E-LEARNING STUDENT ASSISTANCE MODEL FOR THE FIRST COMPUTER PROGRAMMING COURSE

Noman Hasany

Department of Computer Science, Qassim University, Qassim, KSA

ABSTRACT

E-Learning applied to computer programming course design is a promising area of research. The student having clear understanding of the programming constructs can apply it to solve various problems. Because of limited time and availability, the instructor can go back to some extent to cover the weaknesses of their students that hinder the understanding of the problems. As more lessons are covered, the weak students become weaker in programming. To cope up with these problems an e-learning system is devised which the student can use anywhere and at any time as a web application. It comprises of both tutoring and assessment and also provides guiding the students to error correction using back-tracking technique to refine the concepts and reattempt the programming problem.

KEYWORDS

E-learning, computer based learning, first programming language, novice programmers

1. INTRODUCTION

Several universities form help desks for the students to understand computer programming in addition to assigned labs hours. This highlights the need of providing them an electronic 24-hours helpdesk in order to cope up with the difficulties of novice learners of a programming course. The past studies show low passing rates in programming courses especially when this is the first programming course [5]. The necessity of designing e-learning course to help novice learners is due to the dynamic nature of computer programming problems [2]. One of the main problems in this regard is the limited time of the instructor given for theory and lab classes to cover the syllabus [1]. The instructors are assigned limited hours to cover a syllabus for a dynamic nature problem of computer programming to novice learners.

An e-learning system can assist students which they can use anytime and without getting confuse in the class to ask simple questions. The objectives of the study includes analyzing the difficulties of students for the first computer programming course, constructing an e-learning model that helps students in understand the programming and developing a prototype for the proposed e-learning framework and assess it on real students. Our e-learning prototype, c-Learn, is easy to use for new students who never learned any programing languages before. It consists of tutoring and assessment system and a back tracking method that guides the learner to the topic which is necessary to understand in solving the current problem in which the learner is making the mistake.

2. RELATED WORK

Several approaches show that there is an extra working required in attaining attention of students in a programming course [9]. The students feel difficulty in programming and finding solutions to new programming tasks because of the dynamic nature of the usage of computer programming constructs which makes it more difficult [2].

Some recent works on programming courses with respect to e-learning have been reported. Some uses project based learning technique in which projects are assigned, guided and evaluated to teach high-level programming skills to students [9]. This type of project based learning is useful for long-term evaluation but for short term semester-based teaching and evaluation of course

DOI :10.5121/ijite.2017.6101

contents understanding, as in our case, can be analyzed for short term guidance and assessment. Also, [9] highlighted different levels to be achieved to complete the project. In our scenario, the course dependencies make it difficult to learn various courses at the same time to acquire knowledge to complete a project.

Another recent system by [6] has more emphasis on the evaluation of answers rather than design features to assist students in the course. Most of the students passing the programming course with baseline found it difficult to take additional programming course [10]. So, there is a need of assessment at each level of course regarding the students programming skills required by that particular course.

A system is presented in [8] that can bridge up the gap between theory lectures and labs using multimedia animations of the topics. The strategy to course design and evaluations are not discussed. A system is suggested in [7] with more emphasis on evaluation strategy for the students who already have attended the java course lectures. The research discusses the type of questions the user like to answer and the success rate associated with the answers.

Automatic assessment for programming languages is very important as many program features may be overlooked during manual assessment but it is done with entirety in a tool based assessment comprising the compiler also [4]. We have also followed the idea of automatic testing in our c-Learn prototype by incorporating and transmitting to the learners, the compiler error messages.

It is suggested that many program examples can enhance the learning process of novice learners in a programming language course [10]. A system for programming is used to get the learning styles used by students [11]. In our system, we have provided a lot of examples and exercises to the student as well as we have a focus to apply a specific technique to backtrack the student to learn the error made and reattempt the question to achieve the learning.

3. RESEARCH METHODOLOGY

Our work is based on an experimental study. For this purpose, we have waited for the semester to enter in the phase of first midterm exam which often comes after 6 weeks of teaching. The students selected for the experiment have completed the pre-midterm regular teaching with lectures and labs. After the midterm examination, we selected 11 students mostly with average and below average marks. For the experimental purpose, we developed an e-learning prototype names it as c-Learn. This c-Learn prototype, has three major phases with an emphasis on the back-track phase. In our approach, we will back-track a programming problem the student is attempting and making mistakes in it, to the problem which he needs to know it first in order to complete the task. At any point when the student understands the problem, he has no need to go back further. This type of eLearning system will cope-up with the time limitation of the theory/lab instructor and cover up the problem of shy students that want to ask related simple questions during the class to understand the programming problem. The selected students are then passed through a phase of tutoring and learning the c-Learn. They are given two and a half hours to use the e-learning prototype and then the marks from the last chapter is collected. We then compare the marks obtained in the midterm exam and the marks which the students obtained by using the c-Learn e-learning prototype. We have cumulatively compared the leaning improvement of students and also the standard deviation to observe the difference in the learning behaviour among various students. Here we introduce the different section of our e-learning prototype c-Learn that is used for the evaluation of the students.

3.1. C-LEARN MODEL

The c-Learn model consists of three major phases including tutoring, assessment and backtracking, as shown in figure 1. The methodology includes selecting the students that appeared in the midterm-1 examination with conventional tutoring comprising of lectures and lab work and then retest them again after being trained by the c-Learn prototype. To keep c-Learn as a complete e-learning solution for the students, tutoring portion is also included. The benefit of tutoring section is two-fold. As a first advantage, the students can learn various concepts through reading and attempting some intuitive questions. Second, in case of failure in a chapter or lesson, the learner is backtracked to the lesson where he can understand the mistake he made. The next important part is the assessment section, which can be attempted by the student at any time. The assessment consists of multiple choice questions and interactive program tests. These program tests comprises of questions to write the output of the program, or to correct the syntax or logical errors of the program. Following the assessment, either the student is directed to the next chapter or backtracked to the last one in case of marks less than 70%. This decision of redirection to the previous content is done by the backtracking phase.

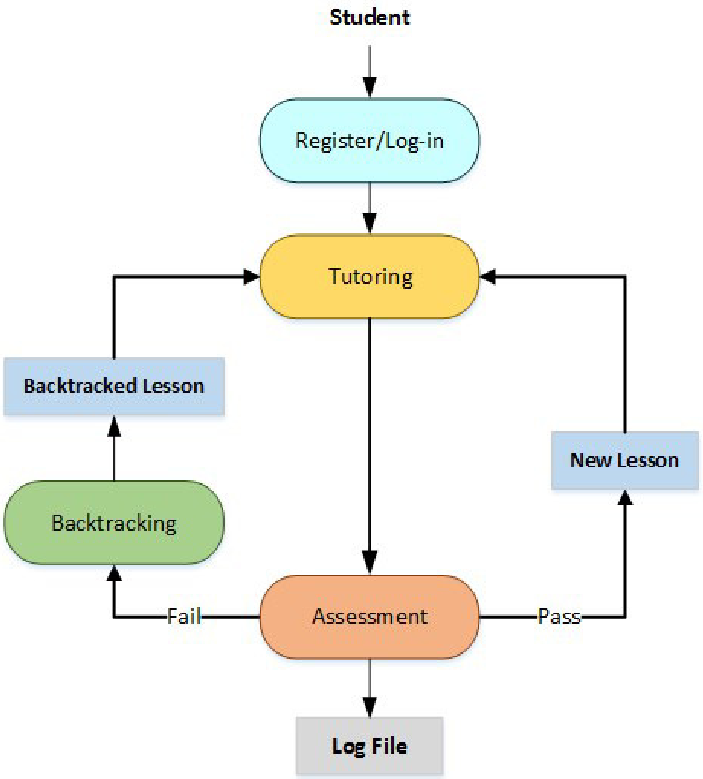
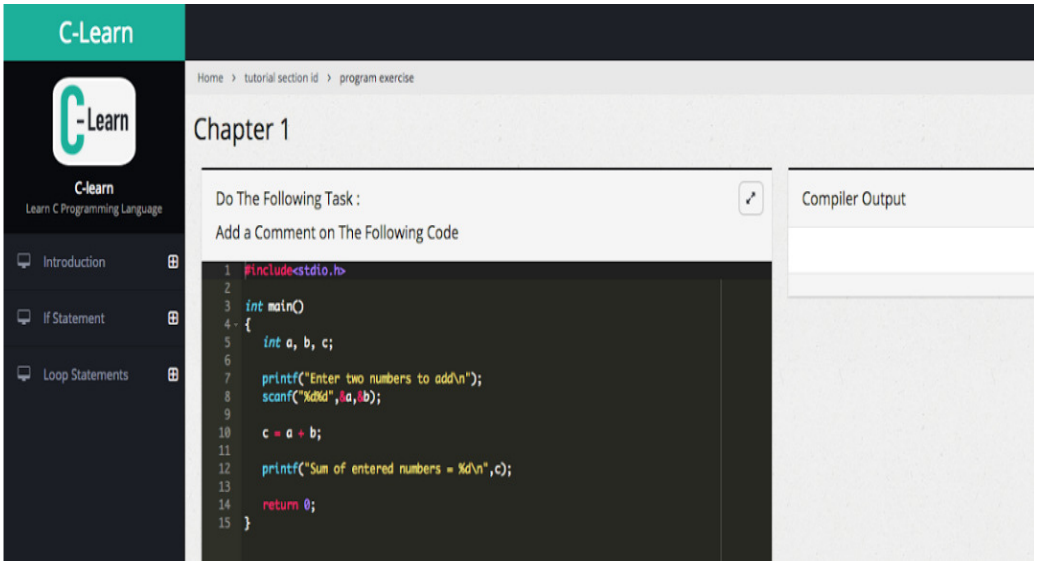


Figure 1. c-Learn Model

3.1.1. Tutoring

In our e-learning program, the student cannot adapt to a new lesson or chapter until he passed the previous lesson. Lesson is considered to be a learning unit in a C-programming course, for example, variable declaration is considered to be a lesson or topic in a programming language course. The main page of c-Learn is the dashboard where student can see his current status while learning through the system. It shows the sections of the course attempted by the student along with the number of errors made in each section. Following the home page is the tutorial page in our c-Learn web application. The main design of tutorial page that can facilitate easy learning for the student is the colouring used for the C language syntax as shown in the fig.2. Each section of the tutorial page is designed to have a brief introduction of the topic and then sample programs with coloured syntactic words. The students' tutorial section is made interactive where the students can write the program and see the output. These type of interactive learning is also supported by [3].


C-Learn
C-Learn
C-learn
Learn C Programming Language
Introduction
If Statement
Loop Statements
Home › tutorial section id › program exercise

Chapter 1

Do The Following Task :

Add a Comment on The Following Code

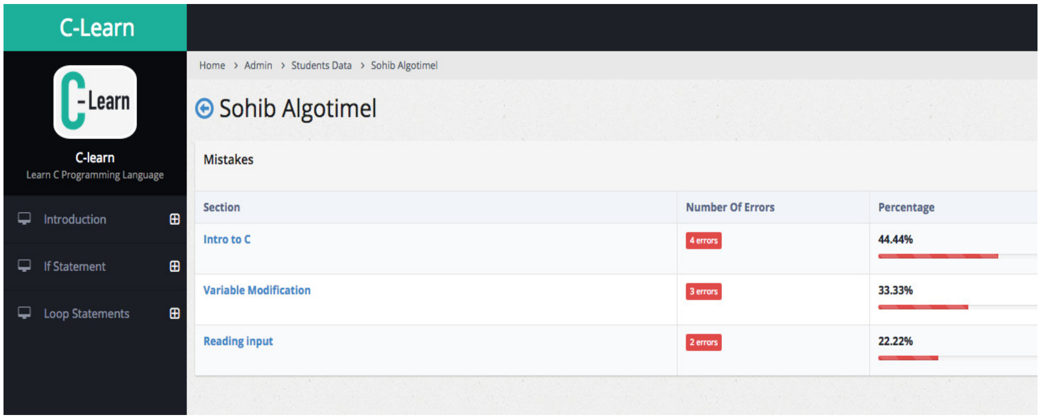
```
#include<stdio.h>

int main()
{
   int a, b, c;

   printf("Enter two numbers to add\n");
   scanf("%d%d",&a,&b);

   c = a + b;

   printf("Sum of entered numbers = %d\n",c);

   return 0;
}
```

Compiler Output

Figure 2. Tutorial Section

3.1.2. Assessment

The assessment is performed by given a time of two and a half hours, after that the results of the students are computed automatically by the prototype. The assessment questions are not only comprised of multiple choice questions but also programs for which the students have to remove either some logical error, complier error or to determine the output. The C language compiler is incorporated in the prototype to analyse the errors of the user amendments in the program. The final assessment is done based on the marks obtained in the last chapter of arrays and loops. It was clearly defined that the student cannot attempt the next chapter until he finished the previous one. In order to attempt the array and loops chapter, it is understood that the student has passed the conditional statements chapter.


C-Learn
C-Learn
C-learn
Learn C Programming Language
Introduction
If Statement
Loop Statements
Home › Admin › Students Data › Sohib Algotimel

Sohib Algotimel

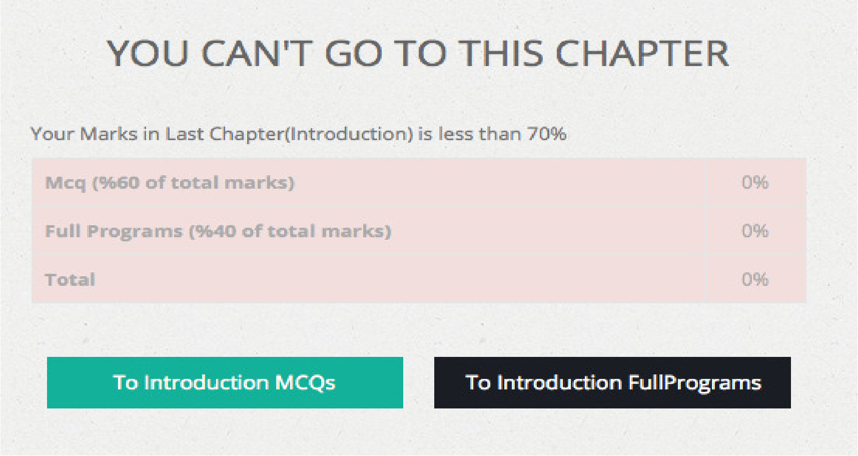
Mistakes

Section	Number Of Errors	Percentage
Intro to C	4 errors	44.44%
Variable Modification	3 errors	33.33%
Reading input	2 errors	22.22%

Figure 3. Results from Assessment Section

3.1.3. Backtracking

The log file generated from the assessment phase also contains the learning sequence of the students with all the details of when an exam is attempted or re-attempted, what type of mistakes have been done, and in how many turns the student is directed to the next lesson. If the score obtained by the student is less than 70% then the student is backtracked to the topic which is the previous in the sequence. The screen that hinders the user to pass to the next chapter in case of less threshold is shown in fig 4.



YOU CAN'T GO TO THIS CHAPTER

Your Marks in Last Chapter(Introduction) is less than 70%

Mcq (%60 of total marks)	0%
Full Programs (%40 of total marks)	0%
Total	0%

Figure 4. Message screen showing result of student having less threshold

4. Results of Research

The c-Learn project is applied to the novice programmers who get less marks in the first Midterm exam of the introductory programming course. The marks are obtained from the midterm1 assessment and are used as before c-Learn training score. In the second stage, these students are provided with a training of how to use the c-Learn prototype. Following the training, the students are advised to pass through a two and a half hours learn and assessment phase and stop to the lesson of arrays and loops. The score obtained in the last test, comprising of material from the first to the last lesson is considered as the score after using the c-Learn prototype. The improvement of the students in the score is shown in the figure 5. In order to move to the next chapter, the student has to acquire at least 70 percent marks in the current chapter. If the marks are less than the criteria, the student is backtracked to the topics that need to be known by the student in order to avoid mistakes identified in the evaluation process.

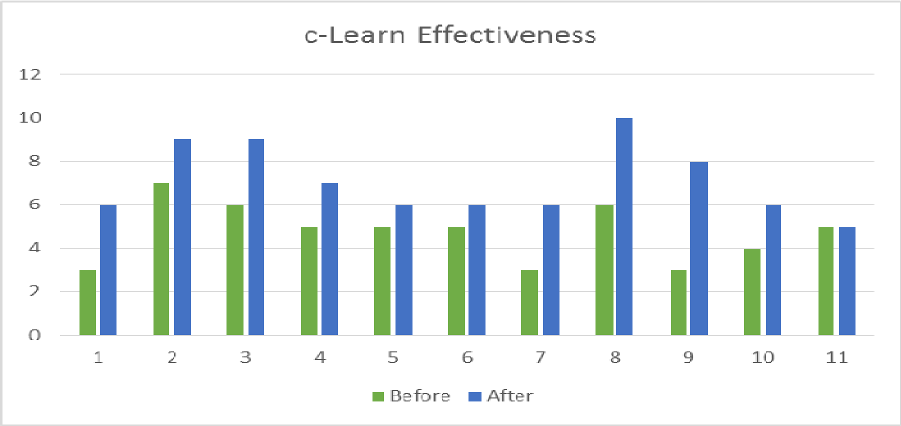


Figure 5. Learners marks pre and post use of c-Learn

The percentile improvement shown in figure 6 as an improvement in using the c-Learn prototype has made towards the learning of the students.

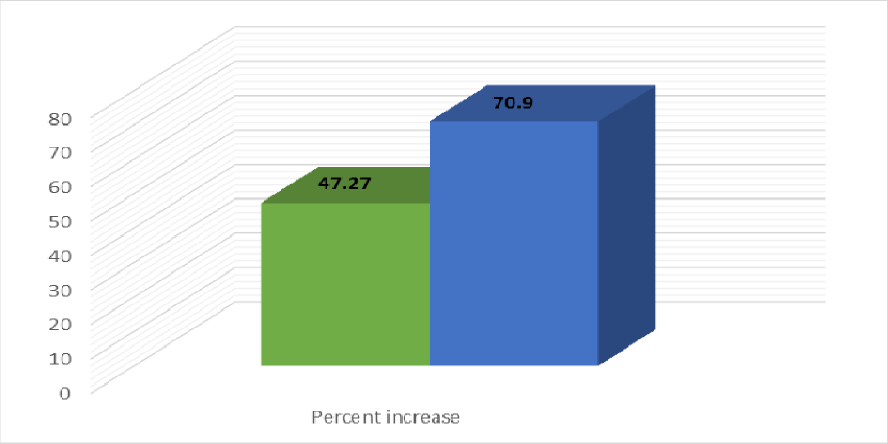


Figure 6. Percent increase for the users of c-Learn

We have also calculated the standard deviation to study the improvement which is increased from 1.35 to 1.64 (rounded to 2 decimal places) as the pre and post c-Learn training. As the marks of the student are either same for one student or in other cases improved, the increased value of standard deviation shows a positive impact of c-Learn in the learning process of the students. This increase in the value means that the students improved their skills based on their capabilities after using the c-Learn program.

5. Conclusions and Future work

An electronic help-desk available 24/7 to learn a programming language can be beneficial to the students. It can solve the limited availability problem of instructors and student shyness to ask questions in the class. We have presented c-Learn in which the student can gradually learn programming and in case of errors it can guide back to the topics which the student should understand to solve the problem. The assessment of the students through the prototype has shown improved results in the learning of programming language by the students who were studying the first programming course in the university level. This learning model can further be enhanced to learn the behaviour of different type of learners, based on the categories described in [10] as effective and non-effective novices. More extensive evaluations can be performed at various stages of learning and between different lessons. The prototype can be enhanced to various topics beyond loops and arrays and evaluations can be performed on different topics. Also, the technique can be used to evaluate different programming paradigms like structured and object-oriented programming.

Acknowledgements

I would like to thank Deanship of Scientific Research (SRD) at Qassim University, for the grant number 2726 under which we completed this project. Special thanks to Mr. Sohib Algotimal and Mr. AbdulAziz Aljameeli for working on the prototype as part of their final year project.

REFERENCES

[1] Law, K. M., Lee, V. C., & Yu, Y. T. (2010). Learning motivation in e-learning facilitated computer programming courses. Computers & Education, 55(1), 218-228.

[2] Pea, R. D., & Kurland, D. M. (1984). On the cognitive effects of learning computer programming. New ideas in psychology, 2(2), 137-168.

[3] Cavus, N., Uzunboylu, H., & Ibrahim, D. (2007). Assessing the success rate of students using a learning management system together with a collaborative tool in web-based teaching of programming languages. Journal of educational computing research, 36(3), 301-321.

[4] Douce, C., Livingstone, D., & Orwell, J. (2005). Automatic test-based assessment of programming: A review. Journal on Educational Resources in Computing (JERIC), 5(3), 4.

[5] Chalk, P., Boyle, T., Pickard, P., Bradley, C., Jones, R., & Fisher, K. (2003, August). Improving pass rates in introductory programming. In Proceedings of the 4th Annual LTSN-ICS Conference (pp. 6-10).

[6] Chrysafiadi, K., & Virvou, M. (2013). PeRSIVA: An empirical evaluation method of a student model of an intelligent e-learning environment for computer programming. Computers & Education, 68, 322-333.

[7] Hsiao, I. H., Sosnovsky, S., & Brusilovsky, P. (2010). Guiding students to the right questions: adaptive navigation support in an E□Learning system for Java programming. Journal of Computer Assisted Learning, 26(4), 270-283.

[8] Othman, A. A., Pislaru, C., & Impes, A. M. (2014). Improving the quality of technology-enhanced learning for computer programming courses. International Journal of Information and Education Technology, 4(1), 83.

[9] Vega, C., Jiménez, C., & Villalobos, J. (2013). A scalable and incremental project-based learning approach for CS1/CS2 courses. Education and Information Technologies, 18(2), 309-329.

[10] Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. Computer science education, 13(2), 137-172.

[11] Klašnja-Milićević, A., Vesin, B., Ivanović, M., & Budimac, Z. (2011). E-Learning personalization based on hybrid recommendation strategy and learning style identification. Computers & Education, 56(3), 885-899.

Author

Dr. **Noman Hasany** is working as an Assistant Professor in College of Computer, Qassim University, KSA. He did PhD in Computer Systems Engineering from Universiti Putra Malaysia; and M.E. and B.E. in Computer Systems Engineering from NED University, Karachi. His research interests include Semantic Web, Natural Language Processing, Information retrieval, learning systems and Knowledge based systems.