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

Merrik Wright

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

## C++ Code Functionality

| **C++ Line of Code** | **Explanation of Functionality** |
| --- | --- |
| int width = 10; | Declares an integer variable width and assigns it the value 10. |
| int height = 5; | Declares an integer variable height and assigns it the value 5. |
| int area; | Declares an integer variable area without assigning a value. |
| area = width \* height; | Calculates the product of width and height, and stores the result in area. |
| cout << endl << area; | Outputs a newline followed by the value of area to the console. |
| return 0; | Ends the main() function and returns 0 to the operating system. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

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

g++ -S Module1Files/assignment1\_1.cpp -o Module1Files/assignment1\_1.s  
The code above is what I used in Step 2

**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 $10, -12(%rbp) |
| int height = 5; | movl $5, -8(%rbp) |
| int area; | Used later, reserved. Later at –4(%rbp) |
| area = width \*   height; | movl -12(%rbp), %eax imull -8(%rbp), %eax movl %eax, -4(%rbp) |
| cout << endl << area; | <br>movq \_ZSt4endl... -> %rax<br>movq %rax -> %rsi<br>leaq \_ZSt4cout ->  %rdi<br>call \_ZNSolsEPFRSoS\_E<br>movq %rax -> %rdx<br>movl -4(%rbp) ->  %eax<br>movl %eax → %esi<br>movq %rdx → %rdi<br>call \_ZNSolsEi |
| return 0; | movl $0, %eax  Leave  ret |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 $10, -12(%rbp) | Store the value 10 in the stack slot for width. |
| movl $5, -8(%rbp) | Store the value 5 in the stack slot for height. |
| movl -12(%rbp), %eax | Move the value of width into register eax. |
| imull -8(%rbp), %eax | Multiply the value in eax (width) with height; result stays in eax. |
| movl %eax, -4(%rbp) | Store the result of the multiplication into the stack slot for area. |
| movq \_ZSt4endl... -> %rax | Load the address of endl into rax. |
| movq %rax → %rsi | Move endl into argument register rsi for cout. |
| leaq \_ZSt4cout -> %rdi | Load address of cout into rdi. |
| call \_ZNSolsEPFRSoS\_E | Call function to print endl. |
| movq %rax → %rdx | Store the stream in rdx to chain output. |
| movl -4(%rbp) → %eax | Load area into eax. |
| movl %eax → %esi | Set area as argument for cout. |
| movq %rdx → %rdi | Set output stream as cout. |
| call \_ZNSolsEi | Call function to print integer value. |
| movl $0, %eax | Set return value of main() to 0. |
| leave / ret | Clean up and return from the function. |