Relevance of Immediate Feedback in an Introduction to Programming Course

Dr. Raquel Landa, Tecnologico de Monterrey (ITESM)

Part-time teacher at Tec de Monterrey since 1999, with a Ph.D. in Education, a Master in Information Technology Management and a Major in Electronic Systems. Currently involved in Innovation projects related to engineering and programming courses.

Yolanda Martinez-Treviño, Tecnológico de Monterrey

Relevance of Immediate Feedback in the Introduction to Programming Course

María Raquel Landa Cavazos
Computer Science Department
Tecnologico de Monterrey,
Campus Monterrey.
Monterrey, México
rlanda@tec.mx

Yolanda Martínez Treviño
Computer Science Department
Tecnologico de Monterrey,
Campus Monterrey.
Monterrey, México
yolanda.mar.tre@tec.mx

Abstract

This paper presents the results of integrating the use of an auto-grader tool in a Computer Science 1 course to personalize the learning process of students by allowing them to advance at their own pace when solving problems in class sessions. The use of this tool allowed us to give them immediate feedback for their solutions, once they verified their solution was correct they can move forward to more complex exercises without having to wait for the professor individual approval.

To evaluate the strategy a statistic experiment has been carried out comparing grades of CS1 courses where the auto-grader tool has not been used with courses in which the tool was implemented. Finally, a questionnaire was applied to identify the student's satisfaction. The results show that, although the grades have not significantly improved, the perception of the strategy by students and professors is positive.

Keywords: Educational innovation, mastery learning, automated grading tools, immediate feedback, pace of student learning

Background

It is widely accepted that learning to program is challenging for many students; it requires the understanding of abstract concepts and the ability to combine statements into valid programs [1, 2, 3].

This represents a challenge also to every programming teacher. Traditional teaching methods are no longer enough to involve students in the learning process actively because it is only through constant practice that the necessary skills to become a competent programmer can be developed. Bay says, "The more you practice, the more you develop skills" [4], and Lahtinen states that students need practical experience to understand the concepts [3].

To meet this challenge one interesting option is *Mastery Learning*. With this approach, students advance in content and activities at their own pace with the particularity that they can only move forward to more advanced topics or problems when they demonstrate that they have mastered the previous ones. Mastery Learning is not a new concept; it was developed by Benjamin Bloom, who considered that professors could improve student learning by personalizing the instruction.

He suggested that nearly all students could reach a high level of achievement if the appropriate learning conditions and adequate time were given to them [5].

For this method to work within the programming context, it is necessary to give immediate feedback to students about their activities, which can only be done through the use of automated grading tools. Koile says that feedback improves learning, especially when it occurs as the new concept is being introduced [6]. Bay writes that students need an efficient assessment tool to inform them if they gave a correct or a wrong solution [4].

In this paper, we present a strategy supported by the use of an automated grading tool to give immediate feedback to students regarding their programming exercises in a Computer Science 1 course. We compare the results of the grades between the groups where the strategy was used with the groups where it was not used. And finally, we report the opinions of the students and the professor participating in the experimental groups.

The implementation of this learning experience was not defined as a formal experiment, and the students of the groups were not selected by the professors; instead, they were the students who enrolled in those courses; therefore, the data collected from this innovative practice was analyzed using a quasi-experimental approach.

Methodology

I. The course

This strategy was applied in the fall semester of 2017 in the first programming course for Computer Science students. The term was 16 weeks long, and it had 3 class hours per week. The language used to teach programming was C++. The strategy was applied in two groups taught by the same professor; namely, a standard group with 27 students and an honors program group with 24 students.

The main difference between the honors program group and the standard group was the difficulty level of the exercises provided. While in the standard group the exercises started at a basic level, in the honors program group, they started at an intermediate level. In both groups, the exercises were designed to have different difficulty levels that ascend up to the challenge level. In the honors program group, it is common that most students reach the challenge level, while in the standard group just a few students achieve that performance.

II. The tool

For this innovative practice, the Vocareum Programming Lab was used [7]. In this tool, the professor uploads the description of the exercise and the test cases she wants to use to check the students' solutions.

The students have an editor area and a console area where the programming exercises can be typed and tested (Figure 1). When the student finishes the exercise, he uploads the solution. Then the automated tool compares the student's program output with the output for each one of the test cases provided by the professor and gives feedback to the student indicating for each test case if the solution submitted passed or failed.

Fig. 1. Students' view. On the left side, the files. On the right side, the editor. At the bottom, the console area.

The professor has a dashboard where he can monitor the submissions of the students (Figure 2). He can see which students have sent the solutions and which have not. He also can see the program of an individual solution and the grade assigned by the tool; he can run the program in the command line area to test the solution, and if it is necessary, he can modify the grade given by the tool.

III. The strategy

In each class session, several programming exercises are published in the tool, each one a little bit more difficult than the previous one, and some at the challenging level.

Students solve the exercises during class time while the professor is available to answer their questions. Prior to using the automated tool, the professor tried to implement the same approach but had to go to each student to check the solutions before they could move to the next exercise. Now, thanks to the use of the automated tool, the student receives immediate feedback on whether the code is correct or not. As a consequence, he can move forward to the next problem or ask for the teacher's support if needed; hence the professor's time is focused on helping the students who are struggling with the concepts.

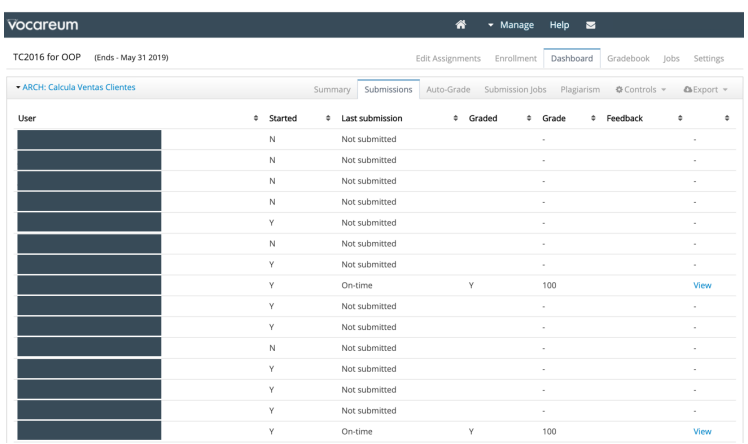

Fig. 2. Professor's dashboard view. On the left side, the student's name. In the right column, a link to the students' solutions (View).

Using the tool enables the learning pace to be personalized because the students who have the correct solutions can continue with the subsequent exercises, while the students who are having problems can work to develop correct solutions at the pace they need for learning. This is especially relevant in this introductory class because there are differences in the mastery levels

that entering students have in the basic programming concepts; i.e., while there are some students who learned to program in high school or using online courses, there are others who have never written a line of code.

IV. Assessments

Comparison of students' grades

We compared the groups where the immediate-feedback strategy was applied (standard and honors experimental groups) with similar groups taught by the same professor during the previous semester where the strategy had not been used (standard and honors control groups). We chose a monthly exam covering the same topics for each one of the four groups.

Due to the size of the groups, we decided to use the Kruskal-Wallis test as a non-parametric option to ANOVA to find out whether the implementation of the mastery-learning approach, supported by the immediate feedback tool, has a significant impact on the students' exam grades.

Students' perception

To know the students' perceptions of the strategy used, we developed a survey and gave it to both experimental groups using Google Forms. The survey included 8 closed questions with a 5-point Likert scale, where 1 corresponded to the lowest level and 5 to the highest; in addition, there were two open-ended questions to collect the students' opinions. The results of the closed questions can be seen in Table I.

The Mann-Whitney test was used to compare the students' answers because some of the questions do not fulfill the normality assumption required to make a t-student parametric test. On the other hand, the open questions were classified by keywords and topics to identify the most mentioned concepts.

Additionally, as complementary information, the institutional evaluation that is applied to the students at the end of each semester was analyzed, specifically looking at the comments related to the implementation analyzed in this work.

Question number	Question
1	According to my experience in this course, I consider that the automatic tool allows me to solve more exercises during the class session compared to sessions where the tool has not been used.
2	The immediate feedback obtained when using the tool makes me feel more motivated to do the course exercises.
3	The use of the tool allows the professor to interact more personally with the students during the class session.
4	I consider that the tool allows me to perform the class exercises at my own pace.
5	I believe that doing the class exercises at my own pace has a positive impact on my learning process.
6	I believe that the fact that the tool allows me to do the exercises at my own pace, without forcing me to follow my classmates' pace, has a positive impact on my learning process.
7	I prefer to have an exercise, exam or programming challenge using the tool than the traditional way.

8	In general, I believe that the use of the tool impacts my learning process in a positive way.

Table I. Questions in the survey applied to the experimental groups.

Results

Grades Analysis

For the Kruskal-Wallis test, the null hypothesis H_0 was that the samples come from the same population, while the alternative hypothesis H_A was that the samples do not come from the same population. The results of the analysis are that the p-value (two-tailed) is 0.1118. As this is higher than the significance level alpha = 0.05, one cannot reject the null hypothesis H_0. The risk of rejecting the null hypothesis H_0 while it is true is 11.18%. This means that there is no significant difference in the students' grades because of the innovation.

Students' Perception Analysis

A Mann-Whitney test was used to compare student responses to each question in the survey. The null hypothesis H_0 was that the distribution of the two groups was equal, and the alternative hypothesis H_A was that the means of the two groups were not equal.

In Figure 3, the means of all the question responses are shown.

In the experimental standard group, the means for all questions are lower than in the experimental honors group. So, it is possible to say that the students' satisfaction is lower in the standard group than in the honors program group. Also, it can be seen that the mean for all the questions in the honors group is greater than 4, which can be interpreted as a higher level of students' satisfaction with the strategy in the honors group.

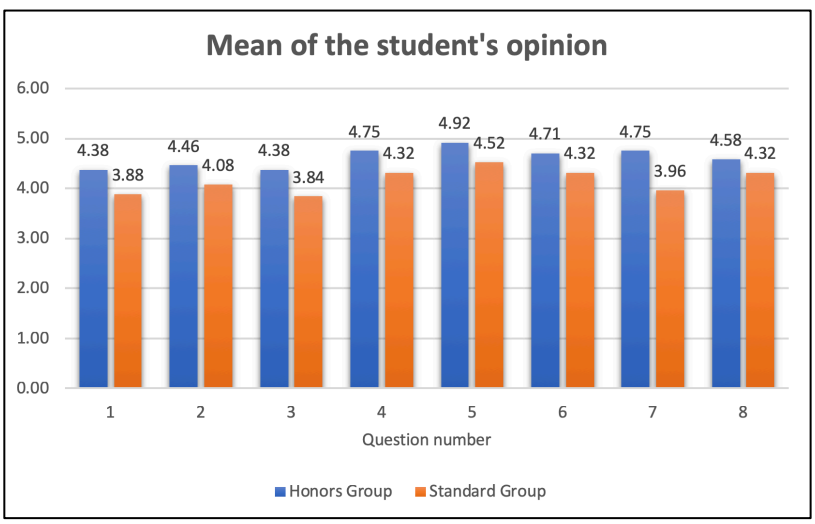

Fig. 3. Means of the students' opinions for the questions of the survey

The previous finding is confirmed by the result of the Mann-Whitney test, which shows that there is a significant difference in the means between the groups for questions 1 (p-value = 0.015), 3 (p-value = 0.035), 4 (p-value = 0.013), 5 (p-value = 0.010) and 7 (p-value = 0.008).

To complement the quantitative results, we analyzed students' answers to the open-ended questions based on the key ideas expressed in each paragraph. Figure 4 shows the three main concepts mentioned by the standard and honors group students for the question, "What do you like most about the automated tool?"

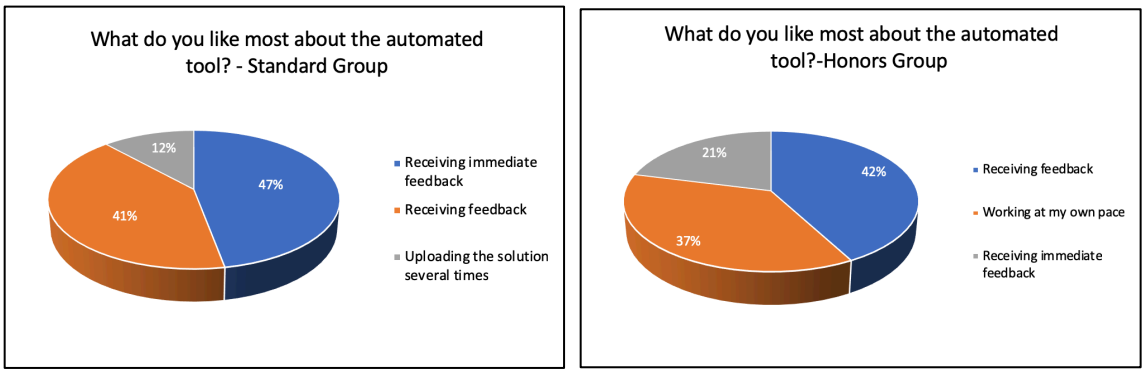

Fig 4. The standard and honors group students' responses to the question, "What do you like most about the automated tool?"

Comparing students' responses, it is possible to notice that both groups identify feedback as a primary advantage of the automated tool, but the importance they give to it varies; i.e., while for the standard group, the fact that the feedback is immediate is what they liked the most about the tool; in the honors group, working at their own pace exceeded receiving immediate feedback. Feedback has been identified as one of the pillars of mastery learning. The results of this implementation show that students recognize it as a main advantage in their learning process. Another interesting finding is that for the more advanced students (honors), feedback is not seen as the main advantage but as an enabler to work at their own pace. In other words, it is through this they are able to personalize their learning process.

Figure 5 shows the three main concepts mentioned by the standard and honors group students for the question, "What do you like least about the automated tool?"

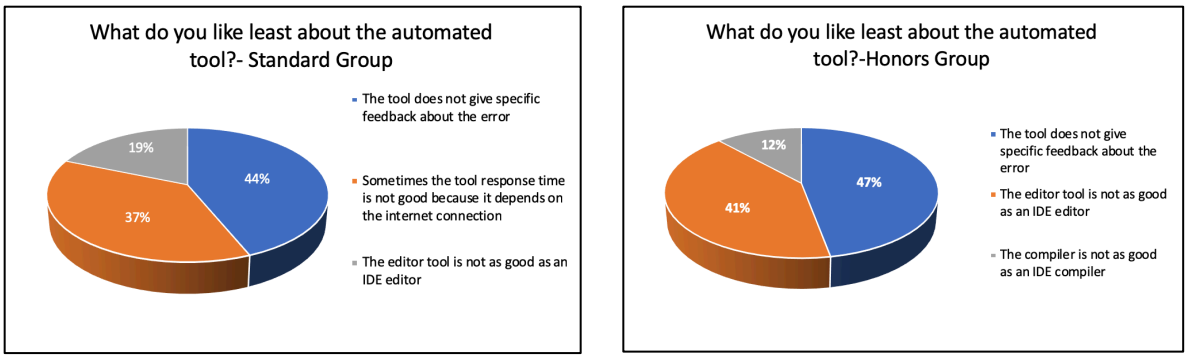

Fig 5. The standard and honors group students' responses to the question, "What do you like least about the automated tool?"

Analyzing the responses for this question, we see that the feedback concept appears as the top answer in both groups, confirming its relevance for students. In this case, they mention that receiving more specific feedback about the errors identified by the tool is desirable. Fortunately,

more specific feedback about errors is something that can be configured by the teacher to improve the experience in later implementations. The other responses refer to the specific capabilities of the tool and the internet connection and are not explicitly related to the learning process nor the methodologies used in the class.
In addition, the comments received from the students in the institutional evaluation were considered. Table II shows the students' comments received in the institutional evaluation of both the experimental standard and honors groups. As can be seen in the table, students recognize the possibility of moving at their own pace and being able to perform challenging problems according to their own skills. Given the number and types of comments received from each of the groups, it is possible to say that the students in the honors program expressed a better perception of the innovation. This confirms the results of the statistical analysis.

Group	Comment
Standard	• The professor uses software tools to make the class more interactive. • She always seeks to challenge you and allows each student to advance at his own pace.
Honors	• Allows each person to move at his own pace • She makes you try as hard as you can so that you can learn better. • The class has been challenging. • If the professor realizes that you are having problems solving your exercise, she helps you personally.

Table II. Comments in the final institutional evaluation of the course.

Discussion: Professor's Opinion

In a Computer Science 1 course, the professor usually defines several exercises for each topic to solve in class or as homework assignments. Thanks to the use of the automatic tool, it has been possible for professors to increase the number of additional programming exercises. Before using the automatic tool, the same exercises were available through the LMS Blackboard, which is the official tool in the institution to deliver content to students, but it does not give the students the possibility to know whether their solutions are correct immediately. The feedback depends 100% on the teacher's availability to grade and comment on each individual submission after the activity has been delivered.

During this implementation, this situation changed a lot. Now, thanks to the use of the automatic tool, students can evaluate their work when they submit the program and know whether they are ready to move on to the next, more complicated exercise. This represents a significant improvement in the teaching and learning process. According to the professor's perception and to the survey results, students have shown more interest in solving the problem exercises, and, usually, they keep working until they get them correct for all the test cases (which are configured by the teacher).

As mentioned earlier in this paper, each topic has several exercises, ordered by level of difficulty and available through the automatic tool during each class session. This allowed students to advance at their own pace and solve challenging exercises using an incremental approach. In this way, students have been able to work individually to solve each problem. This represents an improvement in their personal learning process because the professor does not show the exercise solution to the complete group; instead, she goes to each student who is struggling. This kind of personal interaction helps the student to focus on his own solution and leaves open the possibility of asking for advice from the professor when he needs it.

This personalization of the learning process is the most significant advantage that the automated tool offers to students and teachers because it is possible to recognize the fact that each person learns at his own pace.

Finally, this methodology allows the professor to make better use of her time. Thanks to the use of the automated tool, her time is mostly spent with the students who have not been able to obtain correct answers; the rest of the students already know their solutions are correct and they have moved on to more challenging problems.

Conclusions

Using an automated tool to give immediate feedback to students in a Computer Science1 class allows them to advance at their own pace, which has been a positive experience for both the students and the professor.

The students of the honors program group have a better perception of the strategy used. According to the comments of the survey, they perceive that this strategy has a positive impact on their learning process.

In the opinion of the professor, the possibility of each student advancing at his own pace is the main advantage offered by the use of the automated tool.

It is important to consider that the results of this implementation can't be generalized but can be considered as initial input for further implementations that would extend and analyze the effects of immediate feedback in the mastery learning approach.

Acknowledgment

The authors would like to acknowledge the f support of Writing Lab, TecLabs, Tecnologico de Monterrey, Mexico, in the production of this work.

Bibliography

[1] C. Ott, A. Robins, and K. Shephard, "Translating Principles of Effective Feedback for Students into the CS1 Context," *Transactions on Computing Education,* vol. 16, no. 1, pp. 1:1-1:27, January 2016.

[2] J. Bennedsen and M. E. Caspersen, "Failure Rates in Introductory Programming," *inroads - The SIGCSE Bulletin,* vol. 39, no. 2, pp. 32-36, June 2007.

[3] E. Lahtinen, K. Ala-Mutka and H.-M. Jarvinen, "A Study of the Difficulties of Novice Programmers," in *ITiCSE*, Monte de Caparica, Portugal., 2005.

[4] A. Bey, P. Jermann and P. Dillenbourg, "A Comparison between Two Automatic Assessment Approaches for Programming: An Empirical Study on MOOCs," *Educational Technology & Society,* vol. 21, no. 2, pp. 259-272, 2018.

[5] T. Guskey, "Lessons of Mastery Learning," *Educational Leadership,* vol. 68, no. 2, pp. 52-57, 2010.

[6] K. Koile and D. Singer, "Improving Learning in CS1 via Tablet-PC-based In-Class Assessment," in *ICER*, Canterbury, United Kingdom, 2006.

[7] "Vocareum programming lab," [Online]. Available: https://www.vocareum.com/home/programming-lab/. [Accessed 9 2017].