

Name: Christopher Carl
Email: ccarl2@fau.edu
Znumber: Z23146703

Lab Title: Simple Types

Answers:

1. A boolean uses 1 byte. A char uses 1 byte, an int uses 4 bytes, a float uses 4 bytes, a double uses 8 bytes. No errors to report.
2. Running this on the FAU engineering virtual machines causes a notification that UNC paths are not supported, and that CMD.exe is using the virtualized windows directory rather than the internet directory that drives the virtual machine.
3. bool: 0x01, or 1 as unsigned
char: 0xFF, or 255 as unsigned
int: 0xFFFF FFFF, or 4294967295 as unsigned
float: 0x7FFF FFFF, or 2147483647 as unsigned
double: 07FFF FFFF FFFF FFFF, or 9223372036854775807 as unsigned
4. Line 1: booleans can only have values of 0 or 1, so any number greater than 1 is truncated to 1, which corresponds to the boolean value true.
Line 2: 'D' is the 68th character in the ASCII char table
Line 3: Integers can only be whole values so the compiler automatically truncates the decimal.
Line 4: IEEE 754 Single Float Precision: Sign (1bit), biased Exponent (8bit), and Mantissa (23bit) results in ~8 significant bits
Here, the number starts rounding at the ~8th decimal place, and the rest is garbage
Line 5: IEEE 754 Double Float Precision: Sign (1bit), biased Exponent (11bit), and Mantissa (54bit) results in ~16 significant bits
Here, the number starts rounding at the ~16th decimal place, and the rest is garbage

- The boolean changes and stays at 0, which is one of the two allowed values. It corresponds to the boolean value false.

Lab Title: Declaration of Structures

Answers:

- The output is a prompt "Enter the firstname and lastname: "
- Output does not change, and the program behaves as previously.
- The assignment operator works exactly as it does for regularly typed variables.
- The logical equality operator does not translate to structures because they are many-typed collections of variables. What could be done is comparing a `struct1.var1 == struct2.var1`, and `struct1.var2 == struct2.var2`, for each of the variables in the structure.
- The same problem occurs as question 4.
- The same problem occurs as question 4.
- What could be done is comparing a `struct1.var1 == struct2.var1`, and `struct1.var2 == struct2.var2`, for each of the variables in the structure.

Lab Title: Pointers & References

Answers:

1.

Severity	Code	Description	Project	File	Line	Suppression State
Error	C4700	uninitialized local	COP3014L_2016R_LAB2	\\engvault01.eng.fau.edu\ccarl2\$\profile_documents\my documents\visual studio	11	

		variable 'p' used		2015\projects\cop3014l_2016r_lab2\cop3014l_2016r_lab2\ptrsa ndrefs1.cpp.cpp		
--	--	----------------------	--	--	--	--

It should be noted that `*p` was never initialized. `*p` is an address due to the `*` operator and must be initialized to something.

cout 1: prints the address of int l to the screen (changes every time)
 cout 2: prints the value of int p to the screen (would have been 0, but this is a `*int`, so it doesn't have a default declaration, and causes the error)
 cout 3: prints the address of int p to the screen (changes every time)
 cout 4: prints the value of of int l to the screen (90)
 cout 5: prints the value of int p to the screen (nothing was declared)

2. cout 1: prints the address of int l to the screen (changes every time)
 cout 2: prints the value of int p to the screen (0)
 cout 3: prints the address of temp to the screen (changes every time, but is assigned to the address of int i)
 cout 4: prints the value of of int i to the screen (90)
 cout 5: prints the value of a pointer to the variable p, which has been assigned to the address of int l, to the screen (90)
 cout 6: prints the value of a temp, which has the same address as int l, and therefore holds the same value as that address. (90)