

We Solve It! Project

CPSC 1302K - Fall 2024

Teams can be up to a total of 3 students.

Purpose

The purpose of this assignment is to use the programming skills we have learned to develop a program that addresses a real-world problem. Outside of a classroom, whether you go to work for a company or start your own business, you will use programming to develop solutions to problems that have not been previously coded. Part of your job will be to identify problems and apply computational thinking to develop solutions. For this project, you must write Java code that uses all of the following to solve a real-world problem:

- Javadocs
- Class inheritance (using classes that you/your team wrote)
- A collection (array, ArrayList, Vector, etc.)
- Input validation
- Exception handling
- Files
- Testing (*e.g.*, unit testing)

Practice in these fundamental computer science skills as well as in real world problem solving are essential for success in future courses.

Tasks:

- 1) **Proposal:** Identify a problem that you or a team member encounters in their daily or weekly routine that a Java program would help with. You should identify a specific problem that can be addressed in a small application, working prototype or a proof of concept. Write a proposal (approximately one to two paragraphs) listing your team members and your selected problem. Include multiple (two to four) strategies for addressing the problem. Strategies can include solutions that are not possible yet. Propose one or more solutions for each strategy. Include the benefits and disadvantages of each solution. Indicate which solution your team will implement and why it was selected. For your solution, identify what will be the input(s) (for example, files, a user typing, etc.) and what will be the output(s). Also, indicate what the inheritance hierarchy would be (*e.g.*, UML diagram) for your chosen strategy (including what is unique to each class), what collection (array, ArrayList, etc.) that you're planning on using and how you're going to incorporate files. Here's a potential outline for your proposal:

Project title

Team members' names, their roles and what each one will do on the project

Background

Problem statement

Potential Strategy #1 (with advantages and disadvantages)

Potential Strategy #2 (with advantages and disadvantages)

Potential Strategy ... (with advantages and disadvantages)

Chosen strategy:

- Inputs and output
- Use of inheritance (also include name of parent class and child classes [make sure there is an "is-a" relationship] and unique fields of each one)
- Use of file(s)
- Use of one or more collections (arrays, ArrayList, etc.)



Your proposal must be approved by your instructor before moving on to coding your solution. Submit one per team in one of the following formats: .txt, .doc, .docx, or .pdf. **Submit in CougarVIEW by Saturday, November 2nd at 10:00 PM.**

Grading Criteria: 15 points

- a. Lists all team members' names
- b. Demonstrates the ability to construct a clear problem statement with evidence of many relevant contextual factors as it relates to real world scenarios. (Contextual Factors: Constraints (such as limits on cost), resources, attitudes (such as biases) and desired additional knowledge which affect how the problem can be best solved.)
- c. Identifies multiple possible strategies to the problem that apply within a specific context
- d. Identifies advantages and disadvantages for each possible strategy
- e. Identifies the chosen strategy that the team will implement (and why it was chosen)
- f. Identifies the input(s) and output(s) of solution
- g. Identifies incorporating inheritance, collection and files

- 2) **Code & Supporting Files:** Write a Java program that implements a solution to the problem identified in the proposal. Utilize all the coding elements listed above. Use only relative file names (so that they will work on my computer). **Also submit any files needed to run your program. Submit in CougarVIEW by Saturday, December 7th at 10:00 PM.**

Grading Criteria: 85 points

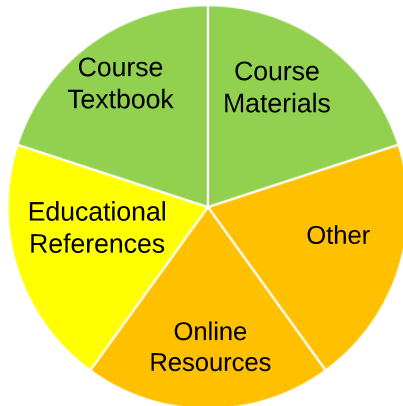
- a. Addresses address the problem selected and provide a solution
- b. Uses all of the Java elements listed above. For child classes, each one needs to have at least one unique field/instance data or overridden method.
- c. Execution is robust (meaning it can gracefully handle invalid inputs) and runs without errors (syntax, runtime, and logic) [so use input validation, etc.]
- d. Provides a usable interface. (Does the user know what is expected of them in terms of input? Can the user understand the output?)
- e. Code is properly documented with Javadocs at the top of each code file and with explanations throughout the code where necessary. description, parameters, and return value (when applicable).
- f. Tests are included in their own file (for example, `UserInterfaceTest.java`)

Note, have one team member submit each assignment and the other team members submit something that just indicates who submitted the assignment. This will allow for feedback in CougarVIEW.

This culminating assignment allows you to apply the concepts learned throughout the course. Consequently, authorized sources are limited. Additionally, you are encouraged to work extensively with your team members. Other authorized sources include the TA, CSU CS tutors and Dr. Carroll.

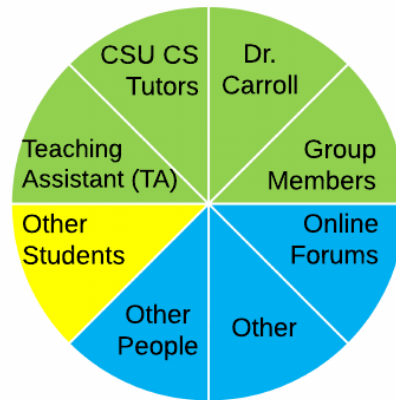


Where can you get help?



Encouraged Ask Dr. Carroll
 Need to Cite

Who Can Help You?



Encouraged Not Acceptable
 Need to Cite