

ECE 231

Intermediate Programming and Engineering Problem Solving

Overview

Course Catalog Description:

Introduction to elementary data structures, program design and computer-based solution of engineering problems. Topics include use of pointers, stacks, queues, linked lists, trees, graphs, systems and device-level programming and software design methodology.

Prerequisites: C or better in ECE 131 or equivalent.

Textbook:

ADTs, Data Structures, and Problem Solving with C++, by Larry Nyhoff, the Second Edition, ISBN: 0131409093, Prentice Hall

Class Goals:

Upon the completion of this course, students are expected to know ADTs and implement ADTs with the major data structures used in computer programming. These include lists, stacks, queues, hash tables, trees, and graphs. Students should clearly see the advantages of the object-oriented programming (OOP) over the procedural programming (old style). Students should also be able to design algorithms based on the data structure to solve application problems; and evaluate the algorithms using the asymptotic notation. The student will have a complete view of the major objectives of object-oriented programming, which are: *reusability*, *inheritance*, and *polymorphism*.

Course Coordinator: Prof. Vincent D. Calhoun

Table I: Objectives, Implementation, and Assessment

Objectives		Implementation	Assessment	A	B	C	D	E	F	G	H	I	J	K
O ₁	Understand the software development phases and approaches. Object-oriented design vs. top-down design. Emphasize the guidelines for developing good source code	2.5 hrs. lecture in 1 st week	Quiz 1 Exam I	X				X						
O ₂	Understand the abstract data type (ADT) and its implementation. Review pointers and pointer operations, and be able to use pointer to represent multi-dimensional static/dynamic array	5 hrs. lecture in week 2-3	Quiz 1 Programming assignment 1 Exam I	X	X	X		X						
O ₃	Understand Object-oriented programming vs. procedural programming; Understand the difference between ADT and its implementation. Understand class and be able to build a class	7.5 hrs. lecture in weeks 4-7	Quiz 2 Programming assignment 2 Exam I	X	X	X		X					X	X
O ₄	Understand and be able to use recursion. Understand the importance of algorithm efficiency and be able to analyze the running time of simple algorithms (eg, sorting algorithms)	6.25 hrs. lecture in week 7-8 and in week 16	Quiz 3 Programming assignment 3 Exam I Final Exam	X	X	X		X						X
O ₅	Understand function overloading, function template, class template and basic components in STL	3.75 hrs lecture in week 9-10	Programming assignment 6 Final Exam	X		X		X						X

**Table II: Expectation and Assessment Outcome
Fall 2008,
Dr. Honggang Yu**

General expectations:

Quizzes

Quizzes are pop-up quizzes. For each quiz, 90% of the students are expected to take it. 80% of the students who take the quiz are expected to score 60% or better.

Programming Assignments:

Six programming assignments are given to help the students understanding the class topics, using the knowledge they learned and gaining programming experience on ADT implementation

Six programming assignments are available online:

http://www.ece.unm.edu/~honggang/ECE231_F08/231_Fall_2008.html

80% of the students are expected to be able to score 80% or better.

Exams:

Expect 80% of the students to score 60% or better on all exams.

	Objectives	Outcome assessment	Evaluation
O ₁	Understand the software development phases and approaches. Object-oriented design vs. top-down design. Emphasize the guidelines for developing good source code	Quiz 1 Exam I 90% students took quiz1. Av g. on Quiz 1 was 45.96%. Avg. on Exam I was 61.26 %. 52.17% of the students score 60% or better on Exam I	Some students lacked basic prerequisite programming knowledge from the previous programming course (ECE131) They have difficulty during the transition at the beginning of the semester
O ₂	Understand the abstract data type (ADT) and its implementation. Review pointers and pointer operations, and be able to use pointer to represent multi-dimensional static/dynamic array	Quiz 1 Programming assignment 1 Exam I 90% students took quiz1. Av g. on Quiz 1 was 45.96%. Avg. on Exam I was 61.26 %. 52.17% of the students score 60% or better on Exam I. Avg. on Programming assignment 1 was 81.59	The first programming assignment provides header file and a couple of coding examples to help the students getting start coding by themselves

O ₃	Understand Object-oriented programming vs. procedural programming; Understand the difference between ADT and its implantation. Understand class and be able to build a class	Quiz 2 Programming assignment 2 Exam I Av g. on Quiz 2 was 58.04%. Avg. on Exam I was 61.26 %. 52.17% of the students score 60% or better on Exam I. Avg. on Programming assignment 2 was 86.25	The second programming assignment provides header file and a couple of member function examples to help the students building a class by themselves
O ₄	Understand and be able to use recursion. Understand the importance of algorithm efficiency and be able to analyze the running time of simple algorithms (eg, sorting algorithms)	Quiz 3 Programming assignment 3 Exam I Final Exam Av g. on Quiz 3 was 71.69%. Avg. on Exam I was 61.26 %. 52.17% of the students score 60% or better on Exam I. Avg. on Programming assignment 3 was 76.75	
O ₅	Understand function overloading, function template, class template and basic components in STL	Programming assignment 6 Final Exam	
O ₆	Understand and gain experience implementing ADTs: list, stack, queue, tree, hash table etc.	Quiz 4, 5 & 6 Programming assignment 4,5 & 6 Final Exam Av g. on Quiz 4 was 61.98%. Av g. on Quiz 5 was 71.60% Avg. on Programming assignment 4 was 82.25	

Course Schedule Fall 2008

Week	Date	Lect.	Topic	Assignment
1	26 Aug	1	Introduction to software development model and approaches	Ch. 1
	28 Aug	2	Compare top-down design and Object-oriented design and Emphasize good coding style	Ch. 1
2	02 Sep	3	Introductions to ADT and its implementation	Ch. 2
	04 Sep	4	Review pointers and pointer operations	Ch. 2
3	09 Sep	5	ADT and data structure, dynamic memory allocation	Ch. 3
	11 Sep	6	Pointers and multi-dimensional arrays	Ch. 3
4	16 Sep	7	OOP programming vs Procedural programming	Ch. 4
	18 Sep	8	Class and class building	Ch. 4
5	23 Sep	9	Class data member, member function, friend function	Ch. 4
	25 Sep	10	Operator overloading	Ch. 4
6	30 Sep	11	Pointers to class objects and pointer operations in class	Ch. 4
	02 Oct	12	Other class operations	Ch. 4
7	07 Oct	13	Recursion	Ch. 10
	09 Oct	14	Recursion and algorithm efficiency	Ch. 10
8	14 Oct	–	Midterm Exam	
	16 Oct	–	<u>Fall Break</u>	
9	21 Oct	15	List ADT and its implementation	Ch. 6
	24 Oct	16	List ADT and its implementation and application	Ch. 6
10	28 Oct	17	Stack ADT and its implementation	Ch. 7
	30 Oct	18	Stack ADT and its implementation and application	Ch. 7
11	04 Nov	19	Queue ADT and its implementation	Ch. 8
	06 Nov	20	Queue ADT and its implementation and application	Ch. 8
12	11 Nov	21	Function overloading, function template	Ch. 9
	13 Nov	22	Class template	Ch. 9
13	18 Nov	23	BST ADT and its implementation	Ch.12
	20 Nov	24	BST ADT and its implementation and application	Ch.12
14	25 Nov	25	Hash table	Ch.12
	27 Nov	–	<u>Thanksgiving Holiday</u>	
15	02 Dec	26	Algorithm analysis, Asymptotic Notation	Ch 10
	04 Dec	27	Sorting algorithms and analysis	Ch 13
16	09 Dec	28	More OOP and ADTs, Inheritance, and Polymorphism	Ch 14
	11 Dec	29	Final Exam	