

Data Structures and Algorithms

1. Introduction

- Understanding the Environment
- Programming Practices
- Declaring and Initializing Variables
- Arithmetic and Math Library Functions
- Decision Constructs
- Repetition Constructs
- Functions
- Variable Scope
- Recursion
- Objects and Object-Oriented Programming

2. Arrays

- Arrays Defined
 - Using Arrays
 - Creating Arrays
 - Accessing and Writing Array Elements
 - Creating Arrays from Strings
 - Aggregate Array Operations
 - Accessor Functions
 - Searching for a Value
 - String Representations of Arrays
 - Creating New Arrays from Existing Arrays
 - Mutator Functions
 - Adding Elements to an Array
 - Removing Elements from an Array
 - Adding and Removing Elements from the Middle of an Array
 - Putting Array Elements in Order
 - Iterator Functions
 - Non-Array-Generating Iterator Functions
 - Iterator Functions That Return a New Array
 - Two-Dimensional and Multidimensional Arrays
 - Creating Two-Dimensional Arrays
 - Processing Two-Dimensional Array Elements
 - Jagged Arrays
 - Arrays of Objects
 - Arrays in Objects

3. Lists

- A List ADT
- A List Class Implementation
 - Append: Adding an Element to a List
 - Remove: Removing an Element from a List
 - Find: Finding an Element in a List
 - Length: Determining the Number of Elements in a List
 - toString: Retrieving a List's Elements
 - Insert: Inserting an Element into a List
 - Clear: Removing All Elements from a List
 - Contains: Determining if a Given Value Is in a List
 - Traversing a List
- Iterating Through a List
- A List-Based Application

4. Stacks.....	
	Stack Operations
	A Stack Implementation
	Using the Stack Class
	Multiple Base Conversions
	Palindromes
	Demonstrating Recursion
5. Queues.....	
	Queue Operations
	An Array-Based Queue Class Implementation
	Using the Queue Class: Assigning Partners at a Square Dance
	Sorting Data with Queues
	Priority Queues
6. Linked Lists.....	
	Shortcomings of Arrays
	Linked Lists Defined
	An Object-Based Linked List Design
	The Node Class
	The Linked List Class
	Inserting New Nodes
	Removing Nodes from a Linked List
	Doubly Linked Lists
	Circularly Linked Lists
	Other Linked List Functions
7. Dictionaries.....	
	The Dictionary Class
	Auxiliary Functions for the Dictionary Class
	Adding Sorting to the Dictionary Class
8. Hashing.....	
	An Overview of Hashing
	A Hash Table Class
	Choosing a Hash Function
	A Better Hash Function
	Hashing Integer Keys
	Storing and Retrieving Data in a Hash Table
	Handling Collisions
	Separate Chaining
	Linear Probing
9. Sets.....	
	Fundamental Set Definitions, Operations, and Properties
	Set Definitions
	Set Operations
	The Set Class Implementation
10. Binary Trees and Binary Search Trees.....	
	Trees Defined
	Binary Trees and Binary Search Trees
	Building a Binary Search Tree Implementation
	Traversing a Binary Search Tree
	BST Searches
	Searching for the Minimum and Maximum Value
	Searching for a Specific Value
	Removing Nodes from a BST
	Counting Occurrences

11. Graphs and Graph Algorithms.

- Graph Definitions
- Real-World Systems Modeled by Graphs
- The Graph Class
 - Representing Vertices
 - Representing Edges
 - Building a Graph
- Searching a Graph
 - Depth-First Search
 - Breadth-First Search
- Finding the Shortest Path
 - Breadth-First Search Leads to Shortest Paths
 - Determining Paths
- Topological Sorting
 - An Algorithm for Topological Sorting
 - Implementing the Topological Sorting Algorithm

12. Sorting Algorithms.

- An Array Test Bed
 - Generating Random Data
- Basic Sorting Algorithms
 - Bubble Sort
 - Selection Sort
 - Insertion Sort
 - Timing Comparisons of the Basic Sorting Algorithms
- Advanced Sorting Algorithms
 - The Shellsort Algorithm
 - The Mergesort Algorithm
 - The Quicksort Algorithm

13. Searching Algorithms.

- Sequential Search
 - Searching for Minimum and Maximum Values
 - Using Self-Organizing Data
- Binary Search
 - Counting Occurrences
- Searching Textual Data

14. Advanced Algorithms.

- Dynamic Programming
 - A Dynamic Programming Example: Computing Fibonacci Numbers
 - Finding the Longest Common Substring
 - The Knapsack Problem: A Recursive Solution
 - The Knapsack Problem: A Dynamic Programming Solution
- Greedy Algorithms
 - A First Greedy Algorithm Example: The Coin-Changing Problem
 - A Greedy Algorithm Solution to the Knapsack Problem