# Integration of Assessment and Learning Platform in a Traditional Class Room Based Programming Course

P Srilatha<sup>1</sup>, G Vishnu Murthy<sup>2</sup>, P Raja Sekhar Reddy<sup>3</sup>

<sup>1</sup>Department of CSE, Anurag Group of Institutions, Hyderabad, India

<sup>2</sup> Department of CSE, Anurag Group of Institutions, Hyderabad, India

<sup>3</sup> Department of CSE, Anurag Group of Institutions, Hyderabad, India

<sup>1</sup> srilathacse@cvsr.ac.in

<sup>2</sup>hodcse@cvsr.ac.in

<sup>3</sup> rajasekharreddycse@cvsr.ac.in

Abstract : Programming courses have become universally applicable body of knowledge and can be found in the curriculum of a wide range of disciplines. In the teaching and learning of programming courses, the traditional lecture based pedagogies and the experiential learning pedagogies were the primary approaches. In this study, a background review was conducted focusing on the experiential pedagogies used to teach and learn programming courses in engineering education. However, given the different backgrounds neither lecture based nor experiential learning was effective and they were combined in innovative ways to enhance the student learning. In this study, a platform blended with both assessment and learning was described. Furthermore, in the study, a series of analysis of covariance(ANCOVA's) were conducted to assess the effectiveness of the platform in promoting student learning by looking at the performance of the student in view of assimilation of vividly reflected knowledge through their performance in assessment tests and student satisfaction. The results shows that the integration of assessment and learning platform into a traditional classroom based programming course has improved the student programming skills as well as identified the areas in which the students can hone their programming skills.

**Keywords**: Learner Centered Approach, Critical Thinking Skills, E- Learning Tools

#### I Introduction

Now days to be successful in learning and in the work place, every student need to have the logical and problem solving skills in any introductory programming course. Research in computer science and engineering education has mentioned that novice students have not enough logical and program skills to solve the real world problems. Hence it is essential to examine the relation between the student logical and program skills to solving skills and their performance in assessment tests. This study helps to predict students performance and also helps the students to further develop such skills in a right direction. However in spite of research on factors that contribute to success in programming, the research reported here aims to find whether student with good problem solving skills in introductory programming course is relevant to performance of the student in assessment tasks. Our interest was motivated by the increasingly prevailing presence of first year students with varied problem solving skill and the need to develop various teaching learning strategies to make students actively engage in the class. Towards this objective, we address the following research questions.

**Research question 1:** How do students with Analytical and problem solving skills are related to student performance in formative assessment tests?

**Research question 2:** How do students with coding skills are related to students performance in summative assessment test?

**Hypothesis 1(H1):** There are significance differences in perceptions of analytical and problem solving skills in students taught with learner- centered approach those with non learner-centered approach.

**Hypothesis 2(H2)**: There are significance differences in perceptions of coding skills in students taught with learner- centered approach those with non-learnercentered approach.

**Hypothesis 3(H3)**: There are significance differences in perceptions of job satisfaction in students taught

with learner- centered approach those with non-learner-centered approach.

This paper is organized as follows. Section 2 focuses on review of studies based on problem based learning, its impact on student performance and its significance in relation to problem based learning in the introductory programming courses. Section 3 describes our research methodology. Section 4 presents results of the study. Finally section 5 presents our conclusion. The scope of the study is confined to selected assessment tasks. The effects of other casual factors on student performance is beyond the scope of the present study and will dealt with in the future study.

### **II Related Work**

Problem solving skill is a meta cognitive skill used to develop practical and creative solution to many real world problems. Every individual has his own critical and problem solving abilities that are acquired in various situations of daily life(Paideya 2011)(Dostál 2015)(Ozus et al. 2015). Several studies have emphasized the importance of problem based learning(Argaw et al. 2017)(Chi and Glaser 1983). In addition, the authors have mentioned problem solving is listed as one of the key skills to the study of STEM disciplines and needed skill for both novice of programming courses and qualified IT professions. Programming is a complex activity, especially for beginners to programming courses it requires certain cognitive skills as a prerequisite(Kappelman et al. 2016)(Rajaravivarma 2005)(Giordano and Maiorana 2014). Learning to program requires the learner to remember, understand the basic concepts of the programming language and general problem solving skills to analyze, implement and evaluate the problem(Kurland et al. 1986)(Kurland et al. 1986). Several studies have investigated how important to have problem based skill in learning programming and explored the integrated techniques and factors that promote problem based skills in introductory programming courses(O'Grady 2012)(Koulouri, Lauria, and Macredie 2014)(Chao 2016)(Psycharis and Kallia 2017). Despite of mixed results research suggests that students who have better cognitive skills outperforms better in solving the problems to find the solutions to the real world problems and learning how to program improves students coding skills(Guzin, Akar, and Altun 2017)(Trumbull and Lash 2013). However, most beginners to programming courses have difficulty in building the logic and implementing the code. Furthermore students with lack of problem solving skill have difficulty in applying key concepts such as loops and conditionals in programming(Koulouri, Lauria, and Macredie 2014).

To measure the student learning progress most of the educators use formative assessment in order to ascertain the level of understanding or the learning difficulties and recommends remedial methods to learning(Lawton improve student et al. 2012)(Heritage 2010)(Veerasamy et al. 2019). Prerequisite factors are important in predicting student performance(Longi 2016)(Sharma and Shen 2018)(Csizér and Magid 2014). However, study shows that there are no concrete valuable predictor to predict the student performance as they are varied from one context to another context in variety of ways ,including cultural settings and students cohort. Several studies have been conducted on the impact of problem solving skill on student performance in various courses including the programming reported that students with problem solving skill may help academicians to identify and analyze student study habits and attitudes that are important to academic performance though it is theoretically unrelated to academic achievements (Veerasamy et al. 2016)(Bubica and Boljat 2014). Similarly, the author reported there is a strong that there is a strong relationship between problem solving proficiency of sophomore mathematic students in a quantitative technology and their achievements in that course. In addition, several studies reported that gender differences does not have any impact on the learning outcome of the programming courses(Veerasamy et al. 2016)(Bubica and Boljat 2014). Specifically our independent sample t tests results confirmed that there is no gender discrimination in related to their performance in introductory course by adopting the problem based leaning.

Overall, our contribution is novel in that it focuses on finding how students with cognitive skills related to the student performance across the student learner centered approach those with non student learner centered approach and also focuses on the relationship among the cognitive skills.

## **III Methodology**

The aim of the study was to examine the relationship between students cognitive skills and their performance (Quiz) in formative and summative(practical's) assessment tasks in an introductory programming course across the student learner centered approach and student non learner centered approach. The following subsection presents

details about course description and data collection, elearning tool and assessment tasks.

A. Course Description and Data collection

The course on "Programming and Problem solving in C" comprises lecture hours, practical hours and tutorial hours using e-learning tool. Over 16 weeks of the semester, 13 weeks were utilized for delivering the course and 2 weeks were allocated for mid exams, final exam preparation and end exams. Therefore 52 hours were allocated for lectures, 39 hours were allocated for tutorial hours. One completed week was allocated for training on 'C' language. So in total 75 hours were allocated for tutorial hours. In addition, this study also collects the data of final year students who undergone the training in their third year for 1week(42 hours) using the e-learning tool.

# B. E-box: E-Learning Tool

Based on the evaluation of the students performance in various exams, we find out that the subjects need more practical approach. E-box tool has been used as a learner-centered approach there by transitioning from pedagogy driven learning ecosystem to practical oriented ecosystem. It provides custom learning environment that facilitates development of analytical and logical thinking skills of the student. It supports auto evaluation features and automatic grading. Course was designed in 2 parts. Analyzing part and the coding part. In analysis part, questions framed in such a way that they were composed of jumbling code, fill in the blacks and code tracing. In coding part students were supposed to implement the entire program for a given problem. In addition, online conducted exams were to measure student performance and skills.

C. Assessment tests

The assessment tests are comprised of 2 types. One is formative assessment and the other one is summative assessment. Formative assessment test comprising of MCQ's and the exercise types includes code tracing, filling missing parts of the code and jumbling of the code. Summative assessment test includes implementation of the program. The duration of formative assessment is for 30min and the duration of summative assessment is for 180 min. The maximum score for formative test is 30 marks and maximum score for summative score is 75 marks. The formative assessment(quiz) comprised of the three dimensions namely code tracing, finding the missing code, rearrange the jumbling code. Quiz is comprised of 30 questions. 10 questions were on code tracing, 10 questions on finding the missing code and 10 questions on rearrangement of the jumbling code. The skills are measured using a five point Likert type scale ranging from 1 to 5. The student who got 100% score ranked on the scale 5, in between 80% to 100% ranked on scale 4, in between 60% to 80% ranked on scale 3, in between 40 to 60% ranked on scale 2 and less than 40% ranked on scale 1. Similarly, Summative assessment task include 2 dimensions problem solving skill(logic building) and coding skill. The skills are measured with 2 values namely agree(logically valid) and disagree(logically invalid). The problem solving inventory is a questionnaire to measure an individual self confidence levels. The questionnaire consists of 4 parameters : Analytical Confidence - This parameter measures the student analytical abilities, Problem Solving Confidence- This parameter measures the student problem solving skills to solve the problems. Code Building confidence-This parameter measure the student coding skills to implement the solution to a given problem, Student learner centered approach- This parameter measures general tendency of the student to learn using elearning tools. Each dimension were measured using a five point Like rt scale ranging from 1 (strongly disagree) to 5 (strongly agree). In addition, job satisfaction parameter was also considered to measure the group of students got placed and they were ranked on the scale 1(placed) or 0(not placed).

# **IV Results and Discussions**

Data were entered and analyzed in SPPS version 23.0 and M plus 7.0. Two steps of data analysis were conducted to test our hypothesis. First, reliability analysis ,tested how well the measuring dimension for each of the assessment tasks and each of the problem solving inventory represented their underlying construct. Second, a series of analyses of Co-Variance(ANCOVA's) were conducted to examine the differences, if any , between the student performances in learning or practicing different levels of skills in variety of assessment tasks across the learner centered approach and student non learner centered approach. The adequacy of the model was evaluated based on Chi-Square statistic, а Standardized-Root-Mean-Square-

residual(S\_R\_M\_R)and a Root-Mean-Square-Errorof-Approximation(R\_M\_S\_E\_A).

D. Variable measurement

	<b>Coefficient</b> α Value
Formative	0.78
Assessment(FA)	
Code Tracing:	
Filling Missing Parts:	
Jumbling Code:	
Summative	0.80
Assessment(SA)	
Logic Building Skill:	
Coding Skill:	
Survey	0.75
Analytical Confidence:	
Problem Solving	
Confidence:	
Coding Confidence:	
Student learner Centered	
approach:	

# Table 1: Coefficient α values for variables used in the study

# Table 2: Analysis of covariance(ANCOVA)between Learner Centered Approach and non<br/>Learner Centered Approach

	Mean Differences	F	Significance value(p)	Partial
H1:				
Code	-0.86	4.91	0.000	0.067
Tracing				
Filling	-0.78	7.78	0.001	0.070
Missing				
Parts				
Jumbling	-0.90	6.93	0.015	0.042
Code				
H2:				
Logic	-0.36	5.62	0.012	0.37
building				
Skill				
Coding	-0.38	9.25	0.001	0.070
Skill				
H3:				
Job	0.8	12.38	0.008	0.026
Satisfaction				

Table 1 provides the coefficient score of the variables under study. The reliability scores of each of the variable is determined by coefficient alpha. The coefficient alpha values for FA, SA, survey and JS were 0.78, 0.80,0.75 and 0.86. All the values are above 0.7 value which indicates a high level of internal consistency and the dimensions in each variable are highly correlated. Table 2 provides the results of the ANCOVA test. The results of the ANCOVA for Hypothesis 1 was highly supported , indicating that students taught with the different pattern of questions and also by focusing on analytical and problem solving skills improves the students performances. In addition, the results of the ANCOVA for Hypothesis 2 was also highly supported, indicating that students taught with the diagrammatic view of conceptual questions and also by focusing on coding skills improves the students performances. The results of the ANCOVA for Hypothesis 3 was supported, indicates that there is a significant impact on the group of students who learnt the programming course in traditional way blended with the student learning approach in terms of placements than the students who learnt the programming course in a traditional way.

## V Conclusions, Limitations and Future Work

This study identifies the student performance who followed student learner centered approach using E-Box tool performs far better in formative assessment tests and summative assessment tests than the students who have not followed the learner centered approach. Therefore our results provide evidence that existing methods to teach introductory programming may need to change their way of teaching in order to improve the learning and critical thinking skills of the students. It can be concluded that measuring the performance of the students in coding skills can be useful in predicting the student performance in taking or participating in the competitive exams or in placements. This study has several limitations. Firstly, it was limited by relatively small group of students and the data were collected from only one course. Second, this study used self reported data to collect the student feedback which were often subjected to bias, leading to variability in getting the accurate data. Our study may be extended to identify the impact of problem based learning on student performance by examine the following skills: How to improve student performance when they learned collaboratively? . How to improve student performance by using project based learning? . How do students general problem solving skills differ from specific problem solving skills for learning programming?. Identify the nature and strength of the relationship among the cognitive skills.

#### References

Argaw, Aweke Shishigu, Beyene Bashu Haile, Beyene Tesfaw Ayalew, and Shiferaw Gadisa Kuma. 2017. "The Effect of Problem Based Learning (PBL) Instruction on Students' Motivation and Problem Solving Skills of Physics." Eurasia Journal of Mathematics, Science and Technology Education. https://doi.org/10.12973/eurasia.2017.00647a.

Bubica, Nikolina, and Ivica Boljat. 2014. "Predictors of Novices Programmers ' Performance." ICERI2014 Conference. https://doi.org/10.13140/2.1.2287.1845.

- Chao, Po Yao. 2016. "Exploring Students' Computational Practice, Design and Performance of Problem-Solving through a Visual Programming Environment." Computers and Education. https://doi.org/10.1016/j.compedu.2016.01.010.
- Chi, Michelene T.H. H., and Robert Glaser. 1983. "Problem Solving Abilities." *Human Abilities: An Information Processing Approach.*
- Csizér, Kata, and Michael Magid. 2014. The Impact of Self-Concept on Language Learning. The Impact of Self-Concept on Language Learning.
- Dostál, Jiří. 2015. "Theory of Problem Solving." Procedia - Social and Behavioral Sciences. https://doi.org/10.1016/j.sbspro.2015.01.970.
- Giordano, Daniela, and Francesco Maiorana. 2014. "Use of Cutting Edge Educational Tools for an Initial Programming Course." In *IEEE Global Engineering Education Conference, EDUCON*. https://doi.org/10.1109/EDUCON.2014.6826147.
- Guzin, Sacide, Mazman Akar, and Arif Altun. 2017. "Individual Differences in Learning Computer Programming: A Social Cognitive Approach." *Contemporary Educational Technology*.
- Heritage, Margaret. 2010. "Formative Assessment and Next-Generation Assessment Systems : Are We Losing an Opportunity ?" *Education*.
- Kappelman, Leon, Mary C. Jones, Vess Johnson, Ephraim R. McLean, and Kittipong Boonme. 2016. "Skills for Success at Different Stages of an IT Professional's Career." *Communications* of the ACM. https://doi.org/10.1145/2888391.
- Koulouri, Theodora, Stanislao Lauria, and Robert D. Macredie. 2014. "Teaching Introductory Programming: A Quantitative Evaluation of

Different Approaches." *ACM Transactions on Computing Education*. https://doi.org/10.1145/2662412.

- Kurland, D. Midian, Roy D. Pea, Catherine Clement, and Ronald Mawby. 1986. "A Study of the Development of Programming Ability and Thinking Skills in High School Students." *Journal of Educational Computing Research*. https://doi.org/10.2190/bkml-b1qv-kdn4-8ulh.
- Lawton, Daryl, Nancy Vye, John Bransford, Elizabeth Sanders, Michael Richey, David French, and Rick Stephens. 2012. "Online Learning Based on Essential Concepts and Formative Assessment." *Journal of Engineering Education* 101 (2): 244–87. https://doi.org/10.1002/j.2168-9830.2012.tb00050.x.
- Longi, Krista. 2016. "Exploring Factors That Affect Performance on Introductory Programming Courses." *Thesis*.
- O'Grady, Michael J. 2012. "Practical Problem-Based Learning in Computing Education." ACM Transactions on Computing Education. https://doi.org/10.1145/2275597.2275599.
- Ozus, E. Elhan, Mine Celikoz, Melek Tufan, and Filiz Erden. 2015. "Interpersonal Problem Solving Abilities of Students of Professional Education Faculty Dressing Programme of Selcuk University." *Procedia - Social and Behavioral Sciences.* https://doi.org/10.1016/j.sbspro.2015.04.827.
- Paideya, Vino. 2011. "Developing Critical Thinking Skills in First Year Chemistry through Supplemental Instruction Learning Spaces." In SEFI Annual Conference 2011.
- Psycharis, Sarantos, and Maria Kallia. 2017. "The Effects of Computer Programming on High School Students' Reasoning Skills and Mathematical Self-Efficacy and Problem Solving." *Instructional Science*. https://doi.org/10.1007/s11251-017-9421-5.
- Rajaravivarma, Rathika. 2005. "A Games-Based Approach for Teaching the Introductory Programming Course." ACM SIGCSE Bulletin. https://doi.org/10.1145/1113847.1113886.
- Sharma, Ritu, and Haifeng Shen. 2018. "Does Education Culture Influence Factors in Learning

Programming: A Comparative Study between Two Universities across Continents." International Journal of Learning, Teaching and Educational Research. https://doi.org/10.26803/ijlter.17.2.1.

- Trumbull, Elise, and Andrea Lash. 2013. "Understanding Formative Assessment: Insights from Learning Theory and Measurement Theory." WestEd: Improving Education through Research, Development and Service.
- Veerasamy, Ashok Kumar, Daryl D'Souza, Rolf Lindén, Erkki Kaila, Mikko-Jussi Laakso, and Tapio Salakoski. 2016. "The Impact of Lecture Attendance on Exams for Novice Programming Students." *International Journal of Modern Education and Computer Science*. https://doi.org/10.5815/ijmecs.2016.05.01.
- Veerasamy, Ashok Kumar, Daryl D'Souza, Rolf Lindén, and Mikko Jussi Laakso. 2019. "Relationship between Perceived Problem-Solving Skills and Academic Performance of Novice Learners in Introductory Programming Courses." *Journal of Computer Assisted Learning*. https://doi.org/10.1111/jcal.12326.