Matthew Lonis

Jeff Whitmer

CSCI-A 290

August 27th, 2016

MINI 1

**Topic 1: Data Types in C**

Hoover Précis

Hoover states that there are only four basic data types in C. These data types are int, float, double, and char. The int is used to store whole numbers. The float is intended to store real numbers. The double is intended to store real numbers as well but has twice the precision of a float. Finally, the char is meant to store character symbols and controls used to display text.

Hoover Summary

Basically, Hoover claims there is only four primitive data types in C programming. The four data types are int, float, double, and char. In my experience I have used all three of these data types in C programming, however I think it is also important to note how most of the mathematical data types also have a short and a long version of the data type.

2 URLs

<https://www.le.ac.uk/users/rjm1/cotter/page_19.htm>

<http://www.tutorialspoint.com/cprogramming/c_data_types.htm>

Summary of URLs

1. First URL overall covers most of what Hoover covered but included void as a primitive data type. It also briefly discusses naming schemes for C programming saying how all C variables have to start with a letter or a “\_”. Other than this, this website didn’t shed a whole lot more light on C data types.
2. This URL went really in depth on C data types. It created four types. These data types are basic types, enumerated types, the type void, and derived types. The basic types are what was covered in Hoover and the first URL. However, the derived types was a new data type. This type included pointers, arrays, structures, unions, and function types.

Summary of Points in All Resources

There are four or five primitive data types in C but those can be built up into more abstract data types.

One Big Question

How did pointers become a derived data type?

**Topic 2: Conditionals in C**

Hoover Précis

Hoover states most conditionals are an if-else statement. C supports inequality testing as well. The condition in the conditional must be satisfied in order for the then in the if-then statement to be processed. It’s a very simple step-by step process really.

Hoover Summary

Conditionals in C are kind of basic but efficient if you know you can run code through a simple battery of test cases and move on. Loops are generally a more worthwhile way of implementing C functions but conditionals can work. Conditionals are honestly fairly similar to most of the other programming languages so I don’t really have much else to say.

2 URLs

1. <https://www.cs.cf.ac.uk/Dave/C/node5.html>
2. <http://www.cprogramming.com/tutorial/c/lesson2.html>

Summary of URLs

The first URL talks about conditionals. It brings up some things Hoover didn’t because it lists out examples of conditional if statements. It also lists and describes what the different inequality symbols are including the ‘?’ symbol. Overall good resource.

The second URL talks about the same things but gives a difference on ‘else’ and ‘if else’ in conditional statements. This URL gives some good Boolean operation examples.

Summary of Points in All Resources

Conditionals are similar to conditionals in many other programming languages.

One Big Question

Why are conditionals seemingly so one dimensional?

**Topic 3: Loops in C**

Hoover Précis

There are three basic loops in C. for loop, a while loop, and a do-while loop. The for loop is intended to be executed a fixed number of iterations known before the loop is entered. The other two loops do not have a given known set of iterations. Hoover then gives examples.

Hoover Summary

Loops in C resemble loops in other languages. Loops are more versatile than conditionals alone. The for loop is important to know how long you want it to end and be sure to set that when creating the loop. The other loops, since they don’t have a built in end, can become infinite if improperly implemented.

2 URLs

1. <http://www.cprogramming.com/tutorial/c/lesson3.html>
2. <http://www.tutorialspoint.com/cprogramming/c_loops.htm>

Summary of URLs

The first URL talks about the same as Hoover but emphasizes that each type of loop has its specific use. The for loops is used when you want to declare a variable and increment it in the loops. The others are used interchangeably to determine whether or not you want to loop to process its ‘do’ portion of the loop before or after the conditional.

This is honestly the same as Hoover and the above. The only good thing is it brings up nested loops and loop break statements. Both of these concepts I find to be very crucial in one’s understanding to loops in c programming. I find nested loops are good to create a sort of menu in the command line prompt or to simulate choices.

Summary of Points in All Resources

The three loops in C all have their uses.

One Big Question

Who uses do…while loops?