ECE 331
Data Structures and Algorithms

Overview

Course Catalog Description: Analysis of algorithms using recurrence relations. Algorithmic design techniques. Data structures for searching, priority queues, balanced trees, disjoint sets, and graphs. Memory management. Design and implementation of algorithms/data structures using object-oriented programming languages.
Prerequisites: C- or better in ECE 231, ECE 340 (Co-req.) and Math 327
Textbook: G.L. Heileman, *Data Structures, Algorithms, and Object-Oriented Programming*, McGraw-Hill, 1996.
Class Goals: To build on knowledge students have gained from previous programming and mathematics courses so that they may learn how to apply more sophisticated techniques to the design and analysis of data structures and algorithms. This will allow students to understand and/or conduct assessments of the data structures and algorithms used in modern computer operating systems, application programs, etc. To provide an understanding of the fundamental concepts in object-oriented programming, and how they are implemented in the C++ programming language. To gain experience implementing data structures and algorithms, and using these to solve practical engineering problems.
Course Coordinator: Prof. Gregory L. Heileman

Table I: Objectives, Implementation, and Assessment

Objectives		Implementation	Assessment	A	B	C	D	E	F	G	H	I	J	K
O_1	Understand basic analytical tools necessary for rigorous study of data structures and algorithms	9 hrs. lecture in 1st three weeks	HW 1-2, Exam I	X				X						
O_2	Understand particular algorithmic design techniques	2.5 hrs. lecture in 4th week	HW 3, Exam I	X		X								
O_3	Understand fundamental object-oriented (OO) design and programming concepts	6.25 hrs. lecture in weeks 6-9	HW 4 Exam II	X									X	X
O_4	Understand how OO concepts are implemented in modern programming languages	3.75 hrs. lecture in weeks 9-10	HW 5, Proj. 1,2,3 Exam II	X		X						X	X	X
O_5	Understand trade-offs associated with implementations of various Dynamic Set ADTs	11.25 hrs. lecture in weeks 12-16	Proj. 1,2,3 Exam III	X				X						X
O_6	Gain experience implementing data structures and algorithms, and using these to solve practical engineering problems	Portions of lectures in weeks 9-16, programming project assignments	Proj. 1-3	X	X	X	X	X		X				X

Table II: Expectation and Assessment Outcome
Fall 2008, Dr. G. L. Heileman

General expectations:
Homeworks:
These assignments involve two components:

1. sharpen/practice/improve skills and understanding of discrete mathematics,
2. simple programming exercise aimed at preparing students for programming projects (e.g., making sure they are able to compile programs, produce output to computer screen, etc.).

For (1), expect that all of the students will be able to answer at least 70% of the problems assigned. For (2), expect 100% of the students to be able to complete the assignments, otherwise then cannot have success in programming projects.

Programming Projects:
Project 1 – Involves a comparison of C and C++ implementations of some non-trivial ADT
Project 2 – More advanced data structure implementation, tied to some application
Project 3 – Experimental analysis of data structures

Exams: Expect 75% of the students to score 70% or better on all exams.

Objectives		Outcome Assessment	Evaluation
O_1	Understand basic analytical tools necessary for rigorous study of data structures and algorithms	87.5 % of students completed 100% of exercise in HW 1 & 2. Av g. on Exam I was 57.3%, with 53% scoring 70% or better	Some students lacked some prerequisite knowledge in discrete mathematics (Math 327).
O_2	Understand particular algorithmic design techniques	73% of the students completed HW 3 satisfactorily. Av g. on Exam I was 57.3%, with 53% scoring 70% or better	
O_3	Understand fundamental object-oriented (OO) design and programming concepts	100% of the students completed HW 4 satisfactorily. Av g. on Exam II was 57.6%, with 75% scoring 70% or better	Students lacked some prerequisite knowledge from the previous programming course in this sequence (ECE 231).
O_4	Understand how OO concepts are implemented in modern programming languages	93% of the students completed HW 5 satisfactorily. Av g. on Exam II was 57.6%, with 75% scoring 70% or better Avg of 68% on Project 1, 61% on Project 2, and 47% on Project 3.	Time constraints (end of semester) affected the scores on Project 3.
O_5	Understand trade-offs associated with implementations of various Dynamic Set ADTs	Avg of 68% on Project 1, 61% on Project 2, and 47% on Project 3. Av g. on Exam I was 59.1%, with 83% scoring 70% or better	
O_6	Gain experience implementing data structures and algorithms, and using these to solve practical engineering problems	Avg of 68% on Project 1, 61% on Project 2, and 47% on Project 3.	Project 1 – Discrete Fourier Transform in C Project 2 – Discrete Fourier Transform in C++ w/ applications to filtering Project 3 – Experimental Analysis of Data Structures

Sample Course Schedule

Week	Date	Lect.	Topic	Assignment
1	20 Aug	1	Introduction & Course Administration	
	22 Aug	2	Background–Abstract Data Type, RAM Model	Ch. 1
2	27 Aug	3	Background–Asymptotic Notation, Recursion	Ch. 1
	29 Aug	4	Background–Extended Example: Exponentiation	Ch. 1
3	03 Sep	5	Mathematical Review–Recurrence Relations & Induction	Ap. A
	05 Sep	6	Mathematical Review–Sets, Counting, Graphs	Ap. B
4	10 Sep	7	Algorithmics–Time and space complexity	Ch. 2
	12 Sep	8	Algorithmics–Worst-case and Average-case Analysis	Ch. 2
5	17 Sep	9	Algorithmics–Tractability and Intractability	Ch. 2
	19 Sep	10	Algorithmics–Design Techniques	Ch. 2
6	24 Sep	–	**Exam I**	
	26 Sep	11	Review– Important C Programming Concepts	Ch. 3
7	01 Oct	12	Review– Important C programming Concepts	Ch. 3
	03 Oct	13	Fundamental OOP Concepts	Ch. 4
8	08 Oct	14	Fundamental OOP Concepts	Ch. 4
	10 Oct	–	*Fall Break*	
9	15 Oct	15	Object-oriented Design	Ch. 4
	17 Oct	16	C++ Programming–Classes & Objects	Ch. 4
10	22 Oct	17	C++ Programming–Polymorphism & Inheritance	Ch. 4
	25 Oct	18	Extended Example–Matrix ADT in C and C++	Ch. 4
11	29 Oct	–	**Exam II**	
	31 Oct	19	Review–Basic Data Structures: Lists, Stacks & Queues	Ch. 5,6
12	05 Nov	20	Review–Basic Data Structures: Lists, Stacks & Queues	Ch. 5,6
	07 Nov	21	Self-organizing Lists	Ch. 5
13	12 Nov	22	Binary Search Trees	Ch. 7
	14 Nov	23	Random & Optimal Binary Search Trees	Ch. 7
14	19 Nov	24	Hash Functions	Ch. 8
	21 Nov	25	Analysis of Uniform Hashing	Ch. 8
15	26 Nov	26	Balanced Trees–AVL trees	Ch. 9 (Sect. 11.1, 11.2)
	28 Nov	–	*Thanksgiving Holiday*	
16	03 Dec	27	Priority Queues	Ch. 12 (Sect. 12.1, 12.2)
	05 Dec	–	**Exam III**	