

Lesson 2.08 — Programming Project

Overview

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

- **Plan and construct** a structured program containing nested loops.

Assessments — *Students will...*

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

Homework — *Students will...*

- **Outline** Chapter 2, omitting BJP 2.5

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)

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
Introduction & classroom procedures	10min
Programming project	30min
Students trade work, check, & submit	10min

Procedure

To prepare students for the upcoming unit exam, the next few class periods will be devoted to reinforcing concepts and applying the tools, procedures, and code that were introduced this unit.

Bell-work and Attendance [5 minutes]

Introduction and Classroom Procedures [10 minutes]

1. If your computer time requires you to move to another room or to change seating, you should teach and/or review those procedures before introducing the lab material. It's been a few weeks since the last long form programming assignments, so make sure to ask students what the procedures are if they:
 - have gotten stuck (check pseudocode and structure diagram),
 - finished early (move on to challenge questions), or
 - can't remember a coding rule or procedure (check your notes, worksheets, and textbook, C2B4)

Unless you have had students submitting work electronically regularly, you should model and review those procedures before students begin work.

2. Introduce the programming project, taking a moment to talk strategy with your class.

PROGRAMMING PROJECT: Write a program that produces the following figure as its output using nested for loops.

```
| " " " " " " " " |
\ : : : : : /
\ : : : : /
```

```

\:::/
 \:/
  /:\
 /::\
/::::\
/:::::\
/:::::\
| " " " " " " " " |

```

TIPS: Start with a structure diagram or writing out steps in English as pseudocode. Try to isolate repeated tasks into methods. Include comments in with your code so others can easily understand what the code is supposed to do.

3. Ask your class for suggestions as to how to tackle this programming problem. Students should suggest drawing a structural diagram, building the program one method at a time (iterative development), and following the correction steps on their personal algorithms (debugging).

Procedural decomposition is hard! As a group, ask students to discuss what components go into drawing each line.

- What characteristics stay the same for each line? (Slashes, colons, spaces)
- What characteristics could we use a loop for?
- What might we want to make its own method that we can call more than once?

Programming Project [30 minutes]

Get students started on the ASCII art programming project. It can be the hour-glass picture, or something different! Offer students help after they have tried to answer the questions themselves:

- a. Have they checked the book for examples?
- b. Have they asked a friend (or two) for help?

If students seem to be getting stuck on the same segment of code, offer a hint or tip on the board (silently, without disrupting student flow).

If the entire class is stuck, return to whole group and work through the programming challenge together as a class, having students offer an increasing proportion of the answers as you move along.

Students trade work, check, evaluate, and turn in [5 minutes]

At the end of class, have students look over each other's projects before submitting.

Evaluation Question: How many lines of code did this program "save" by using for-loops? How can the code be modified for some other interesting effect, or embellishment?

Accommodation and Differentiation

If you have students who are speeding through this project, you should encourage them to:

- Finish the programming project started in class yesterday.
- 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 2.08 Programming Project (TEALS Discourse account required)