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

COSC 302/307 Data Structures and Algorithms II
Spring 2019 DRAFT Syllabus
(pending in class review)

Time and Location: Tuesday & Thursday 11:10 am-12:25 pm, Min Kao 419

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
  - 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 I are here at every step to help. With this in mind we will have both active learning experiences and in-class presentations. I will also try to attend each lab briefly to answer questions one-on-one.

Course Webpage: http://web.eecs.utk.edu/~semrich/ds19/

Short Course Description:

- This course is a third-semester programming course in C++ that focuses on fundamental data structures and associated algorithms. The course covers basic object-oriented programming (OOP), 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 (and classes) that solve related problems.

Prerequisite:
CS140 (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 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 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 basic 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. Assessed in the final presentation and project.

Major Topics (estimated):

1. Review of C++/OOP basics (2.5 hours)
2. Templates including arrays and lists (2.5 hours)
3. Sorting (5 hours)
4. Disjoint sets (2 hours)
5. Graphs (6 hours)
6. Special topics/DS (4 hours)
7. Network flow (2.5 hours)
8. Dynamic programming (2.5 hours)
9. Complexity (2 hours)
10. In-class examples of DS: BYODS week! (2.5 hours)
11. Collaborative reviews and midterm (3.5 hours)

Grading: I will never give an extension or grade late work. Full stop. Details follow, but -- in short -- I expect you to be able to complete most of the Challenge work and develop a solid skeleton for projects within lab and I will reward you for this.

As an alternative to an extension, you can (re)submit one updated assignment for re-grading by the end of the semester. The revised assignment will be graded without penalty and used for computing your final grade. The implication is you must complete all assigned work, but you can either "punt" or redo a single assignment (if desired).

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

55%: homework (including challenges, projects and reading assignments submitted)
10%: final project
10%: 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, please refer to this document I co-wrote at ND for guidance: http://cse.nd.edu/undergraduates/courses/honor-code. I intend to use the "default" table listed this semester. 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 encourage you to start early and leverage time on campus during the week, challenges will be due the Friday immediately after lab. By default no labs will be accepted late! Please upload partial work, however, as instructed in the lab submission guideline and we will do our best to provide partial credit along with constructive feedback.

Students are expected to attend and contribute regularly in class. This means answering questions, participating in discussions and activities, and helping other students. This is for sure the best way to maximize your performance this term.

<u>That said, I will also assess class participation as follows:</u>

0.5 points – Show up at your scheduled lab on time for attendance (once per week)
1 point –Actively participate in a lab by getting a "sign off" from a TA on your Challenge effort so far (once per assigned lab) *Can be done immediately after attendance*.
1 point –Actively participate in a lab by getting a "sign off" from a TA on your current Project progress (once per assigned lab) *Can be done after all initial Challenge checks*
1 point – Stop by Prof. Emrich's office to say hi/get help (max one point)
0.5 points – Participate in a classroom exercise during lecture. (max six points.)

Your class participation grade will be calculated using a smaller denominator than the total earnable points, <u>so you can earn a few extra credit points by near perfect class and lab attendance throughout the semester</u>. Similarly, being late once or twice or missing a class for personal reasons should have minimal impact. That said, poor lab participation can reduce your grade by one level (A to A-) and poor class attendance can reduce your grade if you are on a boundary.