CSCI1320 Final Review Sheet

The final for this class will be formatted in a manner very similar to the midterm. It will focus primarily on the material from the second half of the semester, though material from the first half will certainly be present on it. The final will have 10 questions and while they might be a bit more challenging than those on the midterm, you still should not have any problems with time.

Topics:

Files

You need to know how to use files in code as well as what they are good for. This includes both text and binary files. If you have to write code involving files you don't have to have the function call syntax perfect, but it needs to be close enough that I can understand. It should definitely include the right names of the functions.

Arrays

Know how to declare and use arrays. You need to know how to store values in an array and get them out. You also need to understand the syntax and semantics of passing arrays to functions. This includes multi-dimensional arrays.

Sorting

You need to understand how each of the 3 $O(n^2)$ sorting algorithms works. You should be able to write the code for them or show the way in which they sort a specified array by writing the state of the array after each iteration or the outer loop. It might also be helpful to know how the O(n log n) sorts work for the extra credit. Those are quicksort and mergesort.

Searching

Know how sequential and binary searches work. You need to know when each one can or should be used and be able to describe why the binary search is faster. Also you should be able to write code for either type of search.

Pointers

You need to understand pointers and pointer syntax in C. This includes passing values by reference and their use with dynamic memory. You also need to understand pointers as types and the types of various pointer expressions. If I give you a variable declaration and an expression you should be able to tell me the type of that expression. This also includes expressions involving structures.

Recursion

Know how to write and trace recursive functions. You should also understand when recursive functions can/should be used to aid in problem solving. How do they compare to loops and when are they superior to loops?

Dynamic memory

You should understand the use of malloc and free in C for dealing with dynamic memory. You also need to understand the advantages and difficulties that come with using dynamic memory.

Strings

You need to understand the concept of making a string from a null terminated array of characters. You should also be able to write any of the main functions for dealing with strings without calling the functions in the string.h library or be able to trace code the I give you that manipulates a string in some way.

Enumerations, Structures and Unions

Know how to declare and use the user-defined types in C. You also need to know how these types are helpful and when you want to use them in your code.

Linked Lists

You won't have to write code for a linked list, but you should understand the concept of them and be able to trace code if it is presented to you.

Note that at this point you know virtually all of the C-language. You should be able to trace any piece of code that is given to you as long as it doesn't include function calls that we haven't talked about.