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

Janera Dobson

Southern New Hampshire University

CS-410-T3249

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 i, j;  j = j \* 8;  j = j – 3;  i = j; | * Movess value in −8(%rbp), %eax * Shifts the value in %eax to 3 bits * Subtract 3 from %eax * Moves value in %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 i, j;  j = j \* 4;  j = j – 1;  j = j/8;  i = j; | * Moves the value in -8(%rbp) to %eax. * Shifts the value towards left stored in %eax to 2 bits. 2\*2 * Subtract 3 from %eax * Adds 7 to the value in %rax and store the value in %edx * %eax is tested to know whether it is positive, negative or zero. * The value %edx is moved to %eax if the sign is negative. * Moves the value towards right stored in %eax to 3bits.2\*2\*2. * Moves the value in %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 i, j;  j = j/8;  j = j \* 4;  i = j; | * Moves the value in -8(%rbp) to %eax. * Adds 7 to the value in %rax and store the value in %edx * %eax is tested to know whether it is positive, negative or zero. * The value %edx is moved to %eax if the sign is negative. * Shifts the value towards right stored in %eax to 3bits.2\*2\*2. * Moves the value in -8(%rbp) to %edx * Shifts the value towards left stored in %eax to 2 bits. 2\*2 * Add the value in %edx to %eax. * Place the value in %eax to -4(%rbp) |