# **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;  int y = ((x \* 8) – 3); | Move contents of -8(%rbp) to %eax  Take value of %eax and multiply it by 8  Subtract 3 from contents of %eax |
| 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;  int y = (((x \* 4)-1)/8); | Move contents of -8(%rbp) to %eax  Take value of %eax and multiply it by 4  Subtract 1 from contents of %eax  Put memory address of 7(%rax) into %edx  Test %eax to see if it’s above zero (AND)  Conditional move if negative for %edx to %eda  Take value of %eax and divide it by 8  Shift 3 bits to the right  Move contents of %eax into -4(%rbp) |
| 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;  int y = (((x / 8) + x) \* 4); | Move contents of -8(%rbp) to %eax  Put memory address of 7(%rax) into %edx  Test %eax to see if it’s above zero (AND)  Conditional move if negative for %edx to %eda  Take value of %eax and divide it by 8  Move contents of -8(%rbp) to %edx  Take value of %edx and multiply it by 4  Add value of %edx and %eax together  Move contents of %eax into -4(%rbp) |