# **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 a, b ;  b = (a \* 8) - 3 ; | 1. Move values -8(%rbp) to %eax 2. Shift left 3 bits which is multiply by 2^3=8 3. Subtract 1 from contents of %eax 4. Move content 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 a, b ;  a = ( a \* 4 ) -1 ;  b = a + 7;  If (a == 0) {  a = b;  }  a = a / 8;  b = a ; | 1. Move values -8(%rbp) to %eax 2. Shift left a value 2 bits which is multiply by 2^2=4 3. Subtract 1 from eax 4. Add 7 to a value of %rax assign it %edx 5. Check of %EAX is zero 6. Conditional Move if ZF is 1 move %edx to %eax 7. Shift EAX right by 3 bits (Division by 8) 8. Assign %EAX value 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) | // assume a in EAX and b in  // EDX  int a , b;  b = 7 + a;  if ( a == 0) {  a = b;  }  a = a / 8;  b = a ;  b = b \* 4;  a = a + b;  b = a ; | 1. Move value -8 to EAX 2. Add 7 to RAX and assign the value to EDX 3. IF EAX is zero then assign EDX to EAX 4. Shift right by 3 bits (2^3), divide by 8 5. Set b equal a value 6. Shift left 2 bits, multiply by 4 7. Add b to a and assign to a 8. Set b to equal a |