COLLEGE of CHARLESTON

DEPARTMENT OF
COMPUTER SCIENCE

Course Description

Computer Science Department, College of Charleston

Course Number: **CSCI 230**
Course Title: **Data Structures and Algorithms**
Course Coordinator: **Walter Pharr**

Catalog Description

This course develops abstract data types as mathematical models. Data structures and algorithms are developed as the objects and operations of abstract data types. Topics include lists, stacks, queues, trees, graphs, searching, sorting, and analysis of the efficiency of algorithms. Lectures three hours per week. Prerequisites: CSCI221 and MATH207.

Prerequisites by Topic

1. Object-oriented programming
2. Discrete structures

Major Topics Covered in the Course (Required Topics**)**

1. Algorithm analysis (8 hours)
2. Abstract data types (2 hours)
3. Stacks (1 hour)
4. Queues (1 hour)
5. Lists (4 hours)
6. Binary trees (2 hours)
7. Balanced search trees (e.g., AVL, red-black, B-tree) (3 hours)
8. Tree traversals (1 hour)
9. Hashing (2 hours)
10. Heaps (2 hours)
11. Priority Queues (1 hour)
12. Insertion and selection sort (1 hour)
13. Merge, quick, heapsort (2 hours)
14. Quicksort efficiency (1 hour)
15. Lower bound for sorting by comparison (1 hour)
16. Graphs (2 hours)
17. Shortest path (2 hours)
18. Graph traversals (1 hour)
19. Minimum spanning trees (2 hours)
20. Testing (3 hours)

Course Narrative (optional)

Laboratory projects

1. Array-based list project (1 week)
2. Array-based stack or queue project (1 week)
3. Linked list project (1 week)
4. Linked stack or queue project (1 week)

5. Binary search tree project (2 weeks)
6. Balanced search tree project (2 weeks)
7. Hashing project (1 week)
8. Graph project (2 weeks)
9.

Course Outcomes

Upon successful completion of the course, students will be able to:

Course Outcomes	Program Outcome Linkage
1. Understand the analysis of algorithms using Big-O notation.	abj
2. Understand the Abstract Data Type as a model of data and operations.	abcij
3. Understand the definitions of stack, queue and list.	abc
4. Develop insertion, retrieval and deletion algorithms for array and linked implementations of stack, queue and list.	abci
5. Evaluate the algorithms for stack, queue and list for the array and linked implementations using Big-O.	abcj
6. Understand the definitions of binary tree, binary search tree and some balanced search tree.	abc
7. Develop insertion, retrieval and deletion algorithms for binary search tree and some balanced search tree.	abci
8. Evaluate the algorithms for unbalanced and balanced search trees using Big-O.	abcj
9. Understand the definition of hashing.	abc
10. Develop insertion, retrieval and deletion algorithms for hashing.	abci
11. Evaluate the hashing algorithms using Big-O.	abcj
12. Understand the definitions of heap and priority queue.	abc
13. Develop insertion, retrieval and deletion algorithms for heap and priority queue.	abci
14. Evaluate heap and priority queue algorithms using Big-O.	abcj
15. Develop selection, insertion, merge, quick and heap sorting algorithms.	abci
16. Evaluate the sorting algorithms using Big-O.	abcj
17. Understand the definitions of graph, node, edge and other relevant graph terminology.	abc
18. Develop traversal, shortest path and minimal spanning tree algorithms for graphs.	abci
19. Evaluate the traversal, shortest path and minimal spanning tree algorithms using Big-O.	abcj

COLLEGE *of* CHARLESTON

DEPARTMENT OF
COMPUTER SCIENCE

Course Description

Computer Science Department, College of Charleston

Oral and Written Communications
Every student is required to submit at least _1_written reports (not including exams, tests, quizzes, or commented programs) of typically _2_ pages and to make _0_oral presentations of typically _0__ minute's duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

Notes
e.g. special pedagogy, online component, etc.