Automated Assessment & Feedback System for Novice Programmers

Jagadeeswaran Thangaraj*
School of Computing
Dublin City University
Dublin, Ireland
Jagadeeswaran.Thangaraj@dcu.ie

ABSTRACT

The 'Introductory Programming' module is the first step in software development courses and a number of activities have been introduced to motivate novice students in programming modules. An assessment and feedback system is one such activity. This system checks the granular parts of programs and notifies students where they have errors which they can correct. It is widespread, but it is still an emerging area to improve its process as well as to improve novice programming skills. So the objective of this research is to plug this gap in the assessment and feedback systems of these introductory programming modules.

CCS CONCEPTS

• **Social and professional topics → Student assessment**.

KEYWORDS

Automated Assessment & feedback; Novice Programming

ACM Reference Format:
Jagadeeswaran Thangaraj. 2021. Automated Assessment & Feedback System for Novice Programmers. In *26th ACM Conference on Innovation and Technology in Computer Science Education V. 2 (ITiCSE 2021), June 26-July 1, 2021, Virtual Event, Germany*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3456565.3460021

1 INTRODUCTION AND MOTIVATION

Programming modules are essential for any software development related courses at higher education institutions all over the world. The 'Introductory Programming' module is the first step in the software development courses [2]. The learning outcomes of them are fundamental concepts of programming such as basic computation, simple I/O, conditional and iterative structures and usage structures. When assessing these concepts, the examiner asks students to design, implement and test a project which is typically complex [7]. It consists of usage of various concepts from the module. The objective of these courses is to give the basic knowledge to writing programming languages by introducing syntax and semantics. Therefore these modules play an important role to make them comfortable in continuing their education. There are a number of activities introduced to motivate the novice students in programming modules.

Assessment is an important activity of computer education which leads programming learning. Most of the assessment systems evaluate the program's output, not the code. They do also not cover each and every concept the student attempts. The feedback system is not enough to show where they missed the concepts as it reports back as final results. So the major problem in the assessment and feedback system of these introductory programming modules is that the assessments do not justify the granular parts of the projects and fails to notify where they went wrong. So the objective of this research is to plug this gap in the assessment and feedback system of these introductory programming modules. As [7] recommended, independent components of programming will increase opportunities to diagnose difficulties of novices.

2 RELATED WORK

There are a number of solutions that have been found in order to make assessment easy and to feedback students where they made mistakes or found difficulties. Some of them are automated assessments, rubric styled evaluation. An automated assessment system[12] is a tool which evaluates the student's program and gives the feedback immediately. There are a variety of automated assessment techniques available to evaluate student programming assignments [5]. Automatic assessment is the process of student's assessment using computerised testing. They support automatic evaluation of student submissions based on different approaches such as natural language processing, machine learning, Image recognition and targeted feedback [1][8][9]. Targeted feedback systems [12] help the students to correct common misconception errors in the introductory programming.

Multiple Choice questions (MCQ) is the most common approach in computerised testing. In the case of programming modules, MCQ testing is not enough to assess the student's coding knowledge and it does not engage students to write their own code when they are at an advanced level. Therefore, code writing exercises recommended for this issue. CodeRunner[6] is an automatic assessment tool which tests the code submitted by students and provides feedback immediately. It operates as a question type Moodle plug-in and checks the correctness of programming in most of the text based languages. Virtual programming Lab (VPL) is another Moodle plug-in tool which enables to assess the student programming assignments in different languages [13]. It automatically executes the student programs and provides formative feedback and generates the report by evaluation. It is used to encourage programming

ITiCSE 2021, June 26-July 1, 2021, Virtual Event, Germany

ACM ISBN 978-1-4503-8397-4/21/06.
https://doi.org/10.1145/3456565.3460021

skills for Novices. MULE[4] and Eienstien[3] are other automated assessment and feedback tools used in Ireland. These are providing immediate feedback to motivate the novice programmers and helping in assessing and marking of programming assignments.

3 PROBLEM STATEMENT

Independent components of programming will increase the difficulties for novices. An assessment and feedback system needs to check the granular parts of the programs and to notify students where there are errors so they can correct the results. There is a gap between the assessment and feedback system of programming assignments. Therefore the assessment system should help in this scenario as assessment plays a vital role here. Based on the literature review carried out so far, there is no effective assessment system for evaluating the codes and feedback system for encouraging student's programming skills.

In conclusion, most of the studies have concluded that their primary motivation is to provide the feedback immediately as a student centered approach to make them understand where the code has errors, what types of the errors and possibly to make necessary changes and submit it again. Therefore these assessment tools help the students to reflect and enable them to learn from the mistakes. The first drawback is that it assesses the same questions for all students with different abilities. The next drawback is that these systems fail to motivate the novices in programming assignments apart from notifying the errors as compilers.

The targeted system should enable accurate error correction from the teaching point of view and provide the feedback immediately to encourage the students. The targeted modules are fundamental programming in the university entry level. The aim of this study is to help the educators assess process improvement, in particular, software programming assessment and support student's ability to learn programming languages.

4 RESEARCH GOALS

The goal of this research is to explore the current automated assessment solutions in terms of their ability scaffold and support novice programming students. The aim of this research is to enhance current assessment processes for novice programming students. The goals or research questions for this aim are:

- Given that an automated assessment and feedback (AAF) system can indicate some of the most common programming errors, how does AAF support the ability of novices to understand and correct the errors?
- Does AAF encourage novices to improve their programming skills?
- Does AAF evaluate modular parts of programming to help novices effectively learn the introductory programming concepts?

5 RESEARCH METHODS

In general, Adaptive assessment [10] is defined as testing a participant's knowledge based on their previous attempts in the assessment process. Adaptive assessment process assesses the students with different abilities with different sets of questions. In this testing, a list of questions is classified in three cognitive levels based on the complexity (like easy, moderate and difficult). Here easy questions assess the basic concepts, moderate questions assess comprehensive knowledge and difficult questions do the applications of the knowledge. Although adaptive testing is in use in the computer education sector and it has started to get more attention in recent years with the use of computer assisted learning.

We plan to introduce a progressive approach to assessing and providing feedback on the granular parts of programming projects by using adaptive assessment. In this assessment, if a student answers correctly for a question, then the next question is hard. Otherwise the easy one will be asked. It continues until the system predicts the level of student's proficiency [11]. Difficulties in programming are classified as Bloom's taxonomy of programming. The shortcoming of this work is that it gives only minimal feedback on their work. Therefore the proposed research method is applying the adaptive assessment technique to assess the novice students in their skill level and encourage them to learn to achieve the required knowledge.

6 CONTRIBUTIONS

The short-term contributions of this work includes an investigation into targeted error messages that novice programmers encounter in the introductory programming. In the longer term, the research will provide the feedback immediately and help the students to reflect and enable them to learn from the mistakes. Therefore the aim is to design an automated assessment and feedback framework to improve their programming proficiency.

REFERENCES

[1] Varun Aggarwal, Vinay Shashidhar, and Shashank Srikant. 2013. Principles for using Machine Learning in the Assessment of Open Response Items: Programming Assessment as a Case Study.

[2] Brett Becker. 2019. A Survey of Introductory Programming Courses in Ireland. (07 2019), 58–64. https://doi.org/10.1145/3304221.3319752

[3] Ben Coleman. 2018. Docker Einstein. *Technical Guide* (2018). https://github.com/BenjyC/Docker-Einstein-4YP

[4] Natalie Culligan and Kevin Casey. 2018. Building an Authentic Novice Programming Lab Environment. *Irish Conference on Engaging Pedagogy* (12 2018).

[5] Anabela Gomes, Fernanda Correia, and Pedro Henriques Abreu. 2016. Types of assessing student-programming knowledge. https://doi.org/10.1109/FIE.2016.7757726

[6] Richard Lobb and Jenny Harlow. 2016. Coderunner: A Tool for Assessing Computer Programming Skills. *ACM Inroads* 7, 1 (Feb. 2016), 47–51. https://doi.org/10.1145/2810041

[7] Andrew Luxton-Reilly, Jacqueline Whalley, Brett Becker, Yingjun Cao, Roger McDermott, Claudio Mirolo, Andreas Mühling, Andrew Petersen, Kate Sanders, and Beth Simon. 2017. Developing Assessments to Determine Mastery of Programming Fundamentals. 47–69. https://doi.org/10.1145/3174781.3174784

[8] Phil Maguire, Rebecca Maguire, and Robert Kelly. 2018. Using automatic machine assessment to teach computer programming. *Computer Science Education* 27 (02 2018), 1–18. https://doi.org/10.1080/08993408.2018.1435113

[9] Eerik Muuli, Kaspar Papli, Eno Tonisson, Marina Lepp, Tauno Palts, Reelika Suviste, Merilin Säde, and Piret Luik. 2017. Automatic Assessment of Programming Assignments Using Image Recognition. 153–163. https://doi.org/10.1007/978-3-319-66610-5_12

[10] Elena Papanastasiou. 2014. *Adaptive Assessment.* 1–2. https://doi.org/10.1007/978-94-007-6165-0_3-4

[11] Julia Phelan. 2019. Review of Adaptive Testing, Mastery-based Learning and their Educational Applications. *white paper* (2019). https://doi.org/10.1177/2158244019885136

[12] Yizhou Qian and James D. Lehman. 2019. Using Targeted Feedback to Address Common Student Misconceptions in Introductory Programming: A Data-Driven Approach. *SAGE Open* 9, 4 (2019), 2158244019885136. https://doi.org/10.1177/2158244019885136

[13] Juan Rodriguez-del Pino. 2012. A Virtual Programming Lab for Moodle with automatic assessment and anti-plagiarism features.