CS 210 Project Two Guidelines and Rubric

Competency

In this project, you will demonstrate your mastery of the following competency:

- Design functional programs that comply with industry regulations and best practices

Scenario

Congratulations! You have completed the interview process and have been hired as a junior developer at Chada Tech. Now that you have successfully completed your new-hire orientation and have been introduced to the rest of your team, you are ready to jump in and start working on your first project.

You are asked to collaborate with Airgead Banking, one of Chada Tech's clients. Airgead Banking is well known in the community. They often sponsor schools and have recently decided to partner with the local high school to develop a program that will teach students the concepts of fiscal responsibility (such as living within their means and spending less than they make) via an interactive system. The initial focus for this project will be on investing and the power of compound interest. You will develop an application that allows users to see how their investments will grow over time. Airgead Banking has provided you with a list of functional requirements that describe what they need their application to do.

Directions

1. Review the Airgead Banking App Functional Requirements, located in the Supporting Materials section. Create pseudocode or a flowchart to plan your coding project. Outline your code step-by-step so that you can use it as a guide when coding. This will be submitted along with your zipped application.
    a. Do not write code yet. You will do that in Step 3. For this step, write your thoughts in English of what the program should do.
    b. Don't be concerned with syntax, just list statements, each describing a single action.

c. List all steps.

d. Use proper naming conventions.

e. Keep it simple—use only one statement per line.

2. Develop an object-oriented programming (OOP) application using secure and efficient C++ code. Make sure that your application:

   a. Meets all specifications listed in the Airgead Banking App Functional Requirements

   b. Follows best practices described in the Airgead Banking Standards document

   c. Includes in-line comments

3. When your solution is finished, zip your project including all components (CPP, H, and any other files used).
4. Upload your zipped project file and pseudocode or flowchart to the project submission area.

What to Submit

Pseudocode or Flowchart

Submit a pseudocode or flowchart that clearly outlines your program logic step-by-step.

Investment Code

Submit your zipped project files, including all components (CPP, H, and any other files used). Be sure to include in-line comments.

Supporting Materials

The following resource(s) may help support your work on the project:

Reading: Airgead Banking App Functional Requirements

Airgead Banking provided you with this document, which includes all of the functional requirements necessary for their desired investment application. Your code must meet all of these functional requirements. A text version for all images and tables is available.

Reading: Airgead Banking Standards

This document outlines Airgead Banking's requirements in relation to code development best practices and standards.

Reading: Visual Studio Export Tutorial

This guide will walk you through how to download all of your work from Visual Studio as a ZIP folder.

Project Two Rubric

Criteria	Exemplary	Proficient	Needs Improvement	Not Evident	Value
Pseudocode or Flowchart	N/A	Pseudocode or flowchart effectively plans, organizes, and sequences program flow (100%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include using proper shapes for flowcharts, etc. (55%)	Does not attempt criterion (0%)	10
Input Validation	N/A	Code effectively validates user input (100%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include input validation for positive numbers, etc. (55%)	Does not attempt criterion (0%)	20
User Interface	N/A	Console contains all required input elements (100%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include enabled input boxes, etc. (55%)	Does not attempt criterion (0%)	25
Balance and Earned Interest Charts	N/A	Charts display correct balances and earned interest by year with and without additional monthly input (100%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include accurate earned interest calculations, etc. (55%)	Does not attempt criterion (0%)	25
Industry Standard Best Practices	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Uses industry standard best practices such as in-line comments and appropriate naming conventions to enhance readability and maintainability (85%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include naming conventions or in-line comments (55%)	Does not attempt criterion (0%)	20
				Total:	100%