Kellie Keeling

Business Information & Analytics

Programming Instruction Tips: Using Code Reviews and Rewind/Rev Up Materials

The Challenge

When teaching programming courses, several issues arise:

- Making sure the students are prepared before class.
- Allowing for multiple levels/speeds of students to be able to interact with the material.
- Providing feedback quickly to help students move forward with the material.

The Approach

Making sure the students are prepared before class:

- Must watch video lecture before class
- Must complete multiple-choice quiz before class
- Complete Top Hat (in class quizzing) during class

Multiple levels/speeds of students to be able to interact with the material

- Students are provided:
  - **REWIND materials**: additional practice problems/materials – Extra Credit for Top Hat/Case Problems
  - **REV UP materials**: more advanced materials and problems related to the material, but not essential for everyone to learn
- During class, additional explanations/demos are presented, and Case Problems to be completed at the end of class are available before class

Providing feedback quickly to help students move forward with the material

- Students submit answers to Case Problems at end of class and are immediately given my answer **OR**
- Students participate in **CODE REVIEWS** bringing completed problems to next class and students self and team grade and also fix everyone's code before leaving.

How it Went

CODE REVIEWS: Having each person present their code gave a sense of responsibility to make sure it was mostly complete before class. Letting students pick their own groups relieved some anxiety. The better students were able to learn more by debugging the struggling students code and seeing other ways to complete problems. The struggling students got quick feedback from a couple of peers. I also walked around during this time and could help if the group could not come up with a solution.

REV UP/REWIND: A good source for REV UP were answers to questions generated during class. Students seemed motivated to complete REWIND questions for very small extra credit points.

QUIZZES ON LECTURE MATERIAL: I think making the students complete a quiz on the material before class and also during class (using quizzing software like Top Hat) helps to increase the accountability for the material and helps them to be more prepared to listen to my demos and lecture "wrap-ups" during class and be able to get started/complete the in-class Case Problems quicker.

Advice for Others

CHEATING ON QUIZZES: It takes a lot of work to create the multiple-choice questions with different versions, I would set this up once and then not stress about students sharing answers this quarter or another quarter. To alleviate concerns, just make the point values a small part of their grade.

SELF-GRADING: Having the students "self-grade" gives you a great chance to just focus on folks that are really "not getting" it. I have either had a few multiple-choice questions to check answers if I think they can complete it all or have used a rating system like shown:

Ratings					
On Task Whole Time 100.0 pts	On Task and completion acceptable 90.0 pts	Completion Average 80.0 pts	Completion Below Average 65.0 pts	Minor Work 45.0 pts	No Marks 0.0 pts

CODE REVIEWS:

Provide some type of "code checks" for the Code Reviews to help students make sure they have the correct answers; encourage them to complete most of the code.

Provide an online rating in Canvas for students to rate themselves and team members. I have found them to be quite honest. The reflective questions were informative.

Rating System:

- Assignment Complete at Beginning of Class/Passes Code Checks (Yes, Mostly, No)
- Actively Helped Discuss Code During Review (Yes, Mostly, No)
- Score for Today (100=almost all working, 90: tried everything/good effort, 80: participated)
- What is one thing you found valuable today/What did you learn that helped fix your code?
- What is one thing you can do to improve before the next class/code review?