International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 12, December-2021

The Effect of Massification on Student Experiences: A Case of Higher and Tertiary Students Studying Introduction to Computer Programming Course in A Large Class Set-Up

Tirivashe Mafuhure[1], Gabriel Kabanda[2], Maria Tsvere[3]
Chinhoyi University of Technology,
Zimbabwe

***Abstract*:- This paper explored current experiences of computer students learning an introductory computer programming course in a large class set-up. The study utilized the descriptive survey design and data was collected by making use of a structured questionnaire which was distributed to students in the school in introduction to C programming language at Chinhoyi university of technology. Data that was gathered in this study was quantitative in nature and SPPS version 20 was used to analyze it. Results obtained in the study show that large class environments are not suitable for complex and hands on modules such as computer programming. The majority of student find it complicated to grasp concepts in computer programming and as a result fail the module or memorize content instead of engaging into critical thinking. Since enrolling students in large numbers to fund operations in Zimbabwean universities is the order of the day, this study recommended the development and use of teaching tools that will assist students develop skills such as Self-regulated learning strategies (SRL) to navigate through challenges of large numbers. Despite the challenges that computer programming students face SRL strategies will enable them to be critical thinkers and problem solvers.**

Keywords:- Massification, Computer programming, higher and tertiary Education, large class, Student experiances

1. INTRODUCTION

There is high demand for higher and tertiary education in Africa and as a result public and private universities are enrolling many students a phenomenon now known as massification (Adetiba, 2019) [1]. The term massification was defined as the increase in the enrollment of students without proportionate boost in resources (Lee, 2016) [2]. Daniels (2016) [3], postulates that universities in Africa are incapacitated enough to meet the huge enrollments of students coming from high schools. The massive numbers of students result in a high student lecturer ratio and as a result some lecturers will become overwhelmed with work (Dougherty et al., 2016) [4].This study was done when financial support from government to universities dwindled and enrolling many students to fund operations became the order of the day. Large class numbers have brought forward a number of challenges to complex and hands on courses such as computer programming(Safari, 2017) [5]. Computer programming is not popular with many students because it is very complex to learn (Panadero et al., 2016) [6].Students needs to be motivated so that they become critical thinkers and develop problem solving skills to be successful in the programming field (Pedrosa et al., 2016) [7]. The major purpose of this study is to establish student experiences when learning introduction to computer programming in a large class setup. Since this is part of a PhD study currently underway, the experiences obtained will assist in coming up with technological solutions such as developing teaching tools that will help enhance the learning and teaching of computer programming in a large class setup.

RESEARCH OBJECTIVE

This study sought to:
Establish students' current experiences when learning introduction to computer programming in a large class setup.

LITERATURE REVIEW

Massification has changed the traditional way were universities were seen as centers for the elite and only very few students would gain access (Daniels, 2016) [3]. The increased enrollment in higher and tertiary education is promoted in many countries due to its links to improved health, economic development and empowerment (Mok, 2016) [8]. According to Lee (2016) [2], massification in many countries is as a result of democratisation of the education sector and the social justice agenda. Democratization of higher and tertiary education refer to the practice were access to education at this level is for everyone irrespective of social and economic status in society (Ward, 2016) [9]. Universities in Zimbabwe are enrolling many students as a way tofund their operations since government financial grands have dwindled (Garwe, 2013) [10].Learning computer programming is very difficult (Robins et al., 2003). Effective learning of computer programming requires motivation and involvement (Nunes et al., 2015) [11]. In higher and tertiary education, computer programming records higher academic failures compared to other modules(Safari , 2017) [5]. Panadero (2016) [6] postulates that there is greater need for a strong student lecturer relationship inorder for students to be successful in the field of computer programming. Students needs to develop problem solving skills rather than to memorise content such as sythax of programs (Pedrosa et al., 2016) [7]. Chen (2011) [12] further states that many computer programming students are not competitive in the job market and in many cases fail to meet the expectation of industry due to l;ack of hands on experiance, teamwork and cooperation skills.According to Mohamedbhai (2008) [13],big class environments appear to be there for a very long time hence the purpose of this research was to investigate the current experiences of students who are learning computer programming in a large class environment in higher and tertiary education. The study want establish whether the experiances faced by students who are learning computer programming in a large class setup are supportive to enhance critical thinking and poblem solving skills.

DATA COLLECTION METHOD

This study targeted students who were enrolled in introduction to computer programming module at Chinhoyi University of Technology (CUT). This study used a structured questionnaire to gather data from the students and the questionnaire was administered in person by the researcher after getting an approval letter from CUT management. Sample size was calculated using the sample size determination table by Krejcie and Morgan. The population was 160 students doing introduction to computer programming. Using the table, the sample size was 113. Out of 113 questionnaires distributed to students 90 were returned and utilised. The data was analysed using SPSS version 20.

FINDINGS AND DISCUSSIONS

From table 1 above the percentage of female students is 41% compared to 58% for male students. Male students dominate the field of computer programming because computer programming is a very complex subject hence few female students enrol in IT or computer science

Table 1: Frequency table

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Male	53	58.9	58.9	58.9
	Female	37	41.1	41.1	100.0
	Total	90	100.0	100.0	

REGRESSION TEST

This measures the association amid independent and dependent constructs.

Table 2: Regression model summary

Variable	Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Change Statistics				
						R Square Change	F Change	df1	df2	Sig. F Change
Lecturer attends to individual programming needs of each student	1	.481[a]	.231	.175	.665	.231	4.153	6	83	.001
Instant feedback is received from the lecturer	2	.460[b]	.211	.154	.644	.211	3.704	6	83	.003
The lecturer helps me to be a problem solver	3	.226[c]	.051	.-018	.566	.051	0.744	6	83	.616
Class size is optimum for interactive teaching and learning	4	.226[d]	.071	.004	.631	.071	1.057	6	83	.395

a.Predictors: (Constant), Lecturers allow us to make plans in a large class, There is close lecturer student relationship in a large class, It is motivating to study programming in a large class, There is proper involvement and engagement leaning programming in large class, Lecturers allow us to set goals in a large class, Critical thinking and problem solving are enhanced in large computer programming class social media, media sharing networks and consumer review networks.

b. Dependent Variable: student experiences

As shown on table 1 above, from model 1 up to model 4, the regression co-efficient, the regression square and the adjusted regression square are all below 0.5. This confirms that massification does not improve student positive experiences when studying computer programming. On the first variable which states that the lecturer attend to individual programming needs of each and every student, the regression co-efficient is 0.481. This implies that it is very complicated for a lecturer to attend each student due to high student lecturer ratio. Modules such as computer programming are complex and a number of student struggle to comprehend concepts when they are taught for the first time. There will be need for the lecturer to attend each student taking note of their strengths and weaknesses.

Receiving instant feedback from the lecturer is very critical in computer programming. As show on the second variable which states that instant feedback is received from the lecturer, the regression co-efficient is 0.46, the regression square is 0.211. Since both the regression co-efficient and the regression square are all below 0.5, this shows that students do not get feedback instantly from their lecturers. It will be very complicated for students to self-evaluate themselves and take note of areas of strengths and areas of weaknesses as the module is being taught. Lack of feedback demotivates students and as a result high rates of academic failure are experienced in computer programming

Problem solving skills are very critical in computer programming. On the third variable which states that the lecturer helps the student to be a problem solver, the regression co-efficient is 0.226 and the regression square is .051. Both the regression co-efficient and the regression square are below 0.5. This implies that lecturers face challenges to assist students to be problem solvers through participation and engagement. It is difficult to involve and engage over 160 students in a lecture especially when the lecturer is 1. Lack of involvement and engagement demotivates students and many of them end up memorising content learnt to pass the exam instead of engaging into critical thinking. When students finish their studies, they often face challenges even in industry when assigned programming tasks.

On the fourth variable which states that the class size is optimum for interactive teaching and learning again the regression co-efficient is 0.226 and the regression square is 0.071. This clearly shows that students are not comfortable to learn a module such as computer programming in a large class setup. Large classes are characterised by lack of resources such as computers and teaching aids, lack of concentration and involvement. The student lecturer ration is very high and most of the time students get minimum help and feedback from the instructor and in the end affect quality teaching and learning.

CONCLUSION AND RECOMMENDATIONS

Results from the data analysed reveal that large class setups in computer programming are not proper for students to develop problem solving skills. High lecturer student ratio and lack of resources leads to lack of concentration, motivation, engagement and involvement to programming students. Students end up memorizing content learnt instead of engaging into critical thinking to solve problems. It seems that large classes are there for a very long period of time especially to universities who lack funding from the government. There is need for universities to assist programming students with e-learning teaching tools so that they can learn how to solve programming challenges even outside the classroom. Some authors point to the fact that computer programming students need to develop Self-regulated learning strategies (SRL) to overcome whatever challenges they face in the subject. This study recommends the development of computer programming teaching tools that will help motivate and infuse self-regulating strategies to students learning computer programming in a large class environment. There is need to teach computer programming students to develop skills such as goal setting, planning, understand programming syntax, comprehending written programs, fixing errors and most importantly problem analysis and solving. Although lecturers are urged to train students to be critical thinkers and problem solvers, achieving that in a large class is very complex hence there is need for students and lecturers to interact outside the classroom by making use of e-learning platforms. Students' needs to be given tasks and exercises on the e-learning platform and instant feedback should be given to them after completion. This helps students to self-evaluate themselves and self-evaluation is critical for one to be successful in computer programming (). Specialized e-learning tools should be developed to cater for a specific subject due to their different needs. Computer programming is a complex subject and therefore requires lecturers to develop online teaching tools that specifically for computer programming so that students get motivated and in the end develop problem solving skills

REFERENCES

[1] Adetiba Toyin Cotties, 2019. Massification of higher education in South Africa, the good, the bad and the ugly, Proceedings of International AcadeMmic Conferences 9410873, International Institute of Social and Economic Sciences.
[2] LEE, S. (2016) Massification without equalisation: the politics of higher education, graduate employment and social mobility in Hong Kong. Journal of Education and Work. Vol. 29, No. 1, s. 13-31,
[3] Daniels, R. J., & Spector, P. (2016). Converging paths: Public and private research universities in the 21 st century. TIAA Institute.
[4] Dougherty, K. J., Jones, S. M., Lahr, H., Natow, R. S., Pheatt, L., & Reddy, V. (2016). Performance funding for higher educa-tion. Baltimore: Johns Hopkins University Press.Retrieved at https://www.tiaainstitute.org/public/pdf/converging_paths_daniels_spector.pdf
[5] Safari, E., Hejazi, M. (2017). Learning styles and self-regulation: an associational study on high school students in iran. *Mediterranean Journal of Social Sciences*, 8(1), 463-469
[6] Panadero, E., Jonsson, A., & Strijbos, J. (2016). *Assessment for learning: meeting the challenge of implementation.* Switzerland: Springer International Publishing.
[7] Pedrosa, D., Cravino, J., Morgado, L., Barreira, C., Nunes, R., Martins, P., et al. (2016). Simprogramming: the development of an integarated teaching approach for computer programming in higher education. *International Technology, Education and Development* , 70-81.
[8] KA HO MOK (2016) Massification of higher education, graduate employment and social mobility in the Greater China region. British Journal of Sociology of Education. 2016, Vol. 37, s. 51-71.
[9] Ward, J. D. (2016b). Unintended consequences of for-profit college regulation: Examining the 90/10 Rule. Pullias Center for Higher Education Working Paper.
[10] Garwe, E. (2014). Quality assurance in higher education in Zimbabwe.*Research in higher education*.ISSN 2327-7092
[11] Nunes, R., Pedrosa, D., Fonseca, B., Paredes, H., Cravino, J., Morgado, L., et al. (2015). *Universal access in human-computer interaction: access to learning, health and well-being.* Switzerland: Springer International Publishing
[12] Chen, C. L., & Lin, J. M. C. (2011). Learning styles and student performance in java programming courses. *In Proceedings of the International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS)(p. 1).*The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).
[13] Mohamedbhai, G., 2008, 'Mauritius', in D. Teferra and J. Knight (eds.) *Higher Education in Africa: The International Dimension*, Boston: Centre for International Higher Education and Association of African Universities.