# **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 input;  int output = ((input \* 8) – 3); | -Move 8(%rbp) to eax  -Eax shifted 3 bits left (mult by 8)  -Subtract 3 from eax  -Move eax to 4(%rbp) |
| 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 input;  Int output = (((input\*4)-1)/8); | -Move 8(%rbp) to eax  -Eax shift 2 left (mult by 4)  -Subtract 1 from eax  -Load memory and put 7(%rax) in edx  -Test eax to see if above zero  -conditional move if negative eax test result to put edx to eda  -eax shift right 3, divide by 8  -move eax to -4(%rbp) |
| movl −8(%rbp), %eax leal 7(%rax), %edx  testl %eax, %eax cmovs %edx, %eax  sarl $3, %eax movl −8(%rbp), %edx sall $2, %edx addl %edx, %eax  movl %eax, −4(%rbp) | Int input;  Int output = input /3;  Input = int var;  Var = var \* 4;  Output = output + var; | -Move 8(%rbp) to eax  -load memory and put 7(%rax) into edx  -Test if eax > 0  -If not, conditional moveedx tro eda  -eax shift right 3, divide by 8  -move 8(%rbp) to edx  -shift edx by 2 left, mult by 4  -add edx and eax  -move eax to -4($%rbp) |