



Interview Preparation Course Schedule

DATE	MODULE	CHAPTER	TOPIC
2020-10-13	Module 1:Python Programming	Introduction	Keywords and identifiers, comments, indentation and statements, Variables and data types in Python, Standard Input and Output, Operators, Control flow: if else, Control flow: while loop
2020-10-14	Module 1:Python Programming	Introduction	Control flow: for loop, Control flow: break and continue,Revision Introduction
2020-10-15	Module 1:Python Programming	Data Structures	Lists, Tuples part 1, Tuples part-2, Sets, Dictionary, Strings
2020-10-16	Module 1:Python Programming	Data Structures	Revision Data Structures
2020-10-17	Module 1:Python Programming	Functions	Introduction, Types of functions, Function arguments, Recursive functions, Lambda functions, Modules, Packages, File Handling
2020-10-18	Module 1:Python Programming	Functions	Exception Handling, Debugging Python, Assignment-1,Revision Functions

2020-10-19	Module 1: Python Programming	Object Oriented Programming	Object oriented Programming -1
2020-10-20	Module 1: Python Programming	Object Oriented Programming	Object oriented Programming -2
2020-10-21	Module 1: Python Programming	Object Oriented Programming	Revision Object Oriented Programming
2020-10-22	Module2: Sorting&Analysis	Solved Problems: Big O, Theta, Omega notation.	Sorting & Searching : why bother with these simple tasks?, Satellite data and key, How it works: Card-sorting, Pseudo code, Insertion sort [Python code], Correctness, In-place sorting
2020-10-23	Module2: Sorting&Analysis	Solved Problems: Big O, Theta, Omega notation.	Stable sort, Online sorting, Model of computation, Space and time analysis of Insertion Sort-1, Space and time analysis of Insertion Sort-2, Insertion sort: Big O- notation, Notations: Big O
2020-10-24	Module2: Sorting&Analysis	Solved Problems: Big O, Theta, Omega notation.	Notations: Big Omega, Theta, Notations: Small O, Omega, Theta, Relationships between various notations, Order of common functions & real world applications., Why does asymptotic analysis matter in the real world?

2020-10-25	Module2: Sorting&Analysis	Solved Problems: Big O, Theta, Omega notation.	Solved Problem: Polynomials, Solved Problem: $n > n_0$ case, Solved Problem:1, Solved Problem2, Solved Problem 3, Solved Problem 4, Solved Problem 5, Solved Problem 6, Solved Problem 7,Revision Solved Problems: Big O, Theta, Omega notation.
2020-10-26	Module2: Sorting&Analysis	Solved Problems of Merge sort	Why learn another sorting algorithm?, How it works: Intuition, Pseudo Code, Merge sort [python code], Merge sort: Analysing time & space complexity, Recursion tree method: intuition, External Merge-Sort
2020-10-27	Module2: Sorting&Analysis	Solved Problems of Merge sort	Solved Problem 1, Solved Problem 2, Solved Problem 3,Revision Solved Problems of Merge sort
2020-10-28	Module2: Sorting&Analysis	Bubble Sort	How it works: intuition + code, Space and time complexity, Bubble sort [Python code], When to use Bubble sort?, Solved Problems 1, Solved Problems 2,Revision Bubble Sort

2020-10-29	Module2: Sorting&Analysis	Quick Sort	Why do we need another sorting algorithm?, How it works: intuition, Partitioning, Quick sort by recursion, Quick sort [Python code], Time complexity: Best and Worst cases., Randomized quicksort and amortized analysis, Solved Problem 1, Solved Problem 2, Solved Problem 3, Solved Problem 4
2020-10-30	Module2: Sorting&Analysis	Quick Sort	Solved Problem 5, Solved Problem 6, Solved Problem 7,Revision Quick Sort
2020-10-31	Module2: Sorting&Analysis	Linear Time Sorting	How it works: intuition + code, Space and time complexity., Selection sort [Python code], When to use Selection sort?, Sample Problem 1, Sample Problem 2, Sample Problem 3, Sample Problem 4, Lower bounds on worst case of comparison sorting, Counting sort: intuition + code
2020-11-01	Module2: Sorting&Analysis	Linear Time Sorting	Space and time complexity, Counting sort [Python code], Radix Sort, Radix Sort [Python code], Where to use which sorting algorithm?, Solved Problem ,Revision Linear Time Sorting

2020-11-02	Module3: Data structures	Arrays	What are Data Structures?, One-dimensional array, Multi-dimensional array, Symmetric matrix, Lower triangular matrix & Diagonal matrix, Tridiagonal matrix, Z-matrix, Toeplitz Matrix, Dynamic arrays & Amortized time
2020-11-03	Module3: Data structures	Arrays	Where do arrays fail?, Solved Problem 1, Solved Problem 2, Solved Problem 3, Solved Problem 4, Solved Problem 5, Solved Problem 6, Solved Problem 7, Solved Problem 8, Solved Problem 9, Revision Arrays
2020-11-04	Module3: Data structures	Linked Lists	Structure & memory organization, Code for node, Insertion, Delete, Traversal & Search, Drawbacks, Solved Problem 1, Solved Problem 2, Introduction to Doubly Linked List, Structure & memory organization, Insert, Delete, Drawbacks, Introduction to Circular Linked lists, Circular Singly and Doubly Linked Lists, Circular Linked List: C-Code, Solved Problem 1, Array vs Linked Lists, Revision Linked Lists

2020-11-05	Module3: Data structures	Stacks	Introduction to Stacks, Stack Operations: Push and Pop, How to implement a stack?, Application: Parenthesis check, Infix, Prefix and Postfix, Infix to Postfix, Infix to Prefix
2020-11-06	Module3: Data structures	Stacks	Evaluation of postfix, Evaluation of prefix, More Applications: Call Stack, Solved Problem 1, Solved Problem 2, Solved Problem 3, Solved Problem 4, Solved Problem 5, Solved Problem 6, Revision Stacks
2020-11-07	Module3: Data structures	Queues	Motivation: Why we need them?, Operations: Enqueue and Dequeue, How to implement them?, Linear and Circular Queue implementations., Solved Problem 1, Solved Problem 2, Solved Problem 3, Solved Problem 4, Solved Problem 5, Solved Problem 6, Solved Problem 7, Solved Problem 8, Solved Problem 9, Revision Queues

2020-11-08	Module3: Data structures	Trees	<p>Binary Search Trees Intuition and Terminology, Implementation using pointers/references, Implementation using arrays, Build binary search tree, Operations: Search, insert, min and max, Traversals: inorder, pre order, post-order</p>
2020-11-09	Module3: Data structures	Trees	<p>Operations: Delete, Randomized BST, Logical structure & implementation, Terminology & Traversals, Types of Binary Trees, Properties of a Tree: Depth, Nodes, Leafs, Application: Backtracking for Sudoku</p>
2020-11-10	Module3: Data structures	Trees	<p>Application: Backtracking for Eight Queens, Applications of Trees: Hierarchical information, Websites (DOM), Postfix to Expression Tree, Evaluating an expression Tree, Sample Questions 1, Sample Questions 2, Sample Questions 3, Sample Questions 4, Sample Questions 5, Sample Questions 6, Sample Questions 7, Sample Questions 8, Heap: What and Why, Heapify</p>

2020-11-11	Module3: Data structures	Trees	Build Heap , Time complexity of build_max heap, Heap sort, Time and space complexity of heap sort, Priority queues: Applications of Heaps, Comparision of all sorting menthods, AVL Trees: What and Why?, Height of an AVL Tree & Searching
2020-11-12	Module3: Data structures	Trees	Balancing Tree using Rotations: Single Rotation LL, RR, Double Rotation: RL rotation, Double Rotation: LR Rotation, Insertion with example, AVL Tree: Delete
2020-11-13	Module3: Data structures	Trees	Solved Problem 1, Solved Problem 2, Solved Problem 3,Revision Trees
2020-11-14	Module3: Data structures	Hashing	Hash-Tables: What and Why?, Direct access table, Hash Functions and collisions, Chaining & load factor, Hash Function: Division method (Modulo Hash function), Multiplication method, Collision Resolution: Open Addressing (Closed Hashing), Collision Resolution: Linear probing, Collision Resolution: Double Hashing

2020-11-15	Module3: Data structures	Hashing	Collision Resolution: Quadratic Probing, Applications: Sparse Matrix representation., Applications: Super Fast Search, Solved Problem 1, Solved Problem 2, Solved Problem 3, Solved Problem 4, Solved Problem 5, Solved Problem 6, Solved Problem 7, Revision Hashing
2020-11-16	Module3: Data structures	Graphs	Graphs: Why, What and Basics, Representation of Graphs: Adjacency Matrix, Representation of Graphs: Adjacency Lists, Connectivity in undirected Graphs, Connectivity in Directed Graphs
2020-11-17	Module3: Data structures	Graphs	Breadth First Search: Intuition and example, BFS: Color coding intuition, BFS: Code and Complexity, BFS: Applications, Depth First Search: Intuition and code, DFS: Analysis
2020-11-18	Module3: Data structures	Graphs	DFS: Edge types, Application of DFS: Detect cycles in a di-graph, Application: Strongly connected components, Application of DFS: Topological Sort, Solved Problems 1, Solved Problems 2, Solved Problems 3, Solved Problems 4, Solved Problems 5, Solved Problems 6

2020-11-19	Module3: Data structures	Graphs	Solved Problems 7, Minimal Spanning Tree: What and Why?, Kruskal's Algorithm, Prim's Algorithm, Properties of MST
2020-11-20	Module3: Data structures	Graphs	Solved Problem 1, Solved Problem 2, Solved Problem 3, Solved Problem 4, Solved Problem 5, Solved Problem 6, Solved Problem 7, Solved Problem 8, Shortest paths: What and Why?, Dijkstra's Algorithm
2020-11-21	Module3: Data structures	Graphs	Shortest Paths for DAGs, Bellman-Ford Algorithm, All Pairs shortest paths: Matrix Operations, Floyd-Warshall Algorithm, Solved Problems 1, Solved Problems 2, Solved Problems 3, Solved Problems 4, Solved Problems 5, Solved Problems 6, Solved Problems 7, Solved Problems 8, Solved Problems 9, Solved Problems 10, Revision Graphs
2020-11-22	Module4: Algorithms	Recursion in programming	Recursion tree method, Master theorem, Extended Master Theorem, Inadmissible cases & Shortcuts, Substitution method., Factorial: Time and Space complexity, Recursion vs iteration

2020-11-23	Module4: Algorithms	Recursion in programming	Tail recursion/Tail-Call Optimization, Revision Recursion in programming
2020-11-24	Module4: Algorithms	Searching	Linear Search: intuition and code, Binary Search: Intuition, Pseudo code, Revision Searching
2020-11-25	Module4: Algorithms	Greedy Algorithms	Greedy Algorithms: Fractional Knapsack, Solved Problem 1, Huffman Coding for Data Compression, Solved Problem 2, Solved Problem 3, Solved Problem 4, Job Sequencing with deadlines
2020-11-26	Module4: Algorithms	Greedy Algorithms	Solved Problem 5, Optimal Merge Pattern, Solved Problem 6, Solved Problem 7, Minimum Spanning Trees: Prim's algorithm, Minimum Spanning Trees: Greedy Kruskal's Algorithm, Greedy Algorithm: Dijkstra's algorithm, Revision Greedy Algorithms
2020-11-27	Module4: Algorithms	Dynamic Programming	Introduction: Fibonacci numbers, Longest Common Subsequence (LCS), LCS: Example, 0/1 Knapsack
2020-11-28	Module4: Algorithms	Dynamic Programming	0/1 Knapsack: example, Matrix Chain Multiplication, Subset- sum problem, Traveling Salesman problem

2020-11-29	Module4: Algorithms	Dynamic Programming	Bellman Ford Algorithm as Dynamic Programming, Floyd Warshall Algorithm as Dynamic Programming, Solved Problem 1, Solved Problem 2, Solved Problem 3, Solved Problem 4,Revision Dynamic Programming
2020-11-30	Module5: Problem Solving	Arrays	Find Missing Number, Find Majority Element in an array, Rotate array , Single Number, How many numbers are smaller than current number [leetcode]
2020-12-01	Module5: Problem Solving	Arrays	Sort array by parity [Leetcode], Create Target Array in the Given order [leetcode], Replace elements with Greatest Element on Right Side [leetcode], Shortest Unsorted Continuous Subarray [Leetcode]
2020-12-02	Module5: Problem Solving	Arrays	Find leaders in an array, Search in sorted rotated array, Sort colors, Inversion in an array, Increasing Triplet sequence
2020-12-03	Module5: Problem Solving	Arrays	Partition equal subset sum, Array product problem, Find two missing in a sequence of consecutive nubmers, Find two repeating elements in an array, Merge Overlapping intervals

2020-12-04	Module5: Problem Solving	Arrays	Rotate Matrix by 90 degrees, 3 Sum [Leetcode], Count Negative numbers in a sorted matrix [Leetcode], The K weakest Rows in a matrix [Leetcode], Median of two sorted arrays
2020-12-05	Module5: Problem Solving	Arrays	First missing positive [Leetcode], Find numbers with even Number of Digits [Leetcode], Game of life [Leetcode], Revision Arrays
2020-12-06	Module5: Problem Solving	Searching and Sorting	Sort 0's, 1's and 2's , Kth smallest/Largest Element in sorted array, Wiggle sort [Leetcode], Find peak Element [Leetcode], Count 1's in sorted binary Array
2020-12-07	Module5: Problem Solving	Searching and Sorting	Revision Searching and Sorting
2020-12-08	Module5: Problem Solving	Linked lists	Find Kth node From end of linked list, Linked list Cycle, Remove nth node from end of linked list, Palindrom linked list
2020-12-09	Module5: Problem Solving	Linked lists	Intersection point of two linked lists, Alternative split of single linked lists, Clone linked list with Random Pointer, XOR Linked list [A memory efficient Doubly linked list]

2020-12-10	Module5: Problem Solving	Linked lists	Add two numbers, Split a circular linked list into two halves, Reverse K alternative nodes in a linked list, Merge two sorted linked lists, Flattening a linked list, Merge sort of linked list
2020-12-11	Module5: Problem Solving	Linked lists	Union and intersection of two linked lists, Find middle element in a linked list, Reverse a linked list, Remove Duplicates from sorted linked list, Odd even linked list
2020-12-12	Module5: Problem Solving	Linked lists	Insertion into a sorted circular linked list, Revision Linked lists
2020-12-13	Module5: Problem Solving	Stack	Design stack such that getMin() should be O(1) time and space, Print next greater element, Design and implement special stack Data structure: push(), pop(), getMin(), Check if parentheses are balanced or not, Stock span problem, The celebrity problem, Reverse a stack using recursion, Implement two stacks in single array, Petrol filling problem
2020-12-14	Module5: Problem Solving	Stack	Implement stack using queues, Largest rectangle in Histogram, Trapping Rain water

2020-12-15	Module5: Problem Solving	Stack	Asteroid collision,Revision Stack
2020-12-16	Module5: Problem Solving	Trees	Count number of nodes in the binary tree, Check if two trees are identical or not, Level order traversal, Level order traversal [Spiral form], Convert binary tree into mirror tree, Print Ancestors of a given Binary tree
2020-12-17	Module5: Problem Solving	Trees	Find lowest common ancestors in Binary Search tree, Print Lowest common ancestors in a binary tree, Child sum property , Count leaf nodes in binary tree, Construct a binary tree from inorder and postorder traversals
2020-12-18	Module5: Problem Solving	Trees	Convert binary tree into sum tree, Find the maximum sum leaf to root path in a Binary Tree, Find Diameter of a Binary Tree, Convert a given Binary Tree to Doubly Linked List, Vertical Traversal of binary tree, Inorder Tree Traversal without recursion and without stack (Threaded binary tree)

2020-12-19	Module5: Problem Solving	Trees	Serialize and deserialize of a binary tree, Boundary Traversal of a Binary Tree, Merge Two binary trees [Leetcode], Range Sum of Binary Search Tree [Leetcode], Search in a Binary Search Tree [Leet code]
2020-12-20	Module5: Problem Solving	Trees	Invert Binary Tree [Leetcode], Given a binary tree, find its maximum depth. [Leetcode], Path Sum [Leetcode], Find the sum of all left leaves in a given binary tree. [Leetcode], Given two binary trees, write a function to check if they are the same or not [Leetcode], All Elements in Two Binary Search Trees [Leetcode]
2020-12-21	Module5: Problem Solving	Trees	Maximum Binary Tree, Binary Tree Pruning: Problem Statement [Leetcode], Validate Binary Search Tree: Problem Statement [Leetcode], Binary Tree Zigzag Level Order Traversal [Leetcode], Populating Next Right Pointers in Each Node: [Leetcode]
2020-12-22	Module5: Problem Solving	Trees	Binary Tree Right Side View [Leetcode], Kth Smallest Element in a BST [Leetcode], Revision Trees

2020-12-23	Module5: Problem Solving	Heaps	K'th Largest/Smallest Element in an array, K'th largest element in a stream, Connect n ropes with minimum cost, Convert min heap to max heap, Finding K-Most frequent words in a text-file, K Closest points to origin [Leetcode]
2020-12-24	Module5: Problem Solving	Heaps	Top K Frequent Elements: [Leetcode], Revision Heaps
2020-12-25	Module5: Problem solving	String Matching	Remove all duplicates from the input string., Reverse words in a given string , Run Length Encoding, Remove all adjacent duplicate characters in a string, First Non-repeating character in a string, Find first non-repeating character in a stream
2020-12-26	Module5: Problem solving	String Matching	Find the smallest window in a string containing all characters of another string, Print all anagrams in a list of words, Rearrange Characters to form a palindrome, Reorder Data In log files, Decode Ways [Leetcode]
2020-12-27	Module5: Problem solving	String Matching	Longest Common Prefix: [Leetcode], Reorganize String: [Leetcode]

2020-12-28	Module5: Problem solving	String Matching	Group Anagrams: [Leetcode], Naive Pattern Matching Algorithm, KMP Algorithm - Part 1, KMP Algorithm, Rabin Karp Algorithm- part 1
2020-12-29	Module5: Problem solving	String Matching	Rabin Karp Algorithm - Part2,Revision String Matching
2020-12-30	Module5: Problem Solving	Dynamic Programming	Knapsack problem, Minimum Swaps for Bracket Balancing, Write a Program Activity Selection Problem , Given an array of jobs with different time intervals. Find the minimum time to finish all jobs.
2020-12-31	Module5: Problem Solving	Dynamic Programming	Minimum Number of Arrows to Burst Balloons, Partition Labels: [Leetcode], Largest sum contiguous sub array, Longest Palindromic sub sequence
2021-01-01	Module5: Problem Solving	Dynamic Programming	climbing stairs problem, nth ugly number, Rod cutting Problem, Coin Change Problem
2021-01-02	Module5: Problem Solving	Dynamic Programming	Minimum Cost path Problem, Fill a N*4 wall with 1*4 bricks problem, Levenstein/ Edit Distance Problem, Egg dropping Problem

2021-01-03	Module5: Problem Solving	Dynamic Programming	Word Break Problem, Longest Increasing Subsequence, Longest Increasing Subsequence ($O(n \log n)$), Subset Sum Problem, Count all possible paths in a Grid
2021-01-04	Module5: Problem Solving	Dynamic Programming	Unique Paths [Leetcode], Unique Binary Search Trees [Leetcode], House Robber: [Leetcode]
2021-01-05	Module5: Problem Solving	Dynamic Programming	Longest Palindromic Substring: [Leetcode], Revision Dynamic Programming
2021-01-06	Module5: Problem Solving	Backtracking	Write a Program for N Queen Problem, Write a Program for Sudoku, Write a Program for Rat in a Maze, Letter Combinations of a phone number: [Leetcode]
2021-01-07	Module5: Problem Solving	Backtracking	Permutations [Leetcode], Word Search: [leetcode], Generate Parenthesis: [Leetcode], Revision Backtracking
2021-01-08	Module5: Problem Solving	Graphs	Write a Program for Check whether a given graph is Bipartite or not, Clone Graph: Problem Statement [Leetcode], Rotting Oranges: Problem Statement [Leetcode], Number of Islands: Problem Statment [Leetcode]

2021-01-09	Module5: Problem Solving	Graphs	Critical Connections in a Network: Problem Statement [Leetcode], Revision Graphs
2021-01-10	Module5: Problem Solving	Bit Manipulation	Single Number 11 [Leetcode], Number of 1 Bits [Leetcode], Counting Bits [Leetcode], Maximum Product of Word Lengths [Leetcode]
2021-01-11	Module5: Problem Solving	Bit Manipulation	Total Hamming distance [Leetcode], pow(x, n) [Leetcode], Revision Bit Manipulation
2021-01-12	Module6: Advanced Data structures and algorithms	Advanced DS	Tries, Redblack Trees
2021-01-13	Module6: Advanced Data structures and algorithms	Advanced DS	B-Tree and B+ Trees
2021-01-14	Module6: Advanced Data structures and algorithms	Advanced DS	Ternary Search Tree, Lru Cache
2021-01-15	Module6: Advanced Data structures and algorithms	Advanced DS	Disjoint sets, Segment Trees
2021-01-16	Module6: Advanced Data structures and algorithms	Advanced DS	Complexity classess, Revision Advanced DS
2021-01-17	Module7: Database	Database	Introduction to Normalisation + 1NF and 2NF +examples, 3NF and BCNF, Solved problems + Properties of Normal Forms

2021-01-18	Module7: Database	Database	Introduction to Databases, Why SQL?, Execution of an SQL statement., IMDB dataset, Installing MySQL, Load IMDB data., USE, DESCRIBE, SHOW TABLES
2021-01-19	Module7: Database	Database	SELECT , LIMIT, OFFSET, ORDER BY, DISTINCT , WHERE, Comparison operators, NULL, Logical Operators, Aggregate Functions: COUNT, MIN, MAX, AVG, SUM
2021-01-20	Module7: Database	Database	GROUP BY, HAVING, Order of keywords., Join and Natural Join, Inner, Left, Right and Outer joins., Sub Queries/Nested Queries/Inner Queries, DML:INSERT
2021-01-21	Module7: Database	Database	DML:UPDATE , DELETE, DDL:CREATE TABLE, DDL:ALTER: ADD, MODIFY, DROP, DDL:DROP TABLE, TRUNCATE, DELETE, Data Control Language: GRANT, REVOKE, Learning resources,Revision Database
2021-01-22	Module8: Operating System	Operating System	OS: what does it do?, John von Neumann Architecture + History of Operating Systems, Modes of CPU execution + FORK ()

2021-01-23	Module8: Operating System	Operating System	What is a Process?, Process States & Queues & Schedulers, IPC+Synchronisation: an introduction
2021-01-24	Module8: Operating System	Operating System	Producer-Consumer Problem: Challenges, Synchronisation Mechanisms -I, Detailed explanation about Disabling interrupt, Peterson Solution & TSL based synchronization
2021-01-25	Module8: Operating System	Operating System	Sleep & Wakeup based synchornization, Semaphores, Producer- Consumer with Semaphores, Deadlocks: an introduction
2021-01-26	Module8: Operating System	Operating System	Deadlock handling Methods, Avoidance: Banker's Algorithm, Multi-Threading: an introduction
2021-01-27	Module8: Operating System	Operating System	Kernel and User level Threads,Revision Operating System

Applied Course Wishes You All The Best

Please mail us to interviewprep@appliedcourse.com if you have any queries