# **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> | Allows for the use of cout and cin for character output and input respectively |
| using namespace std; | uses the standard library for names |
| int main() | Starts the main function for the program, with no input variables |
| { | opening bracket of the main function |
| int width=10; | creates integer variable called width with value of 10 |
| int height=5; | creates integer variable called heightwith value of 5 |
| int area; | creates integer variable area, with no initialized value (default is 0) |
| area = width \* height; | sets area to width multiplied by height |
| cout<<endl<< area; | outputs a new line and then the value of the area variable to the console |
| return 0; | returns 0 as the return value of the main function |
| } | end of main function and closing bracket of 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** |
| --- | --- |
| #include<iostream> |  |
| using namespace std; |  |
| int main() | .globl main  .type main, @function  main:  .LFB1493:  .cfi\_startproc  pushq %rbp |
| { |  |
| int width=10; | movq %rsp, %rbp  .cfi\_def\_cfa\_register 6  subq $16, %rsp  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; | movq \_ZSt4endlIcSt11char\_traitsIcEERSt13basic\_ostreamIT\_T0\_ES6\_@GOTPCREL(%rip), %rax  movq %rax, %rsi |
| cout<<endl<< area; | leaq \_ZSt4cout(%rip), %rdi  call \_ZNSolsEPFRSoS\_E@PLT  movq %rax, %rdx  movl -4(%rbp), %eax  movl %eax, %esi  movq %rdx, %rdi |
| return 0; | call \_ZNSolsEi@PLT  movl $0, %eax |
| } | leave |

**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** |
| --- | --- |
|  |  |
|  |  |
| .globl main .type main, @function  main:  .LFB1493:  .cfi\_startproc  pushq %rbp | Starts the main global function and pushes rbp to the top of the stack to use for the variables |
|  |  |
| movq %rsp, %rbp  .cfi\_def\_cfa\_register 6  subq $16, %rsp  movl $10, -12(%rbp) | prepares rbp for storing variables and sets the memory location 12 bytes above the stack to 10 (this is the width variable) |
| movl $5, -8(%rbp) | sets the memory location 8 bytes above the stack to 5 (this is the height variable) |
| movl -12(%rbp), %eax  imull -8(%rbp), %eax  movl %eax, -4(%rbp) | moves the two stored values into eax for multiplication and places those values 4 bytes above the rbp stack |
| movq \_ZSt4endlIcSt11char\_traitsIcEERSt13basic\_ostreamIT\_T0\_ES6\_@GOTPCREL(%rip), %rax  movq %rax, %rsi | performs the multiplication of the two variables |
| leaq \_ZSt4cout(%rip), %rdi  call \_ZNSolsEPFRSoS\_E@PLT  movq %rax, %rdx  movl -4(%rbp), %eax  movl %eax, %esi  movq %rdx, %rdi | prepares the resulting value for display |
| call \_ZNSolsEi@PLT  movl $0, %eax | calls the function and then sets eax back to 0 |
| leave | leaves the function |