EECE 330: Data Structures and Algorithms

Catalog description:
This course covers fundamental algorithms and data structures that are used in software applications today. Particular emphasis is given to algorithms for sorting, searching, and indexing. Data structures such as linked lists, binary trees, heaps, B-Trees, and graphs will also be covered along with their associated algorithms. The course also covers basic algorithmic analysis techniques and seeks to promote student programming skills.

Credit hours: 3 credits

Required or elective:
Required for CCE / ECE students

Prerequisites:
By course: EECE230
By topic: Object Oriented Programming

Textbook(s) and/or required materials
Textbook: A Practical Introduction to Data Structures and Algorithm Analysis, by C.A. Shaffer. Additional material will be placed on reserve in the library and posted on Moodle.

References:
None

Course Objectives
1. Teach students how to implement and apply common data structures in practical applications.
2. Allow students to understand how basic algorithms work.
3. Teach students the skills to analyze the efficiency of algorithms.
4. Teach students the costs and benefits of the covered data structures and allow them to understand the difference between static and dynamic allocation of data structures.
5. Promote student programming skills through weekly exercises that require implementation using a high level language.

Course Topics
1. Lists, Stacks, Queues: Operations and Implementation strategies
2. Basic algorithm analysis: upper and lower bounds plus an introduction to the divide and conquer recurrence.
5. Sorting Algorithms: Insertion sort, Bubble Sort, Selection Sort, Quicksort, Heap Sort, Shell Sort, and Merge Sort.

**Course Learning Outcomes**

1. are able to distinguish between the different elementary types of data structures (arrays and linked lists) and explain their unique properties.
2. know how to implement stacks and queues programmatically using elementary data structures.
3. are able to construct binary tree structures and understand how their managing utilities (e.g., tree traversal, insertions, and deletions) work.
4. understand how heaps work and know how to implement them.
5. are able to associate the different types of data structures (arrays, lists, stacks, queues, and binary trees) with real-life applications and are able to find the suitable structure for a given application.
6. are able to implement simple to medium-level applications programmatically using data structures.
7. understand the application of graphs and have developed knowledge of their implementations and understand the different algorithms used for graph traversal.
8. are familiar with the mathematical notations used in algorithmic analysis and know how to use these notations in representing and analyzing basic algorithms.
9. are able to derive through analysis the best, worst, and average performance of basic algorithms.
10. have developed basic knowledge of sorting and are able to identify the distinguishing properties of the many sorting algorithms.
11. are able to develop applications programmatically that utilize sorting against different data types and records (e.g., structures and classes).
12. demonstrate knowledge of hashing tables and function and understand how they are used in speeding up search.
13. have developed working knowledge of Balanced trees (B and B+ Trees) and their variants, and are able to utilize them for indexing records.
14. are able to write medium to advanced software applications that integrate data structures and algorithms to implement desired functionalities.
15. are aware of the importance of Algorithm Analysis, Software Engineering, and Database Management Systems for Software Development.

**Class/laboratory schedule**

a- Three 50-minute lectures per week
b- One 3-hours lab session per week
c- Use of computer lab is needed for working on the assignments.
Resources of the course
Textbook, C++ compiler, and Moodle.

Computer usage
Programming in C++ language.

Evaluation methods
1- Participation (5%)
2- Assignments/Project (15%)
3- Quizzes (45%)
4- Final Exam (35%)

Professional component
Engineering topics: 75%
General education: 0%
Mathematics and basic sciences: 25%

Person(s) who prepared this description and date of preparation
Ali El Hajj, February 2007

Date of last revision
July 2012