CPCS-204 Syllabus

Catalog Description

CPCS-204 Data Structures (I)
Credit: 3 (Theory: 3, Lab: 0, Practical: 1)
Prerequisite: CPCS-203
Classification: College Required

The objective of this course is to provide students an understanding of abstract data structures, including, but not limited to, arrays, linked lists, queues, stacks, trees, and graphs. The course also aims to give a conceptual understanding of the trade-offs between various different data structures, hence enabling students to choose an optimal data structure for a particular application. The students will also learn concepts of algorithmic design, recursion, and a variety of searching and sorting algorithms.

Class Schedule

Lab/Tutorial 90 minutes 1 times/week

Meet 50 minutes 3 times/week or 80 minutes 2 times/week

Textbook

John Hubbard, , "Schaum's Outline of Data Structures with Java, 2ed", McGraw Hill Professional; 2 edition (2009-06-10)

ISBN-13 9780071702300 **ISBN-10** 007170230X

Grade Distribution

Week	Assessment	Grade %
2	Homework Assignments 1	6
3	Quiz 1	1.25
5	Homework Assignments 2	6
5	Quiz 2	1.25
7	Exam 1	10
8	Homework Assignments 3	6
9	Quiz 3	1.25
11	Homework Assignments 4	6
11	Quiz 4	1.25
12	Exam 2	15
13	Homework Assignments 5	6
15	Lab Exam	15
16	Comprehensive Final Exam	25

Last Articulated

December 17, 2017

Relationship to Student Outcomes

a	b	c	d	e	f	g	h	i	j	k	l	m	n
x	x								x	x			

Course Learning Outcomes (CLO)

By completion of the course the students should be able to

1. Differentiate between static and dynamic data structures, as well as the tradeoffs between different data structures. (b)
2. Design and analyze more efficient algorithms for solving basic problems. (j)
3. Comprehend and trace the output of a given piece of code or algorithm. (a)
4. Demonstrate linear and binary search techniques in problem solving. (a)
5. Represent and implement linked data structures. (j)
6. **Apply different operations, including search, insertion, and deletion, on linked lists. (a)**
7. **Apply recursion in solving simple problems. (a)**
8. Implement stacks using arrays and linked lists. (j)
9. Implement queues using arrays and linked lists. (j)
10. Describe and represent different tree terminologies. (a)
11. **Apply different operations, including search, insertion, and deletion, on trees and binary search trees. (k)**
12. Comprehend and apply basic sorting algorithms in problem solving. (a)
13. **Evaluate and Analyze the running time of small algorithms in terms of the number of operations performed (b)**
14. **Experiment and practice recursive sorting algorithms in problem solving (b)**
15. Describe, represent, and apply hash table terminologies and operations. (j)
16. Describe, represent, and apply heap (priority queue) terminologies and operations. (b)
17. Develop small-scaled programming assignments in Java, taking space and time efficiency into account. (b)

Coordinator(s)

Dr. Muhammad Umair Ramzan, Associate Professor

Dr. Nadine Akkari, Associate Professor

CPCS-204 Syllabus

Topics Coverage Durations

Topics	Weeks
Introduction	1
Array review. Linear & Binary Search.	1
Linked Lists	2
Recursion	2
Algorithm Analysis	1
Stacks	1
Queues	1
Trees and Binary Search Trees	2
Sorting (Selection, Insertion, and Bubble Sort)	1
Sorting (Merge and Quick Sort)	1
Graphs & Hash Tables	1
Heaps (Priority Queues)	1