# **CS 410 C++ to Assembly Activity Template**

**Step 1:** Explain the functionality of the C++ code.

## C++ Code Functionality

| **C++ Line of Code** | **Explanation of Functionality** |
| --- | --- |
| #include <iostream> | Includes the iostream library |
| Using namespace std; | States this file is using the standard library namespace |
| Int main() { | This is the entry point for the file as being the main function which returns an integer |
| Int width=10; | Creates a width variable with value of 10 |
| Int height=5; | Creates a height variable with value of 5 |
| Int area; | Declares the area variable with type of integer |
| area = width \* height; | States the value of area variable is equal to width times height |
| cout<<endl<< area; | Outputs the value of area |
| return 0; | Gracefully exits the program |
| } | End of main function |
|  |  |

**Step 2:** Convert the C++ file into assembly code.

**Step 3:** Align each line of C++ code with the corresponding blocks of assembly code.

## C++ to Assembly Alignment

| **C++ Line of Code** | **Blocks of Assembly Code** |
| --- | --- |
| Int width = 10; | movl $0xa -0x8(%rbp) |
| Int height = 5; | movl $0x5 -0xc(%rbp) |
|  | mov --0x8(%rbp), %eax |
| Area = width \* height; | imul -0xc(%rbp), %eax |
|  | mov %eax, -0x10(%rbp) |
|  | mov 0x2df2(%rip), %rdi |
|  | mov 0x2de3(%rip), %rsi |
| cout << endl << area; | call 0x410 |
|  | mov %rax, %rdi |
|  | mov -0x10(%rbp), %esi |
|  | Call 0x424 |
|  | xor %eax, %eax |
|  | add $0x10, %rsp |
| return 0; | pop %rbp |
|  | ret |
|  | nopl (%rax) |

**Step 4:** Explain how the blocks of assembly code perform the same tasks as the C++ code.

## Assembly Functionality

| **Blocks of Assembly Code** | **Explanation of Functionality** |
| --- | --- |
| movl $0xa -0x8(%rbp) | Moves 10 into address location -0x8 |
| movl $0x5 -0xc(%rbp) | Moves 5 into address location -0xc |
| mov --0x8(%rbp), %eax | Moves 10 from address location into EAX register |
| imul -0xc(%rbp), %eax | Multiplies the EAX register and address location -0xc |
| mov %eax, -0x10(%rbp) | Moves the result into address location rbp |
| mov 0x2df2(%rip), %rdi | Moves cout into address location rdi |
| mov 0x2de3(%rip), %rsi | Moves endl into address location rsi |
| call 0x410 | Calls << |
| mov %rax, %rdi | Moves rax to address rdi |
| mov -0x10(%rbp), %esi | Moves area out of address |
| Call 0x424 | Calls << |
| xor %eax, %eax | Either or address eax or ea |
| add $0x10, %rsp | Adds area to register rsp |
| pop %rbp | Pops the value in register rbp from the stack |
| ret | Return |
| nopl (%rax) | Performs no operation |