Industry Readiness Program

Training Module

Day 1	• Data Types • Operators (Arithmetic(6), relational(6), logical(5), bitwise(6), address of and value at unary operators, ++ and --, dot operator and [] subscript operator, -> operator) • I/P functions (getc, putc, printf, scanf, gets, puts, fgetc, fputc, fgets, fputs, fprintf, fscanf) stdin, stdout, stderr • ASCII character set, Escape sequences, digits as characters (48 to 57), range of small case(65-90) and upper case characters (97-122) • How numbers are stored in memory. How -ve numbers are stored in memory (using 2s compliment technique), How floating point numbers are stored in memory (IEEE standard technique) • Formatting the O/P using %d, %o, %x, %X, %s (left alignment, right alignment) • Type casting (implicit and explicit, downcast and upcast) • Integer division vs floating point division. • If-else (0 and non-zero as boolean values in C programming)
Day 2	Problems with Conditional Programming: • nested if-else • switch-case • ternary operator • O(n) problem solving using Loops (for, while and do-while examples)

Day 3	• Problem-solving on Loops O(n) and O(n-square) problems • Function Call Stack (IR, PC, FP, SP) • Memory management of a program (code area, static and global data segment, stack area, and heap area) • Different stages of a program (macro expansion also called pre-processing, compilation (lexical analysis and synthetical analysis, translation of human-friendly code to machine code). Object code (.obj or .o file) is machine-specific and the interpreter makes it. Loading and Linking. Generation of .exe or .out (executable file) • ASSESSMENT
Day 4	• Recursion and recursive functions. When to avoid and when to use recursion. • Storage classes (scope, storage, life of a variable) • Pointer variable • Address of and value at operators • Pointer, not a number, and vice versa • Pointer can hold only an address • Pointer can only be of a struct data type • Pointer to a pointer • pointer arithmetic • Need for an Array • Array implementation using pointer and pointer arithmetic • const pointer • wild pointer • null pointer • nullptr (C++11 concept) • dangling pointer & const pointer
Day 5	• Problem-solving on Arrays O(n) • Hacker rank, Leetcode, etc. problem-solving on Arrays O(n) and O(n-square) • pointer to an array • pointer to a constant array • constant pointer to an array • constant pointer to a constant array

Day 6	• What is string • Why store a string in an Array • Need of /0 • 2D Array to store multiple strings • String handling functions and their return types and discuss why the return types and function prototypes use const • Implementation of string handling functions • Implement Circular Queue using Array • ASSESSMENT
Day 7	• Hackerrank and Leetcode Problem solving on strings • Complex problem-solving on Arrays and strings • Need for data binding • structure definition • structure definition has no memory allocation • structure within a structure • structure padding • pointer to a structure • pointer within a structure • pointer (advanced concept) • problem-solving on structures

Day 8	• Implement Stack and Queue using Fixed Arrays
	• Implement Stack and Queue using dynamically created Arrays
	• Implement Stack and Queue using Linked List
	• Implement problems on Linked List:
	  ○ reverse an array
	  ○ sort an array using bubble sort
	  ○ merge 2 sorted arrays into a single sorted array
	  ○ search, insert, delete, and list nodes into a linked list
	  ○ implement all the above using the double-linked list
	• ASSESSMENT

Day 9	• Implement the algorithms, find their efficiencies, discuss their efficiencies, and train the students on the need to optimize the algorithms. When to use and when not use the various searching and sorting techniques • Linear search, Binary search, Bubble sort, Insertion sort, Quick sort, Merge sort, BST, BST traversals, add a node into a BST, delete a node from BST (all the possibilities), search for a node in a BST, Implement self-balancing BST which is the AVL Tree (discuss and implement all 4 rotations)
Day 10	• The first half of the day must be used for covering the LEFT OVER topics. • The second half must be used for REVISION, Recap, and Multiple ASSESSMENTS