Enhancing academic achievement in an introductory computer programming course through the implementation of guided inquiry-based learning and teaching

[1]Jose LUKOSE and [2]Kuttickattu John MAMMEN

[1]Department of Information Technology, Faculty of Science, Engineering and Technology, Walter Sisulu University, 19 Manchester Road, Chiselhurst, East London 5200, SOUTH AFRICA

Corresponding Author's E-mail: jlukose@wsu.ac.za

[2]School of General & Continuing Education, Faculty of Education, University of Fort Hare, 50 Church Street, East London 5200, SOUTH AFRICA

E-mail: jose_mammen2002@yahoo.com

Received 20 Aug., 2018
Revised 21 Dec., 2018

Contents

Abstract

Research reports using global data show that the failure rates in introductory programming courses average about 32%. As learners from schools join the university and enrol for different courses, they find the sudden transformation quite challenging. This makes it more challenging for first year university students, especially in difficult courses such as Introduction to Programming. As trends change with advances in technology, the traditional ways of presenting information during teaching and learning interface may not address students' needs. Lecturers and university stakeholders however, do make efforts to address these challenges by proposing innovative teaching methods. One of the common approaches that has been used profitably in certain science and engineering programmes is Guided Inquiry Learning (GIL). It is a form of inductive collaborative learning approach; where students are challenged to accomplish the desired learning outcomes in the course. This research applied GIL to the year-long Development Software 1 (DEV1120) course, which deals with basic programming principles that apply to all computer programming languages. The purpose was to observe whether or not there were more gains through a GIL approach in students' academic achievement than through traditional teaching. The research was located in the pragmatic paradigm using action research design and a mixed method approach. The population consisted of all the 49 first-year students enrolled for the course at a South African university. The students who volunteered to be included in the experimental group were taught using the GIL strategies while the other group were taught using traditional method. Both groups were assessed using the same assessment tools simultaneously. Results from these assessments, together with focus-group interviews, provided the core data for this study. Both quantitative and qualitative analyses were carried out on the data: statistical analysis for the former (chi-square and t-test) and thematic analysis for the latter. Results indicated significant gains in academic achievements for the experimental group over those in the control group.

Keywords: Constructivism, Guided Inquiry Learning (GIL), computer programming, collaborative learning, entry-level students, novice programmers.

Introduction

Programming is an important course taught in higher education institutions as part of the curriculum of the undergraduate programmes in Computer Science (CS) and Information Technology (IT) disciplines. This paper emanates from a larger study which was carried out at a South African comprehensive university. Computer programming (often called programming) is the process of developing and implementing various sets of instructions to enable a computer to do one or more pre-planned task(s). There is general agreement in the literature that learning to program is a difficult task (Jenkins, 2002; Teague & Roe, 2008; Yang, Hwang, Yang, & Hwang, 2015; Malliarakis, Satratzemi, & Xinogalos, 2017). The paper presents the part of the study which sought to find the differences in academic achievement between students who were taught traditionally and those taught using a GIL (Guided Inquiry Learning) approach in learning computer programming. GIL is a type of inductive collaborative learning approach where students are presented with a challenge which challenges them to accomplish the desired learning outcomes in the process of responding to the posed challenge.

Commonly offered as a first-year, core course, introductory programming courses have an alarming failure rate (Bennedsen & Caspersen, 2007; Costa et al., 2017; Iqbal Malik & Coldwell-Neilson, 2017; Koulouri, Lauria, & Macredie, 2015; Malik & Coldwell-Neilson, 2017; Sheard & Hagan, 1998; Watson & Li, 2014). At the selected university the course is known as Development Software 1 (DEV1120). The Innovation and Technology in Computer Science Education (ITiCSE) working group in 2001 assessed the programming ability of an international population of first-year computer science students from several universities (McCracken et al., 2001). According to this study, the majority of entry-level programming students found it difficult to grasp the fundamentals and foundation level programming concepts at their early stages of learning (programming), and they performed poorly on a set of common program-writing problems. Entry-level students often struggle initially to grasp and understand programming and this can lead to frustration and eventually, surrender (Horton & Craig, 2015; Petersen, Craig, Campbell, & Tafliovich, 2016; Shuhaidan, Hamilton & D'Souza, 2009).

The introductory programming course, DEV1120, is offered at entry-level of the three year programme: National Diploma: Information Technology (NDIT). Over the past several years this course has had an alarming failure rate. Since 2010, an average of 51.44% of students has been failing in this introductory programming subject. The assessment of this subject consists of four written tests of incremental difficulty and an end-of-year final examination. Average marks scored in these four tests is calculated to generate a student's year mark. The average year mark (termed in South Africa as Duly Performed mark - DP) averaged between 39.06 and 57.03 with an average of 46.51 for the past five years. Learners are not allowed to do the final examination should they fail to accumulate a minimum of 40% for their year mark (DP). Those who qualified for the final examination averaged a 53.51% examination mark for the past five years. This reflects the depth of the problem in that a great proportion of the students who enrol at this institution have very little or no prior knowledge of basic computer skills.

As alluded to by several researchers, difficulties in learning to program and the high failure rate in introductory programming modules are a global phenomenon and not unique to any specific sector of the population (Bennedsen & Caspersen, 2007; Horton & Craig, 2015; Watson & Li, 2014). Bennedson and Casperson (2007) also argue that low retention rates are often experienced in computer science schools and students' inability to learn programming has been cited as a major cause. Two large-scale global research studies conducted by ITiCSE working groups by McCracken et al. (2001) and Lister et al. (2004) tested entry-level university students' common program-writing problems. The first study assessed the programming ability of students by testing them on common program-writing problems and reported that students struggled to achieve an average of above 30% on their assessments. The second one assessed code-reading and tracing skills of entry-level students and reported that approximately 25% of the students were guessing the solutions. Using global data, Bennedsen and Caspersen (2007) reported that the failure rates in introductory program ming courses averaged about 33% and Watson and Li (2014) concurred by reporting a similar figure of 32%.

Theoretical Framework

An understanding of the phenomena under study and its assumptions and philosophical views are broadly presented through the appropriate theoretical and conceptual frameworks (Devi, Das, Das, & Khandelwal, 2017). The theoretical framework for this study was based on Vygotsky's social constructivism and sociocultural theories and Bandura's social cognitive theory.

Vygotski was a social constructivist and he believed that learning takes place first on an interpersonal level through interaction with others. This is transformed into an intrapersonal one, which is internalized by the individual (Vygotsky, 1978). Understanding, significance and meaning are created through interactions with other human beings (Amineh & Asl, 2015). He introduced the notion of Zone of Proximal Development (ZPD) to explain learning and development. Vygotsky (1978) defined

ZPD as "the distance between the actual development level as determined by independent problem solving and the level of potential development as determined through problem solving under adult guidance or in collaboration with more capable peers" (p. 86). According to him, learning creates the internal proximal development; it awakens a variety of internal developmental processes that are able to operate only when the learner is interacting with people in his/her environment and in cooperation with his/her peers. Once these processes are internalized, they become part of the learner's independent developmental achievement. Social constructivist theory of learning proposes that students must be active participants in their own learning – through conversations and exchange of ideas with educators and other students – that help them reach new horizons of understanding (Harkness, 2009).

Vygotsky's sociocultural theory argues that the environment in which one operates defines one's personality and characteristics (Vygotsky, 1994, p.352). According to the sociocultural theory, the environment comprises the membership or the belonging to a specific social group, living in a specific historical period under specific historical circumstances. Vygotsky (1994, p.176) also states that "the environment that one is in has a greater impact on the personal developments that one gets in life".

Bandura (1986)explains that "human functioning is a model of triadic reciprocality; in which behaviour, cognitive and other personal factors, and environmental events all operate as interacting determinants of each other" (p.18). He is of the opinion that "most human behaviour is learned by observation through modelling" (p.47) (vicarious capability). This implies that when we observe others, we form rules of behaviour, and on future occasions this coded information serves as a guide for our own actions. There is a clear link between one's culture and history with regard to one's behaviour. For learning to be effective, there is a need to observe the behaviour of others (family members, peers, lecturers and role models) in order to determine how to act and what can work for each individual.

Learning basic computer programming principles and applying them in a problem solving domain do not occur for an individual learner alone but is also influenced by his/her peers and/or the teacher. This concurs with social constructivism theory which acknowledges the importance of belonging to a social group (class cohort) and the learning environment. The socio cultural theory refers to the human functioning model of triadic reciprocality and vicarious capability. Vygotski's and Bandura's theories therefore complement each other and both the sample and environment of this study mean the application of this eclectic view of the two theories is apt for the present research.

Literature Review

Overview of programming courses

Computer programming teaches learners to design, develop and manage computer programs with the objective of instructing a computer to carry out specific activities in order to yield the desired intention as required by the developer. Various researchers argue that understanding and learning to code are regarded as challenging tasks (Robins, Rountree & Rountree, 2003; Gomes & Mendes, 2007). According to Winslow (1996), programming needs critical thinking and translation of abstract concepts into real-life application which is not easy for many learners; students who are a bit slow in understanding abstraction therefore always find learning programming difficult both theoretically and practically (Winslow, 1996; Minelli, Mocci & Lanza, 2015).

Following an earlier study by Bennedsen and Caspersen (2007), Watson and Li (2014) analysed the failure rates in introductory programming courses across the world. Their revised study provided results on programming course literature. The data set containing the pass rate data included 161 introductory programming courses from 51 institutions across 15 different countries. The 2014 study indicated a mean global pass rate of 67.7%, which corroborated the finding of the first study, that is, 67% by Bennedsen and Caspersen (2007). The mean global failure and dropout rate was 32.3%. Their 2014 study also found that the mean failure rate in South Africa was 44% which better than the global mean failure rate.

Factors affecting learning on programming

Understanding of programming poses many challenges for first-year students. Ability to read and understand what a piece of code does was observed by Perkins and Martin (1986) to be an important skill required by new programmers. Entry-level programmers are often seen to be having difficulty in grasping the foundation-level programming concepts early in their studies, leading to grief and frustration and ultimately surrender (Shuhaidan et al., 2009.

Programming involves intensive problem solving skills and strategies. Understanding of algorithmic problem solving is considered to be at the core of learning computer programming (Lishinski, Yadav, Enbody, & Good, 2016; Sheth, Murphy, Ross, & Shasha, 2016). Huggard and Goldrick (2009) note that learners even fail to know where to begin their solution when faced with programming problems. While the McCracken group reported on entry-level programming students' poor problem solving skills (McCracken et al., 2001), the Lister working group reported that students even lacked in knowledge and skills that are a precursor or a pre-requisite to problem-solving such as reading and understanding the code and tracing or tracking skills (Lister et al., 2004).

Accessing relevant prior knowledge and adopting an approach to study that will go beyond memorizing, applying and transferring the domain concepts to a new situation are the main issues faced by entry-level programmers (Affleck and Smith, 1999). A large number of students enter ICT programmes with little or no relevant prior knowledge (Falkner & Munro, 2009) . Many students face programming courses for the first time in their lives. Lack of preparation, lack of previous exposure to computers and the level of complexity are some of the inherent problems with entry-level programming courses (Gonzalez, 2006).

Abstraction and abstract thinking are crucial components for learning computer programming (Or-Bach & Lavy, 2004; Bennedsen and Caspersen 2006). Difficulties with the abstract concepts of knowing how to model a solution to a problem, fragment it into manageable and codable subcomponents or sub-problems and then conceive a hypothetical error situation to test and figure out mistakes constitute a major issue faced by entry-level students (Esteves, Fonseca, Morgado, & Martins, 2011) . Students certainly need to learn a number of different skills and processes in learning to programme.

Guided Inquiry Learning (GIL)

Inquiry learning is a "learning process that uses questions and problems to provide contexts for learning" (Prince & Felder, 2006, p.127) where "students learn content as well as discipline-specific reasoning skills and practices by collaboratively engaging in investigations" (Hmelo-Silver, Duncan, & Chinn, 2007, p.100). According to Prince and Felder (2007), in GIL, students are presented with a challenge (such as a question to be answered, an observation or data set to be interpreted, or a hypothesis to be tested) while allowing them to accomplish the desired learning in the process of responding to that challenge. Inquiry Based Learning (IBL) is a form of inductive collaborative learning. According to Lee (2004), IBL enables the learner to formulate good questions, to identify and gather evidence and present them systematically, to analyse, interpret and formulate conclusions and evaluate the worthiness of those conclusions. IBL also involves the ability to identify problems, examine problems, generate possible solutions and select the best solution with appropriate justification. These are seen as critical skills required to be mastered by entry-level programmers. Prince and Felder (2006) state that "Inquiry learning is the simplest of the inductive approaches and might be the best one for inexperienced or previously-traditional instructors to begin with" (p.134).

Traditional ways of collecting, structuring and presenting topics/information do not meet the needs of the students in the ever-increasing/demanding dynamic environment. The results of a survey conducted to study the progress of learning by first-year programming students in a collaborative learning environment by Teague and Roe (2008) indicate that learning in a collaborative environment becomes a social process where students learn by working with others. They further observe that through collaborative learning, students are interactively engaged in the subject material, observing each other's approaches to problem solving, keeping each other focused on the task, and being encouraged to verbalise issues and decisions along the way. With this in mind, McKinney and Denton (2006) conducted an empirical study in the School of Computer Science and Information Sciences, University of Alabama where the students in an introductory programming course were exposed to collaborative learning environments such as team-based problem solving and pair programming. They observed that early use of collaborative learning benefits included deeper learning, developing skills wanted by industry, higher retention, higher achievement, higher course success rates, higher interest and a higher sense of belonging. These benefits were enjoyed by all students but were important for first-year students who were at risk of leaving the discipline.

Several studies conducted on the implementation of the GIL approach within other science- learning fields for example, chemistry also proved the effectiveness of GIL (Farrel, Moog, & Spencer, 1999; Gaddis & Schoffstall, 2007). Several common, and important outcomes observed in all these assessments of implementations were that more students successfully completed the courses, students' mastery of content was at least as high as in traditional instructional methods and students generally preferred the GIL approach to traditional methods. GIL also gives educators enough flexibility to adapt to the environment where it is implemented.

Research Methodology

The research paradigms symbolise "a worldview which describes, for its holder, the nature of the world, the individual's place in it, and the variety of potential relationships to that world and its parts" (Guba & Lincoln, 1994). This research resided under the paragmatic paradigm, thus taking a pragmatic philosophical stand. Pragmatic philosophy refers to the application of both qualitative and quantitative data, depending on the objectives of the study.

The study used a mixed methods approach. With the mixed methods approach to research, researchers incorporate methods of collecting or analysing data from the quantitative and qualitative research approaches in a single research study (Creswell, 2003; Johnson & Onwuegbuzie, 2004; Tashakkori & Teddlie, 2003). The correlation of both the qualitative and quantitative data assisted the researcher in understanding the application of GIL. This study, as noted, made use of both quantitative and qualitative methods within a case study.

Terre Blanche, Durrheim and Painter (2006) state that the research design is a plan of action that acts as bridge between research questions and the actual implementation of the research and which details the conditions for collection and analysis of data. The research design used for this study was Action Research (AR) within a case study. AR is an iterative research methodology that involves the subjects of the research as active participants - the research is done with, rather than on the participants (Carr & Kemmis, 1986).

The population for this study consisted of all 49 students enrolled for the introductory programming course (DEV1120). Twenty learners were selected having given their informed consent. Though it would have been ideal to have 24 students as part of the experimental group, only 20 students voluntarily agreed to participate in the research. The students consisted of males and females of ages ranging between 18 and 22 years. Of the 20 students, one dropped out of the course and the entire programme. This made the effective sample 19 for the experimental group. The remaining 29 students formed part of the control group.

The learner guide for the course, DEV1120 ("Learner Guide - Development Software 1 (DEV1120)," 2016), prescribed that students had to complete four mandatory formative assessments and a final summative assessment during the course of the year. Results of all these assessments formed part of the quantitative data for this study. Focus group interviews with semi-structured, open-ended questions were conducted to collect the qualitative data. This allowed the interviewer to prompt participants for further clarity based on the point of discussion and for respondents to elaborate on their responses. The interview questions were designed to gather qualitative data detailing the learner experience in their own words.

Consistency in producing similar results when repeated measurements are done on the same phenomenon is considered as the reliability factor of an instrument. Cronbach's alpha is a commonly used statistic to ensure reliability of an instrument (Taber, 2017). In any educational research, Cronbach's alpha coefficient value reaching 0.70 is considered as a sufficient measure of reliability (Taber, 2017). Cronbach's alpha coefficient value for this study was 0.883.

Data analysis included grouping and summarising information to establish meaning. Quantitative data were analysed using descriptive statistical methods. T-tests and chi-Square tests were conducted using the statistical analysis software SPSS to measure the significance of the findings. Qualitative data were analysed using thematic analysis.

Intervention process (PAR process)

The purpose of the introductory programming course, DEV1120, is to equip learners with a solid understanding of the basic principles of programming that apply to all computer programming languages. Learners are taught how to analyse problems and how to write algorithmic solutions using pseudocode and VB.net (Console Application) programming language. Students are introduced to the program development cycle and are taught how to use the tools associated with each stage.

A typical GIL class session started with the lecturer introducing new course material. In other words, the lecturer introduced the basic programming concepts, principles and theories pertaining to the topic in the course DEV1110. This was followed by the lecturer gaving an example by applying GIL principles: A Problem was given for which a solution was to be formulated. The procedure then followed an analysis of the problem by asking questions (inquiring) about the problem scenario. Answers to these questions were expected to make the process more understandable or might lead to other questions. Following up on all the new questions would eventually give answers to all the unknown information in the problem. The lecturer provided new problems to the groups. Students (group members collectively) were required to formulate the programming solution using the GIL approach. Students collaborated with one another. During this period, the lecturer walked around the class, listening to the discussions and providing guidance where required. The lecturer's role was to act as a facilitator who made sure the group members did not veer off the problem domain in their efforts to formulate the solution through inquiry (discussions).

Once the class had finished, students were given problems to solve in the form of homework. Though students were expected to find solutions alone while they were out of classroom environment, a virtual collaborative group environment was created through a social networking platform (WhatsApp in this study) to facilitate the GIL environment. Groups of four to five students were created over WhatsApp thus replicating the classroom groups. Students in groups were encouraged to engage on discussions through these virtual groups as they would do in the classroom environment. Since the lecturer was part of these groups, the lecturer could observe, encourage discussions and guide them where required.

Students who were part of the control group were taught using the traditional chalk-and-talk approach. A typical class session for this group started with the lecturer introducing new course material followed by the lecturer showing an example by formulating and writing the solution to a problem/question on the chalk-board. This was followed by the lecturer giving new problems to the students. Each individual student was expected to solve the problem by himself/herself. Once the class had finished, students were given homework to which they were expected to find solutions alone. These students did not collaborate with each other to learn computer programming. There was no opportunity provided for discussion between themselves, or between themselves and the lecturer, while they were inside and outside the classroom environment.

Results

The research question and associated null hypothesis were as follows:

Research Question: ***What are the differences in achievement between students taught using the GIL approach and those taught traditionally?***

The associated null hypothesis was:

H0: The GIL approach, as opposed to the traditional approach used for teaching the introductory programming course for entry-level students at HEIs, has no effect on their course achievements.

Quantitative and qualitative results of the analysis of the data gathered to address the above-mentioned research question and null hypothesis are given in this section.

Quantitative results

These are the findings obtained after an analysis of all the assessments that were given to the learners. The data included marks for the formative assessments - test one, two, three and four. These marks were computed to form the year mark (DP) for examination admission; then there was the summative assessment – examination marks. The final mark was computed as follows: Both year mark and examination mark had to have a sub-minimum of 40%. If a student failed to achieve these separate sub-minimum marks, then he/she failed the course. The final mark was computed by taking 40% of the year mark and 60% of the examination mark. At the end of the course, using this formula, the overall final marks of the students were also calculated. The quantitative data for the study was based on the marks that were captured from the experimental and control groups. (sample questions from all these assessments are given in Appendix 1).

Chi-Square test analysis and bar charts for assessments

Formative assessment 1 (Test 1) was conducted towards the end of first term (first quarter) of the academic year. The same assessment was used for both the experimental and control groups. The analysis of Test 1 results was as follows:

Table 1. Formative assessment 1 (Test 1) Chi-square on Pass rates on two groups

Chi-Square Tests					
	Value	df	Asymptotic Significance (2-sided)	Exact Sig. (2-sided)	Exact Sig. (1-sided)
Pearson Chi-Square	2.859[a]	1	.091		
Continuity Correction[b]	1.338	1	.247		
Likelihood Ratio	4.267	1	.039		
Fisher's Exact Test				.142	.122
Linear-by-Linear Association	2.799	1	.094		
N of Valid Cases	48				
a. 2 cells (50.0%) expected a count of less than 5. The minimum expected count was 1.58.					
b. Computed only for a 2 x 2 table					

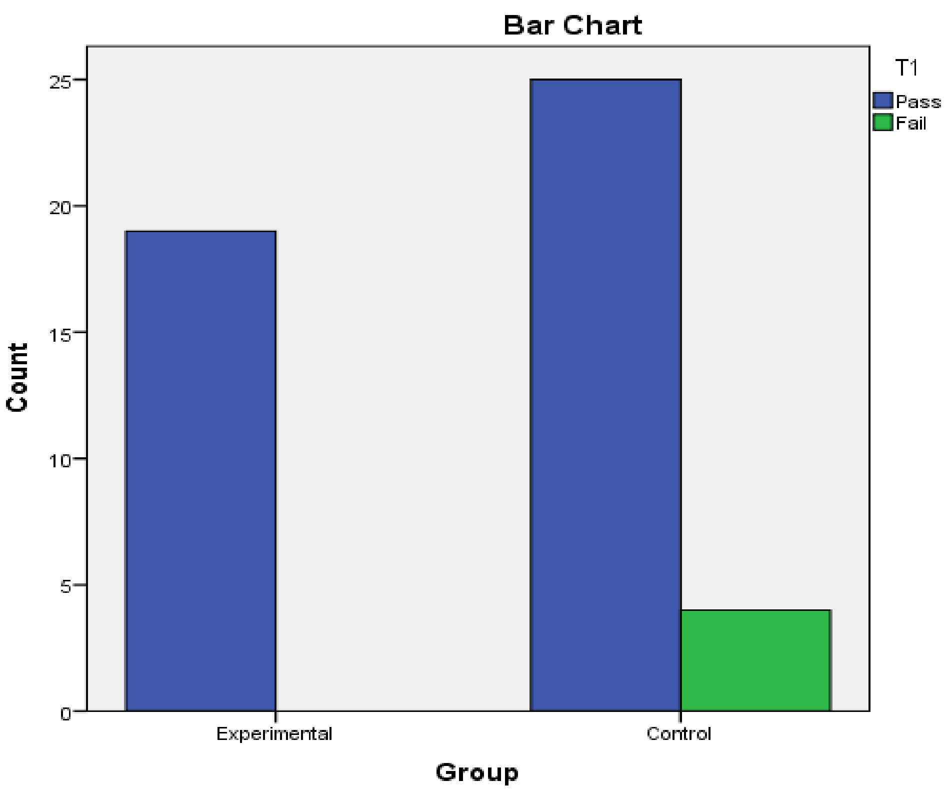


Figure 1. Formative assessment 1 (Test 1) performance for each group.

The value of the test statistic was2.859. Since the p-value ($p=0.091$) is greater than our chosen significance level ($\alpha = 0.05$), we do not reject the null hypothesis. Rather, we conclude that there is not enough evidence to suggest an association between group (teaching methods) and whether a student passes or fails. Based on the results, we can state the following:

- No association was found between groups (teaching methods) and whether or not students passed or failed (chi-square = 2.859, $p = 0.091$).

All students (100%) from experimental group, however, passed test-1 as compared to twenty-five (86.2%) from the control group.

Formative assessment 2 (Test 2) was conducted towards the end of the second term (second quarter) of the academic year. The same assessment was used for both the experimental and control groups. The analysis of Test 2 results was follows:

Table 2. Formative assessment 2 (Test 2) chi-square on pass rates on two groups

Chi-Square Tests					
	Value	df	Asymptotic Significance (2-sided)	Exact Sig. (2-sided)	Exact Sig. (1-sided)
Pearson Chi-Square	18.929[a]	1	.000		
Continuity Correction[b]	16.429	1	.000		
Likelihood Ratio	22.031	1	.000		
Fisher's Exact Test				.000	.000
Linear-by-Linear Association	18.534	1	.000		
N of Valid Cases	48				
a. 0 cells (0.0%) expected a count of less than 5. The minimum expected count was 8.31.					
b. Computed only for a 2 x 2 table					

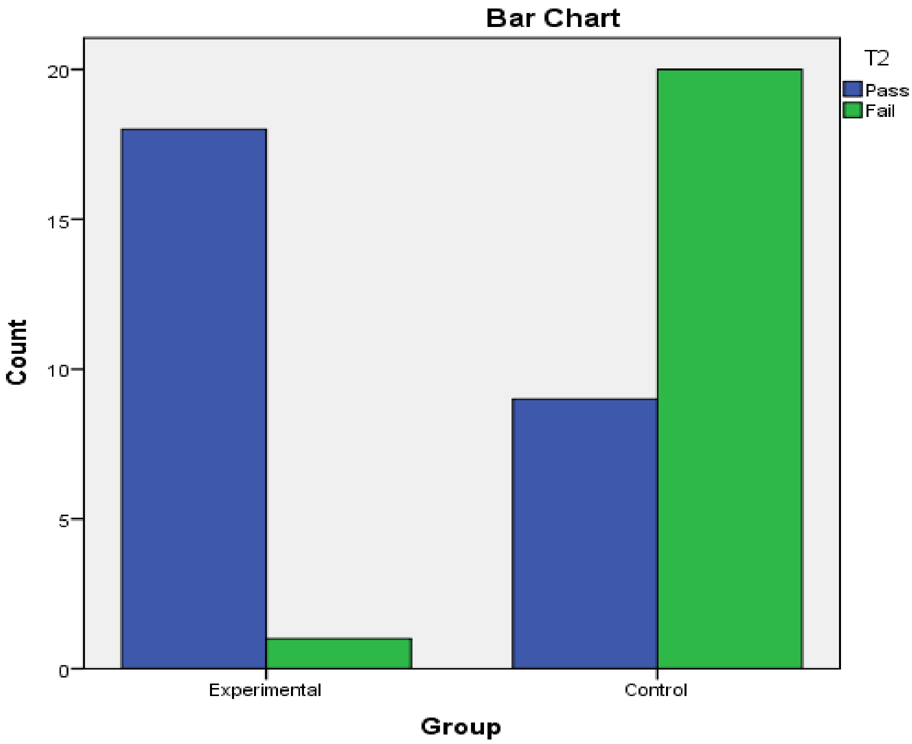


Figure 2. Formative assessment 2 (Test 2) performance for each group.

The value of the test statistic was 18.929. Since the p-value (p=0.000) is less than our chosen significance level α = 0.05, we can reject the null hypothesis, and conclude that there is an association between groups (teaching methods) and whether or not students pass or fail. Based on the results, we can state the following:

- There was a significant association between groups and whether or not students passed or failed (chi-square = 18.929, p =0 .000).

Eighteen students (94.7%) from the experimental group passed the tests as compared to only nine (31.0%) students from the control group. The very high failure rate in the control group contributed significantly to the overall drop of about 35% in the pass rate for test-2 as compared to test-1.

The formative assessment 3 (Test 3) was conducted towards the end of the third term (third quarter) of the academic year. The same assessment was used for both the experimental and control groups. The analysis of Test 3 results was as follows:

Table 3. Formative assessment 3 (Test 3) chi-square on pass rates on two groups

Chi-Square Tests					
	Value	Df	Asymptotic Significance (2-sided)	Exact Sig. (2-sided)	Exact Sig. (1-sided)
Pearson Chi-Square	3.499[a]	1	.061		
Continuity Correction[b]	2.409	1	.121		
Likelihood Ratio	3.714	1	.054		
Fisher's Exact Test				.110	.058
Linear-by-Linear Association	3.426	1	.064		
N of Valid Cases	48				
a. 0 cells (0.0%) expected a count of less than 5. The minimum expected count was 5.94.					
b. Computed only for a 2 x 2 table					

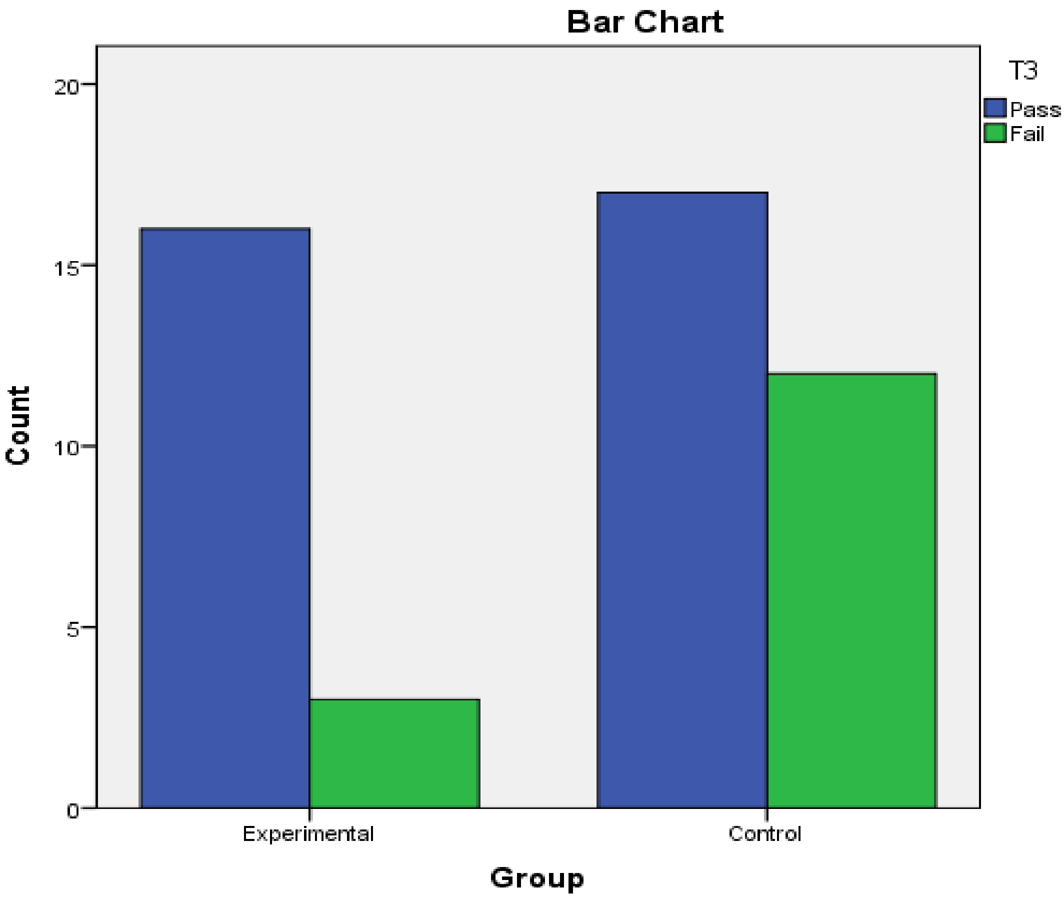


Figure 3. Formative assessment 3 (Test 3) performance for each groups.

The value of the test statistic is 3.499. Since the p-value (p=0.061) is greater than our chosen significance level (α = 0.05), we do not reject the null hypothesis. Rather, we conclude that there is not enough evidence to suggest an association between groups (teaching methods) and whether a student passed or failed. Based on the results, we can state the following:

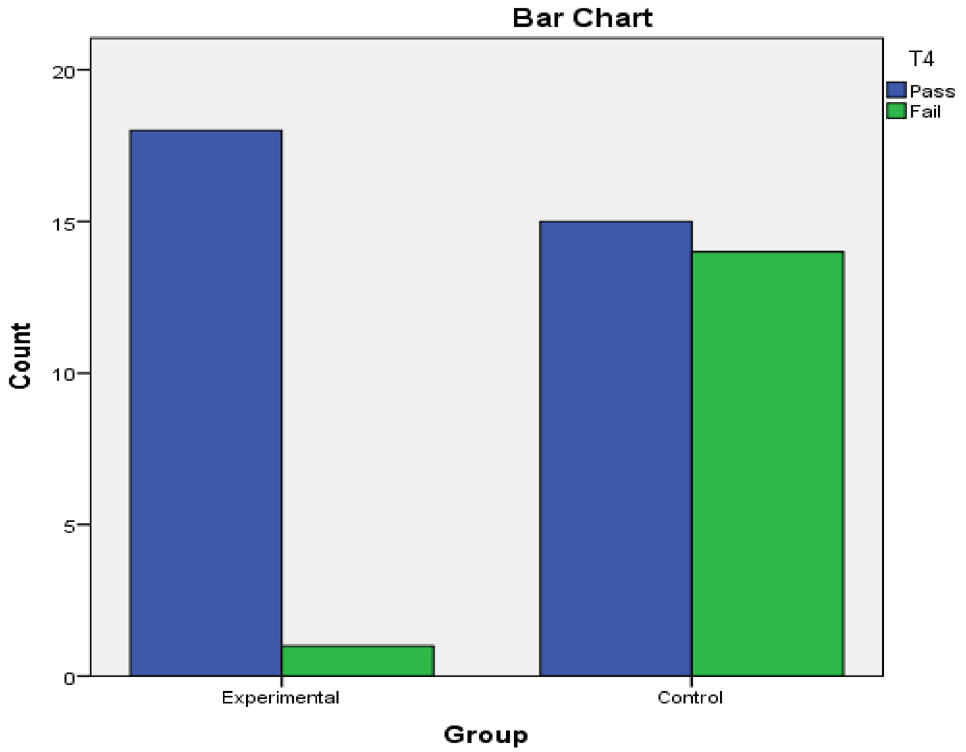
- No association was found between groups (teaching methods) and whether or not students passed or failed (chi-square = 3.499, p = 0.061).

Sixteen students (84.2%) from experimental group, however, passed test-3 as compared to only 17 (58.6%) from the control group.

The formative assessment 4 (Test 4) was conducted in the fourth term (fourth quarter) of the academic year. Same assessment was used for both the experimental and control. The analysis of Test 4 results was as follows:

Table 4. Formative assessment 4 (Test 4) chi-square on pass rates on two groups

Chi-Square Tests					
	Value	Df	Asymptotic Significance (2-sided)	Exact Sig. (2-sided)	Exact Sig. (1-sided)
Pearson Chi-Square	9.885[a]	1	.002		
Continuity Correction[b]	7.984	1	.005		
Likelihood Ratio	11.621	1	.001		
Fisher's Exact Test				.002	.001



Linear-by-Linear Association	9.679	1	.002		
N of Valid Cases	48				
a. 0 cells (0.0%) expected a count of less than 5. The minimum expected count was 5.94.					
b. Computed only for a 2 x 2 table					

Figure 4. Formative assessment 4 (Test 4) performance for each groups.

The value of the test statistic is 9.885. Since the p-value (p=0.002) is less than our chosen significance level α = 0.05, we can reject the null hypothesis, and conclude that there is an association between groups (teaching methods) and whether or not students passed or failed. Based on the results, we can state the following:

- There was a significant association between the groups and whether or not students passed or failed (chi-square = 9.885, p =0 .002).

Eighteen students (94.7%) from the experimental group passed the tests as compared to only 15 (51.7%) students from the control group.

The average marks scored in these four tests constituted the year mark (DP) for students.

Table 5. Admission to Examinations (DP) Chi-square on two groups

Chi-Square Tests					
	Value	df	Asymptotic Significance (2-sided)	Exact Sig. (2-sided)	Exact Sig. (1-sided)
Pearson Chi-Square	10.483[a]	1	.001		
Continuity Correction[b]	8.392	1	.004		
Likelihood Ratio	14.648	1	.000		
Fisher's Exact Test				.001	.001
Linear-by-Linear Association	10.264	1	.001		
N of Valid Cases	48				
a. 1 cells (25.0%) expected a count of less than 5. The minimum expected count was 4.75.					
b. Computed only for a 2 x 2 table					

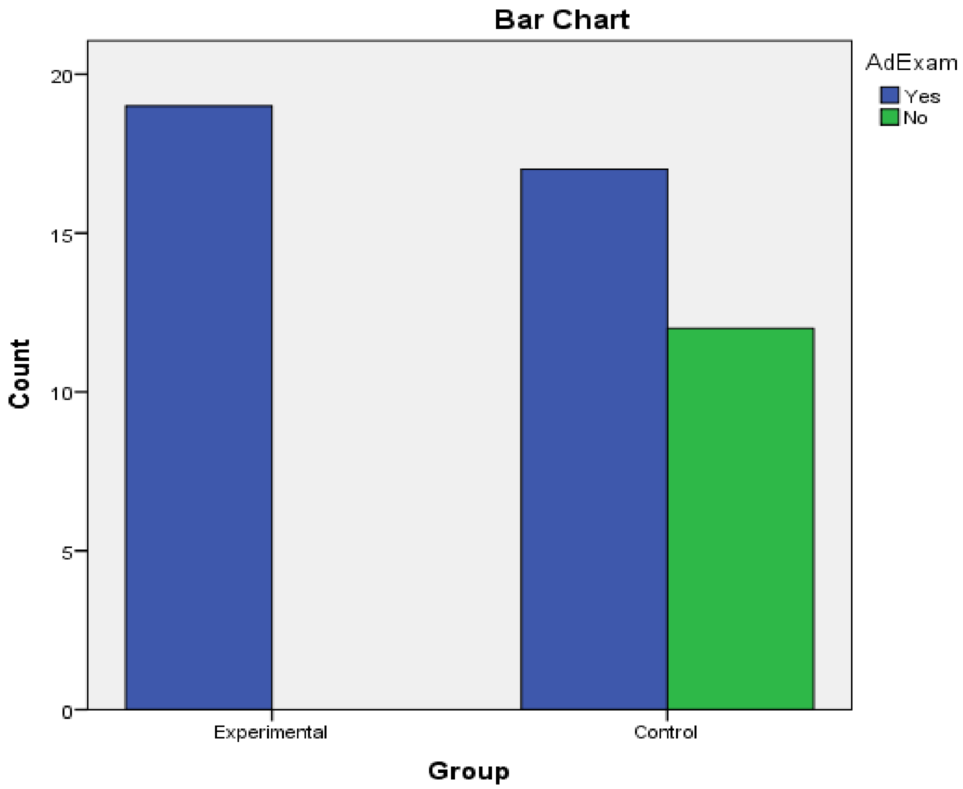


Figure 5. Admission to examination (DP) for each group.

The value of the test statistic is 10.483. Since the p-value (p=0.001) is less than our chosen significance level α = 0.05, we can reject the null hypothesis, and conclude that there is an association between group (teaching methods) and whether or not students pass or fail. Based on the results, we can state the following:

- There was a significant association between groups and whether or not students passed or failed (*chi-square* = 10.483, *p* =0 .001).

All students (100%) from the experimental group qualified for the final examination based on their performances throughout the academic year while only 17 (58.6%) of the students from the control group, which was taught using a traditional approach, managed to qualify for the final examination. From the control group, a staggering percentage (41.4%) could not even qualify for the final examination.

Analysis of final course results are given below:

Table 6. Final results chi-square on two groups

Chi-Square Tests					
	Value	df	Asymptotic Significance (2-sided)	Exact Sig. (2-sided)	Exact Sig. (1-sided)
Pearson Chi-Square	7.361[a]	1	.007		
Continuity Correction[b]	5.760	1	.016		
Likelihood Ratio	8.150	1	.004		
Fisher's Exact Test				.011	.007
Linear-by-Linear Association	7.208	1	.007		
N of Valid Cases	48				
a. 0 cells (0.0%) expected a count of less than 5. The minimum expected count was 6.33.					
b. Computed only for a 2 x 2 table					

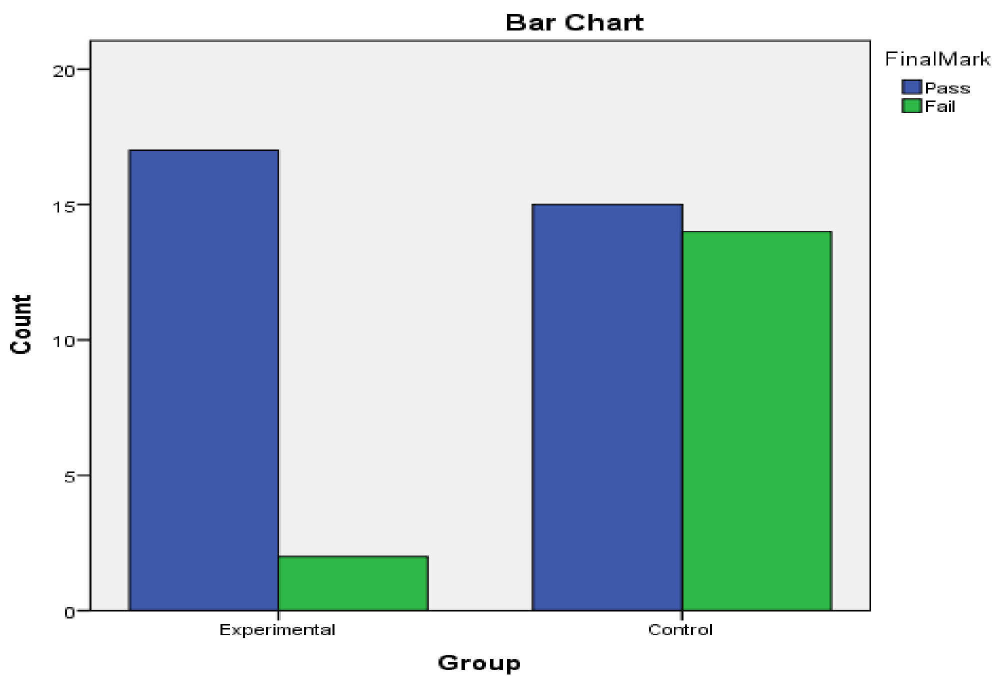


Figure 6. Final Results (Overall Pass/Fail) for each group.

The value of the test statistic is 7.361. Since the p-value (p=0.007) is less than our chosen significance level α = 0.05, we can reject the null hypothesis and conclude that there is an association between group (teaching methods) and whether or not students pass or fail. Based on the results, we can state the following:

- There was a significant association between groups and whether or not students passed or failed (chi-square = 7.361, p =0 .007).

Only two students (10.5%) failed to complete the course successfully from the experimental group as compared to fourteen (48.3%) students from the control group. Nearly all of the students (90%) learning through the GIL approach completed the course successfully as compared to almost 52% of students taught using the traditional teaching approach completed it.

Overall summary of tests and average marks

Table 7. Summary of tests and average mark- Group Statistics

Group Statistics					
	Group	N	Mean	Std. Deviation	Std. Error Mean
Test1	Experimental	19	65.39	6.680	1.533
	Control	29	63.97	15.729	2.921
Test2	Experimental	19	57.37	16.480	3.781
	Control	29	34.07	25.386	4.714
Test3	Experimental	19	56.21	16.301	3.740
	Control	29	46.34	29.472	5.473
Test4	Experimental	19	58.32	16.101	3.694
	Control	29	33.79	22.132	4.110
DP	Experimental	19	59.53	11.187	2.567
	Control	29	45.10	20.106	3.734

t-test analysis for assessments

Table 8. Calculated t-test results for all assessments – The Independent Samples Test

Independent Samples Test										
		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
Test1	Equal variances assumed	13.292	.001	.374	46	.710	1.429	3.826	-6.272	9.131
	Equal variances not assumed			.433	40.737	.667	1.429	3.298	-5.233	8.092
Test2	Equal variances assumed	2.117	.152	3.536	46	.001	23.299	6.590	10.034	36.565
	Equal variances not assumed			3.856	46.000	.000	23.299	6.043	11.136	35.463
Test3	Equal variances assumed	11.758	.001	1.329	46	.190	9.866	7.424	-5.078	24.810
	Equal variances not assumed			1.488	44.994	.144	9.866	6.629	-3.485	23.216
Test4	Equal variances assumed	4.185	.047	4.156	46	.000	24.523	5.900	12.647	36.399
	Equal variances not assumed			4.438	45.412	.000	24.523	5.526	13.396	35.649
DP	Equal variances assumed	10.642	.002	2.845	46	.007	14.423	5.070	4.218	24.628
	Equal variances not assumed			3.183	45.062	.003	14.423	4.531	5.298	23.548

The summary of the t-test results shows the significance of GIL (as shown in Table VII). This test verified that the assumption of equal variances holds in Test 2 and does not hold in the other samples. Significant differences in means between the groups were noticed in Test 2 (p=0.001), Test 4 (p=0.000) and DP (p=0.003).

On average, students' performance was not significantly different in test one and three. However, student performance was significantly different in test two and four. Even though the performances were not significantly different, the average Test one mark shows that students taught using GIL performed slightly better (65.30) than the control group (63.97). Similarly, the Test three average mark shows that the experiment group performed better on average (56.21) than the control group (46.34). Even though Test one and three's average mark is not significantly different for both groups, it is encouraging to note that the standard deviation for the experiment group was small (6.68) compared to that of the control group (15.73) for test one. Though slightly greater, the standard deviation of the experimental group (16.30) in test three was close to half that of the control group (29.47). This implies that, while both groups performed well, especially in test one, GIL closely and collectively improved the performance of all the students when compared to those exposed to the traditional teaching approach. Table VIII shows the details of the marks.

In addition, findings from test two and four show that the experimental group performed significantly better than the control group; for example, the average Test two marks for the experimental group was 57.4 compared to that of the control group which was 34.07. Similarly, Test four showed a significantly high average mark for the experimental group (58.32) compared to that of the control group (33.79). For Test 1 and 3, there was no significant difference as per the p-values, though the average for these two tests was still higher for the experimental group. It should however, be noted that the standard deviation was always lower for the experimental group than the control group (ref. Table 4.3). The lower standard deviation implies that the experimental group was generally performing at the same level. As was the case with Test one and three, the experimental group's standard deviation for Test two was smaller (16.48) when compared to that of the control group (25.39). A similar type of result was found on Test four as the experimental group had a smaller standard deviation of 16.10 compared to that of the control group that was 22.13. There were no common outliers and the mark differences for the assessments among the experimental groups were almost in the same range. The researchers maintain their argument that GIL collectively improves the performance of students with minimum outliers as students help each other understand the subject during the learning process. This argument is supported by the number of students who failed in class tests; for example, none (0%) in the experimental group failed test one as compared to four (13.8%) from the control group. Only one student (5.3%) from the experimental group failed test two compared to twenty (69.0%) from the control group. Twelve students (41.4%) from the control group failed test three compared to three (15.8%) from the experimental group. Lastly, nearly half of the students (48.3%) failed test four from the control group compared to one (5.3%) from the experimental group.

Performances in class tests were extended to their Duly Performed (DP) marks. Results show that all 19 (100%) students from the experimental group achieved at least 40% for the DP that gave them admissions to the final examination. However, just above 50% of the control group got at least a 40% final examination admission. The rest of the students failed to qualify for the examination admission.

In terms of performance, the overall pass rates for the GIL and traditional learning approach students were 89% and 52%, respectively, for the university introductory programming course. Watson and Li (2014) report that the global pass rate for programming courses is 67.7% which is significantly lower than the one achieved for the experimental group of this study. Furthermore, the same authors report South Africa's overall failure rate as being approximately 45% for the same course; and the finding for the control group in this study (48%) was close to their findings and hence confirms the results of Watson and Li (2014). The average year mark obtained by the students in the experimental group and control group were 59.53% and 45.10%, respectively. This supported the findings by another researcher from the same institution who experimented with a similar approach in an Engineering programme (Louw, 2012). The author reported that the class average improved to 55% when students were taught using the guided inquiry based learning approach as compared to the 46% when students were taught using traditional methods.

In conclusion, the GIL approach, as opposed to traditional approaches, used for teaching entry-level students enrolled for computer programming had a positive effect of their course achievements, thus the null-hypothesis (H0) is rejected.

Qualitative Results

Various themes emerged from the qualitative data gathered through the focus group interviews. Focus group interviews were conducted among the students who were part of the experimental group. Of the 19 students, one student did not participate in the focus group interview. Four interview sessions were conducted among the remaining 18 students. They were divided into groups of four (two groups) and fives (two groups) for the focus group interviews. Each focus group interview lasted for about 50 to 55 minutes.

The themes that emerged were:

Theme 1. Group Sessions helped understand the subject content better

Difficulty in understanding the subject content and applying them are viewed as one of the major challenges faced by entry-level programming students by many researchers (Bergin & Reilly, 2005; Derus & Ali, 2012; Schoeman, 2015; Shuhaidan et al., 2009). There was consensus among students that the peer assistance, through group sessions, helped them understand the subject content better. Responses from students as given below in verbatim prove this:

- FG1 P1: May be we won't understand the way the educator explains the things whereas if it is from your peer make more sense. I think it is a really good idea. It works for me, my opinion

- FG1 P4: As my fellow classmates has said - it has helped me a lot because we communicate with group members and understand more complex things than may be the teacher just explains…
- FG3 P3: Interactions within the class mates..its really helped because sometimes that person understand the question more, like an advocate that member can explain to others like when we are done with the lecture. Then like it will help us understand more.

Theme 2. Group session experience assisted students while doing their homework alone.

Entry-level students are required to develop a diverse range of skills such as problem analysis and problem solving at an early stage of their studies (Falkner & Munro, 2009), but studies have proven that some students fail to know where to begin in attempting to get to the solution (Huggard & Goldrick, 2009). Students used their group session experiences and approaches when they were doing their homework. The questions they asked and were being asked during the group sessions worked as a 'guide' on how to analyse the problem. Though there was no fixed rule, through their responses it was evident that they learned to approach a problem scenario systematically questioning themselves to understand and analyse the problem to reach alternate solutions. The following students' responses ascertain this conclusion:

- FG1 P1: ..during the group discussions, we will first analyse the problem. What is it that we want to solve and how can we solve it? Then, I list the alternative way on how to solve it and then I choose the most appropriate one. When I am alone – like…firstly, I have to identify the kind of a problem that I want to solve. Look for the solutions then choose the best one…then okay, vary them then choose the best one. It helped me to fully analyse what is it that I want to solve and how can I solve it
- FG1 P5: If you have a problem sir….when you are alone…. if you ask the right questions to yourself like we did in the groups and follow the process on how to come up with the solutions mostly by answering those questions asked...then you will solve the problem, sir. Yes…it was totally different from how I used to solve any problem before…
- FG2 P2: …before I approach any question, I am trying to use the skills during discussions. Ask myself many questions…to analyse the problems lists all steps in how to solve the problem and find most effective ways to solve the problem.
- FG3 P1: Now you know which questions to ask yourselves to analyse.

From the above listed responses it is clear that GIL sessions give students the belief that there is a solution to every problem and they can engage in conversations through asking critical questions which can eventually lead them to a solution.

Theme 3. Students a ppreciated alternative problem solving approaches.

The study of algorithmic problem solving is at the core of learning computer programming and it is well documented that entry-level students struggle to mastering problem solving skills (Lishinski, Yadav, Enbody, & Good, 2016; McCracken et al., 2001; Sheth, Murphy, Ross, & Shasha, 2016). Students stated that they learned to acknowledge and appreciate the fact that there were different ways to solve a problem as different individuals could approach a given problem scenario differently. Groups sessions taught them that a given problem scenario could be solved in different ways as their peers in the group came up with their own ways of solving the problem. The following responses from students provided valuable insight into this:

- FG1 P2: (GIL) has a positive impact because programming is not always done in a single way. Like for example, I just see how the class members found the problem in different ways, for example you got a question and you argue in the first way, but he gets it in the other way, but you still get the same answer. So you get to see how there are different ways.
- FG2 P5: When we put in our possible solution to a problem, group will come up with the best possible answers to that problem.
- FG3 P2: We've all come up with different solution on how to solve the problem then we look the most appropriate or the most correct way or the one that will be easier for us to understand…., the most appropriate one that will help us to be able to solve even related problems to that one.

Students tend to follow the solutions that are given by lecturers and fail to look for different ways of solving a problem in a typical traditional teaching environment. The active learning environment inspired them to come up with their own ways of solving the problem. This in turn improved their problem solving skills. Students constructed their own solutions rather than acting as receivers of ideas from the lecturer

Theme 4. The social networking platform facilitates GIL sessions (out of the classroom)

Features made available through the virtual groups that are found in social networking platforms such as WhatsApp have the potential to make learners active in the learning process, boost informal communication between learners and teachers and create an anxiety-reducing environment (Awada, 2016; Bouhnik & Deshan, 2014). As part of the study, a social networking platform (WhatsApp in this study), was used to facilitate a GIL environment among the students while they were out of the classroom environment. Groups of 4 to 5 students were created over WhatsApp replicating the classroom groups and they were encouraged to engage in discussions through these groups. Students found this effective and used it extensively to their benefit while away from the classroom, though there was no special incentive given to students such as extra marks for participating in online discussions and collaboratively working on the programming tasks. Responses from the interviews as given below suggest that the experiment to create a GIL environment over virtual space has attracted the new, digital- generation students.

- FG1 P1: …there is no better way to educate a young mind in this generation now, than using social network. Cause, if you put something educative on social network, it will spread, it will be viral. Everyone will learn about it. And, cause, now life now is about technology and everything. So, if it is out there, spreading, more chances are that young adults, as us, will learn about it – enriching our minds and learning everything. So it helps with that for me..
- FG2 P2: There was this WhatsApp group was created. When we do the home work, we type it and post. It helped. My group mates and I discuss…and lecturer did tell us if we were doing completely opposite through WhatsApp and post questions. In a way it was a classroom there in WhatsApp..it helped.
- FG4 P1: We had some WhatsApp groups. We do communicate with others like when we have those lazy moments of hours at home, hei I am not sure about this question like, then we go to WhatsApp and discuss like don't you know this and that. In that way my group members help me on continuing or making sure that my answer is in the right track.

Conclusion

Statistical data analysis shows that there is a significant difference between the students that were engaged using GIL and those taught using the traditional approach, in that the performance of the experimental group was far better than those in the control group. The final achievement result for the experimental group was almost 90% compared to the control group's pass rate of less than 52%. Achievement results obtained by the experimental group was much higher than both the global and South African average achievement rates of 67.7% and 55%, respectively, as reported by Watson and Li (2014) for an Introductory Programming course. It was evident that the overall performance also improved as indicated by the final pass rate of almost 67% for both groups (combined), which is higher than the average pass rate of other previous years (46.88%). This should be attributed to the high pass rates achieved by the experimental group which raised the average percentage. These findings are corroborated through the following themes which emerged from the focus groups interviews:

- Group sessions helped understand the subject content better
- Group session experience assisted students while doing their homework alone
- Students appreciated alternative problem solving approaches
- The social networking platform facilitates GIL sessions (out of the classroom)

As evident through the themes listed above, students overwhelmingly agreed that the GIL approach to teaching the Introductory Programming course assisted them in understanding the underlying theories and principles of the course and improved their problem solving skills.

References

Affleck, G., & Smith, T. (1999). Identifying a need for web-based course support. http://www.ascilite.org.au/conferences/brisbane99/papers/afflecksmith.pdf

Amineh, R. J., & Asl, H. D. (2015). Review of constructivism and social constructivism. Journal of Social Sciences, Literature and Languages, 1(1), 9-16

Awada, G. (2016). Effect of WhatsApp on critique writing proficiency and perceptions toward learning. *Cogent Education, 3*(1), 1264173. doi:10.1080/2331186X.2016.1264173

Bandura, A. (1986). Social foundations of thought and action: A social cognitive theory. Englewood Cliffs, NJ: Prentice Hall.

Bennedsen, J., & Caspersen, M. E. (2006). Abstraction ability as an indicator of success for learning object-oriented programming? *SIGCSE Bull., 38*(2), 39-43. doi: 10.1145/1138403.1138430

Bennedsen, J., & Caspersen, M. E. (2007). Failure rates in introductory programming. SIGCSE Bull., 39(2), 32-36. doi: 10.1145/1272848.1272879

Bergin, S., & Reilly, R. (2005). The influence of motivation and comfort level on learning to program. *Proceedings of the 17th Workshop of the Psychology of Programming Interest Group, PPIG 05. University of Sussex, Brighton, UK.*

Bouhnik, D., & Deshen, M. (2014). WhatsApp goes to school: Mobile instant messaging between teachers and students. *Journal of Information Technology Education: Research, 13*(1), 217-231.

Carr, W., & Kemmis, S. (1986). Becoming critical. Education, Knowledge and action research.

Costa, E. B., Fonseca, B., Santana, M. A., de Araújo, F. F., & Rego, J. (2017). Evaluating the effectiveness of educational data mining techniques for early prediction of students' academic failure in introductory programming courses. *Computers in Human Behavior, 73*, 247-256.

Creswell, J. W. (2003). Research design: qualitative, quantitative, and mixed method approaches. 2nd edition, Thousand Oaks, CA: SAGE.

Derus, S., & Ali, A. Z. M. (2012). *Difficulties in learning programming: Views of students.* Paper presented at the 1st International Conference on Current Issues in Education (ICCIE 2012)

Devi, B., Das, M., Das, M., & Khandelwal, B. (2017). Application of Bandura's social cognitive theory in the technology enhanced, blended learning environment. *International Journal of Applied Research, 3*(1), 721-724.

Esteves, M., Fonseca, B., Morgado, L., & Martins, P. (2011). Improving teaching and learning of computer programming through the use of the Second Life virtual world. *British Journal of Educational Technology, 42*(4), 624-637. doi: 10.1111/j.1467-8535.2010.01056.x

Falkner, K., & Munro, D. S. (2009). Easing the Transition: a collaborative learning approach. Proceedings of the eleventh Australian Computing Education Conference (ACE 2009).

Farrel, J. J., Moog, R. S., & Spencer, J. N. (1999). A guided-enquiry general chemistry course. Journal of chemical education 76.4(1999).

Gaddis, B. A., & Schoffstall, A. M. (2007). Incorporation of Guided-Inquiry Learning into the organic chemistry laboratory Journal of chemical education, Vol. 84 No. 5.

Gomes, A., & Mendes, A. J. (2007). Learning to program-difficulties and solutions. Paper presented at the International Conference on Engineering Education–ICEE.

Gonzalez, G. (2006). A systematic approach to active and cooperative learning in CS1 and its effects on CS2. *SIGCSE Bull., 38*(1), 133-137. doi: 10.1145/1124706.1121386

Guba, E., & Lincoln, Y. (1994). Competing paradigms in qualitative research. In N. Denzin & Y. Lincoln (Eds.), Handbook of qualitative research(pp. 105–117). Thou-sand Oaks, CA: Sage.

Harkness, S. (2009). Social constructivism and the Believing Game: a mathematics teacher's practice and its implications. Educational Studies in Mathematics, 70(3), 243-258. doi: 10.1007/s10649-008-9151-3

Hmelo-Silver, C. E., Duncan, R. G., & Chinn, C. A. (2007). Scaffolding and achievement in problem-based and inquiry learning: a response to Kirschner, Sweller, and. Educational psychologist, 42(2), 99-107.

Horton, D., & Craig, M. (2015). *Drop, Fail, Pass, Continue: Persistence in CS1 and Beyond in Traditional and Inverted Delivery*. Paper presented at the Proceedings of the 46th ACM Technical Symposium on Computer Science Education, Kansas City, Missouri, USA.

Huggard, M., & Goldrick, C. M. (2009). Lessons learnt from a decade of structured support for novice programmers. *International Conference on Engineering Education (ICEE), Belfast, Ireland*.

Jenkins, T. (2002). On the difficulty of learning to program *3rd annual LTSN-ICS conference, Loughborough University*.

Johnson, R. B., & Onwuegbuzie, A. J. (2004). Mixed methods research: A research paradigm whose time has come. Educational Researcher, 33, 14–26. doi:10.3102/0013189X033007014

Iqbal Malik, S., & Coldwell-Neilson, J. (2017). Impact of a new teaching and learning approach in an introductory programming course. *Journal of Educational Computing Research, 55*(6), 789-819.

Koulouri, T., Lauria, S., & Macredie, R. D. (2015). Teaching introductory programming: a quantitative evaluation of different approaches. *ACM Transactions on Computing Education (TOCE), 14* (4), 26.

Learner Guide - Development Software 1 (DEV1120). (2016). National Diploma: Information technology, Faculty of Science, Engineering and technology, Walter Sisulu University.

Lee, V. S. (2004). *Teaching and learning through inquiry: A guidebook for institutions and instructors*: Stylus Pub Llc.

Lishinski, A., Yadav, A., Enbody, R., & Good, J. (2016). *The Influence of Problem Solving Abilities on Students' Performance on Different Assessment Tasks in CS1*. Paper presented at the Proceedings of the 47th ACM Technical Symposium on Computing Science Education, Memphis, Tennessee, USA.

Lister, R., Adams, E. S., Fitzgerald, S., Fone, W., Hamer, J., Lindholm, M., . . . Thomas, L. (2004). A Multi-National Study of Reading and Tracing Skills in Novice Programmers SIGSCE Bulletin, 36(4):119-150.

Louw, C. (2012). *The Effect of a Guided Enquiry Based Learning Approach on Mechanical Engineering Students' Understanding of Thermodynamics.* Walter Sisulu University.

Malliarakis, C., Satratzemi, M., & Xinogalos, S. (2017). CMX: The Effects of an Educational MMORPG on Learning and Teaching Computer Programming. *IEEE Transactions on Learning Technologies, 10*(2), 219-235. doi:10.1109/TLT.2016.2556666.

Malik, S. I., & Coldwell-Neilson, J. (2017). Comparison of Traditional and ADRI Based Teaching Approaches in an Introductory Programming Course. *Journal of Information Technology Education: Research, 16*, 267-283.

McKinney, D., & Denton, L. F. (2006). Developing Collaborative Skills Early in the CS Curriculum in a Laboratory Environment SIGCSE 2006, Technical symposium on computer science education, Houston, Texas, USA.

McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagen, D., Kolikant, Y., . . . Wilusz, T. (2001). A multi-Institutional study of assessment of programming skills of first year CS students. SIGCSE Bulletin, 33(4).

Minelli, R., Mocci, A., & Lanza, M. (2015). *I know what you did last summer: an investigation of how developers spend their time*. Paper presented at the Proceedings of the 2015 IEEE 23rd International Conference on Program Comprehension, Florence, Italy.

Or-Bach, R., & Lavy, I. (2004). Cognitive activities of abstraction in object orientation: an empirical study. *SIGCSE Bull., 36*(2), 82-86. doi: 10.1145/1024338.1024378

Perkins, D. N., & Martin, F. (1986). *Fragile knowledge and neglected strategies in novice programmers*. Paper presented at the Papers presented at the first workshop on empirical studies of programmers on Empirical studies of programmers, Washington, D.C., USA.

Petersen, A., Craig, M., Campbell, J., & Tafliovich, A. (2016). *Revisiting why students drop CS1*. Paper presented at the Proceedings of the 16th Koli Calling International Conference on Computing Education Research, Koli, Finland.

Prince, M. J., & Felder, R. M. (2006). Inductive teaching and learning methods: Definitions, comparisons, and research bases. *Journal of engineering education, 95*(2), 123-138.

Prince, M., & Felder, R. M. (2007). The many faces of inductive teaching and learning Journal of college science teaching, 36(5), 36:14-20.

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. Computer Science Education, 13(2), 137-172. doi:10.1076/csed.13.2.137.14200

Schoeman, M. A. (2015). Enhancing comprehension in open distance learning computer programming education with visualization.

Sheard, J., & Hagan, D. (1998). Our failing students: study of a repeat group. 3rd annual conference on Integrating technology into Computer Science education: Changing the delivery of computer science education ItiCSE 1998.

Sheth, S., Murphy, C., Ross, K. A., & Shasha, D. (2016). *A Course on Programming and Problem Solving*. Paper presented at the Proceedings of the 47th ACM Technical Symposium on Computing Science Education, Memphis, Tennessee, USA.

Shuhaidan, S., Hamilton, M., & D'Souza, D. (2009). Taxonomic study of novice programming summative assessment. Proceedings of the eleventh Australian Computing Education Conference (ACE 2009).

Taber, K. S. (2017). The Use of Cronbach's Alpha When Developing and Reporting Research Instruments in Science Education. *Research in Science Education*. doi:10.1007/s11165-016-9602-2

Tashakkori, A., & Teddlie, C. (Eds.). (2003). Handbook of mixed methods in social and behavioral research (1st ed.). Thousand Oaks, CA: Sage.

Teague, D., & Roe, P. (2008). Collaborative learning – towards a solution for novice programmers. Tenth Australian Computing Education Conference (ACE2008), Wellongong, Australia.

Terre Blanche, M., Durrheim, K., & Painter, D. (2006). Research in Practice: Applied Methods for the Social Sciences. University of Cape Town press (Pty) Ltd.

Vygotsky, L. S. (1978). Mind in Society. Cambridge: Harvard University Press.

Vygotky, L. 1994. The socialist alter ation of man. In Van der Veer, R. & Valsiner, J. 1994. (eds.). *The Vygotsky Reader*. Oxford: Basil Blackwell Ltd. 175 -184

Watson, C., & Li, F. W. (2014). Failure rates in introductory programming revisited. Paper presented at the Proceedings of the 2014 conference on Innovation & technology in computer science education.

Winslow, L. E. (1996). Programming pedagogy — a psychological overview. ACM Sigcse Bulletin, 28(3), 17-22.

Yang, T.-C., Hwang, G.-J., Yang, S. J., & Hwang, G.-H. (2015). A two-tier test-based approach to improving students' computer-programming skills in a web-based learning environment. Journal of Educational Technology & Society, 18(1), 198.

Appendix

Sample questions from formative and summative assessments.

(Extract from) Formative assessment 1 (Test 1)

Question 1

1.1 With the aid of a diagram, show the various steps of the Program Development Cycle.

1.2 Explain the term Algorithm so that its importance in programming is made clear.

Question 2

The ATM (Automatic Teller Machine) is used to withdraw money from your account. Consider that you need to withdraw some money from your account. Answer the following questions from your knowledge of using an ATM.

Note: Assume that there is enough money in your account and the machine is in working order.

1.1 List two inputs into the machine for the job

1.2 List two outputs from the machine.

1.3 Write the various steps needed to perform the operation in the correct order.

(Extract from) Formative assessment 2 (Test 2)

Question 1

Barking Lot is a day care centre for dogs. The centre charges R75.00 per day for dogs below 10 kg, R90.00 per day for dogs 10 kg and above but below 30 kg and R120.00 per day for dogs 30kg and above. The program accepts the name of the dog's owner, breed and weight (in kg) of the dog and the number of days the dog needs to be taken care of. The customer must pay 14% VAT on the total amount. The program must calculate and display the total amount before VAT, the VAT amount and the final amount to pay including VAT.

You are required to do the following:

1.1 Draw the IPO chart of the program

1.2 Write the Pseudocode for the program.

Question 2

Vusi is planning to organise a graduation party at his home. He wants to calculate the cost for his party. The following list shows the cost of hiring the equipment and the cost of food to be provided.

Item	***Cost***
Tent	R 1550.00 to accommodate up to 50 guests. If there are more guests, R1000 for an additional tent.
PA System	R 600
Chairs	R 20 for each chair
Catering	R 150 per person if number of guests is 20 or less. If the number of guests is more than 20, the charge is R 130 per person.

In addition to the above, he needs to pay for the services of a DJ at R 200 per hour. The program will accept the total number of guests he plans to invite and duration of the party in hours. The program must then calculate and display the following with meaningful messages:

- The total cost of catering
- The total cost of hiring the tent and chairs.
- The total cost of hiring the PA system together with services for the DJ
- The final amount he need to pay

2.1 Write the complete program in pseudocode.

<u>(Extract from) Formative assessment 3 (Test 3)</u>

Question 1

Consider the following pseudocode written by a programmer:

```
BEGIN

    Declare P, Q As String

    Declare X,Y,Z As Integer

    Accept Y, Z

    X = Y * 2^ (Z-3) + 15\6 – (18+Y)MOD 7 – 4

    If X>20 Then

        P = “Big Value”

    Else

        P = “Small Value”
```

```
        End If

        If NOT X MOD 4 < Y/(Z – 3) Then
            Q = P
        Else
            Q = "No Value"
        End If
        Display "P = ", P
        Display "Q =", Q
    END
```

1.1 Draw a Trace Table using the input values 5 and 6.

Question 2

Vuyo's couriers is a courier company based in East London. They charge customers to courier their parcels based on the weight of the parcel and the distance between East London and the destination. They charge R30 per kilogram for the weight of the parcel. In addition, for the first 50 kms they charge R80. For up to 100kms they charge R120. For more than 100km they charge R200. They do not deliver parcels if the distance is more than 200kms. For parcels weighing more than 2kg they charge an additional R100. The customer must pay 14% VAT on the total amount. Your program must display the total amount without VAT, the VAT amount and the final amount.

3.1 Draw the IPO chart of the program

3.2 Draw the flow chart for the program

Question 3

South African property owners must pay municipal rates according to the market value of their property. XYZ municipality charges different rates for residential property, business property and education institution. The calculation of property tax is as follows:

Residential Property: The first R 50000 of the market value of the property is exempt from tax. Tax is payable for the balance at a rate of 0.91% for one year.

Business Property: Tax for one year is calculated at 1.2% of the market value of the property.

Educational Intuition: Tax for one year is calculated at 0.25% of the market value of the property.

Divide the yearly tax by 12 to get the tax payable monthly.

Write a program using pseudocode to prompt for meaningful messages. The program must calculate and display (along with meaningful messages) the monthly tax payable by each owner. The program must repeat this for 100 properties. Finally, the program must display the following.

- Total tax for all residential properties for one year.
- Total tax for all business properties for one year.
- Total tax for all educational institutions for one year.

Specific instruction: The user enters R for residential property, B for business property and E for educational institution.

(Extract from) Formative assessment 4 (Test 4)

Question 1

ABC is a company that has two types of employees (Draftsman and Supervisor) and their salaries are calculated as follows: A Supervisor earns a basic salary of R3000 a month. In addition to the basic salary, the employee receives R150 for each year of service for up to 5 years of service. If the years of service are above 5, then the amount is R200 for each year above 5 years of service. A draftsman earns a basic salary of R2800 a month. In addition to the basic salary, the employee receives R140 for each year of service up to 3 years of service; for the service above 3 years, the amount of R190 for each year of service above 3 years is paid.

Your program should receive appropriate data. Thereafter calculate and display (with meaningful messages) the monthly salary of an employee.

1.1 Draw the flow chart for the program

1.2 Write the pseudocode for this program

Question 2

The South African Post Office charges for international parcel a basic charge and an additional charge depending on the weight of the parcel. The following table gives the charge for sending the parcel depending on the zone of the address.

Zone	Basic Rate	Additional Charge

		per 100 gram or part thereof
Zone A	R 111.10	R 3.40
Zone B	R 163.80	R 4.25
Zone C	R 163.80	R 15.30
Zone D	R 169.50	R 14.00
Zone E	R 126.10	R 21.80
Zone F	R 210.40	R 19.50

2.1 Write a Visual Basic Console Application to accept input, calculate and display (with meaningful messages) the amount charged to send the parcel to the specified location/Zone. The program terminates only when the user enters “ZZZ” for the name of the customer