EECS 233: Introduction to Data Structures
Fall 2019

Instructor

Dr. Erman Ayday
Office: Olin402
Email: exa208@case.edu
Phone: 216-368-5029

Course Description

EECS 233 is an introductory course in computer science. Its primary objective is an introduction to various methods of organizing large amounts of data, and basic analysis of algorithms that use these methods. The course will cover linked lists, stacks and queues, hashing, searching and sorting, some tree structures and basics of graphs. Simple analysis of algorithms is also included for hashing, searching, sorting, etc. Java programming language will be used for lectures and assignments.

Prerequisites

ENGR 132, knowledge of Java.

Textbook(s)

- Data Structures and Algorithm Analysis in Java (Second or Third Edition), by Mark Allen Weiss, Addison-Wesley (Optional)

Logistics

Class time: T Th, 1:00-2:15.
Room: Rockefeller 301

Instructor office hours: Th 2:30 – 3:30 pm or by appointment.

TAs: TBD
TA Office Hours: TBD

Course Content and Requirements

Students are required to attend lectures and are expected to master all the material covered in class. Assignments missed due to reasons other than medical conditions cannot be made up. Prior experiences show that students regularly attending lectures usually perform better in the exams than those who do not.

The course will include four programming assignments, four written assignments, a midterm exam, a possible quiz (a short exam), and a final exam.

Students should use electronic submission for both assignment types. We will use a software system to detect code copying. Assignments are due before midnight on the due date. Late submission of assignments is acceptable if no more than three *calendar* days (72 hours) past the due date, but the grade will be discounted 10% per calendar day (24 hours).

Grading Policy

- Programming assignments: 20%
- Written assignments: 20%
- Quizzes (3 - 5): 9%
- Course evaluation: 1%
- Midterm exam: 20%
- Final exam: 30%

Tentative Schedule

Week 1: Course overview, basics of memory and OO programming
Week 2: Basics of recursion, algorithm analysis; Written assignment #1
Week 3: Linked list; Programming assignment #1
Week 4: Stacks and queues; basics of trees
Week 5: Binary (search) trees; balanced trees Written assignment #2
Week 6: Huffman encoding; heaps/priority queue Programming assignment #2
Week 7: Heaps/priority queues
Week 8: Midterm exam; exam review, basics of Hashing
Week 9: Hashing; Written assignment #3
Week 10: Basics of sorting, various sorting methods; Programming assignment #3
Week 11: Various sorting methods
Week 12: Basics of graphs; traversal; Programming assignment #4
Week 13: Shortest-path; Dijkstra's algorithm.
Week 14: Minimum spanning tree: Prim's algorithm; Written assignment #4
Week 15: Network flow; state-space search
Week 16: Final exam