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

Step 1: Convert the assembly code into C++ code.

Step 2: Explain the function of the converted C++ code.

| **Assembly Code** | **C++ Code** | **Explanation of Functionality** |
| --- | --- | --- |
| movl −8(%rbp), %eax sall $3, %eax subl $3, %eax movl %eax, −4(%rbp) | * int x = value; * int result = (x << 3) - 3; | * movl -8(%rbp), %eax loads x into register eax. * sall $3, %eax performs a left shift by 3 bits (multiplies x by 8). * subl $3, %eax subtracts 3 from the result. * movl %eax, -4(%rbp) stores this result back in memory. |
| movl −8(%rbp), %eax sall $2, %eax subl $1, %eax leal 7(%rax), %edx testl %eax, %eax cmovs %edx, %eax sarl $3, %eax  movl %eax, −4(%rbp) | * int x = value; * int temp = (x << 2) - 1; * int result = (temp == 0) ? (temp + 7) >> 3 : temp >> 3; | * sall $2, %eax shifts x left by 2 bits (multiplies by 4). * subl $1, %eax subtracts 1. * leal 7(%rax), %edx calculates temp + 7. * testl %eax, %eax and cmovs %edx, %eax check if temp == 0. If true, eax is assigned temp + 7. * sarl $3, %eax shifts eax right by 3, dividing by 8. |
| movl −8(%rbp), %eax leal 7(%rax), %edx testl %eax, %eax cmovs %edx, %eax sarl $3, %eax movl −8(%rbp), %edx sall $2, %edx addl %edx, %eax  movl %eax, −4(%rbp) | * int x = value; * int temp = (x == 0) ? (x + 7) >> 3 : x >> 3; * int result = temp + (x << 2); | * This block calculates (x + 7) >> 3 if x == 0, otherwise x >> 3. * Then, it adds x << 2 (multiplying x by 4) to the result. |