COURSE OUTLINE

Course Number	Course Title	Credits
COS 210	**Computer Science 2 – Data Structures**	**4**
Hours: Lecture/Lab/Other	**Co- or Pre-requisite**	**Implementation Semester & Year**
3 lecture / 2 lab	**Pre-requisite: COS 102**	**Spring 2022**

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.

General Education Category:
Not GenEd

Course coordinator:

Meimei Gao, 609-570-3483, gaom@mccc.edu

Required texts & Other materials:
Koffman & Wolfgang, Data Structures: Abstraction and Design Using Java, 2nd Edition, ISBN: 978-0-470-12870-1
Java Development Kit (JDK): free download
JGrasp Integrated Development Environment: free download
Eclipse Integrated Development Environment: free download

Course Student Learning Outcomes (SLO):

Upon successful completion of this course the student will be able to:

1. Develop algorithms for manipulating stacks, queues, linked lists, hash tables, trees, and graphs. [Supports ILG # 2, 4, 11; PLO #1, 2, 3]
2. Develop the data structures for implementing the above algorithms. [Supports ILG # 2, 4, 11; PLO #1, 2, 3]
3. Develop recursive algorithms. [Supports ILG # 2, 4, 11; PLO #1, 2, 3]
4. Explain the issues of time complexity and examine various algorithms from this perspective. [Supports ILG # 2, 4, 11; PLO #1, 2, 3]

Course-specific Institutional Learning Goals (ILG):

Institutional Learning Goal 2. Mathematics. Students will use appropriate mathematical and statistical concepts and operations to interpret data and to solve problems.

Institutional Learning Goal 4. Technology. Students will use computer systems or other appropriate forms of technology to achieve educational and personal goals.
Institutional Learning Goal 11. Critical Thinking: Students will use critical thinking skills understand, analyze, or apply information or solve problems.

Program Learning Outcomes for Computer Science AS (PLO)

1. Apply the fundamental concepts and techniques of computation, algorithms, and software design to a specific problem in a variety of applied fields;
2. Provide detailed specifications, analyze the problem, and design a solution that functions as desired, has satisfactory performance, is reliable and maintainable, and meets desired criteria;
3. Apply a firm understanding in areas of mathematics and science.

Units of study in detail – Unit Student Learning Outcomes:

Unit I **Lists and Algorithm Complexity [**Supports Course SLO #1, 2, 4]

Learning Objectives

The student will be able to:

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

Unit II **Stacks and Queues [**Supports Course SLO #1, 2]

Learning Objectives

The student will be able to:

- Describe the algorithms for manipulating stacks and queues.
- Use and Implement stacks and queues.

Unit III **Recursion [**Supports Course SLO #3]

Learning Objectives

The student will be able to:

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

Unit IV **Trees [**Supports Course SLO #1, 2, 4]

Learning Objectives

The student will be able to:

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

Unit V **Sets, Maps, and Hash Tables [**Supports Course SLO #1, 2, 4]

Learning Objectives

The student will be able to:

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

Unit VI **Sorting Algorithms [**Supports Course SLO #3, 4]

Learning Objectives

The student will be able to:

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

Unit VII **Balanced Trees and Graphs [**Supports Course SLO #1, 4]

Learning Objectives

The student will be able to:

- Compare balanced trees to un-balanced trees.
- Describe ALV trees and Red-Black trees.
- Explain graph terminology and algorithms.
- Represent a graph as adjacency matrix, adjacency list.

Evaluation of student learning:

Specific methods for evaluating student progress through the course is up to the discretion of the instructor. Below is an example:

Project Assignments = 30% of the grade
Tests = 30% of the grade
Final Exam = 40% of the grade