Chris Gollnick

7 January 2022

CS410-T3717

# **CS 410 C++ to Assembly Activity Module 1**

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

## C++ Code Functionality

| **C++ Line of Code** | **Explanation of Functionality** |
| --- | --- |
| #include<iostream> | Include library for input output |
| Using namespace std; | Declare namespace usage as std |
| Int main () | Begin main activity type integer |
| { | Start code for main |
| Int width = 10; | Declare width as integer value 10 |
| Int height = 5; | Declare height as integer value 5 |
| Int area; | Decalre area as integer |
| Area = width \* height; | Assign area to width time height |
| Cout << endl << area; | Print new line and then area value |
| Return 0; | Return nothing |
| } | End activity main |

**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 () | 11  .globl main , 12 .type main, @function |
| { | 15 .cfi startproc |
| Int width = 10; | 22 movl $10, -12(%rbp) |
| Int height = 5; | 23 movl $5, -8(%rbp) |
| Int area; | 36 movl $0, %eax |
| Area = width \* height; | 25 imull -8(%rbp), %eax |
| Cout << endl << area; | 29 leaq \_ZSt4cout(%rip), %rdi  30 call \_ZNSolsEPFRSoS\_E@PLT  35 call \_ZNSolsEi@PLT |
| Return 0; | 39 ret |
| } | 40 .cfi endproc |

**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** |
| --- | --- |
| 11  .globl main , 12 .type main, @function | 11 declares a global main, 12 declares main as a type function |
| 15 .cfi startproc | Declares the start of the function |
| 22 movl $10, -12(%rbp) | Stores the value of 10 to register rbp at -12 |
| 23 movl $5, -8(%rbp) | Stores the value of 5 to register rbp at -8 |
| 36 movl $0, %eax | Stores empty value to eax register |
| 25 imull -8(%rbp), %eax | Multiplies the -8 rbp register by the eax registered value |
| 30 call \_ZNSolsEPFRSoS\_E@PLT | Performs call for cout |
| 35 call \_ZNSolsEi@PLT | Performs call for cout |
| 29 leaq \_ZSt4cout(%rip), %rdi | Stores value to register for cout call |
| 39 ret | Return function |
| 40 .cfi endproc | Ends the process or function |
|  |  |