# **CS 410 C++ to Assembly With Loops 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 standard input/output stream library. |
| using namespace std; | Allows usage of standard C++ functions without std:: prefix. |
| int main() | Main function where program execution begins. |
| int num, i; | Declares two integer variables: num and i. |
| int product=1; | Declares and initializes product to 1 (used for factorial). |
| cout<<"Enter a number:\n"; | Outputs prompt asking user to input a number |
| cin>>num; | Receives the number input from the user. |
| for (i=num; i>0; i--) | For loop that starts from num and decrements until i == 0. |
| product = product \* i; | Multiplies product by i during each iteration |
| cout<<"The factorial for "<<num << "is: "<< product<< endl; | Displays the result of the factorial calculation |
| return 0; | Returns 0 to indicate successful execution. |

**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 product = 1; | movl $1, -12(%rbp) |
| cout<<"Enter a number:\n"; | leaq .LC0(%rip), %rsi → call \_ZStlsI... (outputs the string to console) |
| cin >> num; | leaq -20(%rbp), %rax → call \_ZNSirsERi@PLT (reads input into num) |
| for (i=num; i>0; i--) | Loop starts at .L3: with cmpl $0, -16(%rbp) → jle .L2 to check condition |
| product = product \* i; | movl -12(%rbp), %eax → imull -16(%rbp), %eax → movl %eax, -12(%rbp) |
| i-- | subl $1, -16(%rbp) |
| cout << "The factorial for " | leaq .LC1(%rip), %rsi → call \_ZStlsI... |
| << num << " is: " | movl -20(%rbp), %eax → call \_ZNSolsEi@PLT, followed by leaq .LC2(%rip) → call \_ZStlsI... |
| << product << endl; | movl -12(%rbp), %eax → call \_ZNSolsEi@PLT |
| return 0; | movl $1, %eax before leave and 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 $1, -12(%rbp) | Sets product = 1 — the starting value for factorial. |
| leaq .LC0(%rip), %rsi → call ... | Loads and prints "Enter a number" to console using cout. |
| leaq -20(%rbp), %rax → call \_ZNSirsERi@PLT | Reads the user's input into variable num. |
| .L3: → cmpl $0, -16(%rbp) | |  | | --- | |  |  |  | | --- | | Checks the loop condition: if i > 0. | |
| imull -16(%rbp), %eax | Multiplies product by i |
| subl $1, -16(%rbp) | Decrements the loop variable i--. |
| leaq .LC1(%rip)... to call \_ZNSolsEi@PLT | |  | | --- | |  |  |  | | --- | | Prints "The factorial for " followed by the user's number. | |
| movl -12(%rbp), %eax → call \_ZNSolsEi@PLT | Prints the computed factorial result. |
| leave and ret | Ends the function and returns control to the operating system. |
|  |  |