DOI: http://dx.doi.org/10.26483/ijarcs.v13i6.6934

Volume 13, No. 6, November-December 2022

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

ENHANCING THIRD-YEAR COMPUTER SCIENCE STUDENTS' COMPUTER PROGRAMMING SKILLS AT HAWASSA UNIVERSITY, ETHIOPIA

Tewodrose Tilahun
Hawassa University
Institute of Technology
Faculty of Informatics
Hawassa, Ethiopia

Binyam Bayou
Hawassa University
Institute of Technology
Faculty of Informatics
Hawassa, Ethiopia

Workineh Negasa
Hawassa University
Institute of Technology
Faculty of Informatics
Hawassa, Ethiopia

Abstract: Teaching programming in an efficient and effective method has been the most difficult subject in the world of computing for the last couple of decades. Aside from a lack of laboratory equipment, the most prevalent issues are the lengthy time it takes to understand a problem, devising algorithms, writing coding, syntax, and semantic complexity of the programming language. Because of the wide range of students' backgrounds, misconceptions about the course, traditional classroom teaching methodology, and the limited allotted time available to cover the course, it is extremely difficult for a teacher to go beyond the fundamental concepts that impede the development of students' problem-solving abilities. The major goal of this study is to assist students to become better programmers by comprehending code and implementing applications for real-world problems. The study was conducted to uncover specific issues among students who lacked the necessary knowledge and skill sets for computer programming courses for computer science third-year students in the 2022 academic year. A sample of 23 students were chosen using randomized sampling technique for the investigations. Students'primary data were acquired through observation, questionnaires, and readily available data. According to the primary data collected, the researchers discovered that rough teacher-student relationships, disorganized laboratories and a lack of resources, the complicated nature of the course, and a lack of motivation in the teaching-learning process are factors that affect students' performance in understanding and solving real-world problems. The major issues were identified as lack of motivation to learn, persistence, inadequate laboratory facilities, lack of attractive teaching methodology, outdated teaching materials, lack of awareness of the benefit of the course, students' myths about the course difficulty, lack of practical guide materials, lab-assistant level of competence, student bad habit on computer usage, etc. Actions had been taken to solve the aforementioned problems through following well-organized and updated course materials, enabling students to develop a habit of teaching themselves, and establishing well-equipped laboratory facilities for practical demonstrations. The researchers believe that having applied the suggested action will bring a remarkable improvement to the knowledge among students. The researchers proposed different actions to be implementedfor both students and teachers. The recommended action was executed and actively followed up by the researchers. Finally, the researcher discovered that more than half of the students who were underperform get a better performance level and the students'self-confidence was improved, their problem-solving capabilities were enhanced, and they became more excited to engage in project work.

Keywords: Computer programming, coding, real-world problem, performance, boot camp, learning strategy

INTRODUCTION

Background of the Study

Hawassa University (HU) is found in Hawassa City, Southern Region, Ethiopia. It was established in April 2000. Since 1976, the different colleges of HU started with the college of Agriculture. Merging three colleges has formed the University: Awassa College of Agriculture, Wondogenet College of Forestry and Dilla College of Teacher Education and Health Sciences.Hawassa University is located in the Hawassa City, Southern Nation Nationalities and Peoples of Ethiopia. Students enrolled in Hawassa University come from four corners of Ethiopia to pursue their education.

According to (Merriam-Webster dictionary) Learning can be defined as the activity or process of gaining knowledge or skill by studying, practicing, being taught, or experiencing something (Learning is about what students do, not about what we as teachers do). Learning is an active process of appropriation (making one's own) of knowledge, abilities and skills in order to enhance the personal or collective control potential (competence) of shaping reality in a given context or situation. Learning is not only transferring knowledge from instructor to students but also understanding processes where relatively permanent changes are caused by information and experience for the learner.

Computer programming is the process of performing a particular computing task (or more generally, accomplishing a specific computing result), usually by designing and building an executable computer program. Programming involves tasks such as analysis, generating algorithms, profiling algorithms' accuracy and resource consumption, and the implementation of algorithms (usually in a chosen programming language, commonly referred to as coding).

Nowadays, managers and business leaders in giant tech industries don't care about which is the best programming language of all. All they need to care about is efficiency and the speed at which anything can be created or modified. They don't care if the code is 10 lines or 1000 lines long. All they need is something that is easy to create, and customize and need something that will be efficient and requires fewer resources to run the applications. The technology industry as a whole is booming with opportunity as it has been growing immensely over the past decade.

Therefore, there is a strong demand for a highly skilled and potentially adequate number of computer programmers in the field of computer science. However, still, as the industry's job posting report shows, they can't find the expected number of skilled programmers and software engineers. As their demand grows extensively with an enormous number of job postings in hiring for technologyand industriesare not balanced yet.

The U.S. Bureau of Labor Statistics (BLS) projects 13% growth in computer occupations 2020-2030, resulting from demands for improved technologies, security, cloud computing infrastructure, and big data applications. Moreover, according to glassdoor.com and indeed.com/the demand of computer science professional is highly increasing from time to time as job seeking portals reports shows. Specifically, robertsoncollege.com shows that which filed of specialization in computer science is highly the required one compared to the others in order of relevance.

Thus, this study will explore the challenges for acquire the intended skill set of computer programming courses among third-year Computer Science students and identify possible ways to forward.

Statement of the Problem

Computer programming is a fundamental skill that all computer science students are required to learn. There are various kinds of reasons to learn computer programming. Some of the most common of the reasons are to better understand a computer system, to solve a real-world problem, to participate in innovation, to secure financial independence, and others may learn for fun.

However, programming courses are generally regarded as difficult and often have the highest dropout rates in colleges and universities. Most students are not capable of having the appropriate knowledge, skill set, and adequate efficiency in solving real-world problems with regard to producing quality software products, and applications development because of different factors. The most common factors could be that the students may have misconceptions about the learning process of computer programming and they may have the mentality that they are not capable of doing programming. By its nature computer programming requires dedication, persistence, strong critical thinking, and a long duration of time for practicing to better understand. However, most of the students don't want to spend more of their time practicing programming to enhance their programming skills and developing real-world projects.

On the other hand, the low level of adequacy of the practical teaching-learning facilities like laboratory rooms, the number of computers in the room, the performance of available computers, and the conducive nature of the teaching-learning ecosystem are not sufficient. The majority of the instructors are focused on theoretical concepts rather than practical-based teaching-learning. In addition, the larger number of students in a classroom especially during laboratory sessions makes it difficult for the instructors and technical assistants to manage and provide guidance for these large numbers of students.

Research Questions

The following are lists of questions to be investigated in this study:

1. What are the factors affecting students' learning capability of computer programming?
2. What could be the possible learning strategies that enhance student's computer programming skills.
3. Why do most students fail to understand computer programming?
4. What will be the best possible solutions to enhance a student's computer programming skills?
5. What kind of soft skills are required to learn computer programming?

OBJECTIVES OF THE STUDY:

General Objective

The general objective of this action research is to improve students' Computer Programming skills in the case of 3rd-year Computer Science department students at Hawassa University.

Specific Objective

- To identify the factors that affect the learning process of computer programming
- To investigate the reasons behind why most students are not interested to learn programming
- To identify the best learning strategies for enhancing computer programing skills
- To identify the soft skills required to learn computer programming
- For suggesting the possible solution to improve the computer programming learning curve

Significance of the Study

The 21st century is characterized by the presence of technology almost in all our daily lives. New generation students are surrounded by computers and related instruments and will possibly do a job that has not been invented yet. Computing succeeds in conquering most of the aspects of our society and in order to fit in, people need to be adaptable to modern and future technologies. Thus, one

of the aims of teaching computer programming is relevant to solving real-world problems and scenarios related to this.

The significance of this study is to explore the situation relating to the learning of Computer programming in Hawassa University 3rd Year Computer Science students and to suggest techniques, plans, and approaches that will increase students' performance in computer programming skills. Additionally, the study will bring the following benefits:

- Improve teaching learning-styles
- Enhance students' perception of computer programming course
- Help the teacher to develop good strategies of teaching
- Increase the number of students with critical and logical problem-solving skills

- Enhance the quality of education and in turn contribute to the development of the country at the national level

DELIMITATION OF THE STUDY

The scope of this action research is limited only for computer programming courses for 3rd year Computer Science students, Hawassa University in the academic year of 2022.

REVIEW OF RELATED LITERATURE

Computer programming is a fundamental skill that all Information Technology students are required to learn. However, programming courses are generally regarded as difficult and often have the highest dropout rates (Gomes, Areias, Henriques& Mendes, 2008 ;). In the scientific literature, many reasons are pointed out for this, such as the following.

Methodology and tools used—traditional teaching methods, normally based on lectures and specific programming language syntaxes, often fail in what concerns the students' motivation in getting involved in meaningful programming activities (Schulte &Bennedsen, 2006).

Programming languages typically used in programming classes are professional in nature, such as C, C++, C# and Java; they have extensive and complex syntaxes, rendering learning difficult for beginners (Jenkins, 2002; Motil& Epstein, 2000). Students' difficulties with abstract concepts—knowing how to design a solution to a problem, subdivide it into simpler code able subcomponents, and conceive hypothetical error situations for testing and finding out mistakes (Morgado&Martins, 2008); difficulties in understanding even the most basic concepts (Miliszewska& Tan, 2007) such as variables, data types or memory addresses as these abstract concepts do not have direct analogies in real life (Miliszewska& Tan); and not knowing how to use the programming language correctly to create a program (Winslow, 1996).

DeFino and Bardzell(2006)A computer program is a series of instructions written in some computer language that performs a particular task. Many times, beginning students concentrate solely on the language code; however, quality software is accomplished only after careful design that identifies the needs, data, and process and anticipated outcomes. For this reason, it is critical that students learn good design techniques before attempting to produce a quality program. Design is guided by an algorithm, which is a plan of attacking some problem.

A popular definition is that programming is the process of writing, testing, debugging/troubleshooting, and maintaining the source of code of computer programs (Wikipedia, 2007). We will later see that programming is a much broader topic than that described by the latter definition, as for example the ability to solve a complex problem with a top-down approach. Programming is a skill that is considered hard to learn and even after two years of instruction, the level of programming understanding is low (Kurland et al., 1989). However, if supported by suitable teaching strategies and tools it can be mastered by pupils to some extent (Papert, 1980).

Teaching and learning process improvement in universities include using new teaching models, and new technologies. The integration of new technology into computer science (CS) and Information Technology (IT) education programs is often accompanied by studies that aim to understand and improve the teaching and learning process. How we evaluate the potential of emerging technologies and integrate them into teacher education has clearly become increasingly important. (Pears, Daniels, & Berglund).

Course-related activity can be divided into two broad categories, influences and evaluation/research. The influences on a course are classified as follows, tools, stakeholders, and education theory. By tools we mean tools and technologies used to realize aspects of the course environment. Examples of tools are course websites, laptop computers, computer-based teaching products, and wireless networking. Stake-holders refers to the community which has an influence on the content, form and approach taken in designing a course. This category is also intended to capture implicit aspects of the course context such as the expectations of administrator staff, students and teaching staff with respect to the course. Teaching staff, whatever their background, have ideas about how to teach. These ideas might be implicit or explicit.

(Sheard, Hagan, & Macdonald, 1997) found that computer programming courses are more difficult and time consuming than other courses for the majority of students. (Deek&Epsinosa, 2005) believe that studying subjects such as programming languages are difficult because they have been designed without attention to human-computer interaction. Research shows that students in programming courses have difficulty visualizing abstract concepts (McSporran& King, 2005). Emerging research on the use of learning objects to teach AI indicates that deep learning at conceptual, technical and practical levels only occurs when students have opportunities to visualize and engage practically with their programming. Computer Science is recognized as an area of instruction that requires complex conceptual knowledge and understandings, and is potentially

both highly technical and practical in nature. The understanding of essential abstract concepts that underpin the development of a programming mindset present more challenges to teaching and learning than in other courses (Jehng& Chan, 1998). Students in Computer Science learning programming as a subject, must develop competence in several cognitive areas such as syntactic knowledge and conceptual knowledge (Lischner, 2002). They then need to develop strategies and utilize their problem-solving skills to creatively solve programming problems or to create new programs (Bayman & Mayer, 1983). (McSporran& King, 2005) maintain that cognitive development and the development of conceptual understandings rarely occurs in an isolated environment. Hence, studying Computer Science topics in an online environment is very difficult for the students who may be studying in isolation. Programming languages have a highly technical syntax, with complex rules. Like all language learning, computer programming languages are difficult to learn and understand (Lischner, 2002). In their research, (Deek&Epsinosa, 2005) found that most novice programmers find introductory programming courses frustrating and difficult to learn. Computer Science students, particularly those working in programming subjects, are at significant risk when attempting an online course compared to traditional classroom students. Best practice teaching and learning in Computer Science subjects requires students to learn collaboratively in an interactive environment which mimics the workplace; to use a variety of technologies and learning tools; to participate in active learning to develop technical skills and knowledge; and to engage in conceptual discussions with their peers and teachers to become active problem-solvers (Gulatee& Combes, 2006).

METHODOLOGY OF THE RESEARCH

Research design and methodology

To complete the proposed research, the team followed different research methodologies for solving the problem of students' weak technical skills. The research design usedin this study was action research design and also the data collection mechanisms used were observation, readily available data, and questionnaires. A total of 23 students participated in this study consisting of randomly selected both male and female computer science students from Hawassa University.

Description of the Study Area

The study was held on 3rd-year Computer science students at Hawassa University in the 2022 Academic Year. It was started from March until June **2022 G.C**.

Population, sampling, and sampling techniques

The population of this study is 45 students, from which 23 students were selected randomly. The study used a simple random sampling technique due to the small size of the total population.

Instruments of data gathering

In order to collect all the necessary data for this action research mainly primary and secondary data were used. The primary data were collectedfrom students through the questionnaire and observation and the secondary data was gathered from readily available data.

Observation

In this study observation is done by supervising students when they are in the classroom starting from the beginning to the end of the class in order to understand their intentions, who attentively follow the lesson, how they get on the assessments, who do and who don't engage in the learning process, how they score based on their engagement in classroom were observed to assess students class activity score with other qualities including laboratory practical work or project work.

Questionnaires

We prepared a questionnaire in order to understand the student's experience and exposure to the course computer programming, how often they dedicate their time to programming practices and studying, and to know the strong and weak sides of teaching methodology. Thus, a total of 15 questions were prepared.

Readily Available Data

In this study, we used readily available information such as quiz results, student grades, project results, and exam paper results. Therefore, we evaluate the already available data like immediate classroom quiz results, project results and assignments.

Methods of Data Analysis

It is defined as a process of categorizing, transforming, and interpreting data into useful information for decision-makingin aprofessionalpathway for project implementation. In this action research, we evaluate and summarize the data collected through the use of observation, questionnaires, and readily available data by using the Microsoft Excel tool.

OBSERVATION/ EVALUATION

Observation

During the class observation the researchers have found that the teacher delivers the theoretical perspective of the course and tries to make connections to the real-world scenarios. However, we had found that the students did not participate while the teacher gave activities in the classroom. Additionally, we attended the lab session and found that the teacher already prepared a lab guide and worksheet. The teacher was also supervising their activities. The laboratory equipment vs student number ratio is not proportional atall. The computer-to-student ratio was 1:6. The lab environment was not suitable. The researchers observed that there were some students who wrote code on paper and came to the lab to implement it and see the output. Finally, we already

included the observation checklists in the appendix section of the document.

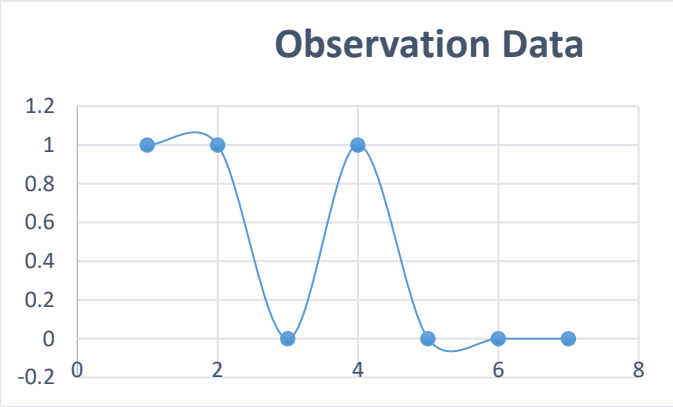

Figure 1: Description of observation results.

Analysis of Questionnaire Response

According to the response from our respondents, 40% of them have prior knowledge about computer programming before joining campus. Among those who do have prior knowledge about programming 89% of them respond that learning programming is important. Among the total respondents, 74% of them agree that learning programming is important. Among those who do have prior programming knowledge, 34% of them think that programming is easy to learn, additionally among those who do not have prior programming knowledge 15% of them think that learning programming is easy.

The researchers believe that those who speak more than two natural languages can benefit from learning computer programming easily. Among the total respondents, 34% of them agree that learning human language and computer programming is somehow similar.

52% of the respondents like to learn programming. Therefore, we tried to investigate the 48% remaining respondents why they hate to learn to program. We found out that among those who hate to learn programming 48% of them dislike to learn due to the difficulty of the syntax, and the remaining 52% believe that the nature of the course is very complicated.

The time spent using computer by the respondent 8.7% uses computer at most 1 hr. per day, 47.8% of them use at most 2 hrs. per a day, 34.7% of them uses at most 4 hrs. per a day, and the remaining 8.7% of them uses a computer at most 6 hrs. a per day. Those who respond on Q6 that they do not like to learn computer programming, 45% of them spend a maximum of 2hrs a day on using the computer. and the remaining 55% respondent spend 3-4 hrs. a day using the computer. Spending time on the computer is not only measured; the researchers ask the respondents what they are doing on the computer. Among them, 21.7% of the respondents spend their time on the computer for social interaction and entertainment purposes. 26 % of them use computers for social interaction, for checking news information, and entertainment purposes. The remaining 52% uses computers for social interaction, checking news, for work/ to study, and for entertainment purposes.

When we came to the effect of programming on increasing the analytical skills of students 60% of the respondents responded to Q13, that learning programming increases their skills to better understand other discipline topics easily.65% of the respondent believed that learning programming through the project, group work, self-thought, participate in boot camp, peer - to peer evaluation, a teacher doing a full project for the students, from reading books, reading other project code, and learning through the game are the best approach if all of them are combined and delivered. The researcher believed that incorporating all these techniques is not an easy task. However, we can't manage to do all the activities within the allotted time due to the need for an extended amount of time which is difficult to finish successfully within a single semester.

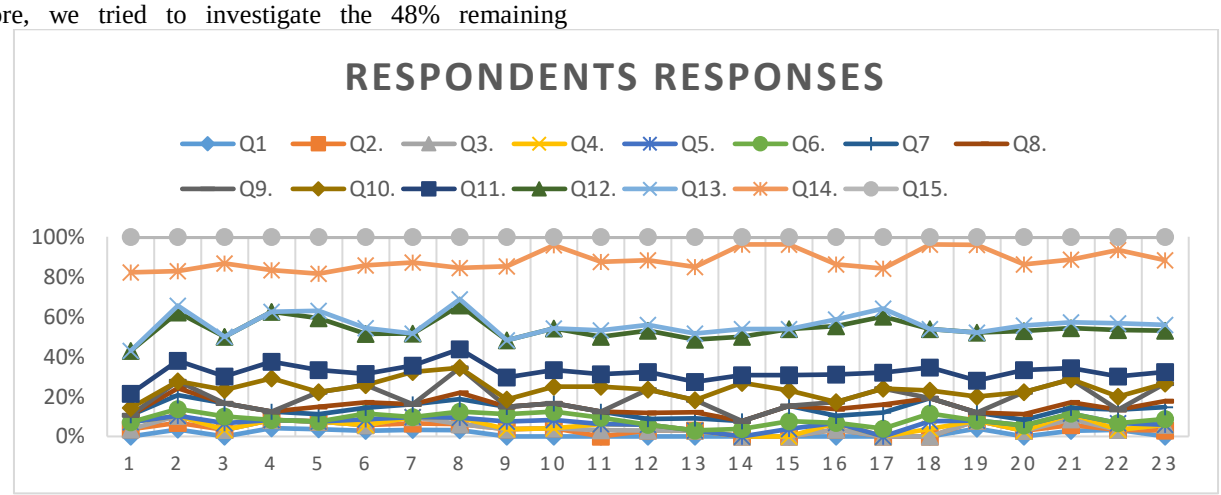

Figure 2: Description of Questioner Response

PROPOSED ACTION

The researchers' proposed actions are categorized into two. These are based on the student and teacher sides.

on the student side, the student must work on the following things. the student must work on their skill.

1. to be **curious**: if you don't have curiosity then you will not succeed in technology

2. to be good at **solving problems**: if you are not good at solving problems systematically you have to work on it.

3. to be **persistent**: if you give up easily in the face of problems, you will not be successful as programmer

4. to be **patient** to learn and understand new things:

5. to be able to **sit for long hour** and focus on one thing at a time

6. to be able to **think outside of the box**

7. to be able to give **attention to details:** read different books and self-thought (e-learning platform, google code jam, Kaggle competition, and another free code camp)

If the student works on these abilities, then programming will be much easier to learn.

on the teacher side: The teacher must deliver the course in different styles. The teacher must work on:

1. giving a more **practical example** as much as possible for every lesson

2. giving **practical homework** (they write code on paper)

3. preparing **Boot-camp:** do one project with the student and the student will do other projects in group

4. **giving them working project code** and make the student read and learn from it

5. **encourage students to self-thought**

6. Preparing worksheets for different topics

Since programming can be a difficult skill to learn, it is certain that most people can learn it. The above list contains attitudes and mindsets that get in the way of learning programming. Most people can overcome the challenges of learning through developing recommended methods and able to increase their level of competency in the area of programming

Implementation

The researchers enforced the implementation of the aforementioned proposed actions through active follow-up, checking their progress, and setting deadlines during the implementation, the researcher tried to work on improving the mindset of the student so that they can easily learn to program and they can be good at it.

Reflection/ Recommended Changes

Reflection in action research makes the teacher aware of the various forms of knowledge and action in education, of the tensions and contradictions between the list of recommended best methods and strategies, and of what is required to manage.

The implementation of the proposed action was not an easy task. Since the teacher teaches different courses and the students take variety of courses, time management was a big concern. However, the implementation was more or less successful. More than half of the students, among under performers, boost their self confidence in terms of computer programming. They had corrected the mentality and removed the myths that they had before.

Having the right mindset is the most important thing, additionally, providing resources for the learning environment is mandatory. The researchers found out that the more the students spend time on the, computer programming tasks, the better they become.

To attain the goal to be 100% successful, there are things that need to be changed. Such as lack of motivation on both the student and the teacher, and resource shortage kills the motivation and time. Additionally, the number of students in one section is very huge and it is very difficult to consultant to one, understand their behavior and assess them at an individual level. Having this in mind, that preparing working project code, practical lab manual, and other learning materials needs to be compiled and updated every time whenever the technology slightly changes.

Applying the methods of teaching that Practice for long period of time, learn though developing project, working in groups, Peer-to-peer evaluation, Boot camp, Self-thought, learn though developing projects with the teacher doing it, learn from Book, learn by reading other project code, and learn through playing game are very difficult to apply all of them on one specific semester. The time duration specified for a semester is not enough to use all the suggested methods. The researchers of this study believe that among a list of ten methods applying six of them are mandatory. The three teaching methods obliged to be considered with higher priority. These are learning through practice for long duration of hour, learn through developing project, and reading books. The remaining three methods can be chosen by the educator. These are either boot camp or doing a project and show the student start to end, either group work or peer evaluation, either learning through games or learning by reading a working project code. The educator can choose one of the options and at least 6 methods must be applied for successful delivery of the course and better understanding of the student. Our success measures are the skill of the

student, confidence in doing projects and analytical and problem-solving capability.

Finally, resource shortage in the laboratory is a very critical level which were 1 computer for up to 6 students. The number of students in a single section is very dense. Therefore, if these things get corrected then we can achieve what we call success.

CONCLUSIONS

In this study, we conducted action research on improving the skill set of computer programming courses for 3rd-year computer science students. We identified problems that were disorganized laboratory facilities, lack of motivation in the classroom, un-updated lecture notes focusing on theory rather than practical based, and the complex nature of the course which were the main challenges for having inadequate knowledge and required skill set. The nature of the course and the shortage of laboratory equipment were critical problems for the students. These problems were assessed by the investigators to provide best practices and appropriate solutions. They were a list of Problems that really existed as the respondent's response. The investigator recommended an action on the aforementioned problems. The recommended best practice to be taken as a measure was well-equipped laboratory facilities, changing the mindset of students about the complicated nature of computer programming course, and also the educator can choose at least 6 proposed methods that must be applied for the successful delivery of the course and for a better understanding of the course. Our success measures are the skill of the student needs to acquire, confidence in doing projects, analytical and problem-solving capability with real-world problems. The result of the action We have observed after we are implementing the proposed action, the student's confidence in programming boosted, their problem-solving capabilities also improved, and their enthusiasm to do projects increased. The researchers also suggest other programming educators to apply these techniques and proposed actions while they deliver the course.

REFERENCES

1. The action research guide book December 2019 https://www.vsointernational.org/sites/default/files/2020-04/vso-cambodia-action-research-gguidebook-english.pdf
2. Sarah Ultan Segal, ACTION RESEARCH IN MATHEMATICS EDUCATION A dissertation submitted in partial fulfillment of a Ph.D. in mathematics,2009
3. SiminGhavifekr, Wan Athirah Wan Rosdy, Teaching and learning with technology: effectiveness of ICT integration in schools, Volume 1, Issue 2, Summer 2015
4. L. Williams and R. Kessler, Pair Programming Illuminated: Addison-Wesley, 2003.
5. L. Williams, R.R. Kessler, W. Cunningham, and R. Jeffries, Strengthening the case for pair programming, IEEE Software, 17(4), July – Aug. 2000, pages 19 – 25, 2000
6. K. Beck, Extreme Programming Explained: Embrace Change: Addison-Wesley, 1999.
7. P. Abrahamsson, J. Warsta, M. T. Siponen, and J. Ronkainen, ―New Directions on Agile Methods: A Comparative Analysis, ‖ International Conference on Software Engineering, 2003.
8. Radermacher, A., Walia, G. "Improving Student Learning Outcomes with Pair Programming" Proceedings of the 8th International Conference on Computing Educational Research - ICER'2012. September 10-12, 2012 Auckland, New Zealand. pp. 87-92
9. Blumenstein, M. (2004). Experience in teaching object-oriented concepts to first year students with diverse backgrounds. Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04) [electronic proceedings].
10. Buck, D., &Stucki, D. (2001). JkarelRobot: A case study in supporting levels of cognitive development in the computer science curriculum. Proceedings of the SIGSCE Technical Symposium on Computer Science Education, 16-20.
11. Dunican, E. (2002). Making the analogy: Alternative delivery techniques for first year programming courses. In J. Kuljis, L. Baldwin & R. Scoble (Eds), Proceedings from the 14th Workshop of the Psychology of Programming Interest Group, Brunel University, June 2002, 89-99.
12. Miliszewska, I., Horwood, J., Tan, G., &Venables, A. (2004). Gender bias in computing? – Student perspectives. Proceedings of the Joint International Conference on Informatics and Research on Women in ICT (RWICT), Kuala Lumpur, Malaysia, 1135-1146.
13. Jenkins, T. (2002). On the Difficulty of Learning to Program. 3rd Annual Conference of the LTSN Centre for Information & Computer Sciences, Loughborough, LTSN-ICS.
14. Bloom, B. S... (1965) Taxonomy of Educational Objectives London: Longman.
15. https://literacy.kent.edu/Oasis/Pubs/0200-08.htmvisied 21/06/2022
16. Dijkstra, E. W.. . (1989) On the Cruelty of Really Teaching Computing Science. Comm. ACM 32: 1398-1404.
17. Syslo, M.M. Kwiatkowska, A.B. (2006). Contribution of informatics education to mathematics education in schools. In: Mittermeir, R.T. (Ed.), ISSEP 2006, LNCS, 4226, 209–219.
18. Szlávi, P. Zsakó, L. (2006). Programming versus application. In: Mittermeir, R.T. (Ed.), ISSEP 2006, LNCS 4226, 48–58.

