

Evaluating Learning Engagement Strategies in a Cyber Learning Environment during Introductory Computer Programming Courses – an Empirical Investigation

Mrs. Mourya Reddy Narasareddygari

I am Ph.D student at North Dakota State University. My research work is to see how different Learning strategies affect the student learning.

Dr. Gursimran Singh Walia

Gursimran S. Walia is an associate professor of Computer Science at North Dakota State University. His main research interests include empirical software engineering, software engineering education, human factors in software engineering, and software quality. He is a member of the IEEE Computer Society. Contact him at gursimran.walia@ndsu.edu

Mr. Alex Radermacher, North Dakota State University

Dr. Otto Borchert, North Dakota State University

Dr. Borchert received his Ph.D. in Computer Science from North Dakota State University in 2015 and has been teaching introductory computer science courses for three years (at Gonzaga and North Dakota State University). His research interests include immersive virtual environments for education, mobile app development, and STEM education through technology. He is currently teaching as a Lecturer at North Dakota State University.

Evaluating Learning Engagement Strategies in a Cyber Learning Environment during Introductory Computer Programming Courses-- An Empirical investigation

1. Introduction:

Introductory programming courses are known for their complexity and difficulty. They are generally accepted as a difficult courses to teach effectively and are similarly regarded as difficult for students, who struggle to learn all the required competencies and skills [1]. Achieving proficiency in programming related capabilities often requires improving aptitude in areas where students are unlikely to possess strong skills. These areas include critical thinking techniques, design strategies, and programming methodologies [2]. Because of this, students may regard programming courses as onerous or as possessing unmanageable work.

Combining those feelings about programming with the notion that many introductory programming courses are dull, uninteresting, or lacking engaging activities results in high dropout rate for these courses [4]. Some researchers have reported attrition rates in computer science are as high as 30 – 40%, with most students leaving after taking CS1 [8].

Much research and effort has gone into decreasing the high dropout rate of introductory programming courses [10]. In order to overcome these issues, instructors in such courses have tried many approaches such as using special development environments [5]. Another approach that has been used is increasing student engagement through intrinsic motivation, which empowers students and increases commitment [9]. A popular approach to intrinsic motivation involves using gamification and other learning strategies to improve student interaction, retention, and collaboration as they learn software programming concepts. These techniques are believed to help students improve their programming knowledge by teaching introductory programming to students in an interesting and motivational way and help in decreasing the attrition rate of the course [10,11].

To address these issues in our own introductory programming courses, we have developed SEP-CyLE – Software Engineering and Programming Cyber Learning Environment, (available at <https://stem-cyle.cis.fiu.edu/>) [3]. SEP-CyLE is a cyber-learning environment that uses assorted learning and engagement strategies including collaborative learning, problem based learning, gamification, and social interaction. The fusion of gamification, social interaction, and collaborative learning in SEP-CyLE helps students explore and discuss topics in software programming content, software tools, and a variety of other areas. The platform provides essential resources to students, allowing them to gain knowledge about programming concepts through the use of integrated Learning Objects, video tutorials, and key engagement features.

This paper describes the details and benefits of the platform and demonstrates how instructors can incorporate software engineering concepts taught in SEP-CyLE into introductory programming courses. Specifically, this paper evaluates the impact of different combinations of the key learning engagement strategies of SEP-CyLE on students' understanding of programming knowledge. This paper discusses a case study that was conducted across four different sections of an introductory computer science course (CS1).

The objective of this research was to evaluate how the different learning engagement strategies (when combined in various ways) improved the programming proficiency of students in an introductory programming class.

2. SEP-CyLE – Software Engineering and Programming Cyber Learning Environment:

SEP-CyLE was created to aid the academic needs of students and instructors in computer science and software engineering courses by giving access to a far reaching and advanced set of learning material on software programming concepts, methodologies and tools [3]. These learning materials include video tutorials and learning objects that cover various software programming and testing concepts, methodologies, and real-world tools. SEP-CyLE aims to improve the conceptual and practical knowledge of students in the area of software programming and testing. The key features of SEP-CyLE include the following:

Learning Objects (LO): Learning objects are small, easily digestible chunks of information. Each LO is divided into two parts: a content section and a quiz. The content section is typically presented as text, but can also contain images or embedded videos. The quizzes allows students to test their understanding of the concept covered by the LO. Instructors can also track student performance and their progress towards completion of the LOs.

Social Features: SEP-CyLE allows students to upload their profile pictures, edit details of their profile, and add comments on an in-platform discussion board. Students are also able to view and track the activity of other students who are enrolled in the same class. Students can also be enrolled in virtual teams and can pose questions to their team, other students in the class, and the larger SEP-CyLE community.

Gamification Features: The gamification design and mechanism has been incorporated in conjunction with the social features of SEP-CyLE. The key elements of the gamification mechanism of SEP-CyLE are reward points and a leader board.

Virtual Reward Points: Reward points are commonly used to enhance student motivation in an effort to improve long term behavior maintenance. In SEP-CyLE, students are given virtual points for completing a learning object / quiz, posting a comment in discussion boards / forums, or other activities within the system.

Leader Board: The leader board in SEP-CyLE provides information about the number of virtual points for all of the students in the course. This can harness the competitive nature of students and motivate them to access additional content or to participate in the various system activities which reward them with virtual points. This can be seen in the Figure 1.

Learning Object Assignment(s)		Tutorial Assignment(s)
First Assignment Description: Lorem Ipsum Start Date: 09-21-2016 @ 05:17PM -0500 End Date: 05-31-2017 @ 10:59PM -0500 Minimum Passing Score: 80 Allowed Attempts: 3 Grading Scheme: maximum score Go to Assignment	Third Assignment Description: Lorem Ipsum Start Date: 09-21-2016 @ 05:17PM -0500 End Date: 05-15-2017 @ 10:59PM -0500 Minimum Passing Score: 80 Allowed Attempts: 3 Grading Scheme: maximum score Go to Assignment	JUnit Description: This is about JUnit Start Date: 05-28-2017 @ 11:00PM -0500 End Date: 05-30-2017 @ 10:59PM -0500 Go to Assignment
My Course Activity View Full List View Report <div> Student One First Name Student One Last Name posted a new comment in thread: Polymorphism 3 05/21/2017 @ 10:12 -0500 </div> <div> Student One First Name Student One Last Name successfully completed the quiz assessment for assignment: Demo LO 99 05/21/2017 @ 10:10 -0500 </div>	Course Activity View Full List <div> Student One First Name Student One Last Name posted a new comment in thread: Polymorphism 3 05/21/2017 @ 10:12 -0500 </div> <div> Student One First Name Student One Last Name successfully completed the quiz assessment for assignment: Demo LO 99 05/21/2017 @ 10:10 -0500 </div>	Course Leader Board View Full List <div> Student One Last Name, Student One First Name 300 </div> <div> Student One Three Name, Student One Three Name 0 </div> <div> Student 5 Last Name, Student 5 First Name 0 </div> Course 1 Team 1 <div> Student One Last Name, Student One First Name </div> <div> Student Two Last Name, Student Two First Name </div>

Figure 1: Student Home Page

3. Study Design:

The study was designed to investigate the effect of different learning engagement strategies (LEs) (e.g., Social Interaction – SI; Gamification – G; and Collaborative Learning – CL) of SEP-CyLE on undergraduate students’ acquisition of general software programming and testing concepts, knowledge of programming and testing techniques, and mastery of the programming and testing tool utilization in an introductory programming course at North Dakota State University. The study was conducted across four different sections of an introductory programming course (CS1) taught by two different instructors. Each section represented a different experimental condition. The study utilized a pre- and posttest instrument to measure the impact of using SEP-CyLE with different combinations of LEs on student learning outcomes when using SEP-CyLE.

The pretest was conducted at the beginning of the semester prior to the introduction of SEP-CyLE. Students then progressed through the course as normal, but were periodically (approximately once per week) assigned a SEP-CyLE LO that was related to the material being covered in the course over that week. At the end of the semester, students were tested again using a posttest instrument which was identical to the pretest. We also conducted a user survey to see if students felt as though using SEP-CyLE improved their success on assignments or understanding of course concepts. This user survey also provided us with feedback regarding student use of SEP-CyLE and their eagerness to use it in future programming courses. Details of the study are provided below:

a. *Study Goals:* This study has two principal objectives. The primary objective is to analyze the effects resulting from a fusion of the different learning engagement strategies of SEP-CyLE on student learning. More formally, it is expressed as follows:

Investigate the effect of learning engagement strategies included in SEP-CyLE on students' acquisition of software programming and testing conceptual knowledge, tools and techniques in an introductory computer programming course.

The second objective was to evaluate the overall satisfaction of the features of SEP-CyLE and its usability in a programming course and to improve its future usage.

b. *Participating Subjects:* The case study was conducted across four different sections of an introductory programming course (CSCI 160 at North Dakota State University), taught by two different instructors (each instructor taught two sections using the same lecture material, assignments, exams, and schedule). A total of 145 students were enrolled across all four sections of the course. Of those students 68 elected to participate in the study.

c. *Study Procedure:* The study procedure included five major steps:

Step 1 – Pretest: A pretest was conducted at the beginning of the semester (prior to introducing SEP-CyLE) to assess students' baseline knowledge of programming concepts.

Step 2 – Introducing SEP-CyLE: The students were then trained on how to use SEP-CyLE. They were taught how to browse video tutorials and learning objects and track their performance, how to change their profile pictures and interact with their peers in the cyber learning environment, and how to post to the discussion forums. Each section was trained separately depending upon the LEs that were enabled for them. The following combination of LEs were used in the four sections. Section 1 had everything enabled (E); Section 2 had everything except social interaction (E-SI); Section 3 had everything except collaborative learning (E-CL); Section 4 had everything except gamification (E-G)

Step 3 – Assigning LO's to Students: Every week students were assigned a LO to be completed as part of their course. The LO's that are used in this case study are listed in Table 1. These LO's had been previously created by the SEP-CyLE development team or were developed by researchers specifically for use in this course.

Step 4 – Posttest: At the end of the semester, the students were retested on the knowledge of their proficiency on software programming and testing concepts, methodologies and tools using the posttest instrument which has the same set of questions as that of the pretest instrument. This was done for all four sections to understand the influence of different LEs on improvement in their conceptual knowledge. We also evaluated students' course performance (end-of-semester grades) to understand the impact SEP-CyLE had on their course performance.

Step 5 – Survey: At the end of the study, students were asked to fill out a survey to evaluate the usability and usefulness of SEP-CyLE in an introductory programming course. We also collected feedback on improving SEP-CyLE for future courses.

Table 1: List of LO's used in case study

LO Name	Description	Week #
Introduction to NetBeans	An introduction to the NetBeans IDE and some of its basic functionality	1
Advanced NetBeans	A look at some of NetBeans's advanced features and functionality	2
Using Subversion with NetBeans	A coverage of basic SVN commands including conflict resolution	3
Introduction to Equivalence Classes	An introduction to the concept of equivalence classes and how they relate to unit testing.	4
Equivalence Classes	An explanation of creating equivalence classes for a Java method	5
Equivalence Classes for Methods with Multiple Parameters.	An explanation of the process for creating equivalence classes for methods with more than one parameter	6
Using Debuggers	An introduction to using the debugger in NetBeans	7
White Box Testing	An overview of white box testing	8
Advanced Debugging	An overview of advanced techniques for using NetBeans debugger.	9
LA Array LO	A learning object related to arrays.	10
SOLID Design Principles	A learning object related to good software design principles	11

4. Data Collection and Evaluation Criteria:

The primary data source are student responses to the pre- and posttests described above. The students responded to the questions using a 5-point Likert scale, where 1 = Strongly Disagree, 2 = Disagree, 3 = Neither Agree nor Disagree, 4 = Agree and 5 = Strongly Agree. This Likert scale data has been treated as interval scales. We also collected SEP-CyLE data for each student in terms of number of LO's attempted, number of LO's passed, time spent on SEP-CyLE, and number of virtual reward points earned. In addition, we collected the student's end-of-semester grade to correlate against SEP-CyLE metrics

5. Data Analysis and Results:

We compared the pre- vs. post-test scores of each section, to assess the impact of different LEs on student acquisition of software programming concepts. The mean of student pretest vs. posttest scores for all the four sections are shown in Figure 3:

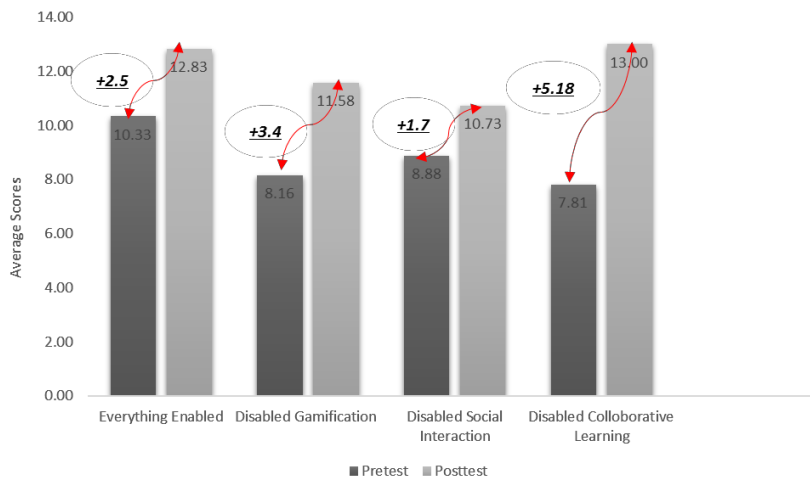


Figure 3: Pretest vs Posttest scores across all sections

From the above results, there was an increase in post-test scores for all sections, but the largest increase was for the section that had everything enabled except collaborative learning. This is consistent with our earlier results that have shown that collaborative learning (or team based learning) in SEP-CyLE did not have any positive contribution to improvement in students' understanding of programming concepts. In other words, Gamification (G) and Social interaction (SI) played the larger roles in influencing the student's engagement and learning in an introductory computer programming course.

Next, we evaluated the relationship between the different variables to evaluate the impact of SEP-CyLE usage metrics on the student's final grade in different sections. The independent variables are listed below and the dependent variable was end-of-semester course grades:

1. Number of LO's Attempted in Sep-CyLE.
2. Number of LO's passed (i.e., completed with at least 80% questions correctly answered).
3. Total time spent on SEP-CyLE.
4. Virtual Points earned.

Table 2 provides the correlation coefficient (strength of correlation) and p-value (statistical significance) between the above 4 variables and students' course grades for all four sections (E; E-G; E-SI; and E-CL. The major result is that, when everything is enabled the usage of SEP- CyLE had a strong positive correlation (significant at p-value <0.1) with the course performance. This was expected as the combination of all different learning engagement strategies can positively contribute to the overall success of students in an introductory computer programming course.

Table 2: Relationship between SEP-CyLE Usage Metrics vs. Course Grade

	E	E-G [without gamification]	E-SI [without social interaction]	E-CL [without collaboration]
# of LO's attempted vs. Course grade	Coeff. = 0.762 P-value = 0.07	Coeff.= 0.14 P-value = 0.55	Coeff. = 0.306 P-value = 0.13	Coeff.= 0.013 P-value = 0.5
# of LO's Passed vs. Course grade	Coeff. = 0.699 P-value = 0.12	Coeff.= 0.145 P-value = 0.55	Coeff. = 0.303 P-value = 0.13	Coeff.= 0.013 P-value = 0.96
Time Spent vs. Course grade	Coeff. = 0.78 P-value = 0.06	Coeff.= 0.418 P-value = 0.4	Coeff. = 0.32 P-value = 0.113	Coeff.= -0.24 P-value = 0.36
Virtual points earned vs. Course grade	Coeff. = 0.792 P-value = 0.06	Coeff.= 0.5 P-value = 0.8	Coeff. = 0.326 P-value = 0.104	Coeff.= -0.09 P-value = 0.73

6. Discussion of Results:

The results from this study indicate the promise of using SEP-CyLE with gamification and social interaction learning strategies included in order to provide better results in teaching software programming and testing concepts for introductory computer programming courses. Student comprehension of testing ideas, their insight into programming and testing techniques and their expertise of tool usage showed a significant improvement when a combination of gamification and social interaction learning strategies are enabled in SEP-CyLE.

In terms of student feedback, there was positive feedback of SEP-CyLE in terms of overall satisfaction with the website, its ease and clarity of information as shown in Figure 6. Remember that students rated their responses on a 5-point Likert scale [1= Strongly Disagree; 2= Disagree; 3= Neither Agree nor Disagree; 4= Agree; 5= Strongly Agree].

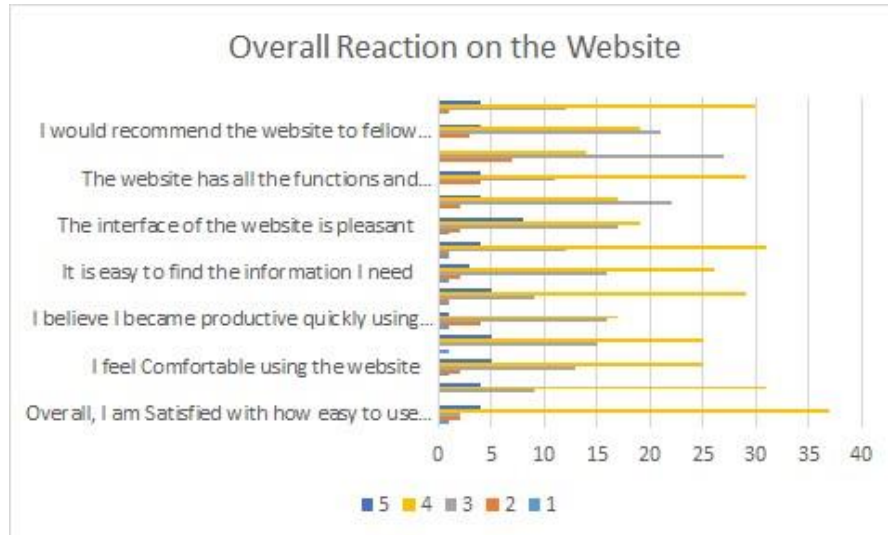


Figure 4: Survey results of Overall Reaction on the Website

7. Concluding Remarks:

The results presented in this paper demonstrate that there is a meaningful relationship between the gamification and social interaction learning strategies used in SEP-CyLE. Additionally, we found that the collaborative learning strategy did not have much influence on the students which was an unexpected result. When the virtual points were analyzed, it was found that the students were more productive at completing LO's and those students who earned more virtual points received better grades. While gamification produced positive results, we are still investigating additional elements that could be of further use in SEP-CyLE. We will conduct additional studies to measure the impact of additional gamification elements on student performance and learning while using SEP-CyLE. Also, in future studies we will use SEP-CyLE in advanced level programming courses and evaluate how these different learning strategies improve advanced understanding of software programming and testing techniques.

8. Acknowledgements:

This work is supported in part by the National Science Foundation under grants DUE-1225742 and DUE-1525112. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

9. References:

- [1] I. F. de Kereki, "Scratch: Applications in Computer Science 1", in *Frontiers in Education Conference*, Saratoga, Springs, NY, USA, pp. 7-11, 2008.
- [2] A. Robins, J. Rountree, N. Rountree, "Learning and Teaching Programming: A Review and Discussion", *Computer Science Education*, 13(2), Swets & Zeitlinger, pp. 137-172, 2003.
- [3] R. Chang-lau and P. J. Clarke. Software engineering and programming cyberlearning environment (SEP-CyLE), 2018. <https://stem-cyle.cis.fiu.edu/instances> (retrieved Jan. 2018).
- [4] H.-T., Lin, T.-H., Kuo, "Teaching programming technique with edutainment robot construction", in *2nd International Conference on Education Technology and Computer (ICETC)*, Shanghai, China, pp. 226-229, 2010.
- [5] M. Kölling, "Teaching Java with BlueJ – A Sequence of Assignments." University of Southern Denmark Technical Report No 1, September 2002, ISSN No. 1601-4219.
- [6] Q.H. Mahmoud, "Integrating Mobile Devices into the Computer Science Curriculum." *Proceedings of the Frontiers in Education Conference (FIE 2008)*, Saratoga Springs, NY, USA, pp. S3E-17- S3E-22.
- [7] A. Vihavainen, J. Airaksinen, C.Watson, "A systematic review of approaches for teaching introductory programming and their influence on success", *Proceedings of the tenth annual conference on International computing education research*, August 11-13, 2014, Glasgow, Scotland, United Kingdom.
- [8] T. Beaubouef, and J. Mason, 2005. "Why the high attrition rate for computer science students: some thoughts and observations". *SIGCSE Bulletin* 37, 2 (Jun. 2005), 103-106.
- [9] M. T. Wang, & J.S. Eccles, (2013). "School context, achievement motivation, and academic engagement: A longitudinal study of school engagement using a multidimensional perspective". *Learning and Instruction*, 28, 12-23.
- [10] S. Kurkovsky, "Engaging students through mobile game development", *Proceedings of the 40th ACM technical symposium on Computer science education*, March 04-07, 2009, Chattanooga, TN, USA [doi>10.1145/1508865.1508881].
- [11] C.S.Gardiner, "Improving the student success and retention of under achiever students in introductory computer science", *Journal of Computing Sciences in Colleges*, v.26 n.6, p.16-22, June 2011.