COURSE OUTLINE

COS210	Computer Science II - Data Structures
Course Number	Course Title
4	3 Lecture Hours/Week and 2 Lab Hours/Week
Credits	Hours: lecture/laboratory/other (specify)

Catalog Description:

Study of advanced programming topics focused on logical structures of data as well as the design, implementation and analysis of algorithms operating on these structures. Topics include linked lists, stacks, trees, queues, graphs and analysis of efficiency. Also covers searching, sorting and hashing techniques.

Latest Review: Spring 2019

Prerequisites: COS102 Corequisites: None

Required texts/other materials:

Koffman & Wolfgang, Data Structures: Abstraction and Design Using Java, Wiley

Course Coordinator: Meimei Gao

Course Objectives.

1. Develop algorithms for manipulating stacks, queues, linked lists, hash tables, trees, and graphs.
2. Develop the data structures for implementing the above algorithms.
3. Develop recursive algorithms.
4. Familiarize the student with the issues of Time complexity and examine various algorithms from this perspective.

Course-specific General Education goals and objectives.

Critical thinking, problem solving and information literacy: Students will use critical thinking and problem-solving skills in analyzing information gathered through different media and from a variety of sources.

Students will identify a problem and analyze it in terms of its significant parts and the information needed to solve it.
Students will use computers to access, analyze or present information, solve problems, and communicate with others.
Students will formulate and evaluate possible solutions to problems, and select and defend the chosen solutions.
Students will recognize weaknesses in arguments, such as the use of false or disputable premises, suppression of contrary evidence, faulty reasoning, and emotional loading.

EVALUATION

To ensure academic freedom for the individual faculty member the following is only suggested:

Project Assignments:	30%
Tests (2):	30%
Final Exam (covering all the topics):	40%

Unit 1 – Lists and Algorithm Complexity

The student will be able to

1. Use and implement Array Lists and Linked Lists.
2. Describe the algorithms for manipulating singly, doubly, and circular linked lists
3. Describe time-complexity issues - definitions of Big-O.
4. Analyze algorithms to determine their running time and the order of their running time.

Unit 2 – Stacks and Queues

The student will be able to

1. Describe the algorithms for manipulating stacks and queues.
2. Use and Implement stacks and queues.

Unit 3 - Recursion

The student will be able to

1. Unfold the recursive program by coding it non-recursively.
2. Create the stack frames for a recursive program.

Unit 4 - Trees

The student will be able to

1. Describe the algorithms for tree traversals, insertions, deletions.
2. Implement and use Binary Search trees, Heaps/Priority Queues.
3. Represent a priority queue using an array.

Unit 5 – Sets, Maps and Hash Tables

The student will be able to

1. Describe the algorithms for Sets and Maps.
2. Implement and use Sets, Maps and Hash Tables.

Unit 6 – Sorting Algorithms

The student will be able to

1. Implement and compare different sorting algorithms.
2. Analyze several sorting algorithms to determine their running time and the order of their running time.

Unit 7 – Balanced Trees and Graphs

The student will be able to

1. Compare balanced trees to un-balanced trees.
2. Describe ALV trees and Red-Black trees.
3. Understand graph terminology and algorithms.
4. Represent a graph as adjacency matrix, adjacency list.