SUNY OLD WESTBURY

DEPARTMENT OF MATHEMATICS AND COMPUTER & INFORMATION SCIENCE

SYLLABUS

DATA STRUCTURES AND ALGORITHMS CS3810

Prerequisite: A grade of C or higher in **CS 2511** and **MA 3030**

COURSE DESCRIPTION: Introduce abstract data structures and their implementations, including lists, stacks, queues, trees, hash tables, heaps, and graphs. Application of these data structures in solving real problems through algorithmic design. Analyze and design various sorting and searching algorithms using different data structures. Introduce advanced algorithmic techniques such as recursion and dynamic programming. Analyze runtime and space complexity of algorithms. Use advanced level Object-Oriented Java programming for assignments.

COURSE OBJECTIVES:

Understand the features of each different data structure and

- Know how to select a proper data structure to solve a specific application problem
- Be able to evaluate performances of operations of different data structures when applying algorithms to problems
- Be able to abstract the operations of data structures by pseudo-language
- Use OO programming language to represent data structures and writing algorithms

PROGRAM-LEVEL STUDENT LEARNING OUTCOMES (PSLO):

1. Problem Solving through iterative design and algorithmic thinking including

   B. Expressing solutions computationally by selecting proper data structure.

   C. Analysis and Evaluation of Solution Execution including memory and runtime efficiency.

TEXTBOOK: **ZyBooks: Data Structures and Algorithms** (an interactive online textbook)

Subscribe online or buy a subscriber code through the SUNY Old Westbury bookstore using your financial aid or book credits

1. Sign in or create an account at learn.zybooks.com
2. Enter ZyBooks code given by your instructor
3. Subscribe

(you should be able to get chapter 1 without subscribing but after that, you have to subscribe)

In addition, lecture slides and supplementary materials will be made available on Brightspace.

PROGRAMMING LANGUAGE: Java - version 1.8 or above of the Java SDK.

TOPICS TO BE COVERED*

1. **Chapter 1 Introduction to Data Structures and Algorithms**
   1.1 Data Structures
   1.2 Introduction to Algorithms
   1.3 Relation between Data Structures and Algorithms
   1.4 Abstract Data Types (ADT)
   1.5 Applications of ADTs
   1.6 Algorithm Efficiency

2. **Chapter 2 Searching and Algorithm Analysis**
   2.1 Searching and Algorithms
   2.2 Binary Search
   2.3 Java: Linear and Binary Search
   2.4 Constant Time Operations
   2.5 Growth of Functions and Complexity
   2.6 Big O Notation
   2.7 Algorithm Analysis
   2.8 Recursive Definitions
   2.9 Recursive Algorithms
   2.10 Analyzing the Time Complexity of Recursive Algorithms

3. **Chapter 3 Sorting Algorithms**
   3.1 Sorting: Introduction
   3.2 Selection Sort
   3.3 Java: Selection Sort
   3.4 Bubble Sort
   3.5 Insertion Sort
   3.6 Java: Insertion Sort
   3.7 Merge Sort
   3.8 Java: Merge Sort
   3.9 Quicksort
   3.10 Java: Quicksort
   3.11Radix Sort
   3.12 Java: Radix Sort
   3.13 Bucket Sort
   3.14 Overview of Fast Sorting Algorithms
   3.15 Java: Sorting with Different Operators

4. **Chapter 4 Lists, Stacks, and Queues**
   4.1 List abstract data type (ADT)
   4.2 Singly-linked lists
   4.3 Singly-linked lists: Insert
   4.4 Singly-linked lists: Remove
   4.5 Linked list search
   4.6 Java: Singly-linked lists

7.4 AVL removals
7.5 Java: AVL Trees
7.6 Red-black tree: A balanced tree
7.7 Red-black tree: Rotations
7.8 Red-black tree: Insertion
7.9 Red-black tree: Removal
7.10 Java: Red-black trees

8. **Chapter 8 Heaps**
8.1 Heaps
8.2 Heaps using arrays
8.3 Java: Heaps
8.4 Heap sort
8.5 Java: Heap sort
8.6 Priority queue abstract data type (ADT)

9. **Chapter 9 Algorithms**
9.1 Dynamic programming
9.2 Java: Dynamic programming

10. **Chapter 10 Graphs**
10.1 Graphs: Introduction
10.2 Applications of graphs
10.3 Graph representations: Adjacency lists
10.4 Graph representations: Adjacency matrices
10.5 Graphs: Breadth-first search
10.6 Graphs: Depth-first search

* The topics may vary slightly and need to be adjusted as we move through the semester.