nurliana Musa,

and Yahya Buntat, 2007). Therefore to meet industry's' needs, programming course is an essential component of the curriculum to be studied, not only in the field of Information Technology, but also required in most program areas such as Science, Mathematics, and Engineering at tertiary levels McCracken et. al. As a leader in the field of 'Technical and Vocational Education and Training' (TVET), polytechnic was also involved in this challenge. Thus, the engineering programs offered are not only engineering-based curriculum alone, in fact, the subject of ICT also inserted, for instance the subject of Fundamental Programming (EC201) need to be studied under the programs offered in the field of Electrical Engineering. However, the question arises, were the students were really being prepared to meet the critical needs of industry expertise in programming? Basically, the subject is said to be difficult, complex and categorized as one of the seven grand challenges in computing education. Students especially novices, have problems in reading, tracking, writing and designing a simple code fragment. This causes students to have the attitude of depending on others to complete the task given and consequently become indolent and do not have the courage to learn and just expect marks of sympathy from lecturer

Computer programming courses are among the main requirements of study plans in higher education not only in the field of Computer Science and Information Technology, but also in Science, Mathematics, and Engineering field. This is because, in this modern age, knowledge of computer technology and programming in various fields of work is a necessity to fulfil the demand of the industry, which is to have expertise in the field of Information and Communications Technology (ICT) such as software programming, database, software engineering, computer networking and creative multimedia as stated by Rosminah and Zamzuri (2012). Nevertheless, they added that learning computer programming is very complicated and brings a huge challenge to many students. Writing a computer program is not an easy task. According to Moström (2011), novice students must understand the problem, formulate the solution using standard problem solving techniques, and write down the solution to solve the problem using a programming language in such a way that a computer can follow the instructions. In reality, learning a programming language is much more complex than this very simplified description since (Baist & Pamungkas, 2017) mentioned that this skill requires other skills from designing the algorithm, writing program as well as understanding the syntax of programming language. Moreover, (Abdul et al., 2018) mentioned that other than learning the basic concept in language syntax, the students are required to comprehend the language structure and style, able to interpret requirement into an algorithm, translate the program code with the correct syntax, find bugs, apply correct logic and use program development environment. As stated by (Gomes et al., 2012), several causes that contribute to learning difficulties in computer programming include the students' background knowledge and attitudes, teaching and learning methods, as well as social context. They also added that the lack of problem solving skills and mathematical background in students as well as programming question proposed by lecturers beyond the students' cognitive development are among the factors of poor performance in programming courses. Another study by (Xinogalos, 2016) pointed out the significance of preparing high-quality instructional materials that are not only based on the teaching and learning design of the course but more importantly, dispels students' difficulties and misconceptions on computer programming. Furthermore, the study suggested that changes are necessary for students' study methods and attitudes as well as the traditional teaching approaches. Instructional methods such as hands-on practice on learning programming and contemporary innovative teaching and learning techniques must be used to increase students' interest in computer education.

Statement of Problem

The learning of Programming language is problematic causing students to experience phobia during learning process. Lack of problem-solving skill and limited surface knowledge of programs is among the main factors that lead to difficulties in learning computer programming. As stated by (Abdul et al., 2018), novice students often approach programming line by line instead of using meaningful structured programs. In a study by (Mohamed et al., 2011), they found that novices feel programming is very complicated as it requires too much knowledge and skills to be mastered simultaneously and at an early

stage. Therefore, most of them start to write computer program without analysing, designing and understanding the problem thoroughly as mentioned by (Oroma et al., 2012). Many of them do not have the abilities to interpret the problem appropriately since they have a weak approach to problem-solving, thus making most students take the easy way out by plagiarising their friends to complete the assignment given or even worse, (Rosminah & Zamzuri, 2012) said that some of them just expect marks out of sympathy from their programming subject instructors. Their study also discovered that students' difficulties in programming are due to three interrelated issues, which are:

- (1) Difficulty in understanding the basic concepts of programming structure,
- (2) Problem in designing a complete task, and
- (3) Syntax of programming languages.

They also added that novices are not only lacking the abilities to apply abstract concepts of programming to solve the problem given, but also skill in mastering various processes such as planning, analyzing, designing, editing, compiling and debugging program code.

Purpose of the Study

The study looked at the assessment of students' academic achievement in learning HTML, PHP, JAVASCRIPT, JQUERY, AJAX programming languages. Specifically, the study seeks to:

1. Determine the mean achievement scores of students taught HTML and PHP.
2. Determine the average scores of students learning basic JAVASCRIPT, JQUEERRY and AJAX programming language.

Research Questions

The following research questions were adopted and used for the study:

1. What are the mean achievement scores of students taught HTML and PHP.
2. What are the average scores of students learning basic JAVASCRIPT, JQUEERRY and AJAX programming language?

Hypothesis

The null hypothesis was adopted and used for the study. The null hypothesis was tested under 0.05 level of significance.

There is no correlation between scores of students basic JAVAscript and AJAX programming language.

Scope of the Study

The study is limited to the use of HTML, PHP, JAVASCRIPT, JQUERY, AJAX programming languages. The study used basically undergraduate students at level two hundred in selected polytechnics in Rivers State.

Literature Review

Programming Language

The activity in reviewing a program source code is not the same as reviewing ordinary documents and many problems in program comprehension arise due to the use of textual representation as the primary source of information. Programs are often in the form of a hierarchical structure, but the actual behaviour of a program cannot be reflected as it is represented in textual forms. Although many methods and tools have been proposed to represent source code including cross-referencing, developers' program domain knowledge, syntax highlighting and tools, comments, dependence graph, slicing, ripple analysis and program decomposition, experience have shown that textual presentation is the most suitable to represent a program (Krinke, 2004). However, the problems still exist if the source code is used in a form of text-based due to the source code. Thus, this section discusses the issues that arisen in learning a program based on the nature of the programming language. One of the issues discussed regarding the effectiveness of students mastering a programming language is the nature of programming, which plays an important role in determining the effectiveness of students' ability to master a programming language. Pears et al. (2007) reported that most institutions use an object-oriented language, but many use Java, C and C ++, languages to teach procedural programming, whereas less than 10% of institutions teach functional programming. Despite the popularity of such languages, there has been much debate about the suitability

of these languages for education, especially when purposes, in contrast to others designed with this specific purpose (such as Python, Logo, Eiffel, and Pascal). There are interrelated types of programming's nature of difficulties while learning to program as stated in the previous studies. Rosminah and Zamzuri (2012) identified three issues that should be addressed by educators, which are: the lack of understanding of the basic concepts of programming structure; problem in designing a program to complete a specific task and; inability to identify the syntax of programming languages. This is in contrast to Bosse and Gerosa (2017), which is more focused on the ability of students in using computers and performing system development tasks. The present discussion focuses more on the opinions outlined by Xinogalos (2016), which has outlined five problems faced by students related to programming nature, namely: developing an algorithm, transferring an algorithm to a programming language, programming structures, modularisation, and; testing and debugging. Guided by Xinogalos (2016), these five issues were discussed based on the study on previous work as well as observations of more than 10 years in the world of programming education and introducing programming to novices. These languages are not designed specifically for educational.

Another factor that influences the difficulties of students in learning programming subjects is poor learning style. Each student has a different learning style. Some of the students prefer group discussion, while others favour independent study. Nevertheless, regardless of any type of learning style, the most important thing is the way students' think. According to (Rahmat et al., 2012), learning to programming involves a different way of thinking. A study by (Oroma et al., 2012) discovered that most people find it easier to learn a certain subject that they are familiar with rather than learning a new subject. This is because the learning process is usually built on previous knowledge as well as experiences, and since computer programming subjects are not related to any former subjects in their primary or secondary schools, many students feel that this new subject is complicated. By thinking this way, (Rahmat et al., 2012) said that most students make a minimal initiative in studying programming subject and depend on assistance from their lecturers, friends or copying computer programs from Internet sources to solve programming assignments without exactly understanding the task given. Novices also rely completely on lecture notes, slides and answer schemes prepared by lecturers although these reference materials are insufficient to increase students' understanding of computer programming. Finally, a study by (Gomes et al., 2012) revealed the correlation between programming subject grades and student motivation. Due to the negative perception of students on programming, most students put little effort and are demotivated to learn programming with an open mind. This attitude accompanied by weaknesses in problem-solving skills has resulted in low grades in introductory programming courses. For that reason, (Gomes et al., 2012) suggested that programming instructors should observe closely their students' motivation level and try to inspire them through innovative way of teaching styles. This can be done by increasing the number of practical hands-on sessions and two-way interactive tutorial discussion on the basic concepts of programming instead of just delivering the theory through abstract and dull lecture sessions. However, they added that this instructional method is only suitable for a small group of around 20 students or less in a class. With this approach, lecturers can get to know their students more closely and aware of their strengths and weaknesses in computer programming.

METHODS

The Quasi experimental research design was adopted for the study. The researcher used a population of ten (10) year two undergraduate students from different polytechnics. The students had a pre-knowledge on the use of programming languages. The population was adopted as sample for the study. After two weeks intensive training on basis of HTML, PHP, JAVASCRIPT, JQUERY, AJAX programming languages, a post test was administered to students to ascertain their level of comprehension on the various programming languages. The scores obtained was analyzed using simple mean and the hypothesis was analyzed using Pearson Product Moment Correlation Coefficient.

RESULTS AND FINDINGS

Table 1: Achievement Scores of Students taught HTML, PHP, JAVASCRIPT, JQUERY, AJAX programming languages

S/No	HTML	PHP	JAVASCRIPT	JQUERY	AJAX
1	75	55	34	56	54
2	65	67	67	75	65
3	55	56	65	60	43
4	50	70	77	56	55
5	56	34	88	54	67
6	70	67	65	87	75
7	65	87	85	54	44
8	35	24	67	55	50
9	40	25	55	65	55
10	45	35	65	45	60
Total	556	520	668	607	568

Table 1 shows the entire scores of students taught HTML, PHP, JAVASCRIPT, JQUERY, AJAX programming languages.

Research Question 1: *What are the mean achievement scores of students taught HTML and PHP?*

Table 2: Mean achievement scores of students taught HTML and PHP

S/No	HTML	PHP
1	75	55
2	65	67
3	55	56
4	50	70
5	56	34
6	70	67
7	65	87
8	35	24
9	40	25
10	45	35
Total	Mean = 556/10= 55.6	Mean= 520/10= 52.0

Findings obtained from research question 1, table 1 showed that the mean scores of students taught using HTML and PHP programming language are 55.6 and 52.0 respectively.

Research Question 2: *What are the average scores of students learning basic JAVASCRIPT, JQUEERRY and AJAX programming language?*

Table 3: Average scores of students learning basic JAVASCRIPT, JQUEERRY and AJAX programming language

S/No	JAVASCRIPT	JQUERY	AJAX
1	34	56	54
2	67	75	65
3	65	60	43
4	77	56	55
5	88	54	67
6	65	87	75
7	85	54	44
8	67	55	50
9	55	65	55
10	65	45	60
Total	Mean=668/10 = 66.8	Mean=607/10=60.7	Mean= 568/10=56.8

Findings obtained from research question 2, table 2 showed that the mean scores of students taught using JAVASCRIPT, JQUEERRY and AJAX programming language are 66.8, 60.7 and 56.8 respectively.

Hypothesis

There is no correlation between scores of students basic JAVAscript and AJAX programming language.

Table 4: Correlation between scores of students basic JAVAscript and AJAX programming language

S/No	JQUERY (X)	AJAX (Y)	(X) ²	(Y) ²	XY
1	56	54	3136	2916	3024
2	75	65	5625	4225	4875
3	60	43	3600	1849	2580
4	56	55	3136	3025	3080
5	54	67	2916	1156	3618
6	87	75	7569	5625	6525
7	54	44	2916	1936	2376
8	55	50	3025	576	2750
9	65	55	4225	625	3575
10	45	60	2025	1225	2700
Total	Σ=607	Σ=568	Σ =38,173	Σ =23,158	Σ=35,103

Using the Pearson Correlation Formula

$$r = \frac{N \sum XY - (\sum X)(\sum Y)}{\sqrt{((N \sum X^2 - (\sum X)^2)((N \sum Y^2 - (\sum Y)^2))}}$$

r= -0.001

Based on the findings obtained from the calculated value of -0.001, which is less than r-tabulated value of 0.432858, the null hypothesis was accepted at 0.05 level of significance. This implies that there is no significant correlation between scores of students basic JAVAscript and AJAX programming language.

DISCUSSION OF FINDINGS

Findings obtained from research question 1, table 1 showed that the mean scores of students taught using HTML and PHP programming language are 55.6 and 52.0 respectively. This is in line with the view of Rahmat et al., 2012 that opined that programming language creates variation among students in academic achievements. Also, findings showed that the mean scores of students taught using JAVASCRIPT, JQUEERRY and AJAX programming language are 66.8, 60.7 and 56.8 respectively. The high performance may be as a result of teaching method adopted by the teacher. Further findings show that calculated value of -0.001, which is less than r-tabulated value of 0.432858, the null hypothesis was accepted at 0.05 level of significance. This implies that there is no significant correlation between scores of students’ basic JAVA script and AJAX programming language.

CONCLUSION

In all, the work was able to establish that students performed averagely in their first interface on the use of various programming language. Students’ skill on programming improved more on the use of JAVAscript.

RECOMMENDATIONS

Finally, it was recommended that tertiary institutions should create the integrated cloud platform to accommodate various programming language to enable students interact and educate themselves as future programmers.

REFERENCES

Abdul, T. F. B., Anuar, N., & Said, M. R. F. (2018). How the nature of programming and learning materials affects novice learner’s motivation and programming ability. ACM International Conference Proceeding Series, 124–128. <https://doi.org/10.1145/3178158.3178184>

- A. Ungku Harun Al'Rashid, "Meeting the demands of global firms: Survey Finding," 2004. [Online]. Available: http://www.epu.gov.my/html/themes/epu/images/_common/pdf/seminars/slide-ungku_harun.pdf.
- Ahmad Rizal Madar, Nurliana Musa, and Yahya Buntat, "Kesan model pembelajaran berasaskan kaedah penyelesaian masalah ke atas pelajar berbeza gaya kognitif dan kemahiran logik," (The effect of using learning model based on problem solving method on students with different cognitive style and logic ability) in 2nd International Malaysian Educational Technology Convention, 2007.
- M. McCracken et. al, "A multi-national , multiinstitutional study of assessment of programming skills of first-year CS students," ACM SIGCSE Bulletin, vol. 33, no. 4, pp. 125– 180, 2001.
- A. McGetrick, R. Borle, R. Ibbett, J. Llyod, G. Lovegrove, and K. Mander, "Grand challenges in computing: Education-A Summary," The Computer Journal, vol. 48, no. 1, pp. 42–48, Jan. 2005.
- Rosminah, M. D., Zamzuri, M. A. (2012). Difficulties in learning Programming: Views of students. 1st International Conference on Current Issues in Education (ICCIE2012), October 2014, 74–78. <https://doi.org/10.13140/2.1.1055.7441>
- Moström, J. E. (2011). A study of Student Problems in Learning to Program. <http://urn.kb.se/resolve?urn=urn:nbn:se:umu:diva-48216%5Cnhttp://umu.divaportal.org/smash/get/diva2:447104/FULLTEXT02%5Cnhttp://umu.divaportal.org/smash/record.jsf?pid=diva2:447104>
- Baist, A., & Pamungkas, A. S. (2017). Analysis of Student Difficulties in Computer Programming. VOLT : Jurnal Ilmiah Pendidikan Teknik Elektro, 2(2), 81. <https://doi.org/10.30870/volt.v2i2.2211>
- Bosse, Y., & Gerosa, M. A. (2017). Why is programming so difficult to learn? ACM SIGSOFT Software Engineering Notes, 41(6), 1–6. <https://doi.org/10.1145/3011286.3011301>
- Abdul, T. F. B., Anuar, N., & Said, M. R. F. (2018). How the nature of programming and learning materials affects novice learner's motivation and programming ability. ACM International Conference Proceeding Series, 124–128. <https://doi.org/10.1145/3178158.3178184>
- Krinke, J. (2004). Visualization of program dependence and slices. 20th IEEE International Conference on Software Maintenance, 2004. Proceedings., 168–177. <https://doi.org/10.1109/ICSM.2004.1357801>
- Gomes, A. J., Santos, Á. N., & Mendes, A. J. (2012). A study on students' behaviours and attitudes towards learning to program. Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE, 132–137. <https://doi.org/10.1145/2325296.2325331>
- Rahmat, M., Shahrani, S., Latih, R., Yatim, N. F. M., Zainal, N. F. A., & Rahman, R. A. (2012). Major Problems in Basic Programming that Influence Student Performance. Procedia - Social and Behavioral Sciences, 59, 287–296. <https://doi.org/10.1016/j.sbspro.2012.09.277>
- Mohamed, S., Hamilton, M., & D'Souza, D. (2011). Understanding novice programmer difficulties via guided learning. ITiCSE'11 - Proceedings of the 16th Annual Conference
- Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., Devlin, M., & Paterson, J. (2007). A survey of literature on the teaching of introductory programming. ACM SIGCSE Bulletin, 39(4), 204–223. <https://doi.org/10.1145/1345375.1345441>
- Oroma, J., Wanga, H., & Ngumbuke, F. (2012). Challenges of Teaching and Learning Computer Programming in a Developing Country: Lessons From Tanzania. INTED2012 Proceedings, October, 3820–3826. <https://doi.org/10.13140/2.1.3836.6407>
- Rosminah, M. D., Zamzuri, M. A. (2012). Difficulties in learning Programming: Views of students. 1st International Conference on Current Issues in Education (ICCIE2012), October 2014, 74–78. <https://doi.org/10.13140/2.1.1055.7441>
- Xinogalos, S. (2016). Designing and deploying programming courses: Strategies, tools, difficulties and pedagogy. Education and Information Technologies, 21(3), 559–588. <https://doi.org/10.1007/s10639-014-9341-9>