VOL. 18 NO. 9
ISSN: 2278-1897
SEPTEMBER, 2020

African Scholar Publications & Research International

African Scholars Journal of pure and Applied Science (JPAS-9)

Examining Causes of Computer Science Students' Poor Hypothetical Performance in Programming Languages in Nigeria Tertiary Institutions

Adeleke Israel Adewale
Computer Science Department, EACOED, Oyo

Abstract

This study examined causes of poor academic performance of students in programming languages among computer science students in Nigeria Tertiary Institutions. It is clear that students' performance in programming languages yearly is always woeful when compared with other courses and this occurred as a result of many challenges faced by the students in the way programming languages is learned and taught. This therefore demands for research to actually know the cause and the remedy. Three research hypotheses were raised to determine the factors militating against such poor academic performance of students in programming languages among computer science students. It was discovered that factors that influence the poor performance of students is on the part of the students and lecturers owning to their poor attention given to programming languages in the process of learning and practical. Meanwhile, In the first hypothesis, the calculated value which is 63.934 is greater than the critical value which is 10.64, which means non–challant attitude by the students is a major factor militating against poor academic performance in programming languages. In the second hypothesis, the calculated value (90.911) is greater the critical value 14.68, this shows that lecturers' inefficiency in term of availability for practical, method and understanding of concept is a factors that contribute to the poor performance of student in programming languages. Also, in the third hypothesis, the calculated value is 43.999 which is greater than the critical value 14.68 tells that complexity in term of coding, compiling and debugging errors is a root of poor academic performance in programming languages courses.

Keywords: *students, lecturers, learning, programming languages and academic performance*

Introduction

Programming has been a challenging subject for computer science students and a programmer that should write code must have strong analytical and reasoning skills to program effectively (Islam, Sheikh, Fatima and Alvi, 2019). As a result of this challenge most students' dropout and those that manage to continue find programming language as a course difficult to scale through. Meanwhile, the course requires instructor with outstanding skills coupled with the students that are ready to study with extra commitment and logical reasoning. That is, instructor must prepare their lectures accordingly and employ several innovative techniques to develop analytical and problem solving skills into the students. Even with this, it has been noticed that there have not been any identified major problem facing computer science students in programming language. Programming languages are computer languages that can be used by computer programmers to write specific sets of instructions for the computers microprocessor to read in order to accomplish a specific task. Computer programming is the process of designing and building an executable computer program to accomplish a specific computing result or to perform a specific task. Programming involves tasks such as: analysis, generating algorithms, profiling algorithms' accuracy and resource consumption, and the implementation of algorithms in a chosen programming language (Bebbington, 2014). In another vein, it is the process that professionals use to write code that instructs how a computer performs any given task with desired result. Programming languages is important for learning to innovate, create solutions for global problems, to automate, collect, manage, calculate, and analyze the processing of data and information accurately. It is a way of instructing electronic machines to perform tasks, solve problems and provide human interactivity. That is, computer programming breaks down a problem to small, manages masses that can be interpreted by a computer, which is achieved either by a compiler (for instance, C++ scripts are compiled to

make executable programs) or interpreted at run-time (i.e., Javascript is a run-time language).

Today, devices like robots, artificial intelligence, machine learning, Internet Of Things (IOT), Cloud Computing, etc. are the technology through the handy work of programming language developed by computer programmers as professionals. Programming languages help to create solutions that are environmentally friendly and support the highest level of living standard for human. Programmers use programming languages to communicate with computers. Many different languages exist, and each one has its own unique features, though they all share some similarities. Meanwhile, each language is different and each may be best suited for a certain purpose with certain industries. Some programming languages are used to create programs, to solve problems or interpret data. Other programming languages are more suitable for making software or apps that entertain. With a strong need for unique and diverse programming languages, it is virtually impossible to create a single universal programming language that meets all needs (Aman, 2020).

According to Putano (2019), top programming languages are Java, C, C++, Python, C#, Javascript and Vb.Net with higher percentage of usage while Visual BASIC, PL/SQL and Delphi have lowest percentage of usage. Some of the most common languages include HTML, Javascript, Python, CSS, Ruby, PHP, C++ and SQL. Some of these languages are easier and more desirable to learn than others, although that doesn't make them any less useful. According to Aman (2020), these programming languages are categorized as front-end and back-end development. Front-end development involves working with code that produces the elements that users can see and interact with; it's all about how a website looks and feels. Back-end web developers make sure the website works as it should do, and the code they write is normally invisible to users. Back-end web developers work with databases that store information such as customer details, and servers which are where databases (virtually) live. Both front-end and back-end computer programming jobs are creative in their own ways, but front-end developers often need to have a visual eye so they can judge what will work best for site users. Basic examples of front-end development include that pretty-looking font on your favorite web page, or that slider on the homepage of a news site, or even the dropdown menu where you can choose your preferred option. Everything from the color scheme to the layout to the positioning to the typography is a front-end developer's responsibility. The three main languages

front-end developers need to know are HTML, Javascript and CSS while common back-end languages include PHP, Ruby, Python, and .NET.

Literature Review

Programming is a subject been studying in computer science department and other related fields like Information Technology, Electrical/Electronic Engineering, Mechanical engineering and some other courses in Nigeria university. This is so because of its relevance in science and technology. However, both lecturers and students often face difficulties already on the basic courses because of the technicality and logical reasoning that it entails. Essi, Kirsti and Hannu-Matti (2005) studied the difficulties in learning programming in order to support developing learning materials for basic programming courses. (Islam, Sheikh, Fatima and Alvi, 2019) studied to identify the major challenges faced by programming students. The study was to help instructor in development of appropriate materials students who just completed their first programming course. To determine what are the most challenging concepts for students to learn and when do the students feel most comfortable learning. The correlations among students response were determined and k-mode clustering was also performed. It was concluded that flipped/ inverted class room model is the most appropriate model for teaching programming course. Mohammed, Elhaddad and Mohammed (2017) analyzed the problem of teaching programming within a computer science department at Omar Elmukhtar University (OMU) through the application of soft system methodology (SSM), in addition to investigate the effects and benefits of using such an approach in order to suggest a solution. Through constructing a root definition and conceptual model, they were able to provide a standard to which the current organization can be compared against, and therefore, achieved a better understanding regarding which changes need to be carried out in order to attain the system's desired results. Schulte and Bennedsen (2006) compared C/C++, C#; Java, Pascal, GW Basic and JavaScript based on different criteria and gathered teacher's opinion about what language should be taught in programming courses and recommended Java as the most widely used language for teaching programming (Davies, Polackwahi and Anewalt, 2011). Bennedsen and Schulte (2010) reported on an experiment undertaken in order to evaluate the effect of a program visualization tool for helping students to better understand the dynamics of object-oriented programs with the use of

BlueJ's debugger and object inspector as concrete tool. The study was done as a control-group experiment in an introductory programming course. Sheikh and Islam (2016) determined which programming language languages should be taught to computer science students at introductory level and analyzed the selected programming languages based on different parameters and provides recommendations on the selection of programming language. The recommendations are; the language should have documentation, tutorials, and community's support and developer forums widely available, minimum platform requirements, extensions available in abundance for specific tasks such as drivers for hardware interfacing, database connectivity and should have extensions available in abundance for specific tasks such as drivers for hardware interfacing and database connectivity.

STATEMENT OF THE PROBLEM

Finding had revealed that computer programming is gaining a lot of popularity in the past three decades and thus many students these days want to opt for a computer science stream in order to get a job at their dream. However, in computer Science course, a high dropout and failure rate is increasing due to difficulty encountered by students in tertiary institutions when learning programming languages as major course in computer science department. Students' failure rate is very consistent in programming languages which call for both students and lecturers attention for transformation. This might be due to their attitudes and lack of understanding semantics of the program. That is, how to understand the meaning of the program element in programming language. Debugging the already written code is another difficulty confronted with writing program codes and this relates to lack of background knowledge which also include function and parameters, concepts, principles, assigning variable, decision structure, loops and syntax due to poor programming skills. Thus, there is need to carry out research on how such poor performance can be addressed.

RESEARCH HYPOTHESIS

1. There is no significant relationship between students' poor academic performance in programming languages and students' attitudes towards learning programming languages.
2. There is no significant relationship between poor academic performance in programming languages and lecturers' inefficiency in terms of their availability for practical, methodology and understanding of concept.

3. There is no significant relationship between poor academic performance in programming language and the complexity of the programming languages in terms of coding, compiling, detecting and debugging of errors.

SIGNIFICANCE OF THE STUDY

Programming Languages deal with giving a set of instructions to computer to perform specific function. This Research will be of great value to both students and lecturers in Nigeria Tertiary institutions in determining preferable way of teaching programming language courses in Tertiary Institutions in Nigeria. It will equally be of great value to the government and citizen because the result of the findings will help them to identify and recognize solutions to the student poor academic performance in learning programming languages courses in Nigeria Tertiary Institutions. Also, the study enables the researcher to investigate the nature of difficulties while learning programming language courses, discover an overview of the situation that would help learners to learn programming courses and investigate the factors that lead to poor performance of students in programming languages.

RESEARCH METHODOLOGY

The methodology employed to carry out the study are population of the study, sample and sampling techniques, research instrument, procedure for data collection and method of data analysis. The design employed for this study is a descriptive survey research to evaluate the factors militating against poor academic performance in programming languages among computer science students.

POPULATION OF THE STUDY

The population of this study consists of computer science students in Nigeria University. However, for the study, four hundred and seventy five (475) respondents were randomly selected in department of computer science in some randomly selected Universities in Nigeria.

RESEARCH INSTRUMENTATION

The research instrument used to collect information is a standardized questionnaire which was prepared and administered to elicit necessary response

from the students for the investigation. It contained seventeen well-structured questionnaire on scrutinizing the causes of poor hypothetical performance in programming language among computer science students in University of Nigeria.

VALIDITY OF INSTRUMENTS

To ascertain the validity of the instrument, questionnaire was drafted and necessary corrections were made and adjusted that the instrument has content and face validity for suitable use of the study.

RELIABILITY OF INSTRUMENT

The researcher administered the questionnaire to the students in some University campuses. The method used was that each student was given the questionnaire and the students were encouraged to give sincere information to the best of their knowledge. The administered questionnaire was gathered and marked by the researcher with the aid of research assistance randomly among the students of the college. Explanations were made necessary for better clarification on the questionnaire. Effort was made to collect the questionnaire on the same day so as to ensure high percentage return.

METHOD OF ANALYSIS

The data collected was analyzed with the use of Statistical Package for Social Science (SPSS) version 21 while chi square was used to test for the research hypothesis.

HYPOTHESIS 1

- There is no significant relationship between students' poor academic performance in programming languages and students' attitudes towards learning programming languages.

S/N	STATEMENTS	SA	A	SD	D
1.	Poor upbringing of many students contribute to their performance in programming language				
2.	Learning approach that students display to the studying of programming courses is essential to their performance				

3.	Non–challant attitude shown by students when learning programming languages affect their performance
4.	Inability of the student to understand the difference between programming knowledge and programming strategies result to their poor academic performance in programming.
5.	Lack of readiness and eagerness to learn programming on the part of students constitute to their performance in programming.
6.	Laziness of students to regularly and consistently practice coding affect their performance in programming languages

Case Processing Summary

	Cases					
	Valid		Missing		Total	
	N	Percent	N	Percent	N	Percent
Level * ScoreT1	475	100.0%	0	0.0%	475	100.0%

Level * ScoreT1 Cross-tabulation

Count

		ScoreT1			**Total**
		SA	A	D	
Level	100 level	105	35	0	140
	200 level	50	42	0	92
	300 level	69	34	18	121
	400 level	79	43	0	122
Total		303	154	18	475

Chi-Square Tests

	Value	**Df**	**Asymp. Sig. (2-sided)**
Pearson Chi-Square	65.934[a]	6	.000
Likelihood Ratio	62.122	6	.000
Linear-by-Linear Association	5.374	1	.020

N of Valid Cases	475

a. 3 cells (25.0%) have expected count less than 5. The minimum expected count is 3.49.

The chi – square table value at 0.05 level of significance at degree of freedom 6 = 10.64

Since the critical value (10.64) is less than the calculated value (65.934), it shows the hypothesis is rejected and it is significant. This signifies that there is significant difference between poor academic performance in programming languages and students attitudes in learning programming languages.

HYPOTHESIS 2

There is no significant relationship between poor academic performance in programming languages and lecturers' inefficiency in terms of their availability for practical, methodology and understanding of concept.

S/N	STATEMENTS	SA	A	SD	D
1.	Poor teaching methodology by lecturers determine the students' performance in programming language.				
2.	Poor practicals experience by lecturers teaching programming language affect the students to perform well				
3.	Inability of lecturers with good teaching methodology coupled with poor impartation of knowledge of programming affect the performance of computer students.				
4.	Incapability of some lecturers to understand essential concepts and ideas in programming is affecting students' performance.				

Case Processing Summary

Cases

	Valid		Missing		Total	
	N	Percent	N	Percent	N	Percent
Level * ScoreT2	475	100.0%	0	0.0%	475	100.0%

Level * ScoreT2 Cross-tabulation

Count

		ScoreT2				Total
		SA	A	SD	D	
Level	100 level	63	68	9	0	140
	200 level	34	58	0	0	92
	300 level	25	87	9	0	121
	400 level	52	52	0	18	122
Total		174	265	18	18	475

Chi Square Test

	Value	Df	Asymp. Sig. (2-sided)
Pearson Chi-Square	90.911[a]	9	.000
Likelihood Ratio	95.719	9	.000
Linear-by-Linear Association	12.469	1	.000
N of Valid Cases	475		

a. 6 cells (37.5%) have expected count less than 5. The minimum expected count is 3.49.

The chi – square table value at 0.05 level of significance = 14.68
Since the critical value (14.68) is less than the calculated value (90.911), it shows the hypothesis is rejected. This signifies that there is significant difference between poor academic performance in programming languages and lecturers' inefficiency in terms of their availability for practical, methodology and understanding of concept.

HYPOTHESIS 3

There is no significant relationship between poor academic performance in programming language and its complexity in terms of coding, compiling, detecting and debugging errors.

S/N	STATEMENTS	SA	A	SD	D
1.	Difficulties in detecting and debugging errors when learning programming language discourage students.				
2.	Poor programming skills in detecting and debugging programs hinder their good performance.				

3.	Students Inabilities to learn the syntax and the functionality of programming languages leads to their poor performance in programming languages.
4.	Inability of students to understand the difference between syntax and semantics that guides the writing of successful programming languages lead to their poor performance.
5.	Programming languages running under different operating system may exhibit significant differences in speed and accuracy which eventually affect students' performance.
6.	Inability of the student to understand the difference programming knowledge and programming strategies result to their poor academic performance in programming languages.
7.	Students are not exposed to various programming specification, syntax and technique.

Case Processing Summary

		Cases					
		Valid		Missing		Total	
		N	Percent	N	Percent	N	Percent
Level	*	475	100.0%	0	0.0%	475	100.0%
ScoreT3							

Level * ScoreT3 Cross-tabulation

Count

		ScoreT3				Total
		SA	A	SD	D	
Level	100 level	79	53	0	8	140
	200 level	58	26	8	0	92
	300 level	67	54	0	0	121
	400 level	52	52	9	9	122
Total		256	185	17	17	475

Chi-Square Tests

	Value	Df	Asymp. Sig. (2-sided)
Pearson Chi-Square	43.999[a]	9	.000
Likelihood Ratio	56.793	9	.000

Linear-by-Linear Association	5.429	1	.020
No. of Valid Cases	475		

a. 6 cells (37.5%) have expected count less than 5. The minimum expected count is 3.29.

The chi – square table value at 0.05 level of significance = 14.68

Since the critical value (14.68) is less than the calculated value (43.999), it shows the hypothesis is rejected. This shows that there is significant difference between poor academic performance in programming language and its complexity in terms of coding, compiling, detecting and debugging errors.

Discussion of Findings

The result of the first hypothesis signifies that the chi – square result value at 0.05 level of significance at degree of freedom 6 which is equal to 10.64 enable us to understand that the critical value (10.64) is less than the calculated value (65.934). This implies that the hypothesis is rejected and it is significant. This reveals that there is significant difference between poor academic performance in programming languages and students' non–challant attitudes in learning programming languages. The outcome has proved that the non–challant attitudes of students in learning programming languages have contributed to their poor academic performance in programming languages.

Second hypothesis result reveals that the chi–square value at 0.05 level of significant at degree of freedom 9.0 which is equal to 14.68 shows that the critical value (14.68) is less than the calculated value (90.911). This signifies that the hypothesis is rejected and it is significant. This shows that there is significant difference between poor academic performance in programming languages and lecturers' inefficiency in terms of their availability for practical, methodology and understanding of concept. The hypothesis has proved that the lecturers' inefficiency in terms of their availability for practical, methodology and understanding of concept is a major factor that contributes to the poor academic performance in programming languages.

Moreover, the third hypothesis helped us to understand the chi – square table value at 0.05 level of significance of the critical value (14.68) is less than the calculated value (43.999), it shows the hypothesis is rejected. This implies that there is significant relationship between poor academic performance in

programming language and its complexity in terms of coding, compiling, detecting and debugging errors. The hypothesis revealed that complexity in terms of coding, compiling, detecting and debugging errors is one of the factors that result to poor academic performance and its complexity in terms of coding, detecting and debugging errors.

Therefore, the results of the research proved that the aforementioned factors are causes of poor performance of students in programming languages. From the analysis of the first hypothesis, it was found that non–challant attitudes of students to programming languages lead to their woeful performance in programming languages courses. Amoako, *et al*., (2013)asserted during world science foundation that the lackadaisical attitudes of students to programming languages is a major factors that leads students poor performance in programming languages. Robins (2003) opined that poor performance of students in programming languages can come to an end if students' bahaviour to programming languages turns a new leaf and the author emphasized that the students' readiness to acquire knowledge on programming languages should be there which was believed that there would be positive change in their performance in learning programming languages. Also, Sajaniemi and Kuittinen (2003) gave an insight to lecturers' inefficiency in terms of their availability for practicals, this is one of the major factors responsible to students poor performance in programming languages. Moreover, complexity in terms of coding, compiling, detecting and debugging errors is a burden faced by students when learning programming languages. contributed that the major challenge faced by students in learning programming languages, but the ability of students to understand the concept of coding, compiling, detecting and debugging errors will contributed to their performance in programming languages. Since programming languages is complex in learning and applying it.

Conclusion

Learning programming languages is easy when the factors discussed in this research that lead to students poor performance have been dealt with, such as lecturers attitude to the teaching of programming languages should be addressed as discussed in this research. In the same vein, students' attitude to programming languages should turn a new leaf. It is certain that when the

aforementioned points in this research is carried out there will be a turnaround in the performance of students in programming languages.

Recommendations

Based on the findings of this study, the following recommendations were given:

- Lecturers should ensure and focus on teaching the analysis and debugging techniques of requirements or issues.
- Students should be in possession of one laptop with series of programming languages software for their use in order to aid them in adequate understanding of programming languages.
- Lecturers working along with instructor (programmer) should endeavour to solve real life problem using simplest programming language so as to facilitate students understanding in programming language.
- Programmer should use suitable method to enable students to understand programming languages.
- Syntax and semantics in programming languages should be taught for students to understand the basis of writing programming languages.
- Sufficient periods and days should be allocated for practicals.

REFERENCES

Aman, G. (2020). Best Programming Languages to Learn in 2020 (for Job & Future). Last Updated 14 Sep, 2020. Retrieved on 16th of September, 2020.

Amoako, Y. O., Kofi, A., Authur, J., and Adjetey, C. (2013). Performance of Students in Computer Programming: Background, Field of Study and Learning Approach Paradigm. International Journal of Computer Applications. (77), pp. 17-21.https://www.researchgate.net/publication/260549178_Performance_of_Students_in_Computer_Programming_Background_Field_of_Study_and_Learning_Approach_Paradigm/citation/download

Bebbington, Shaun (2014). "What is coding". Tumblr. Archived from the original on 2020-04-29. *Retrieved 2014-03-03*. https://yearofcodes.tumblr.com/what-is-coding

Bebbington, Shaun (2014). "What is programming". Tumblr. Archived from the original on 2020-04-29. *Retrieved 2014-03-03*. https://yearofcodes.tumblr.com/what-is-programming

Bennedsen, J. and Schulte, C. B. (2010). ACM Transactions on Computing Education. BlueJ Visual Debugger for Learning the Execution of Object-Oriented Program.

https:/www.researchgate.net/publication/220094514_bluej_visual_debugger_for_learning_the_execution_of_object_oriented_program

Davies, S., PolackWahl, J.A. and Anewalt, K. (2011). A Snapshot of Current Practices in Teaching the Introductory Programming Sequence. beligin, pp.23.

Essi, L., Kirsti, A. and Hannu-Matti, J. (2005). A study of the Difficulties of Novice Programmers. ACM, SIGCSE Bulletin, September 2005. *https:/www.researchgate.net/publication/220808194_a_study_of_the_difficulties_of_novice_programmers*

Islam , N. Sheikh, G. S., Fatima, R and Alvi, F. (2019). A study of Difficulties of Students in Learning Programming. Journal of Education & Social Sciences, October, 2019. https:/www.researchgate.net/publication/336945988_a_study_of_difficulties_of_students_in_learning_programming

Mohammed, S., Elhaddad, M. E. and Mohammed, A. O. (2017). Proposing a Solution for the Problem of Teaching Programming to Novice Students Using Soft System Methodology.*https:/www.researchgate.net/publication/316883532*_proposing_a_solution_for_the_problem_of_teaching_programming_to_novice_students_using_soft_systems_ *methodology*

Putano, B. (2019). A Look at 5 of the Most Popular Programming Languages of 2019. https://stackify.com/popular-programming-languages-2018/?utm_source=codecademyblog

Robins, A., Rountree, J. and Rooftree, N. (2003) Learning and Teaching Programming: A Review and Discussion. Computer Science Education, vol. 13(2), Springer, p. 137-172.

Sajaniemi, J. and Kuittinen, M. (2003). Role of visualization and engagement in computer science education, A Review and Discussion, Computer Science Education, 13(2), pp. 37-74.

Schulte, C. and J. Bennedsen. (2006). What do teachers teach in introductory programming? IEEE conference proceeding, pp.58-65.

Sheikh, G. S. and Islam, N. (2016). A qualitative study of major programming languages: teaching programming languages to computer science students. Educational Data Mining, International Journal of Information and Communication Technology. Vol. 10, No. 1, (Spring 2016) 24-34.

https://en.wikipedia.org/wiki/Computer_programming

https://news.codecademy.com/what-is-computer-programming/

https://hackr.io/blog/what-is-programming

https://hackr.io/blog/what-is-programming-language

https://hackr.io/blog/best-programming-languages-to-learn-2020-jobs-future

https://www.google.com/search?source=hp&ei=NUVjX67gJKmKlwSOy4P4DQ&q=meaning+of+programming+in+computer&oq=meaning+of+progra&gs_lcp=CgZwc3ktYWIQARgDMgIIADICCAAyAggAMgIIADICCAAyAggAMgIIADICCAAyAggAMgIIADoKCAAQsQMQRhD5AToFCAAQsQM6CAgAELEDEIMBOgkIABBAKEEYQ-QE6BAgAEAo6DQgAELEDEIMBEEYQ-QE6BwgAEEYQ-QFQuR9YysoBYJaCAmgGcAB4AoABgAuIAfGFAZIBCDUtMTEuNi41mAEAoAEBqgEHZ3dzLXdpeg&sclient=psy-ab