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

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

## C++ Code Functionality

| **C++ Line of Code** | **Explanation of Functionality** |
| --- | --- |
| #include <iostream> | Includes the iostream header file for input-output operations. |
| Using namespace std | Uses the standard namespace. |
| Int main() | Defines the main function, which is the entry point of the program. |
| { | Starts the block of code for the main function. |
| Int width=10 | Declares an integer variable "width" and initializes it to 10. |
| Int height = 5 | Declares an integer variable "height" and initializes it to 5. |
| Int area | Declares an integer variable "area" without initializing it. |
| Area = width \* height | Calculates the area by multiplying width and height, and stores the result in the "area" variable. |
| cout << endl << area | Prints a newline followed by the value of the variable "area" to the console. |
| Return 0 | Signifies successful termination of the program and returns control to the operating system. |
| { | Ends the block of code for the 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 $10, -12(%rbp) |
| Int height = 5 | movl $5, -8(%rbp) |
| Int area | movl -12(%rbp), %eax imull -8(%rbp), %eax movl %eax, -4(%rbp) |
| Area = width \* height | movl -4(%rbp), %eax movl %eax, %esi movq %rdx, %rdi call \_ZNSolsEi@PLT |
| cout << endl << area | movl $0, %eax leave |
| Return 0 |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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) | the constant value 10 into the memory location that is 12 bytes before the base pointer |
| movl $5, -8(%rbp) | the constant value 5 into the memory location that is 8 bytes before the base pointer |
| movl -12(%rbp), %eax imull -8(%rbp), %eax movl %eax, -4(%rbp) | Load the value of width into %eax.  Multiply the value of height with what's in %eax, storing the result back in %eax.  Store the result which is width \* height in the memory location designated for area. |
| movl -4(%rbp), %eax movl %eax, %esi movq %rdx, %rdi call \_ZNSolsEi@PLT | Load the value of area into %eax.  Prepare the value for the function call by moving it to %esi.  Prepare additional data for the function call  Call the output function to print the value of area. |
| movl $0, %eax leave | Load the value  0  0 into %eax as the return value for the main() function.  Clean up the function's stack frame to prepare for exit. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |