

Assessment of Students' Achievement in Computer Programming: An Inquiry into some Influencing Factors

Ismail. O. Muraina, Mukaila. A. Rahman

Abstract— This study inquired into some influencing factors that affect students' achievement in computer programming courses. Two hundred students formed the sample for the study. Data were collected through the use of both achievement test and questionnaire. The results of regression and structural path analysis indicate that Practical Class makes the strongest unique contribution towards achievement of students in Programming Languages. Next to this is Student Interest while Laboratory Usage contributed the least. The other two predictors did not make any statistically significant contribution towards students' achievement. The results also made it clear that laboratory usage had positive relationship with availability of compilers while availability of compilers was negatively related to quality of instruction.

Index Terms— Assessment, Computer Programming, Practical Classes, Students Achievement

1 INTRODUCTION

Programming has been reported as a very useful skill that can result into a rewarding career [1]. Learning programming languages is one of the foundations of computer science education and usually one of the first courses novice students take [2]. Despite this, programming courses are generally regarded as difficult and often result into students not showing interest and recording of mass failures. It is generally accepted that it takes about 10 years of experience to turn a novice into an expert programmer [3]. Thus, it is important for the teachers to gain students attention and strengthen their motivation for learning to program with the help of a variety of factors such as student interest, quality of instruction, practical class, laboratory usage and availability of languages' compilers. Fortran is one the oldest problem-oriented languages that was designed essentially for scientific and engineering environments. Fortran still maintains a dominant role in both academic and industrial sectors. Teaching and learning of Fortran programming language is as old as science-based courses in Nigerian tertiary institution.

In Nigeria tertiary institutions, computer science is of the major courses studied in the faculty/ school of sciences. The language (FORTRAN) is one of the few programming languages used to teach students some basics programming concepts before exposing them to other advanced programming skills such as Visual Basic, C++, Java and other object oriented programming languages. More importantly, FORTRAN is used to strengthen students' skills in programming, after they might have had some useful experience with BASIC programming language often meant for beginners.

Based on above, this research is to identify some influencing factors that can improve or hinder students' achievement in programming skills like FORTRAN. Therefore, this study tries to answer the following research questions:

1. How do the student's interest, quality of instruction, practical class, laboratory usage and available teaching compilers predict student's achievement in Fortran programming course?
2. How much variance in student's achievement scores in Fortran can be explained by these preceding five factors?

3. Which of the factors best contribute to the student's achievement in Fortran programming ?

2 RELATED LITERATURE

Computer science and technology is playing an important role in the lives of students since computers have become increasingly available at almost all levels of education. Computers are used in several ways in classrooms. Majorly, computer aided instruction is currently common in most schools as a means of making the computer to teach the students. Students nowadays meet computers and technologies in early ages; still the students' computer interaction regarding programming has not been optimal [4]. In any educational system, curriculum and teaching becomes more student centred to connect the school to real life situations and support students to become better thinkers and problem solvers ([5]; [6]; [7]; [8]; [9])

Although, many different approaches have been suggested for developing students' computer skills by the use of programming languages. Computer program has become a difficult area for students at various level of educational system in Nigeria. The reason for this situation is based on some-thoughts: programming is difficult for most students. That is, many new generation programming languages are object driven and mathematically very elegant, yet they are still difficult to learn and master ([10]; [11]). Also, programming does not promote any students' skills, there is no persuasive evidence "that writing programs will automatically improve the students' creativity or general reasoning ability or higher order cognitive skills" [12] and sometimes the programming in quotes may not directly relate with education curriculum.

In considering the list of many programming languages, software development tools, techniques and supporting technologies which have developed in last decades, it is accepted that computer science is technological profession and a field of study which needs to be promoted in all facets of educational levels. The current challenges as observed in computer programming skills can be viewed as to what extend student interest affect the students' achievement in programming. Therefore, for a beginner programmer, the learning of how to code a program first needs the students' interest ([13], [10]). The interest of the students should be

buttressed with quality of teaching by using pure practical means of instruction. Most attention should be channeled towards the use of laboratory with relevant and most current teaching compilers. Various findings from research have revealed the perception, attitudes, performance, teaching and learning of programming languages. These findings include Donald McCasland and Naney Stewart that tested the relationship between study habits and GPA, Roy D. Goldman and David J. Hudson who conducted a study of the relationship between academic abilities and “strategies” and major field, gender and GPA and James R. Leonard who investigated the effect of different instructional approaches in a first-semester computer programming course on student performance. Also, [14] conducted a research on perception of students and teachers about the use of a kid’s programming languages in computer courses yet much have not been done to really match students’ achievement with some influencing factors such as student interest, quality of instruction, practical class, laboratory usage and availability of teaching compilers.

3 METHOD

The purpose of this study is to assess students’ achievement in computer programming by considering some influencing factors in computer science department at Adeniran Ogunsanya College of Education (AOCOED), Lagos, Nigeria. The institution offers computer science programme at both Nigerian Certificate in Education (NCE) as well as Bachelor of Science Degree levels. With this purpose in mind, this research focuses on a single case study – Fortran programming course. In order to meet the objectives of this study, respondents were selected from computer science NCE and Degree students in AOCOED. Two hundred students involved in this study, comprising 100 level NCE and 200 level degree students respectively. The two sets (Degree and NCE) were taught the programming course in the second semester of a particular session, although by two different instructors. At the end of the classroom teaching and practical classes, the two groups were exposed to the same treatment (achievement test) and later given questionnaire on likely influencing factors that can improve or affect their achievement in Fortran programming generally. Two instruments were used to collect data from the students: the first instrument (achievement test in Fortran Programming) contained ten questions that covered the course contents while the second instrument (Questionnaire) solicited students’ opinion on their interest, quality of instruction, practical class, laboratory usage and availability of compilers towards learning Fortran programming. Each of the variables was rated differently. In realizing the validity and reliability of the two instruments, they were given to some targeted students to see whether they understand the construction of the language to ensure face validity then also per reviewed to see that the content therein is within the scope of the study. The achievement test was subjected to split-half reliability index, the result showed that the instrument was reliable with $r = .99$ while the questionnaire instrument was tested using Cronbach’s Alpha with the $r = .73$. The data generated were analyzed using multiple regression and path analysis via IBM SPSS statistics 19 package.

4 RESULTS AND ANALYSIS

Descriptive statistics and bivariate correlations are presented in Table1. As revealed in the table, students’ interest plays a major role in learning programming languages; also it is equally important to select good quality of instruction for students’ understanding. Descriptive results also indicate that practical class is a catalyst to students’ achievement in programming languages. Laboratory usage and compiler availability were considered less necessary to aforementioned variables. Conclusively, the bivariate correlations between the variables were not statistically significant.

Variables	N	M	SD	Achievement Test	Student Interest	Quality of Instruction	Practical Class	Laboratory Usage	Availability of Compilers
Achievement Test	200	48.94	16.93	1.000	.27	.213	.242	-.056	.003
Student Interest	2	3.00	0.60		1.000	.217	.025	.280	.229
Quality of Instruction	2	2.00	0.50			1.000	.115	-.028	-.146
Practical Class	2	2.00	0.50				1.000	.322	-.180
Laboratory Usage	2	2.00	0.60					1.000	.172
Availability of Compilers	2	2.00	0.50						1.000

Regression analysis results were presented in Table 2 and 3. The table represents how much of the variance in the response is explained by the weighted combination of predictors. Regressing Preference on the five predictors in an R2 is 0.193, indicating that approximately 19% of the variance in the achievement of student in programming languages is explained by the predictor variables in the linear regression. In table 3, the ANOVA result showed that there was statistically significant prediction of all the predictor variables (Student interest, Practical class, Laboratory usage, Quality of

instruction and availability of compilers) over the predicted/dependent variable (Achievement test in programming languages) [F(5,194) = 9.27,P<.0001].

Table 2: Regression Model Summary^b

Model	R	Adjusted R Square	Change Statistics						
			Std. Error of Estimate	R Square Change	F Change	df1	df2	Sig. F Change	
1	.439 ^a	.193	.172	15.403	.193	9.266	5	19	.000
			28					4	

a. Predictors: (Constant), Availability of Compilers, Quality of Instruction, Laboratory Usage, Practical Class, Student Interest

b. Dependent Variable: Achievement Test

Table 3: Analysis of Variance (ANOVA)^b

Model	R	Adjusted R Square	Change Statistics						
			Std. Error of Estimate	R Square Change	F Change	df1	df2	Sig. F Change	
1	.439 ^a	.193	.172	15.403	.193	9.266	5	19	.000
			28					4	

a. Predictors: (Constant), Availability of Compilers, Quality of Instruction, Laboratory Usage, Practical Class, Student Interest

b. Dependent Variable: Achievement Test

Evaluating each of the predictor variables was showed in Table 4. In this case, the largest beta coefficient is .31, which is for Practical Class. This means that Practical Class makes the strongest unique contribution to explaining the dependent variable (Achievement Test in Programming Language), when the variance explained by all other variables (Student Interest, Quality of Instruction, Laboratory Usage and Availability of Compilers) in the model is controlled for. It was also revealed that Practical Class is statistically significant P<.0001. Follow this is Student Interest which contributed .30 (.296) beta coefficient value and also statistically significant at P<.0001. Lastly, the table showed that Laboratory Usage contributed the

least -.24 at P=.0001 level of significant. The other two predictors did not make any statistically significant contribution to students' achievement in programming languages this may be due to overlap with other independent variables in the model. The table also reflected the contribution of each of the predictors to the total R squared. Practical Class uniquely contributed 8 percent (part value = .281 → .281*.281 = 0.08) to the explanation of variance in achievement test, next to it is Student Interest that uniquely explained 7 percent of variance in achievement test and finally Laboratory Usage gives .05, indicating a unique contribution of 5 percent to the explanation of variance in achievement test. Quality of Instruction makes a contribution of only 1 percent while Availability of compilers makes zero unique contribution.

Table 4: Beta, Significant Level and Part

Predictors	Beta	Sig.	Part
Student Interest	.296	.000	.268
Quality of Instruction	.114	.097	.108
Practical Class	.309	.000	.281
Laboratory Usage	-.243	.001	-.215
Availability of Compilers	.049	.482	.045

Table 5: Significant Coefficient of predicted and predictors

Predicted	Predictor	Coefficient	Sig.
Student Interest	Practical Class	.00	NS
Quality of Instruction	(a) Laboratory Usage	-.003	NS
	(b) Availability of Compilers	-.146	S
Availability of Compilers	Laboratory Usage	.172	S

4.1 STRUCTURAL PATH ANALYSIS

Structural path analysis was used to further examine the relationship among students' achievement in programming and other variables like Student interest, Practical class, Laboratory usage, Quality of instruction and availability of compilers. The hypothesized model examined in this study is depicted in Figure 1.

We ordered the variables in our model, on one hand, on the basis of experience of the researchers in the field of

programming languages, on the other hand, on the basis of prior theoretical and empirical research using path analysis. It is perceived that for effective learning of programming language and improvement of students' achievement in programming languages, the variables like Student interest, Practical class, Laboratory usage, Quality of instruction and availability of compilers must come to play significant role. In line with this, researchers first drew the hypothesized model showing relationship of achievement test with other variables.

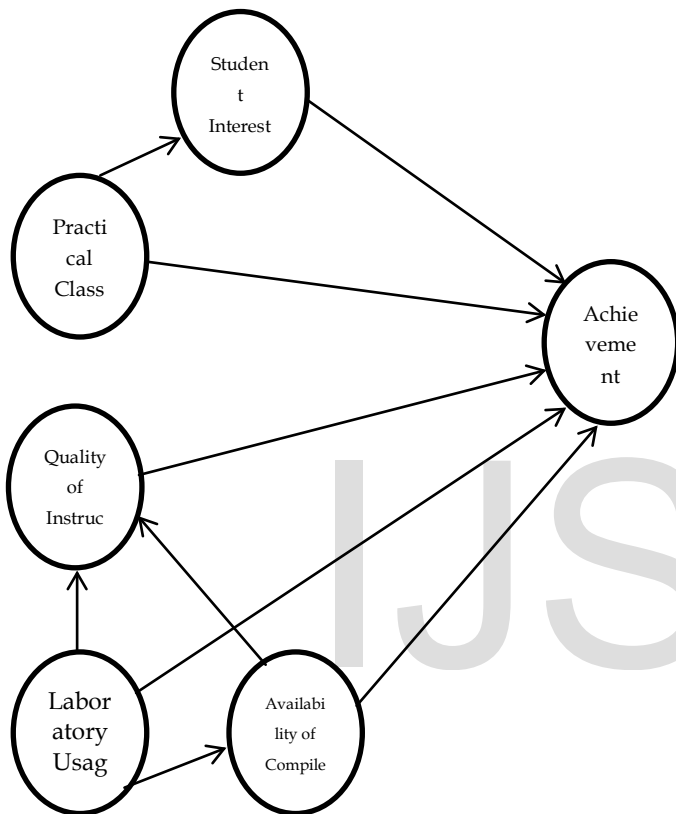


Fig. 1 Hypothetical Model

As indicated in Table 4 three direct path coefficients were statistically significant. The significant direct paths include: Student Interest → Achievement ($\beta = 0.30$), Practical class → Achievement ($\beta = 0.31$) and Laboratory Usage → Achievement ($\beta = -.24$) respectively. Results also indicate that the relationship between Laboratory Usage and Quality of Instruction was completely mediated by Availability of Compilers (Laboratory Usage → Quality of Instruction \implies Laboratory Usage → Availability of Compilers ($\beta = 0.17$) and Availability of Compilers → Quality of Instruction ($\beta = -0.15$)). There were no significant indirect effects for this model (see figure 2)

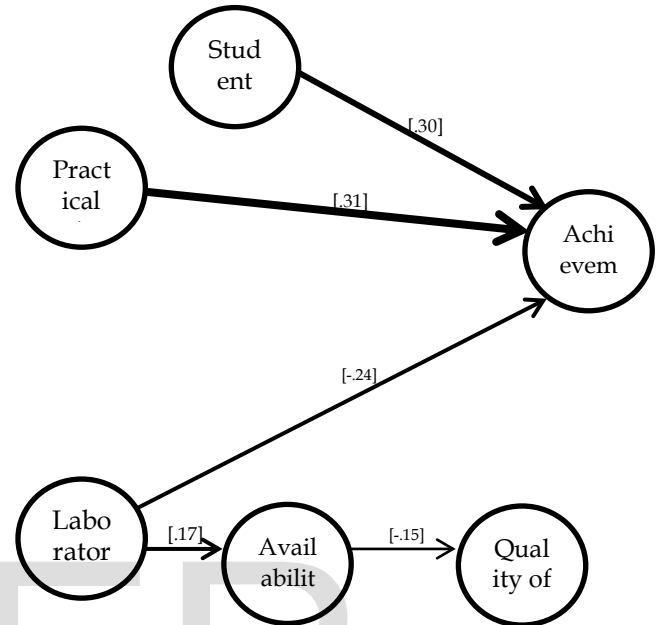


Fig.2 Final Model

DISCUSSION

The results of the analyses presented allow us to answer the three questions posed at the beginning of this study. Our predicted variables, which includes student's interest, quality of instruction, practical class, laboratory usage and available teaching compilers, explains 19.3 per cent of the variance in student's achievement in Fortran programming language. Of these five variables, Practical Class makes the largest unique contribution ($\beta = .31$), although Student Interest and Laboratory Usage also made statistically significant contributions ($\beta = .30$) and ($\beta = -.24$) respectively. The results also anchor it that Laboratory Usage makes unique statistically significant contribution ($\beta = .17$) towards Availability of Compilers while Availability of Compilers in turn makes statistically significant contribution ($-.15$) towards Quality of Instruction.

It is clearly shown that teaching and learning of programming languages highly rest on constant practices. The result of this study shows that practical classes had a positive relationship with the achievement in programming languages. This finding implies that students who engage in practical programming class were more likely to demonstrate higher level of

programming understanding compare with those who might have not been exposed to practical sessions. In support of this [15]made it clear that practical class promotes the basic skills and competencies of doing science; procedural and manipulative skills, observation skills, skills of representing and interpreting data and the accompanying conceptual and critical abilities.

With respect to student interest, the result also indicates that student interest had a positive relationship with the achievement in programming languages. This shows that students who have interest in programming languages are more likely to perform better than their counterparts. This finding is in line with the research of [16]while investigating factors that often impair science education, confirmed that over 80% of failure in science and technology are due to the inability of students to perform well in practical sessions. [17]in his study, identified students' interest as one of major factors to be considered very important as it could cause poor performances of students in science studies. A similar view was reported in [18]. While investigation into factors that affects students' performance in physics. he observed lack of students' interest in physics due to preconceived idea. He concluded that physics being perceived as a difficult subject has affected the performance of students in physics.

On laboratory usage, the result from this study indicates that laboratory had a negative relationship with achievement. This finding suggests that students who had high achievement in programming language may not necessarily be as a result of laboratory usage. This finding against the view of [19]who wrote that "Laboratory activities appeal as a way of allowing students to learn with understanding and, at the same time, engage in a process of constructing knowledge by doing science". This divergent view may be as a result of availability of personal computers to students. This without doubt, make it more convenient for students to practice at their pace with little or no hindrance as a result of power supply failure often observed in our laboratories.

CONCLUSION AND RECOMMENDATIONS

From the aforementioned, the assessment of students' achievement in computer programming can be improved through constant practical class, also by considering students' interest and laboratory usage. Though, quality of instruction plays a major role in teaching and learning programming languages, however this study shows no significance towards quality of instruction and availability of compilers. Conclusively, the results of this study point to influencing factors that can affect students' achievement in computer programming languages. In particular, additional research is needed to explore more influencing factors and also to cover more area than what this study covers.

This study could further investigate students attitudes to modern high level programming languages especially those with object oriented concepts as this may be more appealing to students to practice with.

REFERENCES

- [1] Robins, J. Rountree, and N. Rountree, "Learning and Teaching Programming: A review and Discussion". Journal of Computer Science Education. Vol 13. No.2 Pp 137-172, 2003.
- [2] Matthiasdottir, "How to teach programming languages to novice students? Lecturing or not?" 2006. www.ecet.ecs.ru.acad.bg
- [3] L. E. Winslow "Programming pedagogy- A psychological overview". SIGCSE Bulletin 28, 17-22. 1996.
- [4] A. Druin, The Design of Children's Technology: San Francisco, CA: MorganKaufmann Publishers, 1999.
- [5] J. Piaget, To Understand is to Invent. New York: Grossman, 1973.
- [6] L. S. Vygotsky, Mind in Society: Cambridge, Mass: Harvard University Press, 1978.
- [7] S. Papert, Mindstorms: Children, Computers and Powerful Ideas. New York. Basic Books, 1993.
- [8] S. Vosniadou, "How Children Learn", Educational Practices Series, 7. The International Bureau of Education (UNESCO), 2001.
- [9] Falk, Teaching the Way Children Learn: Re-visiting School to Support Diverse Learners. New York: Teachers College Press, 2009.
- [10] Siegle, Developing Student Programming and Problem Solving Skills with Visual Basic. Gifted Child Today. 32(4) 24-29 (ERIC Document Reproduction Service No. EJ860950), 2009.
- [11] K. Khan, ToonTalk (TM)- An Animated Programming Environment for Children. (ERIC Document Reproduction Service No. ED 392435) 1995.
- [12] S. A. Shafto, "Programming for Learning in Mathematics and Science". ACM SIGCSE Bulletin, 18(1), 296-302, 1986.
- [13] J. Schwartz, J. Stagner and W. Morrison, "Kid's Programming Language (KPL)". The 33rd International Conference and Exhibition on Computer Graphics and Interactive Techniques, Boston, Massachusetts, 2006.
- [14] T. Akcay "Perceptions of students and teachers about the use of a kid's programming language in computer courses". A thesis submitted to the graduate school of natural and applied science of middle east Technical University, 2008
- [15] M. B Ogunniyi "Two decades of science education in Africa". International Science education 70(2): 111-122. 1986
- [16] Ukwuma "Factors Affecting Students' Performance in Science in Nigeria

Schools". Fakejerseys 5. 1990.

- [17] A. T Aiyelabegan, "Effect of physics practical on Students' Academic Performance in Senior School Certificate Physics Examination in Kwara State". Lafiagi Journal of Science Education 5 (1 & 2) 2003.
- [18] L. Bamidele, L. "Students' Poor Performance in Physics.A Bane to our Nation's Technological Development". Nigerian Journal of Science Education and Practice. 2(1), 174, 2004.
- [19] K. G. Tobin, Research on science laboratory activities. In pursuit of better questions and answers to improve learning. School science and Mathematics, 1990.

IJSER