DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
UNIVERSITY OF TENNESSEE - KNOXVILLE

COSC 202 Data Structures and Algorithms I
Spring 2024 Syllabus (Draft)

Time and Location: Tuesday & Thursday 11:20 pm-12:35 pm, Min Kao 622

Instructor:

- Dr. Scott Emrich
  - Office: 608 Min Kao; 974-3891; semrich@utk.edu;
  - Tentative office hours: Mon 2:00pm-3:00pm, after class and by appointment

An Invitation

It is my goal for you to learn simple data structures (then more complex ones in the fall), get better at both problem solving and coding, and along the way enjoy this course! I would like to balance us having a positive experience with developing real, applied skills that will be highly useful to you in interviews, this summer, and beyond. Remember the TAs and I are here at every step to help. To support your learning, we will have both active learning experiences and a dedicated lab period on Wednesday.

Course Webpage: https://web.eecs.utk.edu/~semrich/cs202-24/

Short Course Description

This course is part one of our two course sequence that focuses on fundamental data structures and associated algorithms. During the semester we will cover basic data structures with a focus I/O, files, pointers and basic memory management, lists, queues, stacks, hash tables and an introduction to algorithm analysis. For these topics students are expected to design and implement code with the overall goal of becoming more self-sufficient (C++) programmers. All programming will take place in a Unix environment and will be reinforced by weekly assignments.

Textbook

(optional) *C++ How to Program* by Deitel and Deitel;
additional material will be made available as needed throughout the semester.

Course outcomes

By the end of the second course in this sequence, you will be able to competently design C++ objects/classes, with information hiding, and understand and use more advanced data structures and methods including but not limited to lists, maps, stacks, queues, trees, sorting, graphs, and dynamic programming. Specifically, you will be able to:

1. Demonstrate understanding of, and proficiency in use of C++/object-oriented concepts including data hiding, templates, and design patterns. Assessed via exam and programming assignments.
2. Analyze the performance of data structures in order to select the right one for each situation, as well as create or extend data structures to fit new situations. Assessed in exams and in programming assignments.
3. Combine data structures to solve real world problems, employing abstractions to make them work together cleanly and safely. Assessed through the final lab.

Major Topics:

1. Review of C++/programming basics inc. strings and OOP (5 hours)
2. Iterators, polymorphism, exceptions (3 hours)
3. Pointers, lists and deques (4 hours)
4. Hashing (3 hours)
5. Sets and maps (2 hours)
6. Stacks, Queues, doubly linked list data structures (2 hours)
7. Recursion (3 hours)
8. Running times and Big-Oh (3 hours)
9. Basics of trees: binary search trees and AVL trees (5 hours)
10. Priority queues and binary heaps (2 hours)
11. Special topics / TBD (7 hours)
12. Collaborative reviews and midterm (2.5 hours)

Grading: In the past I would never give an individual an extension or grade late work. This semester you can submit one updated assignment for re-grading by the end of the semester, which will be used for computing your final grade. Labs will be accepted late with a penalty (likely 10%) applied per day. You can remove this penalty instead of uploading an updated assignment if you prefer that option.

Final grades will be computed from a weighted sum of points as follows, although I will probably take informal feedback from you on the weighting:

65%: homework (including programming and written answers submitted)
10%: midterm exam (Thursday, 2/29 tentatively)
15%: final exam
10%: class participation

Course percentages will be translated into letter grades as follows: A: 95% and up; A-: 92-95-%; B+: 88-92-%; B: 85-88-%; B-: 82-85-%; C+: 78-82-%; C: 75-78-%; C-: 72-75%; D: 65-72-%; D-: 62-65; F: 0-62-%. Absences will only be excused in accordance with University policy.

Final lab: A slightly larger lab/project may be due at the end of the semester. For now, these will be solo projects, but we can discuss whether you can also work in pairs.

ADA statement: If you need an accommodation based on a disability can contact Dr. Emrich privately. Full accommodation will be made once approved.

Academic Code of Honor: Any instance of academic dishonesty will not be tolerated. All graded work should go from your head to your fingers to submission; no copying of complete solutions or code that solves a key topic being assessed (group or online).

Like your non-EECS courses, you are encouraged to use "resources" to help solve non-critical elements of the labs. For example, suppose you forget how to convert a string to a number (int) in a lab focused on linked lists. You can google and find info on a site like GeeksforGeeks but you ** must ** cite the URL. You can also ask a generative AI/co-pilot, but you must cite in your comments what you used, and it is your responsibility the solution works as intended (aka passes tests). If in doubt, ask me or the TAs on course management sites/lab/lecture to confirm.

You are encouraged to discuss high level solutions with peers but to help us, please note who you spoke with about what in the comments of your submitted solution.

Attendance and Time management:

No matter how you look at it, programming is a time-intensive activity that is best done throughout the week (and not right before it is due). To be fair to all, all regular labs will be due before the next lab period. Please submit partial work into your drop box.

Students are expected to attend and contribute regularly in any modularity (currently in person). In addition to participating in discussions, some challenges will be provided to reinforce course content and provide a platform to start working with others.

<u>Class participation will be assessed as follows:</u>

0.5 points – Show up at lab on time (once per week)
1 points –Participate in a lab by having a "meaningful" conversation with a TA that indicates that you are making progress (see below; once per week.)
1 point – Stop by Dr. Emrich's "virtual" office hours this semester (max one point)
0.5 points – Participate in a classroom exercise or follow up (max seven points.)

During the pandemic, I shrunk labs from 3 hours down to roughly 1 hour and added a "prep" video. That ended up working very well and the videos were one of the most beloved elements of the last go-round of 202. We will reinforce the video with effectively a forced office hour that you will get easy participation credit for attending. We really want to have a "I've gotten this far and my code does this; I've set it up in this way" conversation, ideally not a "What is lab?" conversation since your time is valuable. Please do feel welcome to ask us for explanation about the instructions, though; we're happy to help during lab and in other venues (Piazza, Discord).

Your class participation grade will be calculated as (points earned / 21 estimated), so you can earn a few extra credit points by actively participating in lectures and pass/fall challenges throughout the semester.