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

COSC 302/307 Data Structures and Algorithms II
Spring 2023 Syllabus (DRAFT)

Time and Location: Tuesday & Thursday 11:20 pm-12:35 pm, SRF 307

Instructor:

- Dr. Scott Emrich
  o Office: 608 Min Kao; 974-3891; **semrich@utk.edu**;
  o Tentative office hours: Mon 2:00pm-3:00pm, after class and by appointment
  o If my office door is open, you are welcome to come in and ask questions.

An Invitation

It is our goal for you to learn more data structures, get better at problem solving and coding, and along the way enjoy this course! We 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 we are here at every step to help. We will have both active learning experiences and in-class presentations. I will also try to briefly attend an occasional lab to answer questions one-on-one.

Course Webpage: http://web.eecs.utk.edu/~semrich/cs302-23/

Short Course Description:

- This course is a third-semester programming course in C++ that focuses on fundamental data structures and associated algorithms. The course will review basic object-oriented programming (OOP) and cover sorting algorithms, disjoint sets, basic graph algorithms including toplological sort, depth-first search, and breadth-first search, shortest path (Dijskreta's algorithm), minimum spanning trees, network flow / minimum cut, and dynamic programming with memoization. For the algorithms listed above students are expected to design and implement C++ programs that solve related problems.

Prerequisite:
CS202 (or equivalent)

Textbook: Main and Savitch, *Data Structures and Other Objects using C++*

Additional material will be made available as needed throughout the semester.

Course outcomes:

At the end of the course, you will be better able to competently design C++ objects/classes, with information hiding, and understand and use new advanced data structures and methods including but not limited to 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, inheritance, 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 presentation and project.

Major Topics:

1. Review of C++/OOP basics (2.5 hours)
2. Templates including arrays and lists (2.5 hours)
3. Sorting (4 hours)
4. Disjoint sets (2 hours)
5. Graphs (6 hours)
6. Network flow (2.5 hours)
7. Dynamic programming (2.5 hours)
8. Complexity (2 hours)
9. Special topics (7 hours)
10. In-class examples of DS: BYODS week! (4 hours)
11. Collaborative reviews and midterm (4 hours)

Grading: By default, I will never give an extension or grade late work. Full stop. It is easy enough to get behind in CS302 and if you need a few extra days it will usually lead to the next assignment being late. However, any student can submit one updated assignment for re-grading by the end of the semester, which is basically an extension for any single assignment. The revised assignment will be used your final grade.

Final grades will be computed from a weighted sum of points as follows, although I will very likely take feedback from you on the weighting. The default weighting is:

45%: homework (including programming and written answers submitted)
15%: final project
15%: midterm exam (Thursday)
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.

Project: A final project involving teams of students will be due at the end of the semester. Teams must consist of two to four students (solo projects won't be allowed).

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. Because I want you to study and discuss course-related work with classmates, but will require you to submit independent assignments/programs. In short, all graded work should go from your head to your fingers to submission; no copying of solutions (group or online).

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 the night before it is due). To be fair to all, labs will be due the Friday after you have lab. By default, no labs will be accepted late! Please submit partial work into your drop box as instructed in the lab submission guideline and we will provide partial credit along with constructive feedback.

Students are expected to attend and contribute regularly to class. This means answering questions, participating in discussions and activities, and helping other students.

Class participation will be assessed as follows:

1.5 points –Participate in a lab by spending at least 1 hour in attendance (once per week.)
1 point – Stop by Prof. Emrich's office hours during the semester. (max one point)
0.5 points – Participate in a classroom exercise during lecture. (max seven points.)

Your class participation grade will be calculated as (points earned / fewer points than possible), so you can earn a few extra credit points from perfect attendance. Because the instructor realizes that sometimes the most recent version may not have been submitted, concepts can be mastered later, or things just happen, the instructor will accept a single lab for regrading without penalty (lab amnesty) at the end of the semester.