Improving computer science lab feedback methods

Sanish Rai
West Virginia University, Institute of Technology, sanish.rai@mail.wvu.edu

***Abstract* - In computer science programming courses such as Java, C, Python, C++, the computer science (CS) lab plays the most significant role in helping freshmen students to learn the coding for the first time. In the labs, students work on some programming assignment problems and submit them on an online platform to be graded by instructors. The labs are designed to get students hands-on coding and implement the programs on the computer. Traditionally, the grading feedback is provided after a week or more, many of which are ignored by the students. As such, in this work, a one-on-one grading feedback methodology on completion of the program in the lab was conducted. Along with feedback, instructors would ask various questions to students related to the problem to understand their knowledge, thinking process and at the same time, enhance the communication skills of students. A quantitative study of the process using survey data showed that this method had a positive impact on students without causing any additional burden on instructors.**

Index Terms – Computer science labs, student feedback, instruction methods.

INTRODUCTION

Basic programming skills are not only necessary for Computer Science majors but are an important skill just as basic Math, Physics and Chemistry for students in all majors. As such Universities are making introductory programming courses as required in all the curriculums. Writing programs and executing them to see the program's output is as necessary as doing physics or chemistry experiments. Universities have recognized the importance of practical lab components for a computer programming course and so most of the courses are accompanied by a separate lab hour. However, in many situations, these labs become just a place for students to write programs and submit for grading. There is much need to research and study on how to efficiently conduct a computer programming lab.

In literature, we find a handful of approaches done by various faculties to test various methods of conducting programming labs in order to increase the lab hour effectiveness. In [1], the authors have presented their work on using pair programming techniques to improve computer science experience. It was observed that when students work in pairs, they had a more positive attitude towards working in collaborative environments and provided better understanding to students and helped in student retention in the CS course. In another work [2], the authors used Microsoft Research's Kodu Game Lab to teach computer science programming concepts to students. It was observed that students were actually being interested in the programming and learning concepts and not only playing games. The authors in [3], developed a new lab-based format for computer science instruction where the course was converted to totally lab-based and instructors' role was converted to a tutor. The results showed that this method enhanced the learning experience for students while improving the communication between instructors and students. In [4], an assessment is done in the effect of using closed labs which showed a qualitative improvement in student learning.

With increasing enrollment in computer science and need of programming skills in everyone, there needs to be more experiments and research on improving computer programming teaching along with diversity in students and the size of the class. In this work, we want to look at how CS labs can be made effective and instructors can use it to not only teach programming syntax to students but improve the coding habits and thinking process as well. This work will mainly focus on providing effective feedback to students without much extra work to the instructors. The CS lab implemented is defined as a closed lab [5] in which students work on programming assignments during a fixed laboratory session. Traditionally, the student submission is graded by the instructor at their convenience and the feedback and grades is returned back to students. In this method, the lab grading by the instructor is done by reviewing the student submission file by sitting together with the student and reviewing it together as soon as the student completes it. The instructor will ask the student to explain their source code by emphasizing various checkpoints. This will allow the instructor to understand the thought process of the students, and provide personalized, efficient feedback to the student based on their submission. This will also allow the instructor to ask other related questions to encourage critical thinking to students. Since the instructor would not need to repeat the grading again, it will not take more time in grading.

A quantitative study of the implementation of this grading process is done where the efficiency of the method has been measured by various instructor/student surveys and feedback. In this work, the computer science lab will be based on Introduction to Computer Science 1 (CS1) and 2 (CS2) using Java for freshmen students.

GOAL

The CS Labs are designed in different ways in different Universities and also depends on the instructor teaching the main lecture course. Traditionally, the main instructor

978-1-7281-7520-1/20/$31.00 ©2020 IEEE

teaches the theory of the programming language in the lecture time and practice is done during the lab. Lab instructor designs the lab where normally students work on various programming questions and code the solutions in certain programming languages. These solutions are graded by the instructors and feedback is provided to the students online, similar to the assignment/quiz grading.

While the CS lab is the most effective place where the students learn to code, many students might not view it the same way. They think CS lab just another class which they need to attend, just like a lecture class. For them, lab assignments are just a task and they want to complete it, submit it and get done with the lab. They just want to get grades back and most students do not even read the feedback provided back to them. As a result, an opportunity for the students to learn and understand the practical aspects of the underlying concepts and an opportunity for instructors to enhance student learning and keep them motivated is wasted. This can seriously hinder student progress as well as motivation in computer science courses and create retention in CS major [6].

In general, these are some of the disadvantages of traditional lab teaching method:

- Difficult to assess if the student is properly understanding the submitted program and the underlying concepts
- Difficult to provide effective feedback on the programming skills of the student
- Difficult for the students to understand and implement the written feedback from the students
- Many students look at only the score and not even read the instructor feedback
- It creates a barrier between instructor and student and does not allow open communication between them
- Students cannot explain their code to other people
- Causes retention problem due to lack of student progress

In this work, we modify the labs so that the feedback is given directly to students and at the same time students are checked for their understanding. These are some of the goals we want to achieve from this work:

- Properly assess a student is understanding the concepts of programming
- Enable a student to be able to explain their source code
- Enable a student to explain the computer science concepts in their own words
- Provide an opportunity for the students to talk one-on-one at least once in the lab and ask questions or present any concern
- Enhance student learning experience as well as their personal development by encouraging them to communicate with instructors

LAB SETUP

Computer science introductory courses, Introduction to Computer Science I and II labs were modified to implement this method of lab instruction. In this course, students use Java programming language to learn introductory computer science courses. The labs are designed to get students hands-on coding and implement the programs on the computer. As such many times, students require one on one support to get help with their code. Lab assistants (undergraduate students) are used to assist instructors to conduct the lab successfully. Qualified lab assistants are hired and trained to provide immediate feedback to the student's source code and help them learn quickly as well as to establish good coding habits. The lab for this study has an average size of 15 students and in one semester there are 4 lab sections in CS1 while 3 sections in CS2. For this size of students, one instructor and a lab assistant are enough whereas any increase in the number of students would require more lab assistants.

In each lab, students are provided with a lab assignment where each assignment is divided into multiple sections, primarily two. The assignment directly relates to the lecture they would have already covered and requires a programming solution code for some given problem. Students work independently on the assignment in the lab, requesting for help from instructor and lab assistant whenever needed. Also, instructors and lab assistants periodically check the students for any issues or support.

When a student completes a section, they inform the instructor. The instructor will use a rubric to check the solution code, at the same time the instructor will check the student's understanding of the source code. The instructor will ask several questions that allow comprehending student's depth of knowledge. Some examples of these questions are:

- Briefly explain your source code
- Could you tell me what this part of the code does?
- Why did you use this method?
- Can you tell me what will happen if we change this part of the code?

While a standard rubric allows the instructor to check if the student's solution is complete or not, the set of questions to evaluate understanding depends on the particular problem and the student source code. As such it is not possible to create a standard set of questions for instructors to ask the students. However, with the examples of the questions given above it is very easy to measure students understanding to some degree.

In the same time, instructors also inform the students of the mistakes in their codes and how it can be improved, other improvements that could be done to make their code efficient, explain the concepts students are having issues with, and also check with students if they have any other questions or difficulty. If students are making errors in their code, they are informed of those errors and given an opportunity to fix it without penalizing. Students liked this approach since they could put effort and get an opportunity to earn points even if they may mistake in the first attempt. Also, they would get instant feedback rather than waiting for a week when they would forget about what code they wrote.

SURVEY RESULTS

The new method of lab assessment was carried out with CS1 and CS2 students for a period of two semesters. To evaluate the results, an anonymous survey was carried out at the end of the semester with both the instructors and students. The student survey consisted of 13 questions with questions tailored to understand their perspective on the lab assignments, the effectiveness of the assessment technique and their suggestions for improvement. Some of the examples of the questions are given below:

1. The instructor graded most of the assignments by checking completion with you and provide you feedback. Do you think it was a good approach? (Select one)
2. Was the feedback provided by instructors during grading helpful? (Select one)
3. Lab TA has been helpful with labs.
4. The difficulty level of lab assignment questions is very high.
5. Any advantage or disadvantage of this style of grading? Write comments.

The response for questions 1 to 4 was collected in a scale of 1 to 5, where 1 is the lowest (worst) and 5 is the highest (best) ranking.

The survey was taken by a total of 45 CS1 and CS2 students. TABLE I presents the result of the survey from CS1 students while TABLE II presents from CS2 students. About 82% of the CS1 students and 88% of the CS2 ranked the instructor feedback method greater 4. 95% of students in both sections thought that the undergraduate lab assistants were helpful. The difficulty assessment of the lab assignment questions showed that 32% of the CS1 students and 22% of the CS2 students thought the lab difficult (ranked more than 4).

The overall survey results were summarized in TABLE III. As seen in the table, on the question relating to the effectiveness of the approach (Survey question 1), 88.89% of the students thought that the method was a good approach most of the time for providing feedback. Similarly, more than 80% of the students thought that the feedback provided by instructors during grading using this method was useful for them. Also, more than 95% of students thought that the lab assistants were helpful in the lab while 4.5% of the students thought the lab assignment questions were very easy, 73% thought it was medium and 4.5% thought the question was difficult.

TABLE I
SURVEY RESULTS FROM CS1 STUDENTS

Question	1	2	3	4	5
The instructor graded most of the assignments by checking completion with you and provide you feedback. Do you think it was a good approach?	0	4.5	13.67	22.7	59.13
Was the feedback provided by instructors during grading helpful?	0	0	23	23	54
Lab TA has been helpful with labs.	0	4.5	0	31.5	64
The difficulty level of lab assignment questions is very high.	9	0	59	23	9

TABLE II
SURVEY RESULTS FROM CS2 STUDENTS

Question	1	2	3	4	5
The instructor graded most of the assignments by checking completion with you and provide you feedback. Do you think it was a good approach?	0	2.22	8.9	22.22	66.66
Lab TA has been helpful with labs.	0	4.34	0	17.39	78.27
The difficulty level of lab assignment questions is very high.	4.5	0	73.3	17.76	4.44

TABLE III
AVERAGE SURVEY RESULTS FOR CS1 AND CS2

Answers	Survey question 1 results (%)	Survey question 2 results (%)
not at all	0	0
a little	2.22	0
sometimes	8.89	20
most of the times	22.22	15
always	66.67	65

For survey question 3, all the students mentioned that the approach had an advantage. A few of the comments are listed below:

"Advantage is that the instructor can evaluate if the student understands the code"
"It was helpful to get one one-on-one time with the professor to look at their code"
"We were given a deeper understanding of the content and assignments "
"I enjoyed receiving feedback based on what I did right and wrong. It allowed me to fix mistakes without losing a good score."
"The instructor explains if your code is wrong and teach you and explain how to fix it you could get instant feedback and help from the instructor"
"An advantage, you still get credit if you are correct, but you are given more efficient ways of writing code so you know for next time"
"The instructor was able to make sure I did things right and gave me advice"
"Helps find better/shorter ways of completing it."

A survey was also done with instructors teaching the CS labs, the results showed that instructors preferred this method of grading compared with the traditional method of writing feedback on the online portal. It was also observed that students did not feel anxious/scared to show their solution face-to-face with instructors. On average, 4.3 minutes was spent on grading with one student which is less than time spent with traditional grading.

DISCUSSION

Besides the goal of achieving improved programming practicing and in-depth understanding of computer concepts, this method also helps students in their professional development. Mainly, it prepares students for technical interviews such as whiteboard coding tests where the interviewee needs to not only write the code but needs to explain the coding process to the interviewer. We believe that this method of grading where students are asked to explain their code improves their confidence as well as the method of conveying their work process.

From the surveys, we observed that students liked this method of receiving active feedbacks rather than getting passive feedbacks through the online grading system which the students often ignore. Also, from the perspective of instructors, the time required for grading is the same as the traditional approach and does not take additional time. One of the disadvantages, however, was that sometimes multiple students would complete the problem at the same time and thus students might be waiting to get feedback. And some other times, students would wait till the end to show their work to the instructor thus the lab duration would exceed the allowed time to complete the grading process. These issues could be resolved if instructors keep on checking if the students completed some portion of the question and grade it immediately thus not causing grading bottlenecks at the end of the lab. Furthermore, the implementation of approaches such as peer-programming could reduce the grading time using this method significantly.

CONCLUSION

In this work, immediate one-on-one grading feedback methodology was implemented in Introductory computer science programming labs. The method showed improved experience in student's programming learning and communication skills compared with the traditional method of grading and writing comments online for students to receive at a later time. This method allowed students to reflect immediately on their code and improve it, as well as learn better ways of writing code. Students also practiced communicating their source code and explaining computer science concepts to instructor thus gaining experience for future interviews.

In the future, other methods of improving computer science labs such as pair programming and feedback in groups, whiteboard explanations, etc. will be explored.

REFERENCES

[1] Nagappan, N., Williams, L., Ferzli, M., Wiebe, E., Yang, K., Miller, C., & Balik, S. (2003). Improving the CS1 experience with pair programming. ACM SIGCSE Bulletin, 35(1), 359-362.

[2] Stolee, K. T., & Fristoe, T. (2011, March). Expressing computer science concepts through Kodu game lab. In Proceedings of the 42nd ACM technical symposium on Computer science education (pp. 99-104).

[3] Clancy, M., Titterton, N., Ryan, C., Slotta, J., & Linn, M. (2003). New roles for students, instructors, and computers in a lab-based introductory programming course. ACM SIGCSE Bulletin, 35(1), 132-136.

[4] Kumar, A.N., 2003. The effect of closed labs in computer science I: an assessment. Journal of Computing Sciences in Colleges, 18(5), pp.40-48.

[5] Parker, J., Cupper, R., Kelemen, C.., Molnar, D., and Scragg, G.: Laboratories in the Computer Science Curriculum, Journal of Computer Science Education, 1990, 1(3): 205-221.

[6] Wooley, B.A., 2003. Utilizing a computing lab to improve retention and recruiting of computer science and computer information science students. Journal of Computing Sciences in Colleges, 18(3), pp.228-234.

AUTHOR INFORMATION

Sanish Rai, Assistant Professor, Department of Computer Science and Information System, West Virginia University Institute of Technology.