TEACHING ADVANCEMENT AT UNIVERSITIES

Final Report

Name: Dr MULEMWA AKOMBELWA Institution: UKZN Enquiry Group 10

HOW TO KILL A KILLER COURSE: Overcoming Challenges of an Introductory Programming Course

1. Introduction

The research project was aimed at exploring the challenges faced by Land Surveying students in computer programming and its applications in the Discipline of Surveying at the University of KwaZulu-Natal. The aim of the research project was to explore and identify the factors affecting learning and to design an intervention that will assist Land Surveying and potentially other students overcome this challenge.

Computer Programming is considered to be a 'killer course' by Land Surveying students. They have exhibited challenges in subsequent courses offered in the discipline where they are required to apply computer programming knowledge and skills. Interventions have previously been applied to help address the situation but these have not helped as they did not include any input from students. The rationale of this research was to establish why the computer programming course is considered to be a killer course and to design an intervention for students to help overcome the challenge.

There was a need to develop a clear understanding of the problem so that evidence-based interventions to this problem could be devised to adequately address the challenges because previous interventions have not worked and students continued to encounter challenges with programming. It is not clear why this module is problematic for Land Surveying students especially given that they are 3rd year students who demonstrate a capability to solve problems in other courses that they are registered for. In order to address the problem, this project employed a ***two-stage process*** of ***firstly,*** finding out what the challenges are using *the Focus Group* approach and then using the information collected to design an intervention based on the perceived challenges. The ***second stage*** is to apply the intervention through *Small Group learning* approaches to another cohort taking the same course and evaluate if any changes would arise. It is hoped that there would be an improvement in the pass rate of Land Surveying students taking the introductory computer programming module in the Computer Science Department.

2. Rationale, Process and Research Objectives

While the ability to program has tremendous potential to support and channel the creative power of people, new programming languages continuously arise as the need to solve new problems emerges. There is a need by industry for more people to help deliver reliable and efficient software solutions to society (Ball & Zorn, 2015).

It has been suggested in the literature that computer programming requires skills in critical thinking, problem-solving, computational thinking and new system designs (Topalli & Cagiltay, 2018). The need for computational thinking is why computer programming is a central part of engineering curricula worldwide (Vaca-Cárdenas et. al, 2015; Ball & Zorn, 2015). Unfortunately, the attrition rate for introductory computer programming courses has been found to be quite high (Beaubouef & Mason, 2005). According to Allan & Colesar (1996), many problems in learning computer programming originate from the complexity of concepts such as variables, loops and so on.

The study seeks to achieve the following objectives:

1. To understand the factors that affect the pass rate for the introductory computer programming module.
2. To design an intervention based on the factors identified by the research participants (staff and students)
3. To test whether application of the intervention will improve student's ability to program and hence change the pass rate?

Research questions

i. Can Focus groups be used to identify and address the challenges that introductory computer programming students face?

ii. Can Small Group Learning be employed to apply the intervention developed from Focus Groups to address challenges in introductory computer programming?

3. Conceptual framework

This research project aims to develop and test an intervention model to address challenges in introductory computer programming by drawing from the theory of Constructive Alignment (Biggs & Tang, 2011), constructivist theory (Schunk, 2012) through Small Group Learning (Edmunds & Brown, 2010) and social learning theory (Bandura, 1971, 1977, 1986). This suggests that small group learning activities in computer programming education programmes can enhance the programming ability of students. These learning activities should be directed at enabling students to achieve computational thinking (Topalli & Cagiltay, 2018) and for the students to be able to write

programming code. This requires students to ascribe, through the focus groups, the factors that lead to an achievement outcome such as passing or failing (Hawi, 2010).

In the context of computer programming, self-efficacy adapted from Bandura, (1986:391) can be viewed as the degree to which individuals believe they have the ability to successfully solve a problem and apply computational thinking to it. The effectiveness of the resulting intervention will be assessed using students' creative programming design solutions to problems. The intervention will be carefully designed and implemented with due consideration of the ***What?*** For ***Whom?***, ***Why?***, ***How?*** and For ***Which*** results? as suggested by Fayolle and Gailly (2008).

4. Methods and Results

4.1 Research design

The study will adopt a Design-Based Research method (DBR) (Figure 1). DBR is regarded as a new approach that is applicable to research in education which is guided primarily by the research problem and objectives with the aim of improving the theory and the educational context (Abdallah & Wegerif, 2014). It is a flexible approach that enables researchers employing DBR to design interventions based on theory, test them and refine the theory which helps improve not only the intervention but practice which is based on the theory (Abdallah & Wegerif, 2014). This study is based on both qualitative and quantitative research methods that will involve conducting interviews and focus groups to collect data from students and staff prior to the design and implementation of the intervention.

The first stage of the study was conducted in terms of the four phases (see Figure 1), as suggested by Herrrington et. al, (2007). The initial sample for this research project involved final year students. The students were requested to participate in the study through a focus group meeting. These students are a convenient and purposive sample that is aimed at identifying the main issues before design of the intervention that will be implemented on a class of 3rd year students who will be attending the course next year. The study also involved two (02) colleagues who lecture computer programming. Their views will help inform the design of the intervention. These colleagues will be interviewed by means of an interview.

Data collection tools

Data was collected using an audio recorder, handwritten notes, and structured questions for the focus group meeting and interviews with the colleagues.

Data collection procedure

Focus group interviews were conducted with 4th year students who have completed the programming module to obtain information about the weaknesses of the current teaching methods and suggestions with regard to what could be improved to enhance creativity

4.2 Results

The Digital recordings were transcribed and handwritten notes typed into a spreadsheet (Leedy & Ormrod, 2018). The qualitative data was then summarised and categorised into themes, and where possible be quantified and analysed using statistical software (Saunders, Lewis & Thornhill, 2016). Quantitative data analysis will be conducted using SPSS Version 25. The main themes identified from the focus group meeting are illustrated in Table 1.

Table 1: Themes Identified from Focus Group Meeting

Duration of Module	Programme Structure	Relevance of Modules	Learning
One semester not enough	Current curriculum structure does not adequately support computer programming	Understand the need for computing	Large Class - not enough interaction with the lecturer
Should run for longer	Should be split in to 3 modules covering Basic, Intermediate and Advanced concepts	Biased to department offering modules	SI tutors & demonstrators not adequately trained to support them
Should begin in 2nd year (to allow students to adjust to university teaching)			Preferably want a Land Surveyor to teach the Programming module with more LS examples

The results indicate that there are 4 main themes that could be drawn from the focus group meeting, namely the *duration of the module*, the *programme structure*, *relevance of modules* and *Learning* which embodies the direct challenges they encounter with the learning process. A closer inspection of the themes seems to suggest that the *Learning theme* is affected by the other themes.

The issues identified require different types of intervention to be devised with some requiring long term action such as those requiring curriculum adjustment or employment of extra staff to deliver

the course within the department. However, for the ones with direct impact, a table with proposed design principles has been created and is outlined in Table 1 in the appendix. The design principles will be implemented in Phase 3 which involves iterative cycles of testing and refinement of the solutions in practice on another cohort of 3rd year students next semester. This will be followed by Phase 4 which is based on a deep reflection of the outcomes of the initial design to produce final design principles and enhance solution implementation. In this phase the intervention will be revised based on the results of the previous phase and then be implemented.

It can be seen from Table 1 that learning from a large class is a big challenge for the students, and that the support during tutorials is ineffective hence it is important to design principles which will address these main learning inhibitors. This is the basis for the design of the design principles.

This is still a work in progress and the results shown are inconclusive until after testing on another cohort of students is done and the results analysed. The work will continue with the possibility of being extended to the Computer Science and Mechanical Engineering departments within the School of Engineering at UKZN where computer programming courses are offered. The lecturers offering the courses have expressed interest to have the study expanded in order that they may also find out how they can improve the delivery of their modules to improve learning in computer programming in their programmes.