

Enhanced Programming Learning Model (EPLM) through Continuous Collaborative Coding (CCC) Practice

Pratibha S. Yalagi¹, Rashmi K. Dixit², Manisha A. Nirgude³

¹Department of Computer Science and Engineering, Walchand Institute of Technology, Solapur, Maharashtra, India.

²Department of Computer Science and Engineering, Walchand Institute of Technology, Solapur, Maharashtra, India.

³Department of Information Technology, Walchand Institute of Technology, Solapur, Maharashtra, India.

¹pratibhayalagi@gmail.com

²rashmirajivk@gmail.com

²nirgude.manisha@gmail.com

Abstract: As the world is moving towards perfection and competition, the requirements of industries have increased. To meet this industry needs conventional teaching learning model has to be enhanced. The real world problems in computing requires problem solving coders in fact, fast problem solver. The teaching learning process adapts the conventional approach of programming skills. But the industry requirements are much higher. It is much focussed on solving real world problem. The market job analysis says that most of the jobs and job hikes require proper coding skills and mastery in programming languages. Conventional teaching method for programming languages needs to be enhanced by practicing industry coding standards and adaption of collaborative learning. To reduce the gap between actual practice and the software industry circumstances and to develop a decisive programmer or coder this technique is useful. To develop programming skills, it needs analytical and rational approach. Hence, to grow the problem solving ability of a student not only effectively but also efficiently Enhanced Programming Learning Model (EPLM) through Continuous Collaborative Coding (CCC) practice is proposed.

The objective of this method is to enhance problem solving ability among the students and to meet the industry needs through proper coding skills. Through the collaborative coding, programming skills have improved among the students. This experiment is conducted for the pre-final year Computer Science Engineering Students for Java Programming Language course. The learner's engagement has improved effectively by this learning mode. The effect of the experiment is verified with Pre-test & Post-test before and after the EPLM through CCC session. Feedback of the students has also proved its effectiveness. It is observed that the collaborative learning skills have

improved. The benefits of collaborative learning include: Development of higher-level thinking, oral communication, self-management, and leadership skills too.

Keywords: Enhanced Learning Model, Continuous Collaborative Coding (CCC), Programming Skills, Problem Solving.

1. Introduction

Programming languages help us to solve the real time problems. They consist of syntax used to communicate with computers. Along with programming language one should also know to think synergistically. The real challenge in teaching programming language is to improve the problem solving ability with the proper application of programming concepts with proper logic. To solve the problem, proper analysis, design, depth of programming knowledge with rational thinking is required. The program writing skills with proper coding techniques plays important role.

For a teacher it is a challenge to inculcate the programming skills with proper logical, rational and development skills. Collaborative learning is one of the teaching methodologies that transforms that traditional lecture or teacher focused classroom into a student or learning centred room. Students work together to help each other understand content, solve problems or create projects and products with the instructor working as a moderator or facilitator.

Enhanced Programming Learning Model (EPLM) through Continuous Collaborative Coding (CCC) Practice focuses on all these aspects. It basically includes a Nonstop coding for 8 hours duration in a group for solving a given real world problem. A group of students have given a

problem to solve.

Java language is preferred for the implementation of the problem solution. Freedom is given to them for using internet resources, books and other resources. At the end of the day, an application in the form of a project with presentation and its report is generated in a group. It is assessed periodically through the panel of examiners. With the suggestions and guidance of the examiners the final product is developed. The student's collaborative learning skills have improved through higher-level thinking, oral communication, self-management, and leadership skills.

2. Literature Review

Basically programming languages are used to solve problems rather real world problems. However, problem solving is beyond the syntax of programming languages and scripts (Linn & Dalbey (1985) [1] and Perkins, Schwartz & Simmons (1988)) [2]. Different methodologies are implemented to teach programming language. There are various negative impacts of poor understanding of the problem solving methods through the programming logic and syntax. Various teaching approaches, models and tools have used to enhance the problem solving ability among the learners. According to Sarpong, Arthur and Amoako (2013) peer programming, peer tutoring and problem solving strategies have improved students' interest and knowledge which therefore will reflect positively on their performance. [3,4]. Ngo-Ye and Park (2014) proposed a teaching approach to orient students with the aspects of programming - Adopt LET US DO IT ALL TOGETHER approach. It has developed interest in students to learn programming [5]. Warren (2001) says, students should be started with Scripting languages such as Java Script. never with system programming languages such as Java or C++. As scripting languages are simpler and more flexible, students will learn easily [6]. Keefe, Sheard and Dick (2006) have focused on Extreme Programming (XP) practices to teaching OO programming to Students. They have proved that the students' programming skills have improved, however, their problem solving technique remains the same [7]. According to Pratibha (2016) the novice learners programming skills can be improved by effective program writing skills and also using debugger tools the bugs can be easily fixed and solved. [8,9]. So, various approaches have developed programming skills. Our focus is to bridge the gap between students learning for problem solving and the actual industry needs.

Now a day's Government of India has started the national level events like Smart India Hackathon for developing the products as collaborative coding efforts. In our EPLM model the Collaborative Coding practice is implemented to enhance the learning abilities among the students.

Problems addressed: It is observed during the conventional Teaching Learning method that,

1. Students are only learning the syntactical structure of the programming language.
2. Students can solve basic problems which are directly given as problem statements. It becomes difficult for them to solve real world problems.
3. Lack of coding practice.

To address the above problems Enhanced Programming Learning Model (EPLM) through Continuous Collaborative Coding (CCC) practice is proposed.

Collaborative learning enhances the engagement of a student in learning and increases interest as learner is actually working as a team member.

The objectives of this method are:

1. To enhance coding skills through collaborative learning
2. To increase the learners engagement in learning
3. To enhance real world problem solving ability

3. Importance of Java

Java is one of the most recommended languages even after decades. It holds strength within itself. Its features are remarkable and that is why it is recommended in today's era. Java has been encouraging the programming aspiration of many. The excellent language java is simple, secure, platform independent, object oriented, robust and portable. [12,15]

In our EPLM model, Java is used as a programming language for implementation. Java is a global programming language which focuses on almost all application development areas. It is used for developing web based applications, normal applications, networking applications, database applications and many more. [16]

4. Enhanced programming learning model (EPLM) through continuous collaborative coding (CCC) practice

The Enhanced Programming Learning Model (EPLM) is based on Continuous Collaborative Coding (CCC) Practice where, students are working in a group for solving a given real world problem statements in a given 8 hours period of time. It is based on collaboration and application of coding proper coding skills effectively. It is based on Think Together Work Together theory.

In collaborative learning students work in groups so as to understand a concept, create a product or solve a problem. Unlike individual learning, students engage with one another to ask for information, evaluate their ideas and monitor their work together. It benefits are:

1. Improves Problem-solving Skills
2. Boosts Critical Thinking
3. Enhances Social Interactions and Supports Diversity

4. Develops Self-management Skills
5. Establishes Oral Communication Skills
6. Fosters the Development of Interpersonal skills

The EPLM through CCC is applied on pre-final year students at the end of the semester. After learning the course Java Programming they have given the task to do in a period of 8 hours. It is a proctored programming coding competition.

The CCC model has basically follows the 10 steps which are shown in Figure 1.

Step 1: Favored group formation:

The students are given a freedom to form their own groups of 4-5 members for solving a problem. Various tasks to be performed are: analyzing a problem, designing, developing algorithm, testing, collaborating all modules and presenting. Based on these students have to form a group.

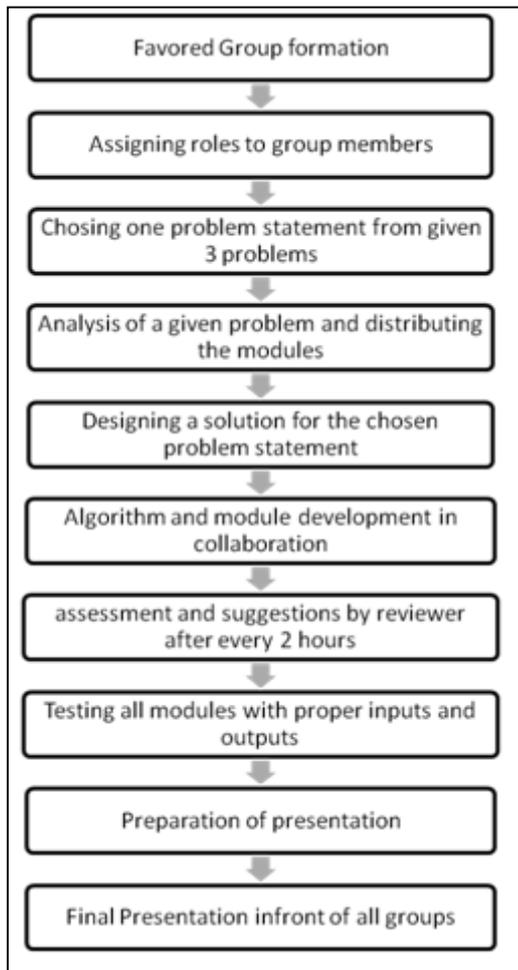


Figure 1: The steps of CCC model

Step 2: Assigning roles to group members:

Based on various tasks to be performed the leader of the group will assign roles to its members for solving the

problem.

Step 3: Choosing problem statement:

Each group is picking up a sheet of a paper which is containing 3 real world problem statements. From the three problem statements the group is choosing one problem for solving. All three problems are at a same difficult level.

Step 4: Analysis of a chosen problem and distributing the modules:

The group members are analyzing the problem and accordingly are designing various modules. Based on the modules, the work is distributed among the group members.

Step 5: Designing a solution for a chosen problem :

Module wise the problem is divided and each group member is working on his/her own module design.

Step 6: Algorithm and module implementation in collaboration:

Every module of the chosen problem solution is developed in collaboration with all group members. Better coding and optimization skills have applied for the implementation.

Step 7: Assessment and suggestions by the reviewers:

The panel of three reviewers is assigned for assessing the activity. After every 2 hours the panel is evaluating the status of each group. The panel is also guiding and suggesting the group members the updates and modifications required for the better product result. This activity iterates/repeats after every two hours of time duration. So, total 4 assessment rounds for 8 hours activity.

Step 8: Module testing:

After the complete implementation of the modules, all modules are combined and tested with proper inputs and outputs for validation.

Step 9: Preparation of presentation:

After solving the given problem, the final presentation is prepared for the visual and oral presentation.

Step 10: Final presentation:

After the implementation and testing, finally it is presented in front of all groups. Every group is presenting its solution for a chosen problem. It is evaluated by the examiners and reviewers.

Finally the best three groups have chosen for the Hackathon contest for the next level.

5. Implementation

This experiment aims to improve the problem solving ability of the students through effective collaborative coding. A continuous coding for 8 hours is implemented in a group. It has improved not only the programming

abilities of the learners but also the presentation and collaborative working skills.

1. *Sample Used :*

The experiment is conducted for pre-final year students of Computer Science and Engineering students. Total 65 students have participated in this activity forming @15 groups for the development of an application.

2. *Experiment Setup:*

For the pre-final year Computer Science and Engineering course named Java Programming, the EPLM through CCC practice is applied. After the completion of the Java Programming theory and lab practices the students have got knowledge of Java programming with its various features. At the end of the course this activity is implemented to enhance the problem solving ability. It's a one day activity of continuous 8 hours activity. Students are given a real world problem to solve in a group as a part of a competition.

3. *Pretest and Posttest:*

A small programming test is conducted on Virtual Programming Lab environment to test the students' learning ability and program writing abilities. One hour coding test is conducted with difficult level problem statements in Java platform.

Continuous Collaborative Coding (CCC) Competition is conducted in a day with a span of 8 hours with proctored and monitored rounds.

The sample problem statements for pretest is shown in the table 5.1

Programming Problem Statement
Magical Number Program

Print a number which is count of magical number in range of 1 to R. A magical number is a number whose **binary representation** has a pattern of '010' N times or greater than N.

Input:

The first line contains an integer K, the number of test cases.
The next K lines contain two space-separated integers, R and N.

Output:

Output K lines, the answer for each test case.

Constraints:

1<=K<=100
1<=R<=10^5
1<=N<=5

Explanation:

Input:

1 (only one test case)
10 1 (test case in which 10 is Range from 1 to 10) and 1 is one time occurrence of 010

Output:

1 (since from 1 to 10 of binary representation only 10 has binary value 1010 which contains 010 one time)

Table 5.1 Pretest sample problem statement

After the competition (implementation of EPLM through CCC practice) again the test is conducted as a post test. Difficult level problem statements are given to the same set of students who appeared for pre test.

The sample problem statements for posttest is shown in the table 5.2

Problem Statement for Programming Test
Hour Glass Number Magic

Given a 6X6 2D Array, :

```
-9 -9 -9 1 1 1
0 -9 0 4 3 2
-9 -9 -9 1 2 3
0 0 8 6 6 0
0 0 0 -2 0 0
0 0 1 2 4 0
```

We define an hourglass in to be a subset of values with indices falling in this pattern in 's graphical representation:

```
-9 -9 -9
  -9
-9 -9 -9
```

There are 16 hourglasses in array , and an hourglass sum is the sum of an hourglass' values. Calculate the hourglass sum for every hourglass in array, and then print the maximum hourglass sum.

For example, given the 2D array:

```
-9 -9 -9 1 1 1
0 -9 0 4 3 2
-9 -9 -9 1 2 3
0 0 8 6 6 0
0 0 0 -2 0 0
0 0 1 2 4 0
```

We calculate the following 16 hourglass values:

```
-63, -34, -9, 12,
-10, 0, 28, 23,
-27, -11, -2, 10,
9, 17, 25, 18
```

Our highest hourglass value is 28 from the hourglass:

```
0 4 3
  1
8 6 6
```

Table 5.2 Posttest sample problem statement

4. Results and Observations:

The experiment is conducted on 65 students of the class. Pretest and posttest results have shown the improvement in student’s problem solving ability.

It is observed that 30% results have improved among the students programming abilities enhanced as compared to pre test results. In pre test out of 65 students 31 have solved the problems 100% and in post test 48 student have solved the problems. Hence the results improved by 30%. It is shown in the table 5.3 and Figure 2.

Total number of students	65
--------------------------	----

No of students successfully completed pretest	31
No of students successfully completed posttest	48

Table 5.3 Pretest and Posttest result data

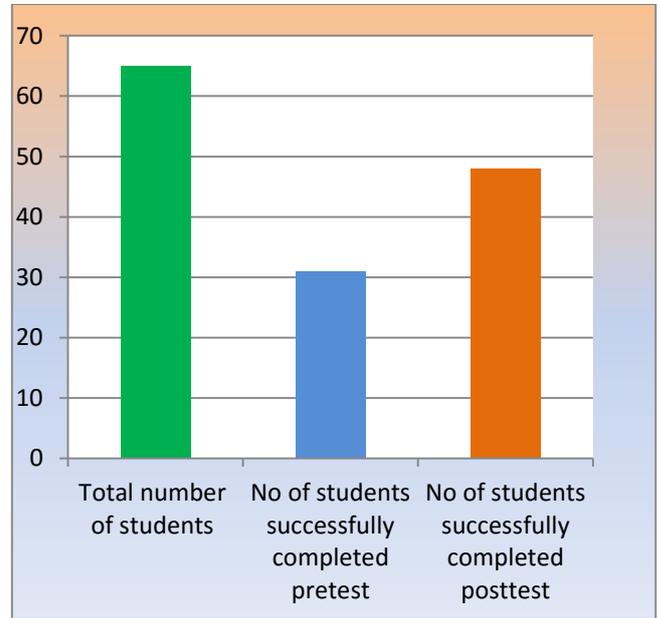


Figure 2. Pretest and Posttest results

6. Conclusion

Teaching programming is challenging. For a student, learning and inculcating logical, analytical & rational thinking is important. Literature shows that a lot of work has been done to improve this; however, it is evident that little effect of this work has had impact on the actual practice of teaching and learning of Software Development programming skills with. Therefore, this gap has been addressed in this EPLM model to enhance the teaching and learning process of programming to students. Continuous Collaborative Coding (CCC) encourages and engages students. There are challenges associated with collaborative learning environments, specifically creating and embracing a new environment and methodology for learning/instruction.

References

[1] Marcia C. Linn & John Dalbey, “Cognitive consequences of Programming Instruction: Instruction, Access, and Ability”, Journal Educational Psychologist Volume 20, 1985 - Issue 4, Pages 191-206, Published online: 08 Jun 2010

[2] Perkins, D. N., Schwartz, S., & Simmons, R. (1988). Instructional strategies for the problems of novice programmers. In R. E. Mayer (Ed.), Teaching and learning computer programming: Multiple research perspectives (pp. 153-178). Hillsdale, NJ, US: Lawrence Erlbaum Associates, Inc.

[3] Kofi Adu-Manu Sarpong, John Kingsley Arthur,

- Prince Yaw Owusu , "Causes of Failure of Students in Computer Programming Courses: The Teacher – Learner Perspective, International Journal of Computer Applications (0975 – 8887) Volume 77 – No.12, September 2013, PP 27-32
- [4] Kofi Adu-Manu Sarpong, John Kingsley Arthur, Prince Yaw Owusu , "Performance of Students in Computer Programming: Background, Field of Study and Learning Approach Paradigm", International Journal of Computer Applications (0975 – 8887) Volume 77 – No.12, September 2013 PP 17-21
- [5] Thomas Ngo-Ye , Sung-Hee Park, “Motivating business major students to learn computer programming – a case study”, Proceedings of the Southern Association for Information Systems Conference, Macon, GA, USA March 21st –22nd , 2014
- [6] Peter Warren, "Teaching programming using scripting languages", Journal of Computing Sciences in Colleges, Volume 17 Issue 2, December 2001 Pages 205-216 ACM DL
- [7] Karen Keefe, Judithe Sheard, Martin Dick, "Adopting XP Practices for Teaching Object Oriented Programming" 2006, Australian Computer Society, Inc.Eighth Australasian Computing Education Conference (ACE2006), Hobart, Tasmania, Australia, January 2006. Conferences in Research in Practice in Information Technology, Vol. 52.
- [8] Pratibha S. Yalagi, Trupti S. Indi, Manisha A. Nirgude, "Enhancing the Cognitive level of Novice Learners using Effective Program Writing Skills "2016 International Conference on Learning and Teaching in Computing and Engineering, IEEE Computer Society, PP 167-171
- [9] Trupti S. Indi, Pratibha S. Yalagi, Manisha A. Nirgude, "Use of Java Exception Stack Trace to Improve Bug Fixing Skills of Intermediate Java Learners " 2016 International Conference on Learning and Teaching in Computing and Engineering, IEEE Computer Society, PP 194-198
- [10] Gomes Anabela and Mendes António José: "An environment to improve programming education", International Conference on Computer Systems and Technologies - CompSysTech'07
- [11] Yang Tzu-Chi, Hwang Gwo-Jen, Yang Stephen J. H. and Hwang Gwo-Haur: "A Two- Tier Test-based Approach to Improving Student's Computer Programming Skills in a Web-Based Learning Environment", Educational Technology Society, 18(1), 198-210 (2015)
- [12] Shukun Liu, Zhen Chen and Jinpeng Tang: "The Improved Methods of Teaching Practice Based on C Language Programming", International Conference on Education Technology and Management Science (ICETMS 2013), pp 1361-1364
- [13] Robins Anthony, Rountree Janet, and Rountree Nathan: "Learning and Teaching Programming: A Review and Discussion", Computer Science Education, 2003, Vol. 13, No.2, pp. 137–172
- [14] Shehane Ronald and Sherman Steven: "Visual teaching model for introducing programming languages", Journal of Instructional Pedagogy Volume 14 – March, 2014
- [15] Said Hadjerrouit, "Java as first programming language: a critical evaluation", Newsletter ACM SIGCSE Bulletin Homepage archive Volume 30 Issue 2, June 1998 Pages 43-47
- [16] Douglas Kramer, "The Java™ Platform" A White Paper, 1996, Sun Microsystems