# **CS241 Homework 1, Winter 20**

Data types and functions

Maximum Points: 120

**Due:** Friday, Feb 14, 2020

***Please answer all of the questions and submit your work through Blackboard. You can also print out a hard copy and turn in your hard copy in class.***

1. (10 points) Give at least 5 keywords that are explicitly reserved in standard ANSI C and briefly explain their use.

**Break – used with any loop or switch case, and will ‘break’ out of the loop or switch, but will NOT exit the function.**

**Int – data type used for a variable**

**If – used for decision control, logic**

**Return – This exits the function**

**Const - This represents a variable that won’t be changing. It’s good to use this to double check yourself if you KNOW you’re not supposed to change a variable.**

1. (10 points) Name 5 of the six types of tokens defined in ANSI C and give at least two examples for each of them.
2. **Identifiers – int, double**
3. **Keywords – break, static**
4. **Constants – const int, const float**
5. **Strings – char \* myString[], {‘h’,’e’,’l’,’l’,’o’,’\0’}**
6. **Operators – increment (++) decrement (--)**
7. (10 points) Which of the following are not identifiers in ANSI C and why? Points will be taken off if your answer is wrong.
   1. 3id ---------🡪 NOT
   2. \_good

These are not identifiers because of the following rule breaks:

1. Must have a letter or underscore in the first slot
2. Cannot have a ‘.’ Symbol
3. Cannot use the ‘\*’ symbol
4. Cannot use the ‘-‘ symbol
   1. cis.gvsu ---------------------🡪 NOT
   2. x2go
   3. YouAreGood
   4. starwall\* -----------🡪 NOT
   5. int
   6. xYshouldI
   7. me\_to-1 ----------🡪 NOT
   8. gogo00
   9. float0int
   10. \_\_yes
   11. ThisOneIsNot
   12. badOne
   13. shouldOK
5. (10 points) Please complete the following calculations in an ANSI C environment.
   1. 7 / 2.0 = ? 3
   2. 2 + 5 \* 2 = ? 12
   3. (5 + 2) \* 2 = ? 14
   4. 5 \* 2 / 3 = ? 3
   5. 5 \* (2 / 3) = ? 0
6. (10 points) We know that different number of bytes are used to represent different data types in C. For example, one byte is used for **char**, two bytes are used for **short**, four bytes are used for **int**, and so on.
   1. What are the tradeoffs of using different data types in your program?

**Using fixed-length data types, for instance using unit32\_t instead of unsigned int can prevent a program from doing unexpected things based. The tradeoff of using int, short, or char, is that it’s easier for the human eye to read.**

* 1. How can you ensure that the value being assigned to a variable is within the range that the data type of the variable can hold?

**Simply look at a chart!**

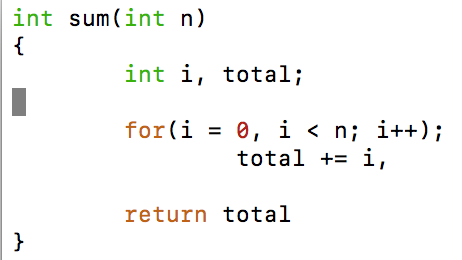
**Byte holds from -128 to 127**

**Short holds from -32768 to 32767**

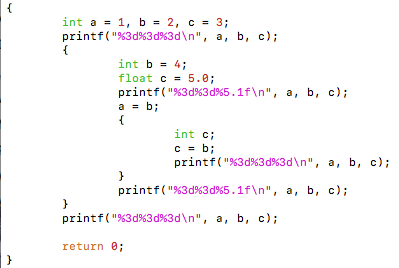
**…..**

**As long as you’re within range, life is hunky dory 😊**

1. (10 points) Find the errors in the following code and briefly explain them. This function is supposed to compute the sum of *1 – n*.



1. There’s no ‘;’ at line 8
2. The comma at the end of line 6 doesn’t belong – should have a ‘;’
3. The ‘;’ will just end the for loop, but it won’t trigger a compiler error.
4. Total should be initialized as ‘0’ to start, in order to accurately calculate the total.
5. If this is supposed to compute the sum of ‘l – n’, then I’m curious where the ‘l’ variable is.
6. Also would this not be computing the difference ‘l – n’?
7. (10 points) What are the differences between a C function and a Java method?
8. Functions can be defined outside of a class, whereas methods must be defined within a class.
9. Functions are defined in structured languages (pascal, c, javascript), and methods are object oriented
10. All data passed to a function is explicitly passed, and methods are used to manipulate instance variables.
11. (15 points) What will get printed when the following piece of code is executed?



**1 2 3**

**1 3 5.0**

**1 4 4**

**4 4 5.0**

**4 2 3**

1. (10 points) Use the library function **sqrt()** to write a function that returns the fourth root of its **int** argument ***k***. The value returned should be a **double**.

Double FourthRoot(int k){

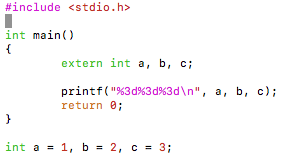
Double value;

Value = sqrt(sqrt(double(k)));

Return value;

}

1. (15 points) Use the program below to answer the following questions.



* 1. What will get printed? Explain your answer.

**1 2 3**

**Extern allows us to reach out to global variables. So we can reach out and touch the values of a, b, and c even though they aren’t defined inside main();**

* 1. Change the last line of the program to

**static int a = 1, b = 2, c = 3;**

Compile the program again. What will happen? Explain your observation.

**We can’t compile the program, because we have conflicting declarations of extern and static!**

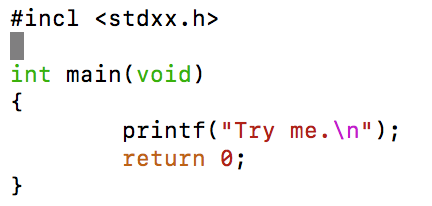
* 1. Change the first line in the **main()** function to

**int a, b, c;**

Compile the program again. What will get printed? Explain the results.

**For me, 32767 0 0. But these numbers change as the program is compiled. That’s because we can’t really reach out and touch the static definitions outside of the main function.**

1. (10 points) When you invoke a C compiler like gcc, the system first invokes the preprocessor. In this exercise, we want to deliberately make a preprocessing error, just to see what happens. Try the following program:



* 1. Type in the above code exactly as what it is. Compile it and observe what happens. Summarize your observation.

**The preprocessor immediately tries correcting us and asks if we meant to say “#include”…it also asks that we include stdio.h for the printf() function.**

* 1. What happens if you change **#incl** to **#include**?

**“#include <stdxx.h>”: No such file or directory**

I mean…it pretty much explicitly states the problem. Header file <stdxx.h> doesn’t exist in the path. We could create our own and replace the <> with “”, but in the current setting it won’t work.