DSA

Data Structure & Algorithm (DSA)

ABOUT US

We offer Cyber Security and Information Security training and Certification in Delhi for Cyber Security and Information Technology aspirants. Since Decade, we have been in the Information Technology and Cybersecurity industry. You can learn more about cybersecurity, Techniques, and Tools to choose a better career path.

DESCRIPTION

Learn the most authentic Data Structures & Algorithms (DSA) Training in Delhi with the high-tech teaching faculties and mentors group possessing great expertise in their respective domains, especially Python, Java, and C++ for DSA. In addition, Craw Security is the Best DSA Institute in Delhi NCR providing all essential training and industry-recognized certifications within their course module during interactive live classroom sessions at Saket and Laxmi Nagar institutes.

Duration -
60Hrs

Language -
Hindi & English

Mode -
Online & Offline

BENEFITS

1. Basic to Advanced Courses
2. Interview Cracking and Proposal-Making Sessions
3. Transparent Syllabus
4. Career-Oriented Courses and Certifications
5. International Accreditation

CRAW
A C A D E M Y

SAKET ADDRESS

1st Floor, Plot no. 4, Lane no. 2, Kehar Singh Estate, Westend Marg, Behind Saket Metro Station, Saidulajab New Delhi - 110030

LAXMI NAGAR ADDRESS

R31/ 32, 2nd floor Jandu Tower, Vikas marg, Shakarpur, New Delhi -110092

www.craw.in

+91 951 380 5401

SAMPLE CERTIFICATE

Cyber Security
DIPLOMA CERTIFICATE

CRAWEN-xxxxxxx | xx
Certification Number

This certificate is awarded to

SAMPLE SAMPLE

XX XX XXXX
Date

for successfully completing all the requirements and criteria for said certification through examination administrated by Craw Security.

CYBER RESEARCH AND ANALYSIS WING

Signature
Director

PARTNERS

EC-Council   CompTIA   CISCO   PECB   CertNexus   OFFENSIVE   Red Hat

DATA STRUCTURE & ALGORITHM (DSA) MODULE

Module 1. Introduction to Data Structures & Algorithms
- Analysis of Algorithm
    - i. Order of Growth
    - ii. Asymptotic Notations
    - iii. Big O Notation, Omega Notation, Theta Notation
    - iv. Analysis of common loops
    - v. Space Complexity

Module 2. Mathematics
- Count Digits
- Palindrome Numbers
- Factorial of Numbers
- GCD (Greatest Common Divisor) & LCM
- Check For Prime & Prime Factors
- Computing Power (Exponentiation)

Module 3. Recursion
- Introduction & Applications
- Checking Palindrome (Recursive)
- Sum of Digits
- Tower of Hanoi Problem
- Visualizing Recursion: Understanding the Stack Trace

Module 4. Arrays (Lists in Python)
- a. Basics: Introduction, Advantages, Types, Operations (Insert, Delete, Reverse)
- b. Searching Algorithms:
    - i. Linear Search
    - ii. Data Structures & Algorithms (DSA) using Python
    - iii. Binary Search (Iterative & Recursive)
- c. Sorting Algorithms:
    - i. Bubble Sort
    - ii. Insertion Sort
    - iii. Selection Sort
    - iv. Merge Sort (Divide & Conquer)
    - v. Quick Sort
    - vi. Heap Sort
    - vii. Counting, Radix, and Bucket Sort

Module 5. Hashing
- a. Hash Functions (Working & Examples)
- b. Collision Handling Techniques
- c. Chaining vs. Open Addressing
- d. Double Hashing
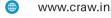- e. Python's dict implementation (under the hood)

DATA STRUCTURE & ALGORITHM (DSA) MODULE

Module 6. Strings
- a. String Data Structure in Python (Immutability)
- b. Pattern Matching Algorithms:
  - i. Naive Pattern Searching
  - ii. Rabin Karp Algorithm
  - iii. KMP (Knuth-Morris-Pratt) Algorithm

Module 7. Linked Lists
- a. Singly Linked List: Traversal, Insertion, Deletion
- b. Doubly Linked List: Forward/Backward Traversal
- c. Circular Linked List

Module 8. Stacks
- a. LIFO Principle (Last In, First Out)
- b. Implementation using Array (List) vs. Linked List
- c. Expression Parsing: Infix, Prefix, and Postfix

Module 9. Queues
- a. FIFO Principle (First In, First Out)
- b. Implementation using Array vs. Linked List
- c. Priority Queue (using Heaps)

Module 10. Trees
- Introduction to Binary Trees
- Tree Traversals:
  - Inorder (Left, Root, Right)
  - Preorder (Root, Left, Right)
  - Postorder (Left, Right, Root)
  - Level Order Traversal (BFS)

- a. Binary Search Tree (BST): Search, Insert, Delete
- b. Self-Balancing Trees:
  - i. AVL Tree
  - ii. Red Black Tree

Module 11. Graphs
- a. Graph Representation (Adjacency Matrix vs. Adjacency List)
- b. Traversals:
  - i. Breadth-First Search (BFS)
  - ii. Depth-First Search (DFS)
- c. Minimum Spanning Tree (MST):
  - i. Prim's Algorithm
  - ii. Kruskal's Algorithm
- d. Shortest Path Algorithms:
  - i. Dijkstra's Algorithm
  - ii. Bellman-Ford Algorithm
  - iii. Floyd Warshall Algorithm

Module 12. Greedy Algorithms
- a. Activity Selection Problem
- b. Fractional Knapsack
- c. Job Sequencing Problem
- d. Huffman Coding