CMSC 495
Trends and Projects in Computer Science

Final Project Description

In this project, you will design, develop, test, and document an approach using a high-level programming language such as Java to solve a specific problem or issue in computer science.

Instructions

This programming project was developed to satisfy several critical learning goals in the CMIS and CMSC programs at UMUC. This project, in addition to challenging you with creating a complex program in a high-level language such as Java, also requires you to analyze the design of your program and its impact on the results.

You will be expected to work in a group of two to four students. One of the primary goals of this project is to gain experience planning, leading, and working within a software development project. Except in special cases, each of you should submit all phases of the project individually to the gradebook.

You will work with your instructor to research and select a current problem in areas such as software design and architecture, systems and application security, mobile applications, database design and implementation, concurrent programming, signal processing, or algorithm performance optimization.

Due Date

Your final project is due by **December 14, 2021 at 11:59 PM EST**. Your instructor's policy on late projects will apply to this project.

Deliverables

1. All source code files (for example, in a Java project, all the dot-java files).
2. Any data files required to run the project.
3. Project documentation, which should include, at a minimum, the following sections: a. project plan.

A. Analysis results
B. Design results
C. Test plan and results
D. Source code (matching desired functionality)

Specific Functional Requirements for This Project

1. Work with your instructor to select a specific problem area in computer science, and provide a high-level description of the issue and the initial approach you envision for solving the problem. Your description should include why the problem is important to you and the field of computer science, along with the steps you will take to solve the problem.
2. The problem must be one that will require some object-oriented code in a high-level programming language, such as Java, to be designed, written, and tested. The final product will look different depending upon your approach and specific problem.
3. After you develop your solution, you should test your product, using appropriate and comprehensive data.
4. You should document your approach to solving the problem with detailed justification for the approach you selected. In this documentation, you should discuss possible issues, strengths, and weaknesses with your approach. Alternative approaches not selected, but considered, should be discussed in your documentation.
5. You should submit all the project deliverables to the proper gradebook entry, and submit them to the Final Project conference to share with the other students in the class.

Format

Textual Documentation Format and Length

The documentation describing and reflecting on your design and approach should be written using Microsoft Word and be of an appropriate length. The font size should be 12 point. The page margins should be 1 inch. The paragraphs should have double line spacing. All figures, tables, equations, and references should be properly labeled and formatted, using APA style.

Code Documentation and Style Requirements
- **Header comments:** The documentation requirement for all programming projects is one block comment at the top of the program, containing the course name, the project number, your name, the date, and the platform/compiler that you used to develop the project.
- **Class comments:** There should be at least one comment for each class in the program describing what that class does.
- **Other comments:** Additional comments should be provided as necessary to clarify the program.

Formatting
- Indentation must be consistent throughout the program.
- Variable and method names should be descriptive of the role of the variable or method.
- All constants, except 0 and 1, should be named.
- Constant names should be all upper-case.
- Variable names should begin in lower-case, but subsequent words should be in title case (e.g., finalSpeed).

Grading

This activity will be awarded 75 percent of the total grade in the course. The final project report elements will be assessed as follows:

Attribute	Value
Project plan	10%
Analysis	15%
Design	15%
Test plan	15%
Source code	20%
Total	**75%**