An Evaluation Framework and Analysis of Auto Assessing the Programming Courses during the COVID-19 Pandemic

Dr.J.Dhayanithi[a], Dr.D.Balamurugan[b], M.Marimuthu[a],Vidyabharathi.D[b], Basker.N[a], Vidhya.G[a] Theetchenya.S[a], Mohanraj.G[a]

[a]Assistant Professor, Department of Computer Science and Engineering, Sona College of Technology(Autonomous), Salem.
[b]Associate Professor, Department of Computer Science and Engineering, Sona College of Technology(Autonomous), Salem.

*dhaya.j@gmail.com,balamurugand@sonatech.ac.in, mari.btech@gmail.com,dvbharathi77@gmail.com,bas2k9@gmail.com, gvidhyacse@gmail.com, theetchenya@gmail.com,mohanraj134@gmail.com,

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 28 April 2021

Abstract: Covid-19 pandemic changed the traditional working dimension of many sectors. Many sectors are still struggling and facing many difficulties to change their current working environment to adapt the Covid-19 pandemic. Education sector is foster in adapting this pandemic and change their teaching learning process from face-to-face methodology to remote learning. UNICEF reported that approximately 800 million learners got affected due to the spread of COVID-19. This outbreak makes a paradigm shift for many educationists from schools to colleges. The educationists are still fighting to survive this pandemic and introduce different approaches to enhance the teaching learning process. This paper presents a framework for auto assessing the programming courses using MOODLE learning management system with the support of four activities (quiz, debugging, fill the missing code and programming). The proposed framework was analysed with the help 176 students enrolled for Python Programming Courses. The grades acquired by all the students areanalysed and the result shows that student programming skill is drastically increases by using this auto assessment framework. The student perception on auto assessment framework was analysed based on the survey conducted with students. The survey responses clearly state that this kind of auto evaluation framework is helping the students to increasing their learning interest in any programming courses. Suggestion on adapting this kind of auto assessment framework in other programming courses is also presented

Keywords: Online Education, Automated Assessment, Evaluation, e-learning, Computer Programming

1. Introduction

Learning computer programs is very essential to all the students from schools to university. Many students still find more difficulties in learning the fundamentals of programming courses. It is evident that there is high failure rate in programming courses examination conducted by various universities. The student learning outcome was measured in the programming courses are by Quiz, Assignments, Project and Viva Voce. To strengthen the student's programming skills several real-time problems should be solved in an adaptive manner. These exercises should be monitored and evaluated. Monitoring and evaluating students' solution will be a very difficult task. The traditional way of assessment is not practice centered. Students receive the question in random manner and start write the logic for the chosen question in a paper. The logic was evaluated by a teacher and if the logic is correct then students are permitted to type the code in appropriate code editor otherwise students are requested to rewrite the logic. The student typed code is manually evaluated by the teacher by giving sample input to the program. If the code correctly typed for the verified logic then 100% grade is given otherwise the grade was awarded based on the correctness of the logic.

The whole process was carried out in stipulated time. If any student is not able to submit the code at particulate span time then only very less grade was given to that particular student. For faculty, it is very difficult to assess all the students in the limited time period. The kind of grading impacts the learning approach of the student. The student assessment spirit and style characterize the curriculum. The baseline of the two authors is that there should be a proper evaluation/assessment method is required. In order to make the evaluation process as more accurate an automated mechanism is required. The auto evaluation process includes compilation error checking, testing the results and logic, efficiency of the code and total time taken to submit the code. Auto Evaluation ensures the same rubric is followed for all the submitted code and also it ensures speedy result declaration. This paper shows the study of how auto evaluation of programming have emphasize the student learning skills and faculty teaching skills compared with the traditional method.

The rest of the paper is organized as follows: Section 2 narrate the related work carried out in automated assessment system. Section 3 explains the framework used for auto assessing the programming courses. Section 4 describes the student performance analyses and also depicts the impact of using auto assessing in the programming courses. In section 5 Conclusion of the article is explained with the outlook of future work to be carried out.

2.Related Work

In the recent years, a quite number of reviews have been carried out on assessing the programming courses in automated manner [18,22–29]. Many students have drop out from the colleges because of difficulty in understanding the programming courses. According to (Tabano ES, Rodrigo MMT &Jadud MC 2011) 30-40% of student drop-out is mainly because of programming courses. The evolution of introductory programming courses for the past fifty years is completely analyzed by (Becker &Quille 2019). Their analyses states how the automated assessment system impacts on student retention and success rate. It also confirms that many research need to be carried out on auto assessing the programming courses based on various characteristics.

Auto Grading of Student programs is started in the late 90's itself. (Jackson & Usher 1997) proposed a system based evaluation tool called ASSYST to assess the ADA and C language programming assignments. Only four characteristics are considered for the assessment namely, complexity, efficiency, style and correctness. (Daly C 1999) proposed another promising tool for assessing C++ program through internet. This tool is limited to analyze the structure and syntax of any C++ program. (Mandal AK, Mandal C & Reade CMP 2006) suggested the architecture for auto evaluating the C programs based on inputs and outputs. (Farrow M, King PJB 2008) utilizes JUnit and BLUEJ software and proposed a framework to assess the Java programming. The improvement to be carried out in auto evaluating the student submitted programming assignments was elaborately listed by (skupas B 2010). In this approach, the auto grading is done by two phases. In the first phase, the student submitted program is auto evaluated based on the output and in the second the programming logic is evaluated manually. A framework for evaluating programming language examinations is carried by (Himani Mittal &Syamala Devi Mandalika 2015). Every student submitted programs is evaluated form a small piece of program to single file programs. This framework is limited to programming languages like C, C++ and Java.

The impact of student performance on auto evaluation of programming course is studied by (Rubio-Escudero&Asencio-Cortes 2018). Their study states that, auto evaluation will enhance the student performance by attending and passing the examinations. It also helps the students to revise their knowledge based on the instant feedback given by the auto assessment tool. The student programming skill is influenced by various factors namely, school mathematics background, prior programming experience, comfort level, intrinsic motivation, formal training carried and understanding the shared material. (Bergin S & Reilly R 2005). The early prediction of lack of understanding in programming courses is done by (Skalka J, Drlík M &Obonya J 2017). A predictive modeling technique was suggested to monitor the student performance from the evaluation carried at mid semester. The auto evaluation alone does not improve the student programming skill. The ideal learning outcome is achieved by coupling effective way of teaching and providing high quality assignments. (Akahane Y, Kitaya H & Inoue U 2015). (Pieterse V 2013) conducted a study on assessing the student performance with and without using the auto assessment tool in programming courses. The results show that performance was not statically significant. The same kind of study was carried by (Wilcox X 2015) and it shows that the exam scores was significantly higher those who are undergone automated grading mechanism. The student dropout also significantly reduced when we use the auto assessment process in the programming courses. (Rubio-Sanchez M, Kinnunen P, Pareja-Flores & Velázquez-Iturbide 2014).

3. Framework for Auto Assessing the Programming Course

The ongoing COVID-19 pandemic came on unexpectedly, causing a change in teaching and learning. Many institutions in India have reacted quickly and created guidelines to emphasize the teaching-learning process without any hurtles. Many research is need to be carried out for efficiently adapt the courses to the new guidelines.

In this paper a framework was describes for the transformation of traditional assessment method of programming courses in engineering education to complete online format by means of student centred automated assessment tool. In order to evaluate the new assessment method, a detailed study was carried out by incorporating new framework for conducting programming lab courses and later student’s perception on the auto evaluation tool is also analyzed. The traditional way of handling programming laboratory courses in many universities and colleges are initiated by providing the lab manual to the students at the commencement of every semester. The lab manual contains detailed description on what experiments need to be completed on every week and supporting material on those experiments. During the lab hour students gone through the lab manual to complete their lab exercises. Once student completed lab exercises faculty will start evaluating the lab exercises manually and mark will be allocated appropriately. The main drawback of the traditional approach is that students need to complete their lab exercise during the lab hour only. Students programming skill is manually assessed by faculty by having interaction with the students. It is very difficult to the faculty to assess all the students in a stipulated time.

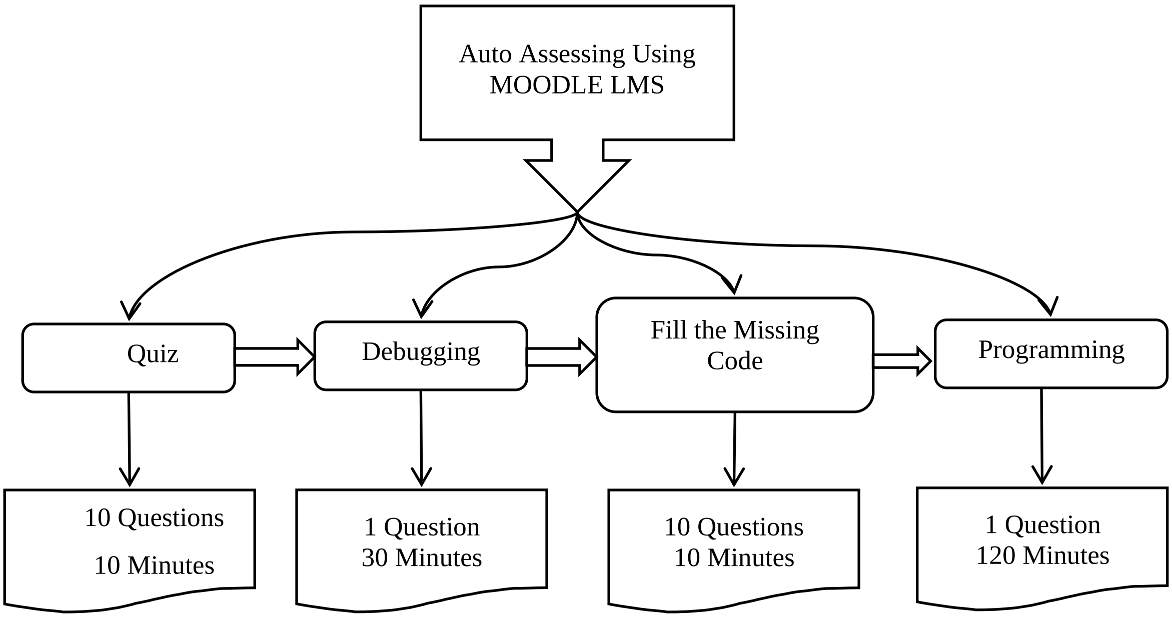

Figure.1 Framework for Auto Assessing the Programming Course

To overcome the aforementioned issues a framework was suggested by incorporating MOODLE for handling programming laboratory courses. Modular Object Oriented Dynamic Learning Environment (MOODLE) is a Learning Management System which is effectively used for delivering content and conducting assessment based on MCQ. The proposed framework was depicted in Figure 1. Each laboratory experiments were auto assessed by completing four activities namely Quiz, Debugging, Filling the Missing Code and Programming. The Quiz comprises of 10 questions and need to be completed in 10 Minutes. Quiz Plugin was exclusively used for handling this activity. Coderunner is an open source plugin for MOODLE and it is used to auto assess the Debugging and Filling missing code activity. One question with 30 minutes duration and 5 questions with 10 minutes duration are used for Debugging and Fill the missing code activity respectively. In the last activity, Virtual Programming Lab (VPL) plugin is used to auto assess the student programming assignment based on the known and hidden test cases. One programming questions with 120 minutes duration is carried out for completing this activity. All the activity was graded against 100 marks. Python programming laboratory course is considered for the study. This course was taught in the fifth semester for Computer Science and Engineering students. 176 students have enrolled for this course and the auto evaluation is carried out for 10 weekly exercises with four activities and one final exam with four activities. The result obtained is enabling us to analyze the creditability of newly introduced auto evaluation tool in the curriculum and how to improvise the same course in the future endowers.

3.1 Analyzing the Auto Assessment Framework

The programming lab session was conducted at any time and student has privilege to access the course material from anywhere. During this Covid 19 Pandemic all the laboratory classes was conducted through online and in every exercise students need to complete four activity as mentioned in the Figure 1. In Table 1 Auto Assessment analysis on Quiz, Debugging and Fill the missing code activity was narrated using two factors namely average completion time of all the students and Average grade secured by the entire student for ten exercises and one final exam respectively. The average completion time of quiz, debugging and fill the missing code by the entire students are 7.27 minutes, 24.63 minutes and 9.18 minutes respectively. From this it was observed that students have actively participated in the entire activity.

Table.1. Auto Assessment Details on Quiz, Debugging and Fill the Missing Code

Activity	Quiz		Debugging		Fill the Missing Code	
	Average Completion Time (minutes)	Average Grade (%)	Average Completion Time	Average Grade (%)	Average Completion Time	Average Grade (%)

Exercise 1	5	83	27	61	9	70
Exercise 2	6	84	26	54	10	61
Exercise 3	8	72	28	72	9	74
Exercise 4	9	89	22	75	9	76
Exercise 5	8	92	19	81	10	81
Exercise 6	7	88	24	85	8	79
Exercise 7	5	92	25	91	10	86
Exercise 8	6	94	21	94	9	90
Exercise 9	8	96	22	95	10	92
Exercise 10	9	98	28	92	9	93
Final Exam	9	97	29	94	8	94

The programming activity was separately analyzed based on number of test case passed, average execution time and number of times student compiled the code in every exercises. The details were illustrated in Table 2. Every programming exercise question should be auto evaluated against 8 known test cases and 2 hidden test cases. The result shows that all the students have successfully passed 9 test cases. The average completion time of the entire program is 108.27 minutes and it reflects that all the students were actively participated in doing programming assignments. From the table it also observed for the final exam student undergone more number of compilations compared with regular programming exercises.

Table.2. Auto Assessment Details on Programming

Activity	Average Number of Test Case Passed	Average Execution Time of all Students (minutes)	No. of Compilation
Exercise 1	8	114	1244
Exercise 2	9	119	991
Exercise 3	10	117	744
Exercise 4	9	103	586
Exercise 5	10	78	355
Exercise 6	10	112	771
Exercise 7	9	118	685
Exercise 8	8	81	378
Exercise 9	8	118	957
Exercise 10	10	117	678
Final Exam	8	114	1342

3.2 Analyzing the Student Responses on the use of proposed framework in the laboratory course

A survey was conducted with students after completing their final examination on python programming laboratory. The survey comprises of 9 question and the responses was collected on likert scale (Strongly Disagree, Disagree, neither agree nor disagree, Agree and Strongly Agree). The student responses over the survey were shown in Table 3. Students feel more comfortable while using the auto assessment tool and only very few percentage of students shows discomfort in using the tool. All the programming questions are auto evaluated against 10 different test cases and 22% of student had faced some difficulties in passing all the test cases. 69% of students responded that auto assessment tool has increased their motivation to do the programming exercises and 74% of students answered that auto assessment tool were increased their programming skill. The grading system provided by the auto assessment tool is more creditable. More than 80% of students agreed that this kind of auto assessment tool can be used for other programming courses and it is the better alternative for traditional programming exams.

Table.3. Results of Student survey on the use of proposed framework in the laboratory course

Questions	Strongly Disagree	Disagree	Neither agree nor disagree	Agree	Strongly Agree
Auto Assessment Tool was easy to use	9	5	28	52	82

	(5%)	(3%)	(15%)	(30%)	(47%)
Passing all the test case is difficult	15 (9%)	17 (10%)	23 (13%)	64 (36%)	57 (32%)
Auto Assessment tool has increased my motivation to work on the programming exercises	0 (0%)	0 (0%)	24 (14%)	31 (18%)	121 (69%)
The auto grades provided by the tool were fair	0 (0%)	21 (12%)	34 (19%)	58 (33%)	63 (36%)
Auto Assessment tool improved my programming skills	0 (0%)	0 (0%)	22 (13%)	24 (13%)	130 (74%)
The grades provided by the auto assessment tool is more creditable compared with the manual assessment method	0 (0%)	14 (8%)	32 (18%)	62 (35%)	68 (39%)
It is better to use the tool in other programming courses	0 (0%)	0 (0%)	0 (0%)	34 (19%)	142 (81%)
It enables to do the programming exercise at my convenient time	0 (0%)	0 (0%)	12 (7%)	74 (42%)	90 (51%)
Auto assessment tool is better alternative to traditional laboratory exams	0 (0%)	0 (0%)	0 (0%)	34 (19%)	142 (81%)

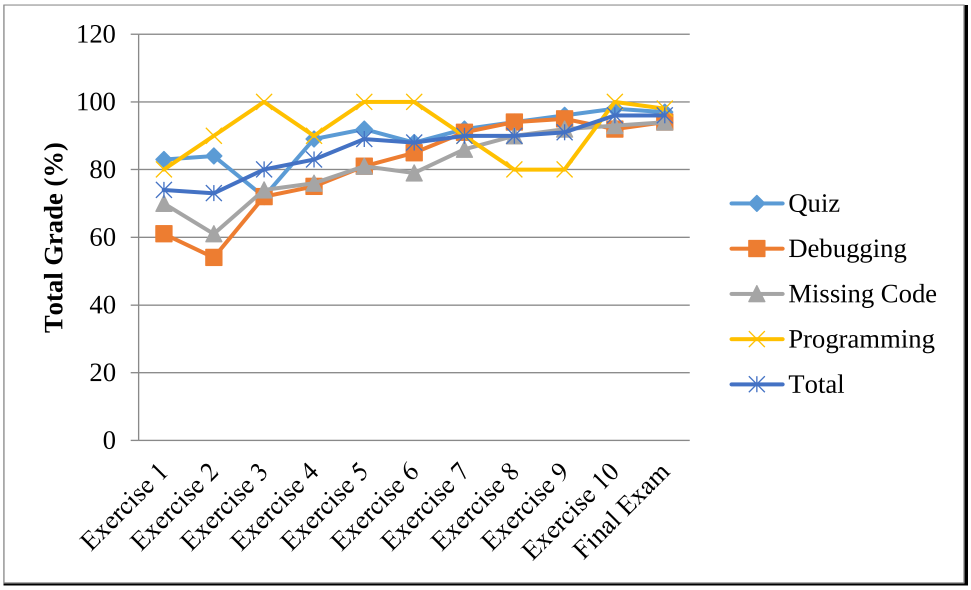

Figure.2 Average Grade of all the Students – Activity wise

Figure 2 shows the results of average grade of entire 176 students all the four activity. The total grade of entire students is gradually increased from the exercise 1 to final examination. Average grade of entire programming laboratory is 96% which is significantly high compared with the traditional way of assessing the programming experiments.

4.Conclusion and Future Work

The evaluation framework for analyzing the auto assessment process of programming courses during the covid-19 pandemic is elaborated in this paper. The proposed framework was incorporated using MOODLE learning management system by influencing four plugins. Every student should complete the four activities created for every lab exercise in a stipulated time. The grading possessed by all the students' shows that the proposed evaluation framework has increased the student programming skills. The framework can be easily configured by any person and it is completely available as open source. The questions prepared for all the four activities in every lab exercises is the key factor for the entire evaluation framework. Faculty need to be more sensible on preparing the questions. The elaborative survey was conducted with the entire student who enrolled in the programming courses. The survey responses show that student feel more comfortable in using this framework and many students have recommended to use this kind of assessment process in other programming courses too. One of the major limitations in this framework is all the activities to be carried out in home and there is no any standard mechanism to ensure whether appropriate student only doing the activity. This will be taken to considered as an interesting future work.

References

1. Jackson D, Usher M (1997) Grading Student Programs using ASSYST. In: Proceedings of the 28th SIGCSE technical symposium, 335–339
2. Daly C (1999) RoboprofAnd An Introductory Computer Programming Course. In: Proceedings of the 4th annual SIGCSE/SIGCUE on innovation and technology in computer science education, Krakow, 155–158
3. Douce et al (2005) A Technical Perspective on ASAP—Automated System For Assessment Of Programming. In: Proceedings of the 9th computer assisted assessment conference
4. Mandal AK, Mandal C, Reade CMP (2006) Architecture Of An Automatic Program Evaluation System. In: Proceedings of CSIE
5. Farrow M, King PJB (2008) Experiences with Online Programming Examinations. IEEE Trans Educ 51(2), 251
6. Skupas B (2010) Feedback Improvement In Automatic Program Evaluation Systems. Inform Educ 9(2) 229–237
7. Himani Mittal and Syamala Devi Mandalika,Framework for Evaluationof Programming Language Examinations, Computational IntelligenceTechniques for Comparative Genomics, Forensic and Medical Bioinformatics, 121-127
8. C. Rubio-Escudero, G. Asencio-Cortes, F. Martinez-Alvarez, A. Troncoso, and J. C. Riquelme, "Impact of Auto-evaluation Tests as Part of the Continuous Evaluation in Programming Courses", The 13th International Conference on Soft Computing Models in Industrial and Environmental Applications, 553--561
9. Becker, B.A.; Quille, K. 50 years of CS1 at SIGCSE: A review of the evolution of introductory programming education research. In Proceedings of the SIGCSE 2019-Proceedings 50th ACM Technical Symposium Computer Science Education, Minneapolis, 1(1), 338–344.
10. Tabanao, E.S.; Rodrigo, M.M.T.; Jadud, M.C. Predicting at-risk novice Java programmers through the analysis of online protocols. In Proceedings of the Seventh International Workshop on Computing Education Research, ICER 2011
11. Bergin, S.; Reilly, R. Programming: Factors That Influence Success. In Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education, 411–415
12. Hellas, A. Nam Liao, S. Ihantola, P.; Petersen, A. Ajanovski, V.V.Gutica, M. Hynninen, T. Knutas, A. Leinonen, J. Messom, C. Predicting academic performance: A systematic literature review. In Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE, 175–199
13. Costa, E.B. Fonseca, B. Santana, M.A. de Araújo, F.F. Rego, J. Evaluating the effectiveness of educational data mining techniques for early prediction of students' academic failure in introductory programming courses. Comput. Hum. Behav. 7(3), 247–256
14. Akahane, Y.; Kitaya, H.; Inoue, U, Design and evaluation of automated scoring Java programming assignments, In Proceedings of the 2015 IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 1–6
15. Pieterse, V, Automated Assessment of Programming Assignments. In Proceedings of the 3rd Computer Science Education Research Conference on Computer Science Education Research, 45–56.
16. Wilcox, C. The Role of Automation in Undergraduate Computer Science Education, In Proceedings of the 46th ACM Technical Symposium on Computer Science Education, 90–95.
17. Rubio-Sánchez, M, Kinnunen, P.; Pareja-Flores, C. Velázquez-Iturbide, Á. Student Perception and Usage of an Automated Programming Assessment Tool.Comput. Hum. Behav. 31, 453–460
18. Asraf Yasmin, B., Latha, R., & Manikandan, R. (2019). Implementation of Affective Knowledge for any Geo Location Based on Emotional Intelligence using GPS. International Journal of Innovative Technology and Exploring Engineering, 8(11S), 764–769. https://doi.org/10.35940/ijitee.k1134.09811s19
19. Muruganantham Ponnusamy, Dr. A. Senthilkumar, & Dr.R.Manikandan. (2021). Detection of Selfish Nodes Through Reputation Model In Mobile Adhoc Network - MANET. Turkish Journal of Computer and Mathematics Education, 12(9), 2404–2410. https://turcomat.org/index.php/turkbilmat/article/view/3720