Functions in C (part 3 of 3)

CMSC 104 Section 02
April 17, 2024

Administrative Notes

Practice quiz 4 is now available on Blackboard

- We'll go over it in class next Monday

Classwork 7 is due tonight

- Finish it before you do Homework 7, since the homework is an extension of the classwork

Classwork 8 is out today

- Due next Wednesday, April 24

So what should you know about functions so far?

All the stuff from last week:

1. Functions in C are blocks of code, or modules, that are part of the program.
2. They have ***names***; they take zero or more inputs as ***parameters***, and they ***return*** zero or more values
3. Every C program has to have a function called main. In this class, main always returns one integer value, and it always takes zero parameters. That's why we always see
4. Every other function has a ***function prototype*** in the ***main*** function. It warns the C compiler that there will be function that looks like this defined later in the prog
5. In this class, the ***function definition*** comes after the full ***main*** function. It starts with a type identifying what type of values are returned, followed by the function name, followed by a list of ***formal parameters*** including the type and name of each parameter; and then the code that makes up the function.
6. A function that has no return value is of type `void`. If there are no parameters to the function, you can either write (`void`) as the parameter list or you can leave an empty parameter list ( )
7. The ***function call*** occurs in the main program, or in another function*. The function call has the name of the function, followed by a list of ***actual parameters*** whose values are passed to the function.

*We haven't gotten to one function calling another yet.

… and from Monday's lecture

8. Parameter passing: actual parameters and formal parameters are matched by ORDER, not by NAME

- We'll see an example of that today

9. Parameter values are passed using "call by value" meaning the function uses a different area of memory, has its own symbol table and does not impact the main program except through return values

10. "Variable scope" means in what parts of the program the variable is known/defined. "Local variables" are those defined in functions.

- All formal parameters are local variables
- Additional local variables are defined in the function code

11. Return values are values that are passed from the function back to the main program.

- They are of the type specified in the function prototype
- The value is associated with the function's name in the function call

More stuff about functions

We’re going to write a program that gets the user to input the height, weight and depth of a rectangular box

- Each side is a rectangle; each intersection is a right angle

We’re going to use a function to get each input and make sure that it’s greater than 0.

- The code is on the github repo.