CSCE 3110 - Data Structures and Algorithms

Course Information & Syllabus (Fall 2021)

https://hengfan2010.github.io/teaching/21F-3110/index.htm

Basic Course Information

- **Instructor:** Heng Fan (heng.fan@unt.edu)
- **Office:** Discovery Park F284
- **Office Hours:** Thursday 3:30 – 5:30 pm or by appointment
- **Time:** Fall 2021: Tuesday/Thursday 1:00 - 2:20 pm
- **Classroom:** NTDP B142
- **TA:** Shiva Ebrahimi (ShivaEbrahimi@my.unt.edu)
  Office: TBD
  Office hours: TBD

Recommended Textbooks

- **Data Structures & Algorithm Analysis in C++ (4th edition)**, by Mark Allen Weiss
- **Introduction to Algorithms**, by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein

Prerequisites

CSCE 2100 and **CSCE 2110** or equivalent. You need to know how to write C++ program and compile on your own, and basic knowledge of elementary data structures.

Course Description

The aim of this course is to provide an introduction to the design and analysis of fundamental data structures and algorithms. The lectures will emphasize the theoretical aspects, while the homework assignments will focus more on the programming and hands-on experience, meant to reinforce the theoretical aspects covered in lectures. Topics include:

- Time and space analysis
- Recursion and recurrence relations
- Review of basic data structures, including arrays, lists, stacks, queues, etc.
- Tree-based data structures, including heaps, BSTs, AVL trees
- Hashing
- Data structures for storing graphs, elementary graph algorithms and their applications
- Algorithms for solving minimum spanning tree problem and their implementations

ABET Outcomes

After completing the course satisfactorily, a student is expected to:

- Understand time complexity of algorithms
- Be able to solve recurrence relations
- Understand and be able to analyze the performance of data structures for searching, including balanced trees, hash tables, and priority queues
- Apply graphs in the context of data structures, including different representations, and analyze the usage of different data structures in the implementation of elementary graph algorithms including depth-first search, breadth-first search, topological ordering, Prim's algorithm, and Kruskal's algorithm.
- Be able to code the above-listed algorithms

Grading

- **Quizzes:** 20%
- **Assignments:** 40%
- **Midterm exam (closed book):** 15%
- **Final exam (closed book):** 20%
- **Attendance:** 5%
- **Course project (optional):** 10% (Bonus)

Quizzes: There will be 5 or 6 in-class quizzes. Each quiz may contain 3-4 questions. There will NOT be any makeup quizzes.

Assignments: There will be five or six homework assignments (mixed with written and programming exercises).

- You are expected to do homework assignments (5 or 6) by yourselves. Even if you discuss them with your classmates, you should turn in your own. Do NOT share your code!
- Each assignment will specify the material to be turned in. All programming will be in C++ and must compile on a University Unix/Linux machine. No credit will be given for programs that do not compile.
- Assignments are due before class on the due date. Assignments may be turned in electrically using Canvas. A late penalty of 10% will be applied to all late assignments for up to 3 calendar days. No credit will be given after 3 days.

Midterm and final exams: The mid-term exam will be during class on TBD. The final exam will be on TBD.

Attendance: Attendance may be checked on randomly selected days. You are responsible for any missed material and completing all work by the assigned due dates. You should notify the instructor of your absence as soon as possible. If you miss more than four lectures, your grade will be dropped to the next level in the grading scale.

Course project: The course project is optional, and you should work on your own for the project. It will be a coding task to implement a practical system using appropriate data structures and algorithms.

Students may discuss the project, but don't share the solution and code. Details will be announced in a certain class.

Grading Scale (based on 100 points)

90-100 = A

80-89 = B

70-79 = C

60-69 = D

below 60 = F

UNT Policies

Academic Integrity and Consequences: According to UNT Policy 06.003, Student Academic Integrity, academic dishonesty occurs when students engage in behaviors including, but not limited to cheating, fabrication, facilitating academic dishonesty, forgery, plagiarism, and sabotage. A finding of academic dishonesty may result in a range of academic penalties or sanctions ranging from admonition to expulsion from the University.

Most lectures in class will have homework assignments. Students may discuss the homework problems and approaches with each other but must work on their solutions individually unless otherwise stated in the assignment. Students must not copy homework from any source, including other students or the internet. No collaboration is allowed in quizzes and exams.

ADA Policy: UNT makes reasonable academic accommodation for students with disabilities. Students seeking accommodation must first register with the Office of Disability Accommodation (ODA) to verify their eligibility. If a disability is verified, the ODA will provide a student with an accommodation letter to be delivered to faculty to begin a private discussion regarding one's specific course needs. Students may request accommodations at any time; however, ODA notices of accommodation should be provided as early as possible in the semester to avoid any delay in implementation. Note that students must obtain a new letter of accommodation for every semester and must meet with each faculty member prior to implementation in each class. For additional information see the ODA website: https://studentaffairs.unt.edu/office-disability-access.

Disclaimer

This syllabus is to serve as a guide and may be subject to changes. Up-to-date information, assignments, and class material can be found in the course space on Canvas. This syllabus may be updated to reflect changes. The updated version will be available in the course space on Canvas.