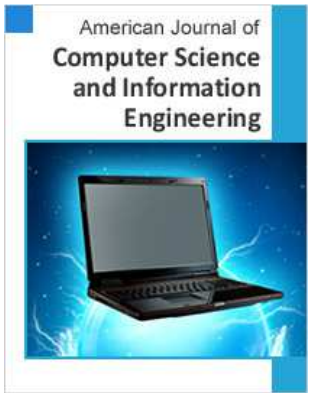
American Journal of Computer Science and Information Engineering
2015; 2(5): 45-50
Published online October 30, 2015 (http://www.aascit.org/journal/ajcsie)
ISSN: 2381-1110 (Print); ISSN: 2381-1129 (Online)

American Association for Science and Technology



Keywords
Computer Science,
Programming,
Academic Performance,
Information Technology,
Improvement,
Skill,
Quality Education

Received: September 23, 2015
Revised: October 13, 2015
Accepted: October 15, 2015

Improvement Strategies for Computer Science Students' Academic Performance in Programming Skill

Akinola Kayode Emmanuel[1], Olanrewaju Grace Oluwadamilare[2], Oyenuga Augustine Yomi[2]

[1]Computer Science Department, Abraham Adesanya Polytechnic, Ijebu-Igbo, Ogun State, Nigeria
[2]Department of Statistics, Abraham Adesanya Polytechnic, Ijebu-Igbo, Ogun State, Nigeria

Email address
kayodewale87@yahoo.com (Akinola K. E.)

Citation
Akinola Kayode Emmanuel, Olanrewaju Grace Oluwadamilare, Oyenuga Augustine Yomi. Improvement Strategies for Computer Science Students' Academic Performance in Programming Skill. *American Journal of Computer Science and Information Engineering*. Vol. 2, No. 5, 2015, pp. 45-50.

Abstract
Programming dexterity is an essential skill for computer science students. However, teaching and learning programming ideas and skills has been recognized as a great challenge to both teachers and students. Therefore, the development of efficient learning strategies and environments for programming courses has become an important subject. The study is intended to recommend improvement strategies which can be used in teaching Computer Programming course effectively to undergraduate students in the field of computer science and information technology. Survey research design was used. Descriptive method, the actual grades of the students, and informal interviews among lecturers were utilized in the study. The recognized assessment results revealed that 47% of students in the course have poor academic performance in the programming course. This concluded that there is a need for improvement strategies which requires planning and implementation to strengthen the motivation of the students to study, boost their academic performance, and uplift the quality of education in the field of computer science. Relevant workable recommendations were suggested.

1. Introduction

The students' academic performance is the outcome of the final examinations, quizzes, assignments, attendance and other graded points related to the course. The primary mission of each tertiary institution in the field of computing and information technology is to offer high quality and relevant education in order to produce skillful and competent graduates. To accomplish this, a number of practical and instructional strategies were designed to improve the students' academic performance.

Computer programming course is one of the requirements to obtain a degree in most computing and information technology academic programs. It is a generally belief that computer programming course is one of the most difficult courses in tertiary institutions because it is too abstract and requires higher analytical and problem solving skills. Thus, to improve the performance of struggling students in programming course, improvement strategies are needed.

Aside from the fact that, teaming number of graduates are unemployable and jobless because they were ill equipped in practical and employable skills in their course. Thus,

there is need to assess and suggest improvement strategies for computer science students' academic performance in programming course.

The purpose of improvement strategies in education is to identify all factors that may advance students' performance academically. Lecturers have usually found ways for classroom strategies that will reach out and help the students to boost their performance.

1.1. Statement of the Problem

Learning programming is a complex task since programming requires new ideas in thinking and creative skills in problem solving. The average of students' academic performance in programming course for over the years in undergraduate program of computing and information technology in many universities and polytechnic can be concluded as poor compared to other courses in the field. However, the percentage of students with poor academic performance every semester is indisputable and do not support the mission of many academic programs. This affects the quality of education the academic programs are trying to keep up. Improving the academic performance of students is part of the continuous effort to uplift the standard of education in the area of computing and information technology.

1.2. Objectives of the Study

The objectives of the study are 1) to identify the behaviors of students in computer programming course, 2) to identify the academic performance of students in the course, and 3) to recommended improvement strategies to advance the students' performance in computer science course.

1.3. Significance of the Study

The study aims to assess the academic performance of students in programming course.

1.4. The beneficiary of the Study Are

1) Students, to boost their academic performance in programming skills and to help them recognize the value of learning each course of their chosen field.
2) Lecturers, to assist them in the area that their students needed, helping them to make the course appealing to their students.
3) Curriculum development committee, to review the course as a whole, focus on the current programming software, hardware and area that is needed by the students, recommend necessary actions to assist the department in maintaining the quality of education.
4) Computing and information technology academic departments and institutions, to help them maintain the standard of the quality of education that they are providing to the community, and
5) Future researchers, to study on how to assist students of computer science in improving their academic performance.

2. Literature Review

The rapid growth of information technology has created high demand for skillful programming specialists. Programming skills have become a core competence for engineering and computer science students (Verdú et al., 2012; Hwang, Shadiev, Wang, & Huang, 2012; Fessakis, Gouli, & Mavroudi, 2013). According to (Brooks, 1999; Govender, 2009; Katai & Toth, 2010; Wang, Li, Feng, Jiang, & Liu, 2012; Yeh, Chen, Hung, & Hwang, 2010), learning a computer-programming language involves the understanding of theoretical background and practice of a range of semantic and syntactic knowledge, coding skills, and algorithmic skills, which are usually complex and difficult for most students to master. Researchers have reported that many lecturers have encountered difficulties in teaching programming languages. Moreover, most students and teachers have the same opinion that learning programming is a challenging task that many students struggle with (Govender & Grayson, 2008; Kordaki, 2010). Therefore, it has become an important and challenging issue to develop improvement strategies or tools for teaching computer-programming languages (Emurian, Holden, & Abarbanel, 2008; Hwang, Shadiev, Wang, & Huang, 2012).

According to Nwanaka and Amaechule (2011) there are three stages in skills acquisition: theoretical, practical and exposure to challenges. An important factor in skills acquisition process is exposure to practical situations where these skills are displayed and utilized. It is thus essential that at polytechnic students be given the required practical skills, which they need to cope with emerging challenges of the modern world.

However, skill is thought of as a quality of performance which does not depend solely upon a person's fundamental, innate capacities but must be developed through training, practice and experience. In addition, skills represent particular ways of using capacities in relation to environmental demands (Adeyemo, 2009). Nwanaka and Amaehule (2011) emphatically states that it is only with skilled men that materials can be harnessed, manipulated and changed into products.

Gumbari (2009) is of the opinion that, there is no issue that should be addressed as a matter of urgency and utmost importance than that of skills acquisition by the youth, considering the failure of our basic education to yield the expected positive results with its attendant consequences such as armed robbery, insurgency, militancy, kidnapping, abduction for ransom and a lot of others.

Researchers have argued that when learning programming, continuous practice is compulsory to ensure that the knowledge is sustained (Chen, Chang, & Wang, 2008; Hwang, Wang, Hwang, Huang, & Huang, 2008). Moreover, actively and periodically scheduled learning is important for students to reach high levels of achievement (Hwang & Wang, 2004). Nevertheless, many computer science students

cannot grasp the most fundamental concepts of programming and are thus unable to produce even the most basic programs (Eckerdal, 2009). Learning strategy, lack of study, and lack of practice have been identified by researchers as the fundamental attributes of success or failure in a computer programming course (Hawi, 2010; Hwang, Wu, Tseng, & Huang, 2011).

3. Methodology

The research was conducted in an academic institute which offers diploma degree in computer science. Descriptive and qualitative method was employed by identifying the characteristics of the students in programming skills. The grades for each student per semester were utilized to determine the progress of students in programming course. In addition to students' grades, informal interviews with some selected faculty members were conducted to gather additional information that is useful for this research.

3.1. Presentation of Data and Discussion

The Cumulative Grade Point Average (CGPA) is the weighted average of the GPA of all the semesters.

Probation:	CGPA between 1.50 – 1.99
Pass:	CGPA between 2.00 - 2.49
Lower Credit:	CGPA between 2.50 - 2.99
Upper Credit:	CGPA between 3.00 - 3.49
Distinction:	CGPAbetween 3.50- 4.00

3.2. Semesters Grade Distributions

The following sections discuss the results of the students academic performance in the programming course for the first and second semester (year one) and first and second semester (year two).

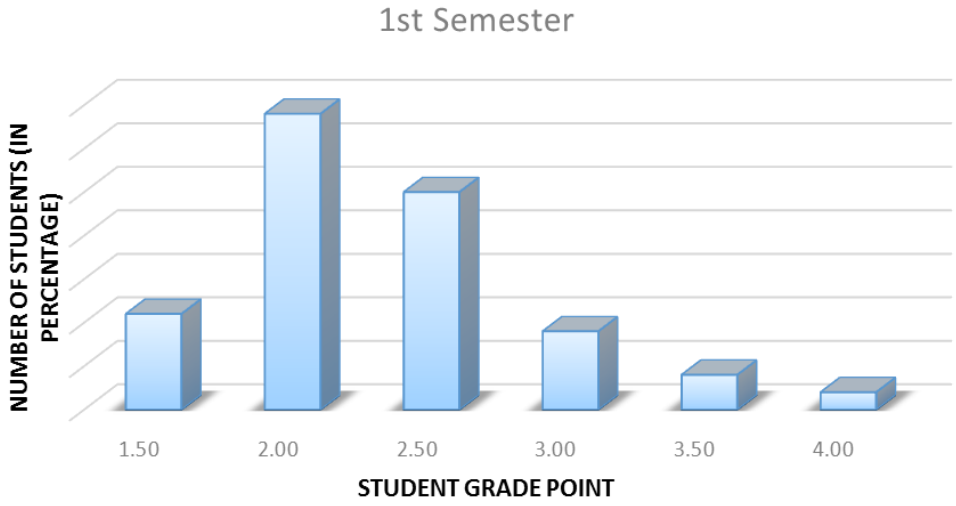


Fig. 1. *Percentage distribution of the student grade for the first semester.*

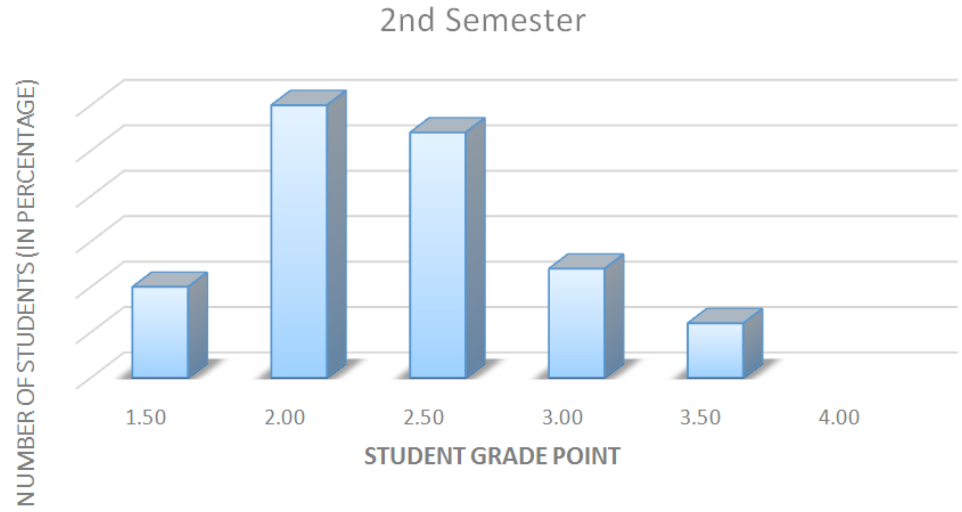


Fig. 2. *Percentage distribution of the student grade for the second semester.*

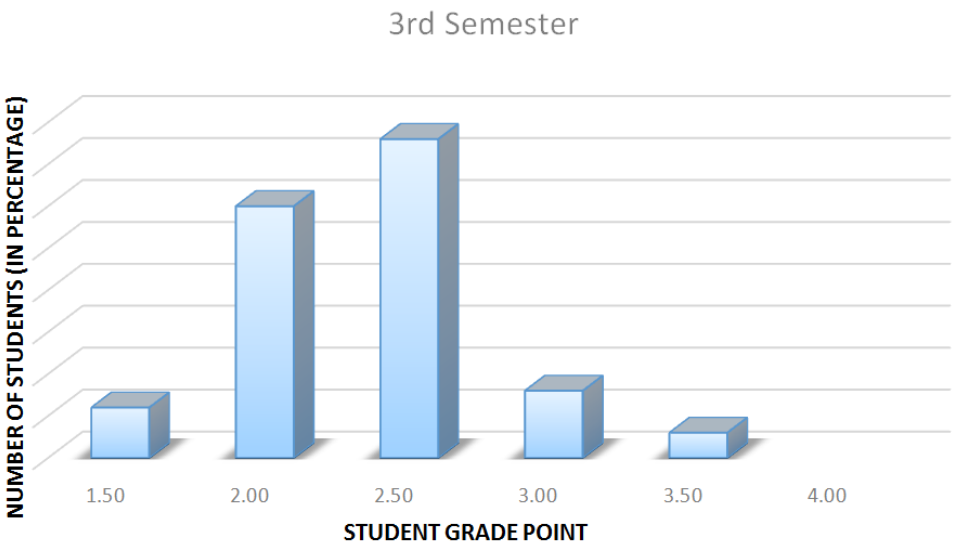


Fig. 3. *Percentage distribution of the student grade for the third semester.*

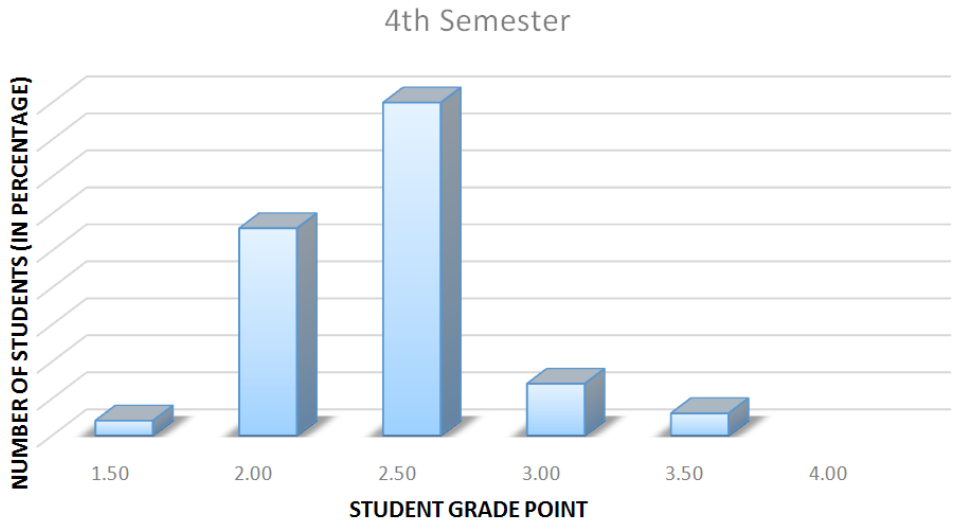


Fig. 4. *Percentage distribution of the student grade for the fourth semester.*

3.3. Analysis of the Results

The following paragraphs are the analysis of the above charts.

During the first semester (fig. 1) the results showed that majority of the students 34(40%) had grade point average of pass, while 25(29%) had lower credit, 11(13%) had probation, 9(11%) had upper credit and 6(7%) had distinction. However, second semester (fig. 2) results revealed that higher percentage of the student had GPA of 30(35%), followed by 27(32%) which is lower credit, 12(14%) had upper credit, 10(12%) had probation and 6(7%)had distinction.

Furthermore, in the third semester (fig. 3) there is slight change as the results showed an improvement, 38(45%) had GPA of lower credit, 30(35%) had pass, 8(9%) had upper credit while 6(7%) and 3(4%) are probation and distinction respectively. Finally, in the fourth semester (fig. 4) 45(53%) had lower credit, while 28(33%) had pass, 7(8%) had upper credit, 2(2%) and 3(4%) had probation and distinction respectively.

The above analysis clearly indicate poor academic performance of the student as only few of them can attain the level of upper credit and distinction. In addition, lack of interest, poor attendance and non-challant attitude to lecture and assignment have been identified by some lecturers for their poor performance. Poor attendance has a negative impact on students academic performance, N. Harb and A. El. Shaarawi (2006).

In conclusion, the above results indicate that improvement strategies are needed in order to boost the students' academic performance in programming skill.

4. Recommendations

1. Computer science courses should be taught by exemplary lecturers/instructors who have the requisite knowledge to teach the curriculum and who continue to upgrade their technical and teaching skills throughout

their careers because of rapid changes and advancement in the world of technology.

2. There is need to implement a national computer science curriculum for high schools(Senior Secondary School and Junior Secondary School). This curriculum must be principle-based, must address core content and key skills, and must incorporate appropriate strategies to reach and teach our students.
3. There is need to support the implementation of this new curriculum with a plan that includes a realistic timeframe and the provision of the resources required to achieve it.
4. There should be program to mentor, counsel, advise, guide, and tutor student. There should be academic support through extra lessons, remedial lessons which should be facilitated or supervised by a qualified and dedicated person who supports the department's vision. Also, in order for the tutor to be effective, the tutoring style should match with the students' learning style. As defined, learning style refers to the individualized ways we take in, process, and organize information J. Truschel. (2012). And the learning style is categorized as visual, auditory, and kinesthetic learning styles. Visual learning style work best, when they are able to see it. Auditory learning style work best, when they are able to hear it. While kinesthetic learning style work best, when they are doing it or physically involved with it.
5. Instructor's Capacity Building. Instructors' should be given opportunity to develop and improve their teaching skills by attending seminars, conferences etc and should be allowed to teach other instructors as well.
6. Using Small Groups. Having students work in small groups or in student pairs is an effective instructional strategy especially for struggling students. In small groups, struggling students are more likely to ask questions and benefit from explanations from fellow students.

This issue requires a national vision, supportive action, and commitment at all levels of the political and educational systems as schools alone cannot achieve these outcomes.

References

[1] Adeyemo, S. A. (2009). Understanding and Acquisition of Entrepreneurial Skills: A pedagogical Re-orientation for Classroom Teacher in Science Education. *Journal of Turkish Science Education.* 6 (3), 57-65.

[2] Brooks, R. (1999). Towards a theory of the cognitive processes in computer programming. *International Journal of Human Computer Studies, 51*, 197–211.

[3] Butler, A. C., Karpicke, J. D., & Roediger, H. L. (2007). The effect of type and timing of feedback on learning from multiple-choice tests. *Journal of Experimental Psychology: Applied, 13*(4), 273–281.

[4] Chen, G. D., Chang, C. K., & Wang, C. Y. (2008). Using adaptive e-news to improve undergraduate programming courses with hybrid format. *Computers & Education, 51*, 239–251.

[5] Eckerdal, A. (2009). *Novice programming students' learning of concepts and practise* (Doctoral dissertation), Retrieved from http://uu.divaportal.org/smash/record.jsf?pid=diva2:173221

[6] Emurian, H. H., Holden, H. K., & Abarbanel, R. A. (2008). Managing programmed instruction and collaborative peer tutoring in the classroom: Applications in teaching JavaTM. *Computers in Human Behavior, 24*, 576–614.

[7] Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education, 63*, 87–97.

[8] Govender, I. (2009). The learning context: Influence on learning to program. *Computers & Education, 53*, 1218–1230.

[9] Gumbari, J (2009). The Importance of Skills Acquisition: A Challenge to Nigerian Legislators. Retrieved September 20, 2013 from www.nasslegislatonline.com

[10] Hawi, H. (2010). Causal attributions of success and failure made by undergraduate students in an introductory-level computer programming course. *Computers & Education, 54*, 1127–1136.

[11] Hwang, G. J., Wu, C. H., Tseng, Judy C. R., & Huang, I. W. (2011). Development of a ubiquitous learning platform based on a real-time help-seeking mechanism. *British Journal of Educational Technology, 42*(6), 992–1002.

[12] Hwang, W. Y., Shadiev, S., Wang, C. Y., & Huang, Z. H. (2012). A pilot study of cooperative programming learning behavior and its relationship with students' learning performance. *Computers & Education, 58*, 1267–1281.

[13] Hwang, W. Y., & Wang, C. Y. (2004). A study on learning time pattern in asynchronous learning environments. *Journal of Computer Assisted Learning, 20* (4), 292–304.

[14] Hwang, W.Y., Wang, C. Y., Hwang, G. J., Huang, Y. M., & Huang S. (2008). A web-based programming learning environment to support cognitive development. *Interacting with Computers, 20*, 524–534.

[15] J. Truschel. (2012). Habits of a Highly Effective Tutor. [Online]. Available: http://www.myatp.org/wp-content/uploads/2012/06/Synergy-Vol-1-Truschel.pdf

[16] Katai, T, & Toth, L. (2010). Technologically and artistically enhanced multi-sensory computer-programming education. *Teaching and Teacher Education, 26,* 244–251.

[17] Kordaki, M. (2010). A drawing and multi-representational computer environment for beginner learning of programming using C: Design and pilot formative evaluation. *Computers & Education, 54,* 69–87.

[18] N. Harb and A. El-Shaarawi (2006). Factors Affecting Students' Performance. Online0 available: http://mpra.ub.uni-muenchen.de/13621/1/mpra_paper_13621.pdf.

[19] Nwanaka, C. R. and Amaechule, S. (2011). Skills Acquisition: Imperative for Business Studies Educators among Secondary Schools in Rivers State. *Mediterranean Journal of Social Sciences*, 2(7) 37-43 Retrieved September 25, 2013 from www.mcser.org/images/stories/MJSS-Special-issues/.../nwanaka-c-r.pdf

[20] Regina A. Garcia and Lilac A. Al-Safadi (2014): Intervention Strategies for the Improvement of Students' Academic Performance in Data Structure Course . *International Journal of Information and Education Technology, Vol. 4, No. 5, October 2014*

[21] Verdú, E., Regueras. L. M., Verdú, M. J., Leal, P. J., Castro, J. P., & Queirós, R. (2012). A distributed system for learning programming online. *Computers & Education, 58,* 1–10.

[22] Yang, T.-C., Hwang, G.-J., Yang, S. J. H., & Hwang, G.-H. (2015). A Two-Tier Test-based Approach to Improving Students' Computer-Programming Skills in a Web-Based Learning Environment. *Educational Technology & Society, 18* (1), 198–210.

[23] Yeh, Y. F., Chen, M. C., Hung, P. H., & Hwang, G. J. (2010). Optimal self-explanation prompt design in dynamic multi-representational learning environments. *Computers & Education, 54*(4), 1089–1100.

[24] Wang, Y., Li, H., Feng, Y., Jiang, Y., & Liu, Y. (2012). Assessment of programming language learning based on peer code review model: Implementation and experience report. *Computers & Education, 59*, 412–422.