# **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> |  |
| using namespace std; |  |
| int main()  { |  |
| int num, i; |  |
| int product =1; |  |
| cout<<"Enter a number:\n"<< endl; |  |
| cin>>num; |  |
| for(i=num;i>0; i--)          product = product \* i; |  |
| cout<<"The factorial for " << num << "is: \n"<< product; |  |
| return 1; |  |
| } |  |

**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** |
| --- | --- |
| "Enter a number:\n"  "The factorial for "  "is: \n" | .LC0:    .string "Enter a number:\n"  .LC1:    .string "The factorial for "  .LC2:    .string "is: \n" |
| int num, i; | movq %rax, -8(%rbp) |
| int product =1; | movl $1, -12(%rbp) |
| cout<<"Enter a number:\n"<< endl; | leaq  .LC0(%rip), %rsi    leaq  \_ZSt4cout(%rip), %rdi    call |
| cin>>num; | leaq  -20(%rbp), %rax    movq  %rax, %rsi    leaq  \_ZSt3cin(%rip), %rdi |
| for(i=num;i>0; i--)          product = product \* i; | .L3:    cmpl  $0, -16(%rbp)    jle .L2    movl  -12(%rbp), %eax    imull -16(%rbp), %eax    movl  %eax, -12(%rbp)    subl  $1, -16(%rbp) |
| cout<<"The factorial for " << num << "is: \n"<< product; | movl -20(%rbp), %eax  movl  -12(%rbp), %eax  movl  %eax, %esi  leaq .LC2(%rip), %rsi  call |
| return 1; | 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** |
| --- | --- |
| .LC0:    .string "Enter a number:\n"  .LC1:    .string "The factorial for "  .LC2:    .string "is: \n" |  |
| movq %rax, -8(%rbp) |  |
| movl $1, -12(%rbp) |  |
| leaq  .LC0(%rip), %rsi    leaq  \_ZSt4cout(%rip), %rdi    call |  |
| leaq  -20(%rbp), %rax    movq  %rax, %rsi    leaq  \_ZSt3cin(%rip), %rdi |  |
| .L3:    cmpl  $0, -16(%rbp)    jle .L2    movl  -12(%rbp), %eax    imull -16(%rbp), %eax    movl  %eax, -12(%rbp)    subl  $1, -16(%rbp) |  |
| movl -20(%rbp), %eax  movl  -12(%rbp), %eax  movl  %eax, %esi  leaq .LC2(%rip), %rsi  call |  |
| ret |  |
|  |  |
|  |  |