COP 3223C Section 4

Final Exam and Lab Exam 2 Overview

Time and Location of the Final Exam and Lab Exam

- **Section 4**
- **Exam date:** Tuesday December 7, 2021
- The written exam will start from **4:00 PM and due at 5:45 PM**
- **As soon as you submit your written exam, you can start your lab exam which will be one hour timed exam in webcourses (Exam password will be provided in the class)**
- **Total duration for the written exam:** 1 hour and 45 minutes
- **Location: Exam will take place in your regular class room**

Exam Environment and Setup

- **The Written Exam:**
  - **For the written exam, I will provide you the question and you have to answer within the question paper**
  - **You can use only 4 function calculator**

- **Lab Exam:**
  - **You must need to bring a laptop in the classroom to take the lab exam. If you don't have a laptop, you can borrow one from the library**
  - **The lab exam question will be at the last page of your written exam. You will separate the lab exam question from your written exam while submitting the written exam**
  - **After submitting your written exam, you can turn on your laptop and start your lab exam.**
  - **It will be like a webcourses quiz with free text box where you will submit your code**
  - **In order to test your code, you will have access to online gdb compiler or repl.it while working on the lab exam to test your code**
  - **You must have to show your submission screen before leaving the classroom**

- **Install UCF Here App:**
  - Please install ucf here app to take attendance during the exam.

- **Total point for the written exam:** 100 (It will cover 22% of your course grade)
  - ***You must have to get at least 35% in the final exam to get C grade or better***

- **Total point for the lab exam:** 100 (It will cover 5% of your course grade)

Lab Exam Topic and Preparation

- The lab exam will cover ***1D array, 2D array, pointers, strings, file I/O, and structure. There will be extra credit for using dynamic memory allocation!***
- Naturally, the other old topics are included too. ***If you have a good understanding of the concepts, if you have done the exercises, weekly coding, lab problems, and the code we have discussed in the class, you should be easily able to write the code. Consider writing more codes, re-write the codes we have in the slide without looking at the solution. Think about how the programs were designed, how the logic was built, and how the programs were implemented at the end. You are not allowed to use any additional help during the exam. Use the provided question paper as scratch paper as generally it is helpful to scratch ideas before writing a code.***
- ***Again, practice!!! Write code that has file io, file reading and writing, string processing, Matrix and 2D array diagonal elements, finding max, min, searching a particular member in structure and printing the remaining members based on that, loading data from file to array of structure, etc. Understanding why your code is working and why it is not working is very important.***
- **Past Coding solutions:**

All past coding solutions will be available by Friday night at the "lab and weekly coding solution module" in webcourses. Please go through them even if you have passed all the test cases and received a good score. Because, a code can be written in many different ways, but looking at our solution would help you to see how the code can be written in a different and better way sometimes.

Topics for Final Term (Not cumulative)

- Lecture slide 9 (Array) to Lecture slide 16 (dynamic memory allocation)
- The topic includes:
  - Arrays
  - Sorting (basic idea)
  - 2D arrays
  - Pointers
  - Character processing and Strings
  - File I/O
  - Structure
  - Dynamic memory allocation(**except the last example-Dynamic memory allocation for a 2D array**)
  - ~~(note that recursion is self study and not part of the exam)~~

Question Pattern (read it and practice accordingly!)

- **You got some ideas about pattern of questions from the quizzes, but as the exam has more time, there will be more questions with various level of difficulties.**
- **There will be enough time to answer them all if you do not mess-up much.**
- **Question can be:**
- **Multiple choice questions/True False/fill in the blanks**
- **Error and bug tracing question**
- **Output tracing question**
- **Conceptual question: for example, differences between typedef and not typed, differences between malloc, calloc, realloc, when to use dot and when to use arrow, explain issues in a give program, memory leak, segmentation fault, double free, various functions in string.h library, nested structure, etc.**
- **Writing C sytax for various types of operations**
- **Programming questions**
  - **Writing a full program with various functions**
  - **Just writing a function for a given problem where you don’t need to write main function**

How to prepare

- Remember, this is **NOT** a course where I will say to memorize the factorial function, memorize the string reverse function, memorize the file matching function, etc.
  - But we saw, how to think about a problem, how various syntax work, how to write functions and call a functions, and how to solve a problem
- So, this course is more about understanding the C syntax and use them for solving various problems.
- So, in the exam, you will demonstrate your understanding by answering the questions.
- Read the next slide for more on preparation

How to prepare

- If you have prepared well for the quizzes, you have already prepared yourself for those part of the topics covered in the quiz3 and quiz4.
- If you have done your coding assignments yourself, completed the lab coding with good understanding, you should have pretty good knowledge already.
- If you have done the webcourses exercises properly and understand them clearly, you should have a good breath of knowledge on the topics.
- Go through the slides, notes or any uploaded codes to understand the concepts. Exercises are not the replacement of the examples of the lecture notes.
- Practice and go through the examples in the slides.
- Go through the uploaded exercises and solutions.
- If there is something in the slide, **try to do it yourself**
- **Ask yourself (Self check):**
  - Can you do the examples in the slide and their varieties without looking at the solutions?
  - Do you know why a line was written in a particular example?
  - Do you know all the syntax discussed in the sides and can you use the syntax for another situation?
  - Do you know the reason behind an output in the slide? Can you derive them on your own by tracing the code?
  - Can you answer the exercises questions accurately without looking at the solution**, not by memorizing**, but by tracing.
  - **Do you know why a particular answer is correct and why the other answers are wrong for the exercises?**
  - **Practice accordingly so that you can answer the questions in a timely manner.**

Any Question?

- It was a great semester with you. I am very happy to see many of you can write pretty good code and how you have improved your coding!
- I know sometimes you got into pressure, but it was expected as mentioned in the syllabus
  - All of the codes and assignments you have done, they will benefit your future coding.
- For those who are CS/IT major:
  - DO NOT STOP CODING!!!
  - Try to solve interesting problems on whatever you have learned. Make sure you are master of structure and pointer (in addition to all the other topic) and have some knowledge of Dynamic memory allocation. You will need them heavily for CS1 course.
- For other major, programming is fun and keep coding! Maybe jump and explore python programming language as you have some programming background now!
  - I say again, knowing programming is an asset. Use it and explore more.
- I wish you all the best
- I love you all and I will miss your curious faces during the lecture!
- And of-course stay safe and take care yourself
- ☺ Good luck for the final exam! ☺