Learning Computer Programming Using Project-Based Collaborative Learning: Students' Experiences, Challenges and Outcomes

Wilson Osafo Apeanti[1], Daniel Danso Essel[1,*]

Abstract

The major concern of teaching computer programming in higher education is to provide students with the necessary skills to integrate theory and practice. One of the methods most suited for this task is project-based collaborative learning (PBCL). This study provides an in-depth analysis of students' experiences, levels of collaboration and challenges in learning computer programming in a PBCL setting. A sample of 428 students was selected from a population of 840 undergraduate computer programming students at all levels using a stratified random sampling technique. It was found that focusing programming courses on authentic problems made the course more interesting for students. The students gained new interpersonal skills and understood the technical concepts of the courses better. PBCL was found to be suitable for teaching lower level (level 100 and 200) undergraduate programming courses compared with higher level (level 300 and 400) courses. Students' challenges in PBCL computer programming courses include time allocation for projects, choosing appropriate problems and piggy riding in project groups. We expect the findings of this study to influence policy on the teaching of computer programming courses at the undergraduate level.

Keywords: Learning Computer Programming, Project-based learning, Project Groups, Project-based collaborative learning

1. Introduction

The dictates of the information society demand students who are equipped with critical thinking and problem-solving skills. To meet this challenge, higher education institutions are adopting teaching and learning methods that allow students to apply academic knowledge in a real-world context. The study of computer programming is not an exception. Computer programming is a skill-based practical-oriented course that requires a lot of real-world examples for students to come to terms with its application. Over the years, several learning methodologies have been experimented with to try and create a learning environment that produces the best learning experiences for students studying programming courses. Some of these methodologies include inquiry-based learning, problem-based learning and project-based collaborative learning (PBCL).

[1] Department of ICT Education, University of Education, Winneba, Ghana
[*] Corresponding author: ddessel@uew.edu.gh

Project-Based Collaborative Learning (PBCL) is a model that organizes learning around the students' collaborative performance of various projects (Jamal, Essawi & Tilchin, 2014; Thomas, 2000). In PBCL, projects are not simple assignments but are complex tasks based on challenging problems that involve students in design, problem-solving, decision making and investigative activities (Thomas, Mergendoller, & Michaelson, 1999). Projects help students to work quite autonomously overextended a period to produce authentic products or presentations. In this way, projects provide practical applications for materials taught initially by the traditional teaching method. PBCL projects do not end up at a predetermined outcome or take predetermined paths but incorporate a good deal of more student autonomy, choice and unsupervised work time (Thomas, 2000). Students can go beyond textbook content to explore, collaborate and apply new knowledge in projects. As a result, students get the feeling of authenticity of the content that they study in the classroom through interdisciplinary studies, real-world applications of academic content and community service (Thomas, 2000). PBCL approach promotes knowledge sharing among students, effective construction of higher-order thinking skills, accountability skills, and collaborative skills (Jamal, Essawi &Tilchin, 2014). Students practice and gain improved soft skills such as leadership skills, social communication skills, and conflict resolution skills which are difficult to gain through the traditional methods of teaching alone.

The adoption of PBCL is growing in the context of computer science courses and chief among them is computer programming (Kizaki, Tahara, & Ohsuga, 2014; Joorabchi, Mesbah, & Kruchten, 2013). Among several reasons, the main reason for the adoption of PBCL in computer programming courses is to train students in principles, methods, and procedures under conditions like those which characterize the development of actual software products (Yadav, & Xiahou, 2010). In the real world, developing software products results from teamwork which requires both technical know-how and soft skills. These skills consist of the ability to assume responsibility for making choices, monitor the progress of the tasks, communication, and teamwork. Projects in PBCL help students to gain knowledge in the project presentation, documentation and technical reports writing which are vital soft skills needed in the real world (Kizaki, Tahara, & Ohsuga, 2014).

1.1 Background

This research was carried out in the Department of ICT Education at the University of Education, Winneba. Since the inception of the Bachelor of Science program in ICT education in 2011, computer programming courses have been taught using a PBCL approach from the first year (level 100) to fourth year (level 400). The goal is to allow students to apply the theories and principles they learn in these courses to create products of real value, particularly for educational use. By the completion of the program, students would have studied five computer programming courses that are taught using the PBCL method. Table 1 shows a list of computer programming courses that students learn according to their level. Students learn one programming course in level 100 and level 300, two courses in level 200 and in the final year, there is an option of selecting one course from three different programming courses.

Table 1: List of programming courses

Course	Level
Introduction to Computer Programming Languages	Level 100
Object-Oriented Programming with Java	Level 200
Programming with Visual Basic	
Educational Software Development using ASP.NET	Level 300
Advanced Object-Oriented Programming with Android or	Level 400
Advanced Software Development or	
Advanced Website Development using PHP.	

The first programming course students study is Introduction to Computer Programming Languages. This course introduces students to the art of programming. It begins with understanding the logic and concepts of programming like pseudocode, algorithms, and flowcharts. C++ is used as the language of instruction in this course because of its simple syntax, elementary nature of classes, objects and its widespread use. Students are also introduced to functions, methods and control structures. The main assessment in this course is a group project. Students work in groups on a selected real-world project topic and develop it to completion at the end of the course after which they present the project in class. The second programming course is Object Oriented Programming (OOP) with Java. This course gives students full exposure to the concept of OOP. The course introduces students to classes and objects. It also exposes students to concepts like inheritance, polymorphism, and abstraction among others. Students are allowed to create a graphical user interface (GUI) applications and encouraged to explore concepts beyond what is taught in class. Database-driven applications are usually the standard accepted for this course. Students work in groups to develop projects by applying the concept they have been taught.

The third programming course is Programming with Visual Basic which allows students to create event-driven applications. Apart from basic concepts like data types and control structures, they are introduced to string manipulation, event handling, file handling, error handling and connecting to several types of databases. Finally, students are taught how to generate reports. Students work in groups to develop their projects. The fourth programming course is Educational Software Development using ASP.NET. This course diverts from desktop application development and delves into web development. This new dimension presents new challenges to the students to equip them for the world. Similar concepts from the previous programming courses are tackled but using different techniques. Concepts like webpage styling using Cascading Style Sheets (CSS) and multimedia authoring are applied in this course. The focus of this course is to expose students to the steps involved in developing educational software for the basic and secondary levels of education through group work.

In level 400, students would have studied different programming languages and developed an affinity for one programming language. Students must select an elective option in their area of interest to pursue. The fifth programming course students take will be either Advanced Object-Oriented Programming with

Android or Advanced Software Development or Advanced Website Development using PHP. Advanced Object-Oriented Programming with Android focuses on developing applications to be used on mobile platforms, specifically the Android platform. Students learn the advanced concepts of mobile app development. The project for assessment in this course is an android mobile app developed using the Android Studio. Advanced Software Development is the elective option that allows students to develop complex projects using the Visual Studio Integrated Development Environment. Students can use either VB.NET or ASP.NET to develop their selected projects. Complex concepts like remote database connections, database backup and recovery are taught in this course among others. Advanced Website Development is for students who have an interest in open-source web development. The programming language of choice is PHP. In groups, students are expected to develop web applications that can connect to databases and generating dynamic reports among others.

1.1.1 Student Grouping

At the beginning of every semester, lecturers assess requisite knowledge of students for the course through diagnostic tests and students' cumulative grade point averages (CGPA). CGPA is calculated as a weighted average of the letter grades that students earn at the end of each semester on a scale of 0 to 4.0. Depending on the CGPA score of the student, they are ranked in terms of First class (3.50-4.00), Second class upper (3.00-3.49), Second class lower (2.50-2.99), Third class (2.00-2.49), Pass (1.00-1.99) and Fail (0.00-0.99). The CGPA takes into consideration the number of credit hours in each course and the scale associated with the grade of students. The assessment of the requisite knowledge of students is used to rank the students by performance. The list of students is then divided into several sub-lists depending on the number of members that have to be in each student group. The first sub-list will be the highest-ranked students, followed by the next sub-list with above-average students and so on till the last sub-list. Each student group will now be constituted by selecting a member from each sub-list. This gives a balance across all student groups with excellent students, average students, and below-average students. Most student groups have four (4) members.

1.1.2 Group Project

Group projects in the PBCL computer programming courses follow three cyclical phases similar to the suggestion of Han and Bhattacharya (2001). Each group of students is expected to demonstrate an understanding of the problem that they have chosen, explore the curriculum for more information and resources on the problem and resolve the problem (see figure 1).

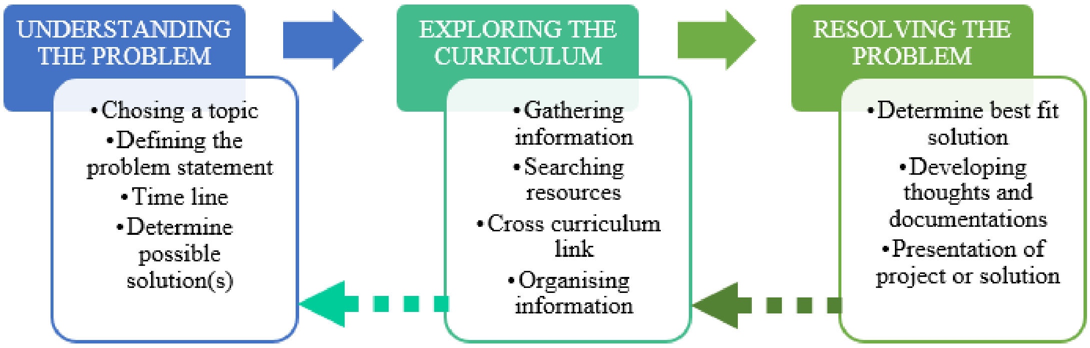

Figure 1: Phases of PBCL project (adapted from Han & Bhattacharya, 2001)

1.1.3 Understanding the Problem

In this phase, students are allowed to choose from a list of projects in computer programming. After the groups have decided which topic they want, they are expected to demonstrate an understanding of the problem by presenting a proposal on how they are going to solve the problem. The proposal should include a well-documented definition of the problem, strategies, timelines and the authentic product that will result from the solution to the problem. Oral presentations are done by each group for the instructors and students. Students are allowed to ask questions and contribute to the group presentations. A final proposal is then submitted as part of the formative assessment.

1.1.4 Exploring the Curriculum

In this phase, students are expected to gather the information that is needed to generate possible solutions to the problem. Each group is expected to apply knowledge in programming such as data structures and coding to resolve the chosen problem. The students are allowed to use reference material from the library and the internet to assist them in finding the solution to the problem. It is a requirement for each group to document their reference material as part of the technical report that will accompany their final project submission. In this way, students are not limited to what is taught in the classroom alone, but they can apply new knowledge from different sources and across courses. However, the projects are expected to be produced in the particular programming language that they are studying.

1.1.5 Resolving the Problem

After exploring the curriculum, students are expected to determine the best fit solution for the problem they have identified. The solution to the problem is expected to be in the form of an authentic computer software product. Each group is expected to document and write a technical report on the solution to the problem that they have discovered. These reports are submitted for assessment. The final product is then presented to the class as part of the end of semester examination grading. During the presentation, each member of the group is required to demonstrate their contribution to the project. Projects typically last between eight (8) to ten (10) weeks.

1.1.6 Assessments

In all the programming courses, formative assessments like quizzes and assignments are used to test theoretical concepts. Code reviews and presentations are used to test student practical knowledge. Code reviews and presentations allow lecturers to gain an insight into the progress the various groups are making and any challenges that the groups are facing in terms of member participation or deviation from desired objectives. It also allows lecturers to give the desired input and direction to all the groups. Other students are also allowed to express their views, suggestions, and criticisms of the projects presented. Finally, the presentation allows all project groups to see the progress other groups are making and serves as a source of encouragement, motivation, and learning. Summative assessments are conducted at the end of the semester to see the solutions each group developed. At the end of the course, the formative assessment makes up 40% of the final score. The breakdown of the formative assessment includes 10% quizzes, 10% assignments and 20% for code review/presentations. The summative assessment which makes up the remaining 60% consists of written examination (30%) and group project (30%).

1.1.7 Evaluation

Students are evaluated for their knowledge exhibited during the code reviews and presentations. A rubric detailing what will be evaluated and the weight for each item is well documented and given out to group leaders at the beginning of the semester. During the presentation, each member must demonstrate or present some sections of the project. Each member of the group is questioned which allows them to express their knowledge on the project. Documentation accompanying the project is also reviewed.

1.2 Problem Statement

Learning is the act of acquiring new or modifying and reinforcing existing knowledge, behaviours, skills, values, or preferences which may lead to a potential change in synthesizing information, depth of the knowledge, attitude or behaviour relative to the type and range of experience (Gross, 2010). We are in the age where learning environments need to prepare learners for the complexities of the professional world by utilizing instructional activities reflecting the problem solving and challenge-meeting process professionals use on their jobs. Problem-based learning has been utilized as a design methodology for teaching in engineering, science, medicine, and economics for over 40 years (Strobel & van Barneveld, 2009). Project-based learning (PBL) is an instructional model that is based on the constructivist approach to learning, which entails the construction of knowledge with multiple perspectives within a social activity, and allows for self-awareness of learning and knowledge while being context-dependent (Duffy & Cunningham, 1996)

In Ghana, many universities offer computer programming related programs. Even though extensive work has been done in the use of PBCL in higher education, to the best of our knowledge, very few attempts have been made to investigate students' experience, level of collaboration and challenges they faced in learning computer programming with PBCL in developing countries like Ghana. It was, therefore, necessary to conduct an empirical study to investigate the experiences of students’ on using PBCL in their programming courses. Students' levels of collaboration and the challenges faced.

1.3 Purpose

The purpose of this study was to investigate students' experiences, levels of collaboration and challenges in learning computer programming via PBCL. We anticipate that the findings of this research will inform stakeholders, especially lecturers on the merits of using PBCL. Generally, we expect this research to influence policy on the teaching of programming languages.

1.4 Research Questions

1. What are the experiences of students on using PBCL in their programming courses?
2. What level of collaboration exists between students in programming courses using PBCL?
3. What are the challenges students face in programming courses using PBCL?

2. Literature Review

Project-Based Learning (PBL) is generally defined as a model that organizes learning around projects. (Thomas, 2000). Many researchers have defined projects by several criteria including the following: Projects are complex tasks, based on challenging questions or problems, that involve students in design, problem-solving, decision making, or investigative activities; allow students to work relatively autonomously over an extended period; and culminate in realistic products or presentations (Jones, Rasmussen, & Moffitt, 1997; Thomas, Mergendoller, & Michaelson, 1999). Projects must include features like authentic content, authentic assessment, teacher facilitation but not direction, explicit educational goals, (Moursund, 1999), cooperative learning, reflection, and incorporation of adult skills (Diehl, Grobe, Lopez, & Cabral, 1999). However, we agree with (Thomas, 2000) on the characteristics a project should exhibit to be considered as PBL; these are centrality, driving question, constructive investigations, autonomy, and realism.

There are a few other learning paradigms that try to achieve somewhat similar results as Project-Based Learning (PBL) but use a different approach. Two of such popular paradigms are Peer Assisted Learning (PAL) and Problem Based Learning (PBL). Peer-Assisted Learning (PAL) involves students who actively assist others to learn and in turn benefit from an effective learning environment. PAL helps students develop theoretical knowledge and skills by interacting with colleagues, and improves communication skills, enhances personal development and boosts performance. PAL has been commonly used in educational areas such as clinical education. According to Romito (2014), many medical schools have used it as a way, in part, to deliver undergraduate teaching. Some of the advantages of PAL are summarized as higher-level cognitive reasoning, improved interpersonal skills, enhanced self-worth, increased motivation for learning, greater active learner engagement, improved group discipline, development of teaching skills, lesser demand on institutional resources, modest curriculum adaptations to accommodate it and cost-effective education. On the other hand, problem-based learning is an instructional tactic that allows students to work communally to investigate and resolve an ill-structured problem based on real-world issues. The steps involved in problem-based learning permit the student to use inductive and deductive reasoning as they probe and refine a possible solution to an identified problem.

Students have been taught computer programming for many years without using a PBL approach. In a traditional classroom setting, the instruction often utilizes simplified, decontextualized examples and problems (Collins *et al*, 1991) leading to an inadequate understanding of and ability to apply the understanding developed. Bransford *et al* (1990) state that the basic problem is that traditional instruction often fails to produce the kinds of transfer to new problem-solving situations that most educators would like to see. Information presented to students has no relevance or meaning for them, they tend to treat new information as facts to be memorized and recited rather than as tools to solve problems relevant to their own needs (Grabinger, Dunlap, & Duffield 1997). Some teachers confuse the well-defined weekly programming tasks given to students as PBL. For example, most computing courses involve setting “problems” which students are required to complete. We will refer to these as exercises because they are small and well-defined. (Barg, et al., 2000). All programming courses will require some form of problem-solving, however, PBL includes significantly more extensive issues, which include a bigger arrangement of critical thinking abilities. PBL places critical thinking and metacognitive abilities at the core of the educational programs. Class time is dedicated to such generic critical thinking abilities as characterizing a learning design, conceptualizing to start on a problem, reflection, articulation of issues and solutions, self-evaluation, practice in active listening, and other communication skills. These aspects are also assessed and contribute to the grade awarded. (Barg, et al., 2000; Grabinger, Dunlap, & Duffield 1997)

3. Methodology

3.1 Research Design

The study used a cross-sectional survey research design with mixed quantitative and qualitative data. Fraenkel, Wallen, and Hyan (2012) explain a cross-sectional survey as a study that involves collecting information at just one point in time from a sample that has been drawn from a predetermined population by administering questionnaires to individuals to find out specific characteristics of the group. Survey research determines and reports the way things are through collecting numerical data to test hypotheses or answer questions about the current status of the subject of study. Surveys are commonly used in assessing the preferences, practices, concerns, or interests of a group of people (Gay, Mills & Airasian 2012). We considered a cross-sectional survey design as appropriate for this study because we sought to investigate students’ experiences and challenges in learning computer programming courses through PBCL.

3.2 Population and Sample

The population for this study is a set of students from the University of Education, Winneba who have participated in collaborative group projects in computer programming courses that were taught using the PBCL method. There were 840 students from level 100 to level 400 of which 772 (91.9%) were male and 68 (8.1%) were female. In all, there were 226 (26.9%) level 100 students, 268 (31.9%) level 200 students, 193 (23.0%) level 300 students and 153 (18.2%) level 400 students.

A sample of 428 students was selected using a stratified random sampling technique. Stratified random sampling is a process in which certain subgroups, or strata, are selected for the sample in the same proportion as they exist in the population (Fraenkel, Wallen & Hyan, 2012). This sampling method was considered appropriate for the study because of the lower number of females (8.1%) compared with male (91.9%) students in the Bachelor of Science ICT Education program. To make the sample representative of the population, we sought to get the same proportion of male and female distribution according to levels in the population. This would have been difficult with a simple random sampling technique. Each level in the population was divided into two strata (male and female) and their respective proportions were calculated. Using random sampling, a corresponding number of samples was then selected based on the proportions of males and females in each level as in the population (see table 2). The random sampling was done by using random numbers and participants were selected if the number matched with their student identification number. In all the sample consisted of 393 (91.8%) males and 35 (8.2%) females.

Table 2. Proportion of males and females in the population and sample.

	Population = 840				Sample = 428			
	Male		Female		Male		Female	
	N	%	N	%	N	%	N	%
Level 100	205	90.7	21	9.3	118	90.8	12	9.2
Level 200	246	91.8	22	8.2	116	91.3	11	8.7
Level 300	180	93.3	13	6.7	136	93.2	10	6.8
Level 400	141	92.2	12	7.8	23	92.0	2	8.0
Total	772	91.9	68	8.1	393	91.8	35	8.2

3.3 Instrument

The instrument for the study was a course experience questionnaire (CEQ) which was adapted and modified from Ramsden (1991). The CEQ is based on a theory of university teaching and learning in which students' perceptions of curriculum, instruction, and assessment are regarded as key determinants of their approaches to learning and the quality of their learning outcomes (Ramsden, 1991). The CEQ consisted of 23 items which were divided into four parts. The first part elicited participant demographic data which are *gender* and *level.*

Part two consisted of 7 items that were used to measure participants' *experiences* in learning computer programming courses using PBCL. Part three consisted of 13 items and measured participants' levels of *collaboration* in projects. Parts two and three were Likert-type scale with 4 response options (1. Strongly disagree, 2. Disagree, 3. Agree, 4. Strongly agree) based on theoretical considerations (Allen & Seaman, 2007). The reason for using a Likert-type scale with even response options is to remove a mid-point (e.g. Neutral) which can be easily misinterpreted as a true neutral option or whether the respondent was unsure about the item. Additionally, it helped to balance the response categories which allows participants to discriminate in the responses. There was one open-ended question in part four which allowed participants to express their challenges about PBCL.

The instrument was vetted by three senior lecturers who have been using PBCL in teaching computer programming. This helped in making corrections with the wording of some of the items to place the CEQ in the context of the study. The questionnaire was piloted on 32 students who had the same characteristics as the sample. These students were identified and excluded from the main study. The pilot data was used to calculate the reliability of the instrument. The Cronbach's Alpha reliability statistics for the CEQ was 0.764. Besides, the two main variables – experiences, and collaboration had Cronbach's Alpha reliability statistics which ranged from 0.712 to 0.833 (Table 3). These values are in the acceptable range of reliability (Schmitt, 1996, Fraenkel, Wallen & Hyan, 2012) and so the instrument used in the study was considered reliable.

Table 3: Reliability statistics of the course experience questionnaire (CEQ)

	Reliability statistics	
Variable	Cronbach's alpha	No. of items
Experiences	0.833	7
Collaboration	0.712	13

3.4 Procedure

To ensure independent responses from participants, the students who were sampled for the study were assembled in a lecture hall and given the questionnaire to fill. Assembling the participants ensured that any anomalies with the questionnaire were answered. The responses from the questionnaire items were coded and keyed into IBM Statistical Package for Social Science (SPSS v.22) for analysis.

3.5 Data Analysis

We sought to use parametric statistical methods which require that the dependent variable is approximately normally distributed for each category of the independent variable. Test of normality was calculated using the overall mean of the answers for experiences, collaboration and preferred assessment constructs on the questionnaire. A Shapiro-Wilk test (with $p>0.5$) (Shapiro & Wilk, 1965; Razali & Wah, 2011), skewness test and a visual inspection of the histograms, and normal quartile-quartile (Normal Q-Q) plots of the three constructs was conducted to investigate their normality.

Table 4: Tests of Normality

			Shapiro-Wilk		
	Skewness	S.E	Statistic	df	Sig.
Experiences	-1.838	.118	.811	428	.000
Collaboration	-.627	.118	.958	428	.000

The Shapiro-Wilk test (table 4 for the contrast: experiences (Shapiro-Wilk=0.811, $p<0.05$) and collaboration (Shapiro-Wilk=0.958, $p<0.05$) indicated that they are not normally distributed. This was confirmed by the Normal Q-Q plot (Figure 2 and Figure 3) and histogram (Figure 4 and Figure 5) of the constructs (Figure 2). Due to the lack of normality of the constructs, median and interquartile range (IQR) were used for the descriptive analysis instead of mean and standard deviation typical of normal distribution (Boone & Boone, 2012). IQR is the difference between the 75th (Q3) and 25th (Q1) percentile which acts as a substitute to the standard deviation in normally distributed data.

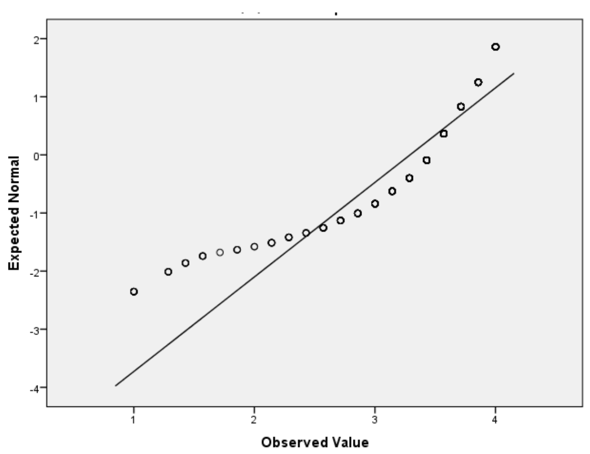

Figure 2: Normal Q-Q Plot of Experiences

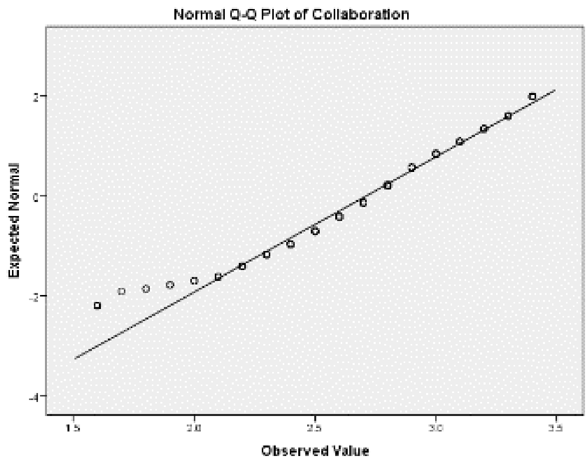

Figure 3: Normal Q-Q Plot of Collaboration

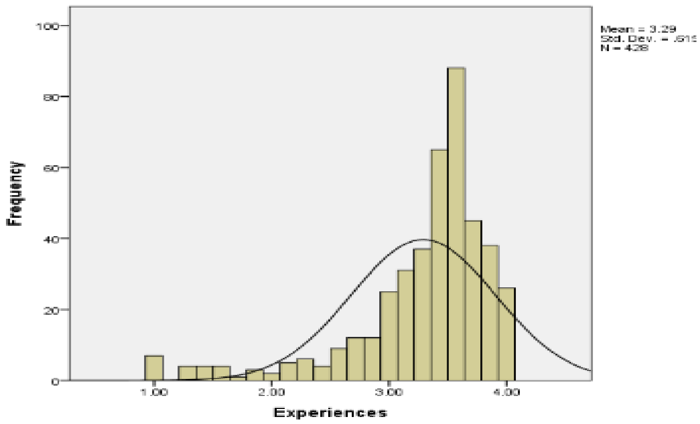

Figure 4: Histogram of Experiences

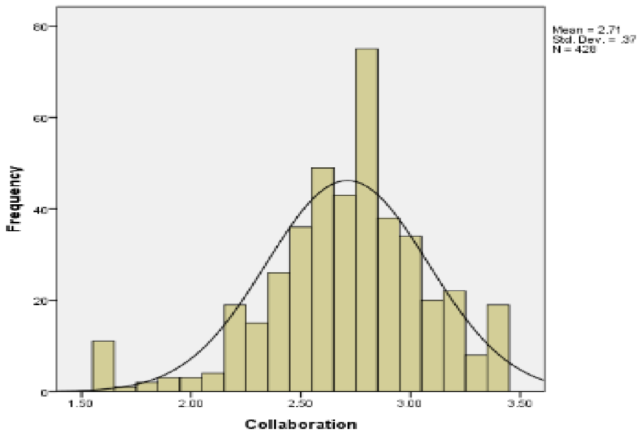

Figure 5: Histogram of Collaboration

4. Results

The results presented below are displayed per the order of the research questions in the study.

4.1 Student's experiences in PBCL computer programming courses

Table 5 presents students' experiences in learning computer programming through PBCL. Median (Mdn) and interquartile range (IQR) were used because of the lack of normality in the data. The results show that students positively evaluated their experiences with all the items having a median of 3 (agree) with IQR of 0 or 1. The cumulative percentage of agree and strongly agree responses show that focusing the courses on real problems made the course more interesting for 85.2% of the students (item 1), 86.6% of students learned a lot more by solving problems they face in their project (item 2) and 78.5% of students understood the technical concepts of the courses better with this teaching approach (item 3). The other items in the experience dimension revealed that the facilitation from the lecturer during the course helped 84.3% of the students to work independently in the project (item 4), 77.1% of the students seeking assistance for other resource persons apart from the lecturer (item 5), 82.4% of students accessed other learning materials such as books, video tutorials to aid them in completion of the project (item 6) and 79.5% of the students prefer group work in future projects (item 7).

Table 5: Students' experiences in PBCL computer programming courses.

	Item	SD N (%)	D N (%)	A N (%)	SA N (%)	Mdn	Q1	Q3	IQR
1	Focusing the courses on real problems made the course more interesting for me	21 (6.3)	36 (8.4)	251 (58.6)	114 (26.6)	3	3	4	1
2	I learned a lot more by solving the problems I faced in my project	25 (6.1)	31 (7.2)	260 (60.7)	111 (25.9)	3	3	4	1
3	I understood the technical concepts of the courses better with this teaching approach.	28 (6.5)	64 (15.0)	247 (57.7)	89 (20.8)	3	3	3	0
4	The facilitation from the lecturer during the course helped me work independently in the project.	25 (5.8)	42 (9.8)	221 (51.6)	140 (32.7)	3	3	4	1
5	I sought assistance from other resource persons apart from the lecturer	36 (8.4)	62 (14.5)	221 (51.6)	109 (25.5)	3	3	4	1
6	I accessed other learning materials to aid the completion of project e.g. books, video tutorials	26 (6.1)	49 (11.4)	221 (51.6)	132 (30.8)	3	3	4	1
7	I will prefer group work in future projects.	30 (7.0)	58 (13.6)	201 (47.0)	139 (32.5)	3	3	4	1
	SD – Strongly Disagree (1), D – Disagree (2), A - Agree (3), SA – Strongly Agree (4), Mdn – Median, Q1- 25th percentile, Q3 - 75th percentile, IQR - Inter quartile range								

4.1 Levels of Student's Collaboration in PBCL Computer Programming Courses

Students' level of collaboration during the PBCL computer programming courses was measured using the CEQ. Table 6 shows descriptive statistics of the different scopes that were used to measure students' level of collaboration. Overall, the result shows the students evaluated their level of collaboration positively. The cumulative percentage of agree and strongly agree responses shows that the use of collaborative groups improved the ability of students (57.2%) to learn (item 8) and 56.1% of the student learned more than they would in a solo project (item 9). A significant number of students (78.2%) preferred doing programming course projects in a collaborative group (item 11) and 58.9% of them favoured groups with the same ability (item 13). On the contribution of members in the collaborative groups, 78.5% of the students trust that their group members contributed their best with respect to the project (item 14) and 89.4% of the students trust they contributed their best to the group project (item 15). More students (65.4%) believed that they gained new interpersonal skills by working in a group (item 20).

Table 6: Students levels of collaboration in PBCL computer programming courses.

	Item	SD N (%)	D N (%)	A N (%)	SA N (%)	Mdn	Q1	Q3	IQR
8	The use of collaborative groups improved my ability to learn	48 (11.2)	135 (31.5)	143 (33.4)	102 (23.8)	3	2	3	1
9	I wouldn't have learnt more if I did the project alone	60 (14.0)	128 (29.9)	154 (36.0)	86 (20.1)	3	2	3	1
10	I do not prefer doing programming course projects alone	104 (24.3)	169 (39.5)	96 (22.4)	59 (13.8)	2	2	3	1
11	I prefer doing programming course projects in a collaborative group	25(5.8)	68 (15.9)	188 (43.9)	147 (34.3)	3	3	4	1
12	I prefer being in a group with member who have varied knowledge distribution. e.g. 1 excellent, 1 average, 1 below average	120 (28.0)	187 (43.7)	74 (17.3)	47 (11.0)	2	1	3	2
13	I prefer being in a group with members of the same knowledge level e.g. (3 excellent students or 3 average students or 3 below average students	71 (16.6)	105 (24.5)	151 (35.3)	101 (23.6)	3	2	3	1
14	I think my group members contributed their best with respect to the project	36 (8.5)	56 (13.1)	224 (52.3)	112 (26.2)	3	3	4	1
15	I contributed my best to the group with respect to the project	24 (5.6)	24 (5.6)	213 (49.8)	167 (39.6)	3	3	4	1
16	I prefer to form my own collaborative group in subsequent programming courses	105 (24.5)	201 (47.0)	82 (19.2)	40 (9.3)	2	2	3	1
17	I am happy with the composition of my collaborative group	38 (8.9)	88 (20.6)	220 (51.4)	82 (19.2)	3	2	3	1
18	I was able to work with the group I was given	107 (25.0)	162 (37.9)	97 (22.7)	62 (14.5)	2	1.25	3	0.5
19	I think my interpersonal relationship skills have improved because i worked in a group	29 (6.8)	43 (10.0)	230 (53.7)	126 (29.4)	3	3	4	1
20	I gain new interpersonal skills by working in a group	48 (11.2)	100 (23.4)	138 (32.2)	142 (33.2)	3	1	3	1
	SD – Strongly disagree (1), D – Disagree (2), A – Agree (3), SA – Strongly Agree (4), Mdn – Median, Q1- 25th percentile, Q3 - 75th percentile, IQR – Inter quartile range								

However, some of the items (10, 12, 16 and 18) recorded a median of 2 (disagree) with IQR of 0.5 to 1. These items were negatively worded to prevent satisficing response. The cumulative percentages of strongly disagree and disagree show that 63.8% of the students do not prefer doing programming course projects alone (item 10) which confirms the finding in item 11. Also, 71.7% of the students do not prefer being in a group with members who have varied knowledge distribution (item 12) which confirms the finding in item 13. Interestingly, 71.5% of the students do not want to form their collaborative group (item 16). Meanwhile, 62.9% of the students claimed they were not able to work with the group they were given.

4.6 Challenges Students Face in PBCL Computer Programming Courses

Students' responses to the open-ended item on the questionnaire were used to investigate the challenges that they faced in learning computer programming through the PBCL method. It was found that the challenges students faced were mainly associated with collaborative groups and assessments. With the groups, the main challenge students faced was the constitution of the group and time for group meetings. On the constitution of the group a student observed that: *We should be allowed to form our collaborative groups rather than forming groups based on performance from previous programming courses.* This suggests that grouping students according to performance do not always work for some students because the level of students' performance does not necessarily indicate their level of commitment in the group projects as observed by a student:

> *I usually find myself in groups where members are good at programming languages and are not committed. They do not come for group discussions and as a result, we usually present an unimpressive project work that I do not like.*

One reason for the lack of commitment to group projects might be that students who are good in programming courses might feel that other students in the group want to "piggy ride" on them for their success in the course as expressed by one of the students:

> *Some students don't actually take part since [the project] is a group based with the mindset that other members will do the work. Some members tend to piggy ride on people with higher knowledge of the subject area.*

Another challenge that students identified was time for group meetings. From the students' responses, most of them find it challenging to get time for group meetings as one student puts it:

I believe the course aids in solving real-world problems but the time allocated for studies and group meetings is limited.

5. Discussion

The result of the study suggests that students have more positive experiences in learning computer programming courses with PBCL. Students found the courses interesting, understood the technical concepts of programming through problem-solving and were able to expand their knowledge by accessing other learning resources outside the scope of the courses to complete projects. These findings are consistent with the finding of other studies which suggests that PBCL engages students in learning knowledge and skills through an extended inquiry process structured around carefully designed authentic products and tasks (Shpeizer, 2019; Markham, Larmer, and Ravitz, 2003). PBCL also enhances student motivation as they work on a preferred topic which are authentic real-world problems (Saukkonen, 2014). The findings of the study also show that students improved their collaborative and interpersonal skills through group projects. Collaborative work encourages students to be more involved and responsible (Donnelly, 2005).

Collaborative work is a difficult aspect of PBCL (Kapp, 2009). Brilliant students tend to collaborate more than low ability students. Low ability students tend to be “hitchhikers” who contribute little in discussion due to the lack of experience and skills in joint learning (Pawson et al., 2006). It is therefore essential for lecturers to create a classroom culture of collaboration, where students feel responsible for helping each other, and of interaction, where they expect to make mistakes to learn from them (Kolodner et al., 2003). More positive experiences were found in level 100 students. In Level 200 students, a more positive assessment of experience and collaboration was found. However, in level 300 and 400, none of the dimensions were found to be significant. The positive experiences at level 100 might be because this type of teaching method is new to the students most of whom are coming directly from senior high school. Collaboration levels in Level 200 are also high because, at that level, students are now acquainted with their colleagues which make it easier for them to collaborate. Level 300 and 400 students had lower collaboration levels possibly because many more students had gained some experience and would prefer working alone. Students' challenges in PBLC computer programming courses include time allocation for projects, choosing appropriate problems and piggy riding in project groups. These findings are consistent with a study by Huysken et al. 2019 which indicates that PBLC is time-intensive for both instructors and students as more time is needed to master specific concepts before applying them to create authentic products.

6. Conclusion

The study has established that students have a positive learning experience in computer programming when PBCL is used. Focussing the computer programming courses on courses on real-world problems makes the course more interesting for students. Students gain new interpersonal skills and understand the technical concepts of the programming better through the use of group projects. For students to continue benefiting from the merit of PBCL, we recommend institutional recognition of the importance of PBCL and the principle it represents. Higher education institutions must also recognize the additional time and effect required by both lecturers and students in the PBCL course in order the inform policy decisions. There is also a need for continuous training of the lecturer in the effective ways of using PBCL.

7. References

Allen, E., & Seaman, C. A. (2007). Likert Scales and Data Analyses. *Quality Progress*, 40, 64-65.

Barg, M., Fekete, A., Greening, T., Hollands, O., Kay, J., & Kingston, J. H. (2000). Problem-Based Learning for Foundation Computer Science Courses. *Computer Science Education*, 109-128.

Boone, H. N. & Boone, D. A (2012). Analyzing likert data. *Journal of extension*. 50(2) pp.

Bransford, J.D., Sherwood, R., Hasselbring, T., Kinzer, C. and Williams, S. (1990), 'Anchored instruction: why we need it and how technology can help' in Nix, D. and Spiro, R. (eds), Cognition, Education, and Multimedia: Exploring Ideas in High Technology, Hillsdale NJ: Lawrence Erlbaum, 115-42.

Collins, A., Brown, J.S. and Holum, A. (1991), 'Cognitive apprenticeship: making thinking visible' *American Educator, Winter* (6-11), 38-46.

Diehl, W., Grobe, T., Lopez, H., & Cabral, C. (1999). Project-based learning: A strategy for teaching and learning. Boston, MA: Center for Youth Development and Education, Corporation for Business, Work, and Learning.

Donnelly, R. (2005). Using Technology to Support Project and Problem-Based Learning. In: T. Barrett, I. Mac Labhrainn and H. Fallon, eds., Handbook of Enquiryy and Problem-Based Learning: Irish Case Studies and International Perspectives, 1st ed. Galway, Ireland: Celt, pp. 157-177.

Fraenkel, J. R., Wallen, N. E. & Hyan, H. H. (2012). *How to design and evaluate research in education* (8th ed.). New York:McGraw-Hill Publishing Co.

Gay, L. R., Mills, G. E. & Airasian, P. (2012). *Educational research, competencies for analysis and application* (10th Ed.) Boston: Pearson

Gross; Richard. (2010). Psychology: *The Science of Mind and Behaviour*. 6th Edition. London: Taylor & Francis

Han, S., & Bhattacharya, K. (2001). *Constructionism, learning by design, and project based learning*. Available at: http://epltt.coe.uga.edu (Accessed 30 March 2019).

Huysken K., Olivey, H., McElmurry, K., Ming Gao, M., Avis, P. (2019) Assessing Collaborative, Project-based Learning Models in Introductory Science Courses. *Journal of the Scholarship of Teaching and Learning,* 19(1). P.6-28. DOI:10.14434/josotl.v19i1.26777

Jones, B. F., Rasmussen, C. M., & Moffitt, M. C. (1997). *Real-life problem solving.: A collaborative approach to interdisciplinary learning*. Washington, DC: American Psychological Association.

Kapp, E. (2009). Improving student teamwork in a collaborative project-based course. *College Teaching*, 57(3), 139-143.

Kolodner, J. L., Camp, P. J., Crismond, D., Fasse, B., Gray, J., Holbrook, J., Puntambekar, S., & Ryan, M. (2003). Problem-based learning meets case-based reasoning in the middle-school science classroom: Putting learning by design into practice. *The Journal of the Learning Sciences*, 12(4), 495-547.

Markham, T., Larmer, J. & Ravitz, J. (2003). *Project Based Learning Handbook: A Guide to Standards-Focused Project Based Learning for Middle and High School Teachers*. 2nd ed. Novato, CA: Buck Institute for Education.

Moursund, D. (1999). Project-based learning using information technology. International Society for Technology in Education.

Pawson, E., Fournier, E., Haigh, M., Muniz, O., Trafford, J. & Vajoczki, S. (2006). Problem-Based Learning in Geography: Towards a Critical Assessment of its Purposes, Benefits and Risks. *Journal of Geography in Higher Education*, 30(1), pp. 103-116.

Ramsden P (1991) A performance indicator of teaching quality in higher education: The course experience questionnaire. *Studies in Higher Education*, 16 (2), 129-150

Razali, N. M., & Wah, Y. B. (2001). Power comparison of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Andersong-Darling tests. *Journal of statistical modelling and analytics* 2(1), 21-33.

Romito, A. (2014). Peer assisted learning. Retrieved from http://www.essentialgptraining

Saukkonen, J. (2014). Effects of Project-Based Learning in Education-Enterprise Collaboration to Learning Experience and Students Engagement. [Online] Finnish Business Review. Available at:http://urn.fi/urn:nbn:fi:jamk-issn-2341-9938-4

Schmitt, N. (1996). Uses and abuses of coefficient alpha. *Psychological Assessment*, 8(4), 350–353.

Scott Grabinger, Joanna C. Dunlap & Judith A. Duffield (1997) Rich environments for active learning in action: *Problem-based learning, ALT-J,* (5)2, 5-17, DOI:10.1080/0968776970050202

Shapiro, S. S., & Wilk, M.B. (1965). An analysis of variance test for normality (Complete Samples). Biomerika, 56(3) 591-611.

Shpeizer, R. (2019).Towards a Successful Integration of Project-based Learning in Higher Education: Challenges, Technologies and Methods of Implementation.*Universal Journal of Educational Research* 7(8): 1765-1771. DOI: 10.13189/ujer.2019.070815

Strobel, J., & van Barneveld, A. (2009). When is PBL More Effective? A Meta-Synthesis of Meta-Analyses Comparing PBL to Conventional Classrooms. *Interdisciplinary Journal of Problem-Based Learning*, 3(1).Available at: http://dx.doi.org/10.7771/1541-5015.1046

Thomas, J. W. (2000). A Review of Research on Project-Based Learning. California: Autodesk Foundation.

Thomas, J. W., Mergendoller, J. R., and Michaelson, A. (1999). *Project-based learning: A handbook for middle and high school teachers*. Novato, CA: The Buck Institute for Education.