Creative Project

Advanced Applications

1. Introduction

Welcome to the Creative Project assignment for CMPG 121. This project is designed to allow you to apply the concepts you've learned throughout the module in a creative and meaningful way. You will have the freedom to choose the type of program you want to develop while adhering to specific technical requirements.

2. Project overview

Your task is to design and develop a C++ application that meets the following requirements:

2.1 Technical requirements:

- Implement at least one array.
- Use structs (you may exclude structs if using classes).
- Define and use at least three self-defined functions.
- Ensure good user interaction through both input and output.
- Use conditional statements (e.g., if, else, switch).
- Implement loops (e.g., for, while)
- Incorporate file handling for both reading from and writing to files.
- Handle possible errors gracefully, including input validation.

2.2 Advanced requirements:

- Use at least one class and create objects from it.
- Implement smart pointers.
- Utilise generic templates (which are necessary for smart pointers).

3. Assignment structure

The assignment is divided into two phases: Planning and Development.

Phase 1: Planning (30% of grade)

This phase focuses on planning your program and laying a solid foundation for development. It includes the following tasks:

1. Program Scope
   Write a brief description of your program, outlining its purpose, functionality, and intended users. Explain how your program meets the technical and advanced requirements.
2. Pseudocode
   Develop pseudocode for your program's main functions or algorithms. Ensure that your pseudocode clearly represents the logical flow and structure of your program.
3. Flowcharts
   Create flowcharts to visually represent the logical flow of your program or specific algorithms. These should complement your pseudocode.
4. Research on Advanced requirements
   Conduct research on the advanced requirements you'll be implementing (classes and objects, smart pointers, generic templates). Include explanations of how these concepts work and how you plan to integrate them into your program. Provide references to all sources you use.

Deliverables for Phase 1:

- A document containing the Program Scope, Pseudocode, Flowcharts, and Research on Advanced Concepts (references included).
- Submit this document by **Monday, 30 September** at noon.

Phase 2: Development (70% of grade)

In this phase, you will develop the program you planned in Phase 1. This involves writing the actual code, ensuring it meets all technical and advanced requirements, and providing detailed comments throughout your code to explain your approach and logic.

1. Program implementation:
   Develop your program according to your plan. Make sure to include the technical- and advanced requirements as mentioned before.
2. Extensive commenting:
   Provide extensive comments in your code to explain (in your own words) what each part does and why you implemented it that way.
3. Reflection on AI usage:
   If you used generative AI tools (like Chatgpt, Gemini, Co-pilot, etc.) during the planning or development phase, include a reflection in your final document. Discuss how AI assisted you, what you learned, and how you ensured the code submitted is your own work.
4. Demonstration video:
   Create a short video (no longer than 5 minutes) demonstrating your program. In the video, you should:
   - Briefly explain the scope of your project.
   - Run your program and demonstrate the key features, focusing on user interaction, navigate through your program's different features, and mention where requirements like file handling, arrays, and implementation of advanced requirements where used.
   - Next, explain parts of your code, specifically the advanced concepts you implemented, as well as parts with file handling and use of arrays.
   - Ensure that your voice is clear and the video quality is good enough to follow your explanation.

Deliverables for Phase 2:

- The Code::Blocks C++ project with all required features.
- A reflection document.
- A demonstration video showcasing your program and explaining parts of the code.
- Submit your project, reflection document, and demonstration video by **Monday, 14 October** at noon.

4. Grading criteria

Your project will be graded on the following:

- Planning (30%)
  - Clarity and completeness of the Program Scope (10%)
  - Accuracy and logic of the Pseudocode (10%)
  - Quality and relevance of the Flowcharts (5%)
  - Depth and clarity of the Research on Advanced Concepts (5%)
- Development (70%)
  - Functionality and adherence to technical requirements (25%)
  - Correct use of advanced concepts (20%)
  - Code quality, including readability and commenting (10%)
  - Error handling and user interaction (10%)
  - Quality and clarity of the demonstration video (5%)
- Bonus (10%)
  Creativity and originality of the project.
  This bonus is awarded for unique, innovative approaches to the project, including novel use of programming concepts, exceptional user interaction design, or other creative features that go beyond the basic requirements.

5. Submission guidelines

- All submissions must be made through the Assignment tool on eFundi by the due dates.
- Late submission will not be accepted.
- Ensure that your code is well-commented and that all required documents, including the video, are submitted together.

Good luck, and enjoy the creative process :-)