# **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 temp;  temp = (temp\*8)-3 | moves the contents of 8(%rbp) into eax  shifts the value of eax 3 bits to the left (\*2^3 or \*8)  subtracts 3 from the value of eax  moves the value of eax into −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 temp  temp = ((temp \*2) - 1) / 8 | moves the contents of 8(%rbp) into eax  shifts the value of eax 2 bits to the left  subtracts 1 from the value of eax  Load effective address -  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  shift the value of eax 3 bits to the right (/2^3 or /8)  moves the value 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 temp  temp = temp / 8  int temp2  temp2 = temp2 \* 4  temp = temp + temp2 | moves the contents of 8(%rbp) into eax  Load effective address -  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  shift the value of eax 3 bits to the right  moves the contents of 8(%rbp) into edx  shifts the value of edx 2 bits to the left  adds the value of edx into eax  moves the value of eax into −4(%rbp) |