AMA COMPUTER ACADEMY

DATA STRUCTURES USING C & C++

For More Detail Contact

+91 9078990958

+91 7735962820

amacomputeracademy.com

Course Offered by OKCL
Certification by OSOU

Created by Ama Computer Academy

OCOC Data Structures using C & C++

Objectives of the Course:

1. Equip students with a comprehensive understanding of data structures and algorithms.
2. Enhance programming skills using C and C++ with a focus on efficient data structure implementation.
3. Develop the ability to solve complex computational problems by applying suitable data structures and algorithms.
4. Provide a strong theoretical foundation on data structures and their applications.
5. Enable students to apply data structures and algorithms in real-world scenarios.
6. Teach various algorithm optimization techniques and their practical implications.
7. Familiarize students with different methods of analysing the efficiency and complexity of algorithms.

Outcomes of the Course:

1. Students will become proficient in C and C++ programming languages, specifically in the context of data structures.
2. Ability to design, implement, and analyse algorithms for solving various computational problems.
3. Proficiency in implementing and utilizing various data structures such as arrays, linked lists, stacks, queues, trees, and graphs.
4. Ability to apply theoretical knowledge to practical problems in software development and other computational fields.
5. A solid foundation for pursuing advanced studies and research in computer science and related fields.
6. Comprehensive understanding of various algorithmic approaches, including iterative, recursive, divide-and-conquer, dynamic programming, and greedy algorithms.
7. Development of critical thinking skills to evaluate and choose the best data structures and algorithms for a given problem.
8. Preparation for careers in software engineering, data analysis, and other technology-driven fields where data structures and algorithms play a crucial role.

Outcomes of the Course:

The HKCL Certificate in Data Structures using C & C++ course offers an in-depth exploration of essential data structures and algorithms, preparing students to tackle complex computational problems. This course covers the fundamentals of data structures, including arrays, linked lists, stacks, queues, trees, and graphs, as well as advanced concepts such as AVL trees, heaps, hashing, and algorithmic techniques like

divide and conquer, dynamic programming, and greedy algorithms. By using C and C++ for practical implementation, students will gain proficiency in programming and analytical skills, positioning them for further studies and careers in software engineering and technology-driven fields.

Course Syllabus:

Topic	Sub-Topics
Introduction and Environment Setup	Introduction to Data Structures
Basic Building Blocks	Mathematical Notations
	Contradiction
	Mathematical Induction
	Mathematical Prerequisites
Introduction to Algorithm	Introduction to Algorithm
	Asymptotic Notations
Programming Essentials I - Basics and Decisions	Variables and Datatypes
	Decisions
Programming Essentials - II	Switch Case
	Array
	Functions
	Pointers
	Object Oriented Programming
Programming Essentials III – Structures	Structure
Programming Essentials IV – Miscellaneous	Data Types
	Union
	Miscellaneous Problems and Solutions
Programming Techniques: Iteration	Introduction
	Loop
Programming Techniques: Recursion and Backtracking	Recursion
	Backtracking
	Recursion and Backtracking: Problems
Array	Introduction to Array
	Array Operations
Two Dimensional Array	Introduction
	Two Dimensional Array Operations
	Array of Pointer
String	String Introduction
	Programming String Operations
	Array of String
	Pointers and Strings

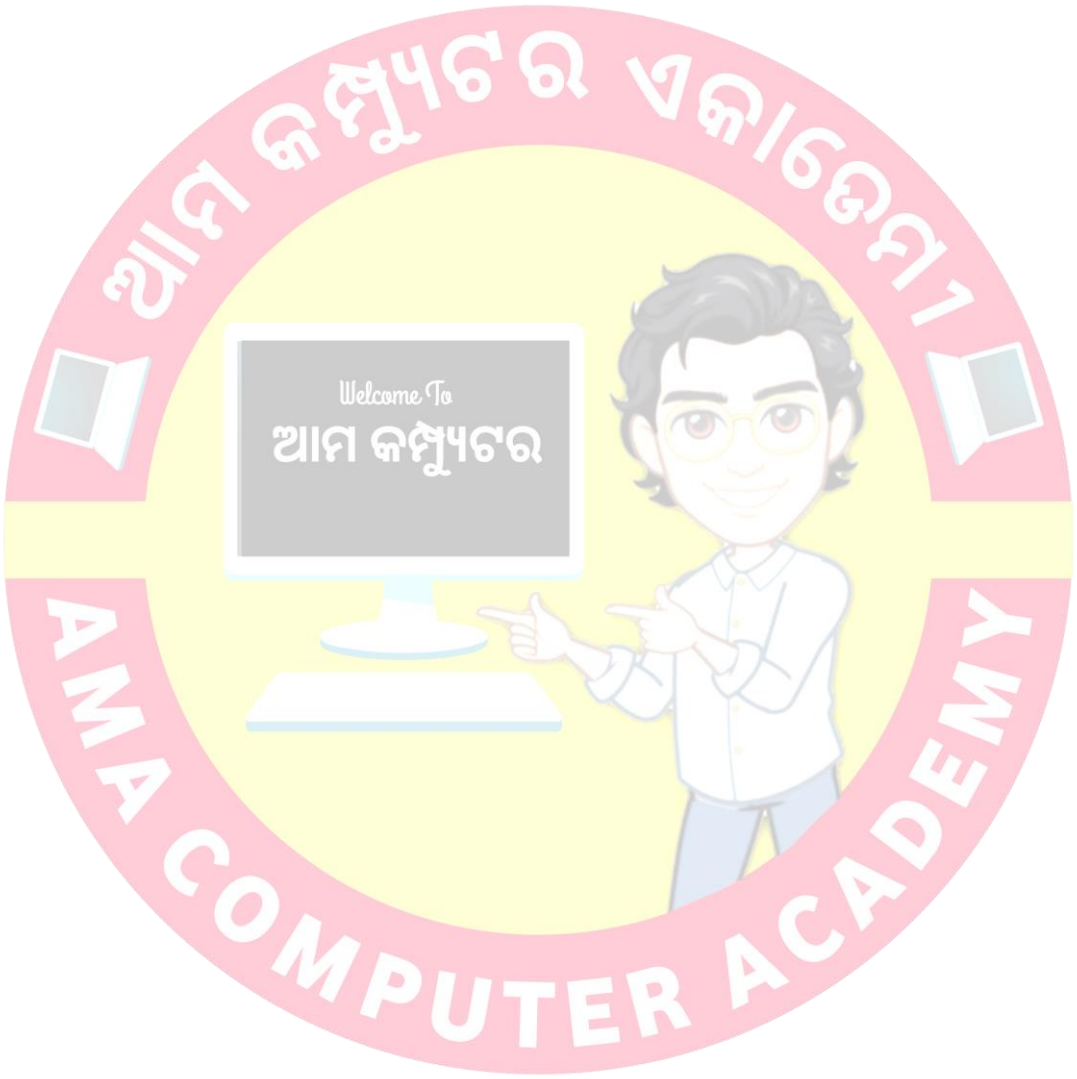
	Pattern Matching
	String Matching
Linked List	Linked List
	Programming Linked List
Circular Linked List	Circular Linked List Introduction
	Programming Circular Linked List
	Doubly Linked List
	Programming Doubly Linked List
More on Linked List	Comparison of Array and Linked List
	Recursive Display of Linked List
	Circular Linked problems and other concepts
Matrices	Matrix Introduction
	Lower Triangular Matrix
	Upper Triangular Matrix
	Symmetric Matrix
	Tri-Diagonal and Tri-band Matrix
	Toeplitz Matrix
	Menu Driven Program for Matrices
Sparse Matrices	Sparse Matrix
	Operations on Sparse Matrices
Stack	Introduction to Stacks
	Stack using Array
	Stack using Linked List
	Parentheses Matching
	Operators
	Infix to Prefix Conversion
	Postfix to Prefix Conversion
	Postfix to Infix Conversation
More on Stack	Infix to postfix conversion
	Evaluation of Postfix Expression
	Stack: Problem and Solution
Queue	Introduction to Queue
	Queue as an array
	Queue as a linked list
	Queue: Problems and Solutions
	Introduction to Double ended Queue
	Programming Deque
	Priority Queue
	Circular Queue
Tree	Introduction to Trees
	Strict and Complete Binary Tree comparison

	Tree: Problems and Solutions
Binary Tree	Binary Tree
	Traversal of a Binary Tree
	Binary Tree: Problems and Solutions
Binary Search Tree	Introduction
	Operation on BST - search and insert
	Deletion from a BST
	Programming: Binary Search Tree
	Generating BST from preorder
	Applications of Binary Tree
	Generating BST from post order
	Binary Search Tree: Problems and Solution
AVL Trees	Introduction
	Operation on AVL tree
	Programming various operations on AVL
	AVL Tree: Problems and Solution
AVL Tree	Balancing AVL tree
	Programming Balancing of AVL
Search Trees	2-3 Trees Introduction
	2-3-4 trees Introduction
	Searching in a 2-3 Tree
	Insertion in a 2-3 Tree
	Deletion from a 2-3 Tree
Search Tree	Introduction to red black tree
	Operations on red black tree
	Operation on red black tree to a 2-3-4 tree
Heap	Priority Queue
	Introduction to Heap
	Insertion in a heap
	Heap: Problems, Solution and Implementation
	Heap as a priority queue and Implementation
	Heap as a priority queue and Implementation
	Replacement of a node in heap
Disjoint Subsets	Introduction to Disjoint subsets
	Fast find implementation
	Fast Union Implementation
Graphs	Definition and terminologies

	Depth First Search (DFS)
	Breadth First Search (BFS)
	Spanning Tree
	Disjoint subsets
Minimal Spanning Tree	Minimum Spanning Tree
	Prim's method
	Kruskal's method
	Programming: Minimal spanning tree
Searching	Introduction
	Linear Search
	Binary Search
	Interpolation Search
Sorting	Introduction to sorting
	Bubble Sort
	Selection Sort
	Insertion Sort
	Shell Sort
	Comparison of Sorting Methods
	Merge Sort
	Quick Sort
Linear Sorting	Counting Sort
	Radix Sort
	Bucket Sort
Symbol Table	Introduction to Symbol table
	Array Implementation of Symbol Table
	Linked List Implementation of Symbol Table
	Other implementation
	Comparison of implementation methods
Hashing	Introduction to Hashing
	Chaining
	Linear Probing
	Double Hashing
	Hash Functions
Single Source Shortest Path	Introduction to Single Shortest Path
	Dijkstra's Algorithm
	Bellman-Ford Algorithm
Greedy Approach	Introduction to Greedy Approach
	Huffman Coding Algorithm
More About Greedy Approach	Huffman Coding Algorithm
	Fractional Knapsack problem

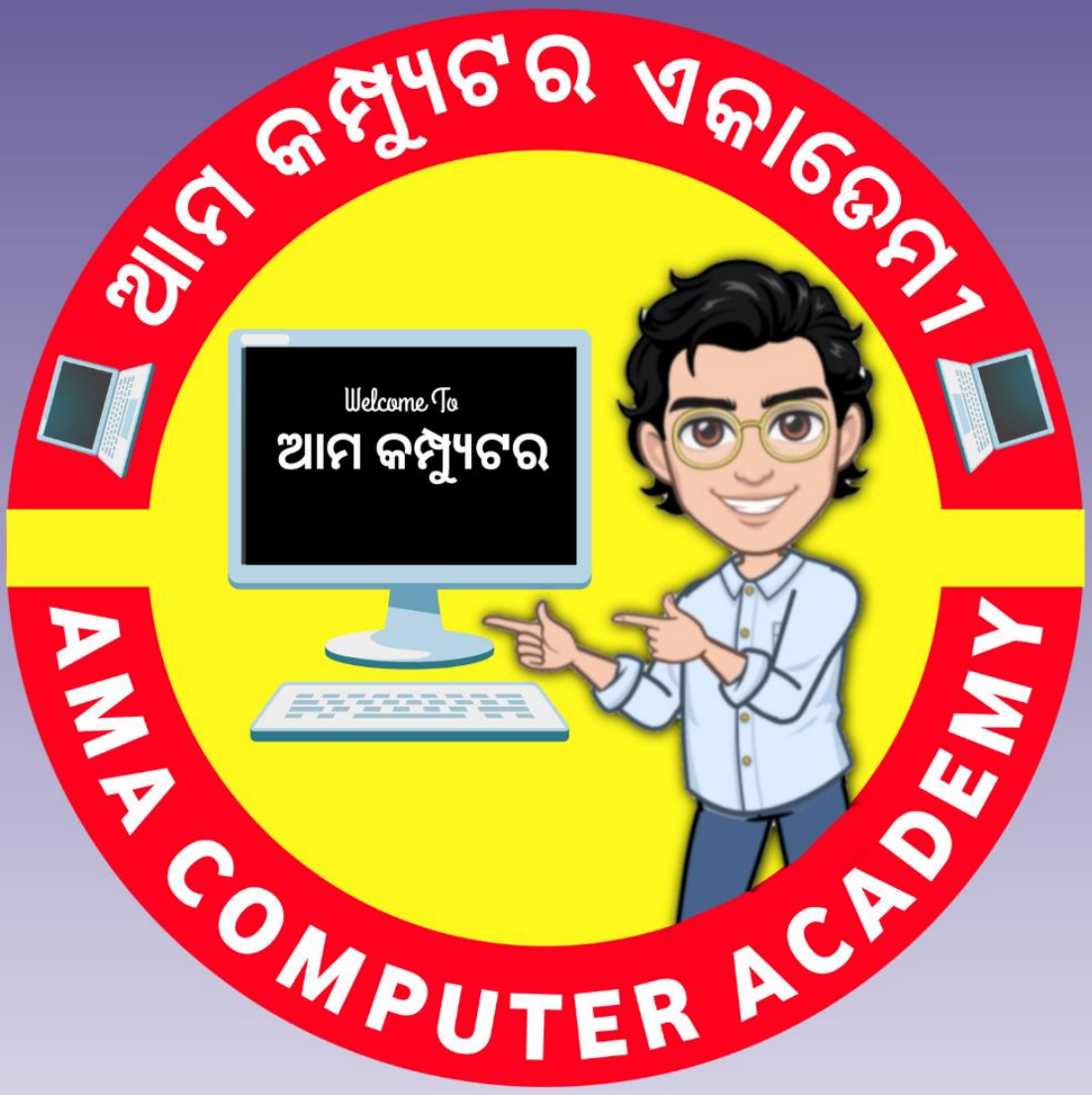
	Dynamic Programming
	Algorithm: 0/1 Knapsack Problem
Dynamic Programming Approach	Matrix Multiplication
	Chain Matrix Multiplication
Divide and Conquer Approach	Divide and Conquer Approach
	Binary Search
	Strassen's Matrix Chain Multiplication
Branch and Bound	Introduction to Branch and Bound Approach
	4-Queens Problem
	Least Cost search
	15-puzzle problem
	FIFO - Branch and Bound
Selection Algorithms	Introduction
	Partition Based Selection
	Linear Search
	Finding K-smallest elements
	Selection Algorithm: Problems and Solutions
More Algorithmic Problems	Introduction to Travelling Salesman Problem
	Introduction to All Pairs Shortest Path Problem
	Job Scheduling Problem
	Coin Change Problem
Computational Complexity	Introduction to NP completeness
	Polynomial Time Reduction algorithms
	NP hard and NP complete Problems
	SAT problem
	NP problems: Examples
Dynamic Memory Management	Compaction of Blocks and storage
	First-Fit
	Best-Fit
	Improvement in the first fit algorithm
	Freeing Storage Blocks
	Dynamic Memory Allocation: Problems and Solutions
Conclusion	Revisiting Programming Concepts

	Revisiting data structures
	Revisiting algorithmic concepts
	miscellaneous problems and solutions



ODISHA'S NO.1 ONLINE LEARNING ACADEMY

TRUSTED BY 2500+ STUDENTS IN JUST 2 YEARS.



FOR ADMISSION CALL TO
+91 9078990958 / 7735962820

Ama Computer Academy - Your Gateway to Govt Jobs & IT Careers.