Ecs36b Spring 2023 Final Exam Review

In general, knowledge about Makefile and g++ compiler options are needed for all the homework assignments. In other words, you might see such elements in several of the final exam questions.

Regarding source code used in the final exam, in the document, I gave examples of code from github. As a warning, you might be seeing source code different from those or something different during the final exam.

HW#1

Working with Files

How to verify if the output file is exactly as you expected (e.g., using the command “diff”)

Can you write a program to verify if two files are the same (i.e., your own implementation of “diff”)?

HW#2

Work with data, in particular, words (like strings), with some given data sets such as Solution.txt or All.txt. During hw2, some of us developed strategies which use the “impossible” array to eliminate the candidate set so we can narrow down the possibilities. For instance, I talked about such a strategy and went over the source code in

<https://www.youtube.com/watch?v=UIrIh6Oj0ss>

So, a possible extended question would be to be given a segment of code to eliminate certain words from the files (not just array in the memory). Related questions: what is/are the difference(s) of information being stored between memory (such as an array or vector) and files? Another question here is the function “strdup”. What is the correct implementation of strdup?

strcpy(malloc(strlen(src) + 1), src);

free()

HW#3

This was our first assignment involving the design of C++ classes.

<https://github.com/sfelixwu/Person-and-Thing-HW3-Reference>

<, >, ==, >=, <=, !=

Call by value, Call by reference (call by name)

A few important concepts here – operator overloading (remember it is actual a template under #include <utility>), difference between an Object (e.g., Person x) and an Object reference (e.g., Person& x), constructors/destructors, public/private. Furthermore, the design and implementation of the member function called “dump2JSON”.

HW4

The main focus here is to bridge the gap between two different object representations, C++ and JSON. Among other things, we also emphasized the differences between compiler time and run time. The source code was at

<https://github.com/sfelixwu/0419_2023>

and

<https://github.com/sfelixwu/mobility>

(Also, this might be the first time, you were given the source code of ecs36b\_JSON.cpp – you might be asked questions related to the implementation of functions in ecs36b\_JSON.cpp – we will provide code segment if this is going to happen).

Here, one important concept is how to convert a JSON into a C++ object with the same design. And, how can this capability be utilized to implement Persistent objects or Mobile objects.

HW5

Homework 5 is all about inheritance, virtualization (virtual functions, virtual inheritance, and vtable pointers), and memory layout. This particular part of the course is to learn the one level deeper into the C++ language and its implementation. The source code is at

<https://github.com/sfelixwu/memory_layout>

Also, you should know how to use the g++/clang to generate the memory layout (it is actually in the Makefile of the above directory).

Finally, you will be asked questions related to the proper usage of Pre-processor options such as #ifdef #endif. Or, the -D, -I, and -L flags in g++ or Makefile.

HW6

This homework is about exception handling –

<https://github.com/sfelixwu/exception>

HW#7

Template functions and classes, lamda, function pointers, and STL –

<https://github.com/sfelixwu/openai_jsoncpp>

<https://github.com/sfelixwu/cpp_stl_map>

map, pair, and vector

iterator