

LEARNING OBJECTIVES / GOALS / OUTCOMES / LEARNING OUTCOMES:

By the end of the course, you should be able to

- describe and calculate space and time complexity for commonly used algorithms
- describe array and linked list representations for lists, stacks and queues, implement these in an object-oriented programming language, and calculate the time complexity for the various methods associated with each data structure
- describe strategies for storing arrays and matrices
- describe how tables, skip lists and hash tables can improve performance of searching algorithms
- describe general and binary trees, and how they can be stored and accessed
- describe heaps, and applications to priority queues and sorting
- describe binary search trees, common operations on them, and strategies for maintaining balanced trees
- describe graphs, storage strategies and traversal algorithms
- implement any of the algorithms studied in this course as an object-oriented programming language application

METHODS:

The course will be delivered in lecture-lab format, with numerous demonstrations and hands-on activities. The lab portion gives students and the instructor the ability to view and interact with current projects.

PRIOR LEARNING ASSESSMENT RECOGNITION (PLAR):

Credit can be awarded for this course through PLAR (Please check:) Yes No

METHODS OF OBTAINING PLAR:

Exam, portfolio or interview.

TEXTBOOKS, REFERENCES, MATERIALS:

[Textbook selection varies by instructor. An example of texts for this course might be:]

Dale, Joyce, Weems; Object-Oriented Data Structures Using Java; Jones and Bartlett, 2002

SUPPLIES / MATERIALS:

none

STUDENT EVALUATION:

[An example of student evaluation for this course might be:]

Midterm	25%
Assignments	30%
Final exam	45%

COURSE CONTENT:

[Course content varies by instructor. An example of course content might be:]

Software Engineering	Chapter 1
Data Design & Implementation	Chapter 2
ADTs Unsorted List & Sorted List	Chapter 3
ADTs Stack and Queue	Chapter 4
Linked Structures	Chapter 5
Lists Plus	Chapter 6
Programming with Recursion	Chapter 7

Binary Search Trees	Chapter 8
Priority Queues, Heaps, & Graphs	Chapter 9
Sorting & Searching Algorithms	Chapter 10