CS1: Final Project

CS 140 with Dr. Sam Schwartz

Brainstorm due: Tuesday, December 4
Final Project due: Sunday, December 14 at 11:59pm via Canvas Upload

1 Purpose

The purpose of this individually produced 200 point assignment (that will not be dropped) is to:

1. Synthesize the concepts learned throughout the course into one cohesive project.
2. Strengthen your ability to independently design, implement, and test a program.
3. Demonstrate clear, professional documentation and problem-solving ability.

2 Tasks

You will create your own program for the final project.

2.1 Program Requirements (140 points)

This program must incorporate and demonstrate mastery of the following concepts:

1. Algorithmic thinking and abstraction (e.g., using methods)
2. Variables
3. Data types and type casting
4. Primitives and Objects
5. Input / Output (e.g., Scanner)
6. Conditionals (If / else if / else structures)
7. Test variables
8. Try/Catch
9. Arithmetic & operator precedence
10. Methods with parameters
11. Methods with returns
12. For loops
13. While loops
14. Nested loops
15. Sentinel-controlled loops
16. Infinite loops
17. Strings and string methods
18. Arrays and array traversal
19. Multiple classes
20. Appropriately commented with JavaDoc

It goes without saying that everything should also be appropriately scoped and visible. (E.g., correct use of public vs private; static vs instance; etc.)

Your code should also be formatted nicely. No programs all squished on one line, please.

Each of the above concepts is worth seven points. These points are split among the key concepts of:

- Validity: e.g., does the code work bug free?
- Readability: e.g., named thoughtfully using standard conventions, commented well, etc.
- Fluency: e.g., it was executed in such a way that an experienced practitioner would not find it "weird-in-a-bad-way"

2.2 Documentation (50 points)

Writing well is thinking well.

In addition to writing code, you will also document the following in a PDF document:

1. (5pts) The name of your project and a brief description of what it does. Also describe how it has changed (if at all) from your initial proposal.
2. (5pts) How to run your project. (E.g., where is the main method?)
3. (5pts) At least one screenshot showing your program running (multiple screenshots are fine).
4. (15 pts) At least three bugs you encountered during the writing of your project, what kind they were (compiler, runtime, logic, etc.) and how you fixed them.
5. (20pts) A brief comment for each of the above 20 requirements for your program describing where I can find at least one example where it's clear you've mastered the concept.

Example:

I demonstrated mastery of each concept in the following ways:

1. *Algorithmic thinking and abstraction by breaking my algorithm out into n methods, which can be found on lines a, b, c.*
2. *Using variables by declaring one on line d and printing it out on line e.*
3. *Data types and type casting by casting an integer to a double on line f.*

and so on, for each of the 20 concepts.

2.3 Project Proposal (10 points; Due Tue, Week 14)

Writing well is thinking well.

Before you begin, you will submit to Canvas a one-to-two paragraph sketch of the idea you propose to implement as your final project. This is due on Tuesday at 10:59am (just before the first CS 1 class of the day begins). This is 5 points.

On Tuesday (and Wednesday, if needed) Dr. Schwartz will meet with each of you individually during class to review your proposal and make adjustments. Attending this meeting is 5 points.

3 Criteria for Success

- Turn in PDF of proposed idea to Canvas (5pts)
- Have a conversation with Dr. Schwartz about it (5pts)
- Turn in a file called `README.pdf` (50 pts) which contains each of the following elements documenting your code:
  - Project name and description
  - How to run it
  - Screenshots
  - Bugs you found and how you solved them
  - List of each concept outlined in the *program requirements* section of this document and where I can find the corresponding implementation in your code.
- Turn in a file called `code.zip` (140 pts) which contains all of your code.