CS 410 C++ to Assembly Activity Guidelines and Rubric

Overview

In this activity, you will convert a simple application from C++ to assembly code. The coding for this assignment will be performed in Codio. You will then download the file from Codio for submission, along with the completed C++ to Assembly Template Word Document. The following resources will help you complete the tasks in this assignment:

- Section 7: How to Generate Assembly Code from C++ Code in the *Guide to Software Reverse Engineering*.
- Section 8: Downloading and Uploading Files in the *Codio Guide*.

Prompt

Specifically, you must address the following rubric criteria:

1. Explain the functionality of the C++ code.
    - Use the C++ to Assembly Activity Template to complete this step.
    - The C++ file is located within the Software Reverse Engineering Playground in the Module One file folder in Codio. It is also in the following table:

```
#include<iostream>
using namespace std;
int main()
      {
int width=10;
int height=5;
int area;
area = width * height;
cout<<endl<< area;
   return 0;
}
```

2. Convert the C++ file into assembly code.
    - The C++ file is located within the Software Reverse Engineering Playground in the Module One file folder in Codio.
3. Align each line of C++ code with the corresponding blocks of assembly code.
    - Use the C++ to Assembly Activity Template to complete this step.
4. Explain how the blocks of assembly code perform the same tasks as the C++ code.
    - Use the C++ to Assembly Activity Template to complete this step.

What to Submit

C++ to Assembly Template

This should be a Word document. Use this template to explain the functionality of the lines of C++ code, align the lines of C++ code with the corresponding lines of assembly code, and explain how the assembly code performs the same tasks as the C++ code.

Assembly File (.S file)

This file is needed to ensure that the C++ code was successfully converted into assembly code.

C++ to Assembly Activity Rubric

Criteria	Exemplary	Proficient	Needs Improvement	Not Evident	Value
C++ Functionality	Exceeds proficiency in an	Explains the functionality of the	Shows progress toward	Does not attempt criterion (0%)	27

Explanation	exceptionally clear, insightful, sophisticated, or creative manner (100%)	C++ code with minimal errors and adequate detail (85%)	proficiency, but with errors or omissions; areas for improvement may include explaining the functionality of the C++ code with minimal errors and richer detail (55%)		
C++ to Assembly Conversion	N/A	Converts C++ file into assembly code (100%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include converting the C++ file into assembly code (55%)	Does not attempt criterion (0%)	15
Translation Alignment	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Aligns each line of C++ code with the corresponding blocks of assembly code with minimal errors (85%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include aligning each line of C++ code with the corresponding blocks of assembly code with fewer errors (55%)	Does not attempt criterion (0%)	22
Assembly Functionality Explanation	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Explains how the assembly code performs the same tasks as the C++ 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 assembly code performs the same tasks as the C++ code with minimal errors and richer detail (55%)	Does not attempt criterion (0%)	27
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%)	9
				Total:	100%