Lesson 2.09 — Programming Project

Overview

Objectives — ***Students will be able to…***

- **Ideate, plan and construct** a structured program containing nested loops.
- **Share and evaluate** one another's code

Assessments — ***Students will…***

- **Submit** a complete, functional program by the end of class.

Homework — ***Students will…***

- **Complete** practice questions with class constants

Materials & Prep

- **Projector and computer** (if you are able to/opt to use Eclipse with your students)
- **Student self-help system** (such as C2B4 or student pairing)

Today is day 2 of the programming project; be sure to remind students that they only have the first 30 minutes of class to finish and submit their program. Be prepared to review the correct code and offer an upgrade using class constants for the second half of class.

If you have not finished grading notebooks, make sure you are set up to grade student notebooks today. If possible, you should only collect 3–5 notebooks at a time so students have their notebooks available to reference during programming time.

Pacing Guide

Section	Total Time
Bell-work and attendance	5min
Programming project	30min
Introduction to class constants	10min
Students update their code	10min

Procedure

Today's lesson will be a combination of drilling the parts of a basic program, and conditioning students to check for common errors. To hook your class, have pictures of punch cards and punch card readers up when students enter. If possible, have physical punch cards available to pass around the room for tactile learners as you explain the origins of the phrase "bug" and "debugging."

Bell-work and Attendance [5 minutes]

Programming Project [30 minutes]

Have students continue the programming project, aiming to finish with about ten minutes left in class.

Introduction to Class Constants [10 minutes]

1. Present on a board or projector a complete Java class that solves the programming project. Ideally, this would be a student solution. Walk the class through the solution, having students help you trace the flow of control and predicting output to confirm that the program works.
    - Suggest that there might be an easier way to update repetitive values at once, and introduce **class constants**.

2. Trace or run the program with the updated class constants to demonstrate that this does indeed work.
3. If you are running the program in Eclipse, show that you can easily change the constants. Change the values several times, running the program each time to drive this home.

Students Update Their Code [10 minutes]

Give students time to update their project code, now including class constants.

Accommodation and Differentiation

If you have students who are speeding through this project, you should encourage them to act as student TAs and help struggling classmates (NOTE: you should specifically direct students NOT to give answers, but to help students think of ideas on their own.)

If you have students that are struggling during this class (and you will), resist the urge to help students too much at this stage. Ask leading questions, direct students to their notes, or an example that demonstrates a similar solution, but don't give students the answer here. Resilience/grit is an important emotional tool for solving complex programming problems: the emotional journey students take during these difficult programming problems is as important as the actual coding challenge.

If students are having trouble due to language, pair students up so those with more advanced English can help those that are emergent language learners.

Forum discussion

Lesson 2.09 Programming Project (TEALS Discourse account required)