CS 410 Binary to C++ Activity Guidelines and Rubric

Overview

In previous activities, you converted code between C++, assembly, and binary while retaining functionality. Now you are ready to take the next step by identifying the functions of legacy code, converting that code into a workable form in assembly, and converting assembly to C++. This real-life task will help you gain the skills you need for the successful completion of Project One, which is centered on working with legacy code. The coding for this assignment will be performed in Codio. You will submit the completed Binary to C++ Activity Template Word Document, as well as a new binary file and CPP file.

Prompt

For this assignment, you are given four legacy binary files. Specifically, you must address the following rubric criteria for each binary file:

1. Convert the binary file to assembly code.
    - These files can be found in the Software Reverse Engineering Playground in the Module Four file folder in Codio.
2. Explain the functionality of the blocks of assembly code.
    - Use the Binary to C++ Activity Template to complete this step.
3. Convert the assembly code to binary.
    - Create a new binary file for submission.
4. Convert the assembly code to C++ code.
    - Use the Binary to C++ Activity Template to complete this step.
    - Compile the C++ code in the Eclipse integrated development environment.
5. Explain how the C++ code performs the same tasks as the blocks of assembly code.
    - Use the Binary to C++ Activity Template to complete this step.

What to Submit

Binary to C++ Activity Template

This should be a Word document. Use the Binary to C++ Activity Template to convert the four legacy binary files to assembly code, explain the functionality of the assembly code, convert the assembly code to C++ code, and explain the functionality of the C++ code.

Binary File

This file is needed to run the application.

C++ File

This file is needed to ensure that the code and explanations are correct.

Binary to C++ Activity Rubric

Criteria	Exemplary	Proficient	Needs Improvement	Not Evident	Value
Binary to Assembly Conversion	N/A	Converts binary file into assembly code (100%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include converting the binary file into assembly code (85%)	Does not attempt criterion (0%)	12
Assembly Functionality Explanation	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Explains the functionality of the blocks of assembly code with minimal errors and adequate detail (85%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include explaining the functionality of the blocks of assembly code	Does not attempt criterion (0%)	23

			with minimal errors and richer detail (55%)		
Assembly to Binary Conversion	N/A	Converts the assembly code to binary (100%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include converting the assembly code to binary (85%)	Does not attempt criterion (0%)	12
Assembly to C++ Conversion	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Converts each block of assembly code into C++ code with minimal errors (85%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include converting each block of assembly code into C++ code with fewer errors (55%)	Does not attempt criterion (0%)	22
C++ Functionality Explanation	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Explains how the C++ code performs the same tasks as the blocks of assembly code with minimal errors and adequate detail (85%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include explaining how the C++ code performs the same tasks as the blocks of assembly code with minimal errors and richer detail (55%)	Does not attempt criterion (0%)	25
Articulation of Response	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Clearly conveys meaning with correct grammar, sentence structure, and spelling, demonstrating an understanding of audience and purpose (85%)	Shows progress toward proficiency, but with errors in grammar, sentence structure, and spelling, negatively impacting readability (55%)	Submission has critical errors in grammar, sentence structure, and spelling, preventing understanding of ideas (0%)	6
				Total:	100%