EE367L – Lab for Computer Data Structure and Algorithms

Designation: Required

Catalog Description: EE367L – Computer Data Structure and Algorithms Lab (1) (one 3-hr Lab) Laboratory for EE367.
Credits: 1
Pre- and Co-requisites:
Pre-requisite = EE160 (Programming for Engineers) or consent of instructor.
Co-requisite = EE367 (Computer Data Structure and Algorithms).
Class/Lab Schedule: 3 hours of lab session per week

Topics Covered: is synchronized with those of EE367.

All of the fifteen 3-hour lab sessions are planned out at the beginning of the semester, as below:

1. Refresh and Practice with C, together with the use of makefiles and GRADE account. Work on the Game of Life individually, but discuss improvements in teams
2. Implement and use the List data type (= P1 on p.62). Assignment 1 on the Game of Life due for demonstration of operations in the lab
3. Implement and use the Stack and Queue data type. Apply Stack to reverse a list of items. Avoid redundancy (by marking items) when building Stacks or Queues.
4. Experimenting with Recursion and Iteration **by team**. Discuss Conversions and Debate their strengths and weaknesses ***Lab Graded***
5. Re-implement the Queue data type with Doubly Linked Lists; Compare its use with the simple Queue implemented earlier
6. Build a Search Tree (1 of 2 **by team**) and build a Trace program to reveal the Comparison Tree; Count comparisons and get running time to compare 2 methods.
7. Implement Insertion and Selection Sort; then trace the programs and count the number of compare (and swap) operations ***Lab Graded (Individually)***
8. Implement a carefully chosen Shell Sort (with your own shell or shells), then compare the performance of your best shell sort with those from other classmates
9. Apply Divide and Conquer to Implement Recursive Sorts **by team**; Trace Calls to compare their performance ***Lab Graded***
10. Build a Heap using Arrays and Iteration + Apply it to Priority Queue and Heap Sort ***Lab Graded (Individually)***;
11. Build a Hash Table and use it for Memory Efficiency; Debate its values and problems **by team**
12. Build a Binary Search Tree, Traverse it, and use it in Tree Sort and the "information retrieval project" of the Textbook (P6 of Sec. 9.2, pp.412-413)
13. Build a non-binary Tree, Traverse it by Recursion and trace your programs to show results from nested calls ***Lab Graded (Individually)***
14. Implement Graph Traversals (depth-first, breadth-first, …) **by team** and compare their performance ***Lab Graded***
15. **In-Lab Project Examination (35% of Lab Grade);** Usually, 2 mini-projects of programming and testing within 3 hours.

Lab Participation and Team Contribution = 15% of Lab Grade; *Lab Graded* = 10% each (dropping the worst grade), add to **50% of Lab Grade**

Textbook and Other Required Materials:
"Data Structures and Program Design in C", Second Edition, R.L. Kruse, C. L. Tondo, B. P. Leung, Prentice Hall, 1997.

Course Objectives and Relationship to Program Objectives:
The objective of this EE367L lab course is to provide practical opportunities and online, hands-on environment for the students to design and develop programs, debug and test their executions with realistic and challenging data, as well as conduct experiments to get a sense of the time and storage (memory) efficiency of the program codes. At the end of the course, students should be able to have a collection of usable (and possibly sharable) programs and data structures that could make their future programming tasks/projects easier.
[Program Objectives this course addresses: A, B, C, D and E.]

Course Outcomes and Their Relationship to Program Outcomes
- Design, implement and test high-level language programs for and based on fundamental data structures. [1,2,3,5,7,11]
- Understand the relationship between programs and the computer hardware (processors, memory, I/O, etc.) they run on. [1,2,3,5,7,11]
- Understand how data is represented in computers. [1,2,3,5,6,7,11]
- Understand how algorithms and the implemented programs can be applied in a broader context. [1,2,3,5,6,7,8,11]
- Develop an understanding of the value and appreciate the skills of algorithm and program performance analysis. [1,2,3,5,6,7,8,11]
- Execute program testing to verify the actual performances and measurements against theoretical predictions and analyses. [1,2,3,5,6,7,8,11]
- Reach an appreciation of how to use computers and the software (skills) to solve engineering problems that have a broader impact in the information-age society. [1,2,3,4,5,6,7,8,9,10,11]

Contribution of Course to Meeting the Professional Component
Engineering topics: 100%.

Computer Usage: 100% percent of assignments use computers, software tools, the Internet, etc.

Design Credits and Features: EE 361L has 1 design credit. All of the laboratory course involves design. The focus of the entire course is the design and implementation of programs, each of which is an engagement in design. In fact, even the analysis, measurements, and testing components of the course are aimed at reaching a full understanding of the designs for data structures and algorithms.

Instructor(s): D. Yun and T. Dobry

Person(s) Preparing Syllabus and Date: D. Yun, Feb. 21, 2003.