CS 410 C++ to Assembly Activity

C++ Code Functionality	
C++ Line of Code	**Explanation of Functionality**
#include<iostream>	Includes standard input/output header file, allowing input and output operations.
using namespace std;	Makes symbols from std namespace accessible without needing to prefix them.
int main()	Starting point of program, returns integer when finished.
{	
int width=10;	Declares integer named width and initializes it with a value of 10.
int height=5;	Declares integer named height and initializes it with a value of 5.
int area;	Declares integer named area with no initial value.
area = width * height;	Multiplies values of width and height and assigns result to value of area.
cout << endl << area;	Outputs a blank line followed by the value of area on the next line.
return 0;	Signals successful program execution and exits main function
}	

C++ to Assembly Alignment	
C++ Line of Code	**Blocks of Assembly Code**
	.file "assignment1_1.cpp"
	.text
	.section .rodata
	.type _ZStL19piecewise_construct, @object
	.size _ZStL19piecewise_construct, 1
	_ZStL19piecewise_construct:
	.zero 1
	.local _ZStL8__ioinit
	.comm _ZStL8__ioinit,1,1
	.text
int main();	.globl main
	.type main, @function
	main:
	.LFB1493:
	.cfi_startproc
	pushq %rbp
	.cfi_def_cfa_offset 16
	.cfi_offset 6, -16
	movq %rsp, %rbp
	.cfi_def_cfa_register 6
	subq $16, %rsp
int width = 10;	movl $10, -12(%rbp)
int height = 5;	movl $5, -8(%rbp)
area = width * height;	movl -12(%rbp), %eax
	imull -8(%rbp), %eax
	movl %eax, -4(%rbp)
	movq _ZSt4endlIcSt11char_traitsIcEERSt13basic _ostreamIT_T0_ES6_@GOTPCREL(%rip), %rax
	movq %rax, %rsi
	leaq _ZSt4cout(%rip), %rdi
	call _ZNSolsEPFRSoS_E@PLT
	movq %rax, %rdx
cout << endl << area;	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
	.LFE1493:
	.size main, .-main
	.type _Z41__static_initialization_and_destruction_0ii, @function
	_Z41__static_initialization_and_destruction_0ii:
	.LFB1979:
	.cfi_startproc
	pushq %rbp
	.cfi_def_cfa_offset 16
	.cfi_offset 6, -16
	movq %rsp, %rbp
	.cfi_def_cfa_register 6
	subq $16, %rsp
	movl %edi, -4(%rbp)
	movl %esi, -8(%rbp)
	cmpl $1, -4(%rbp)
	jne .L5
	cmpl $65535, -8(%rbp)
	jne .L5
	leaq _ZStL8__ioinit(%rip), %rdi
	call _ZNSt8ios_base4InitC1Ev@PLT
	leaq __dso_handle(%rip), %rdx
	leaq _ZStL8__ioinit(%rip), %rsi
	movq _ZNSt8ios_base4InitD1Ev@GOTPCREL(%rip), %rax
	movq %rax, %rdi
	call __cxa_atexit@PLT
	.L5:
	nop
	leave
	.cfi_def_cfa 7, 8
	ret
	.cfi_endproc
	.LFE1979:

	.size _Z41__static_initialization_and_destruction_0ii, .-_Z41__static_initialization_and_destruction_0ii
	.type _GLOBAL__sub_I_main, @function
	_GLOBAL__sub_I_main:
	.LFB1980:
	.cfi_startproc
	pushq %rbp
	.cfi_def_cfa_offset 16
	.cfi_offset 6, -16
	movq %rsp, %rbp
	.cfi_def_cfa_register 6
	movl $65535, %esi
	movl $1, %edi
	call _Z41__static_initialization_and_destruction_0ii
	popq %rbp
	.cfi_def_cfa 7, 8
	ret
	.cfi_endproc
	.LFE1980:
	.size _GLOBAL__sub_I_main, .-_GLOBAL__sub_I_main
	.section .init_array,"aw"
	.align 8
	.quad _GLOBAL__sub_I_main
	.hidden__dso_handle
	.ident "GCC: (Ubuntu 7.5.0-3ubuntu1~18.04) 7.5.0"
	.section .note.GNU-stack,"",@progbits

Assembly Functionality	
Blocks of Assembly Code	**Explanation of Functionality**
.file "assignment1_1.cpp"	
.text	
.section .rodata	
.type _ZStL19piecewise_construct, @object	
.size _ZStL19piecewise_construct, 1	
_ZStL19piecewise_construct:	
.zero 1	
.local _ZStL8__ioinit	
.comm _ZStL8__ioinit,1,1	
.text	
.globl main	Main function declared.
.type main, @function	
main:	
.LFB1493:	
.cfi_startproc	
pushq %rbp	
.cfi_def_cfa_offset 16	
.cfi_offset 6, -16	
movq %rsp, %rbp	
.cfi_def_cfa_register 6	
subq $16, %rsp	
movl $10, -12(%rbp)	Assign value of 10 to -12(%rbp).
movl $5, -8(%rbp)	Assign value of 5 to -8(%rbp).
movl -12(%rbp), %eax	Move contents of -12(%rbp) to %eax.
imull -8(%rbp), %eax	Multiply -8(%rbp) by %eax.
movl %eax, -4(%rbp)	Move result of %eax to -4(%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	Move contents of -4(%rbp) to %eax.
movl %eax, %esi	Move contents of %eax to %esi to call cout.
movq %rdx, %rdi	
call _ZNSolsEi@PLT	
movl $0, %eax	Assign value of 0 to %eax.

leave	Leaves the main function.
.cfi_def_cfa 7, 8	
ret	Return %eax value.
.cfi_endproc	
.LFE1493:	
.size main, .-main	
.type _Z41__static_initialization_and_destruction_0ii, @function	
_Z41__static_initialization_and_destruction_0ii:	
.LFB1979:	
.cfi_startproc	
pushq %rbp	
.cfi_def_cfa_offset 16	
.cfi_offset 6, -16	
movq %rsp, %rbp	
.cfi_def_cfa_register 6	
subq $16, %rsp	
movl %edi, -4(%rbp)	
movl %esi, -8(%rbp)	
cmpl $1, -4(%rbp)	
jne .L5	
cmpl $65535, -8(%rbp)	
jne .L5	
leaq _ZStL8__ioinit(%rip), %rdi	
call _ZNSt8ios_base4InitC1Ev@PLT	
leaq __dso_handle(%rip), %rdx	
leaq _ZStL8__ioinit(%rip), %rsi	
movq _ZNSt8ios_base4InitD1Ev@GOTPCREL(%rip), %rax	
movq %rax, %rdi	
call __cxa_atexit@PLT	
.L5:	
nop	
leave	
.cfi_def_cfa 7, 8	
ret	
.cfi_endproc	
.LFE1979:	
.size _Z41__static_initialization_and_destr	

uction_0ii, .-_Z41__static_initialization_and_destruction_0ii	
.type _GLOBAL__sub_I_main, @function	
_GLOBAL__sub_I_main:	
.LFB1980:	
.cfi_startproc	
pushq %rbp	
.cfi_def_cfa_offset 16	
.cfi_offset 6, -16	
movq %rsp, %rbp	
.cfi_def_cfa_register 6	
movl $65535, %esi	
movl $1, %edi	
call _Z41__static_initialization_and_destruction_0ii	
popq %rbp	
.cfi_def_cfa 7, 8	
ret	
.cfi_endproc	
.LFE1980:	
.size _GLOBAL__sub_I_main, .-_GLOBAL__sub_I_main	
.section .init_array,"aw"	
.align 8	
.quad _GLOBAL__sub_I_main	
.hidden__dso_handle	
.ident "GCC: (Ubuntu 7.5.0-3ubuntu1~18.04) 7.5.0"	
.section .note.GNU-stack,"",@progbits	