Southern
New Hampshire
University

CS 410 C++ to Assembly Activity Template

Tabitha Tallent
Southern New Hampshire University
CS-410: Software Reverse Engineering
Professor Jonathan Norman
March 9, 2025

Step 1: Explain the functionality of the C++ code.

C++ Code Functionality

C++ Line of Code	Explanation of Functionality
#include<iostream>	Library to allow for input and output functionality via streams.
using namespace std;	Indicates the code uses the C++ namespace called std, this namespace is then used across the entire code as a global namespace.
int main() {	Declares the main function
int width=10;	Declares an integer variable named width and assigns it a value of 10
int height=5;	Declares an integer variable named height and assigns it a value of 5
int area;	Declares an integer variable named area
area = width * height;	Assigns area the value of width multiplied with height
cout<<endl<<area;	Outputs a new line and then the value of area
return 0; }	Indicates that the main function executed successfully

Step 2: Convert the C++ file into assembly code.

Step 3: Align each line of C++ code with the corresponding blocks of assembly code.

C++ to Assembly Alignment

C++ Line of Code	Blocks of Assembly Code
#include<iostream>	
using namespace std;	
int main() {	main: pushq %rbp movq %rsp, %rbp subq %rsp, 16
int width=10;	movl $10, 12(%rbp)
int height=5;	movl $5, -8(%rbp)

int area;	
area = width * height;	movl -12(%rbp), %eax imull -8(%rbp), %eax movl %eax, -4(%rbp)
cout<<endl<<area;	movq _ZSt4endlIcSt11char_traitsIcEERSt13basic_ostreamIT_T0_ES6_@GOTPCREL(%rip), %rax movq %rax, %rsi leaq _ZSt4cout(%rip), %rdi call _ZNSolsEPFRSoS_E@PLT movq %rax, %rdx movl -4(%rbp), %eax movl %eax, %esi movq %rdx, %rdi call _ZNSolsEi@PLT
return 0; }	movl $0, %eax leave .cfi_def_cfa 7, 8 ret .cfi-endproc

Step 4: Explain how the blocks of assembly code perform the same tasks as the C++ code.

Assembly Functionality

Blocks of Assembly Code	**Explanation of Functionality**
main: pushq %rbp movq %rsp, %rbp subq %rsp, 16	Declares the main function and allocates space on the stack
movl $10, 12(%rbp)	Stores the value of width (10) in a memory location on register rbp
movl $5, -8(%rbp)	Stores the value of height (5) in a memory location on register rbp
movl -12(%rbp), %eax imull -8(%rbp), %eax movl %eax, -4(%rbp)	Move value of width (10) from current location into the eax register Multiply the value of the eax register (10) with the value of height (5) Move the result (50) into the memory location for area on register rbp
movq _ZSt4endlIcSt11char_traitsIcEERSt13basic_ostreamIT_T0_ES6_@GOTPCREL(%rip), %rax movq %rax, %rsi leaq _ZSt4cout(%rip), %rdi call _ZNSolsEPFRSoS_E@PLT movq %rax, %rdx movl -4(%rbp), %eax movl %eax, %esi movq %rdx, %rdi call _ZNSolsEi@PLT	Move the value of area (50) from rbp to eax and call cout and endl
movl $0, %eax leave .cfi_def_cfa 7, 8 ret .cfi-endproc	Move return value (0) into the eax register and use leave and ret to reset the stack and end the main function