|  |  |
| --- | --- |
| **Program Duration:** | **For Second Year Plan**  \* Technical Coding Training - **18 Days - 108 Hrs**  Phase – I – 6 days - Basic C Programming to While Loop Implementation and Problem Solving for All the students.  \*Mid Assessment  Phase – II – 6 Days – Arrays to Structures and Pointers  \*Mid Assessment  Phase – III – 6 Days – Data Structures till Linked List  \*Product Batch - Assessment  (Product Batch Student Split up and Service Company student Split up based on their performance in Phase I Training + Assessment Consideration)  **A green text on a white background  Description automatically generated**Product Batch Students – Advanced Data Structures and Advanced C programming, Problem Solving using Algorithmic Thinking.  Service Batch Students – Problem Solving Skills in Portal.  Daily Practice Assessment in Portal for Both Product and Service Batch Students.  **NOTE: Class will be 3 hrs. Hands-On Session 3 hrs in Dedicated Hackerrank Server.**  **(Portal access will be available till the candidate is passed out)**  **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ONLY LIVE CODING\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*** |
|  | **For Third Year Plan till Placement**  Hands-On Training in Real-Time Problem Solving in **Leetcode** (Based on Slots available).  (OPTIONAL)  \* Technical Coding Training – **6 days –** Basics of Python with OOPs  \* Technical Coding Training – **6 days –** Basics of JAVAwith OOPs |

**SEED - C – Program Syllabus/Schedule**

**Phase I:**

|  |  |
| --- | --- |
| **Day** | **Topics** |
| 1 | Introduction to C- Programming, Number Systems, Basic Input/Output, Operators |
| 2 | Conditional Statements |
| 3 | Looping |
| 4 | For Loop Implementation (Patterns – I) |
| 5 | Advanced Patterns – II |
| 6 | **A green text on a white background  Description automatically generated**While Loop Implementation  (Number Crunching & Number Based Problems) |

**Phase II:**

|  |  |
| --- | --- |
| **Day** | **Topics** |
| 1 | Arrays (SSID – (Searching/Sorting/Insertion/Deletion) |
| 2 | Multi-dimensional Arrays (2d Arrays) |
| 3 | String Concepts |
| 4 | Advanced String Problem solving |
| 5 | Functions & Pointers |
| 6 | Structures and Pointers |

**Phase III:**

|  |  |
| --- | --- |
| **Day** | **Topics** |
| 1 | Introduction to Data Structures  Stack and Queue with Arrays,  Dynamic Memory Allocation |
| 2 | Linked List Introduction |
| 3 | SSID in Linked List with Types of Linked List |
| 4 | Stack and Queue with Linked List |
| 5 | Problems Solving with Linked List |
| 6 | Sorting Algorithms,  Introduction to Non-Linear Data Structures  (Trees and Graphs) |

**NOTE: Only Live Coding will be taught throughout the   
 training.**

**Advanced Phase for Product Batch:**

**Advanced C:**

|  |  |
| --- | --- |
| **Day** | **Topics** |
| 1 | Introduction to Bit Manipulation with  Problems Solving |
| 2 | Introduction to Recursion and Types |
| 3 | Recursion – Problem Solving |
| 4 | Recursion – Taylor Series – Optimization I & II |
| 5 | Fibonacci Series, Tower of Hanoi with Implementation |
| 6 | Matrices – Diagonal, Lower/Upper Triangular, Symmetric, Tri-Diagonal, Square band, Toeplitz Matrices, Addition of Sparse Matrix |

**NOTE: Topics will be added and covered based on no of days.**

**Advanced Phase for Product Batch:**

**Advanced Algorithm:**

|  |  |
| --- | --- |
| **Day** | **Topics** |
| 1 | Introduction to Algorithms and Backtracking  - N Queen Problem  - Knight’s Tour Problem  - Rat in a Maze Problem  - Subset Sum Problem |
| 2 | **A green text on a white background  Description automatically generated**- Hamiltonian Cycle  - Sudoku Solver  - Prime Numbers after Prime P with sum S  - Print all possible paths from top left to  bottom right of a m x n matrix. |
| 3 | Introduction to Graphs  Representation of Undirected Graphs  Representation of Directed Graph  Breadth First Search,Depth First Search |
| 4 | Graph Coloring Problem  – Kruskal’s Minimum Spanning Tree Overview about algorithms  – Prim’s Minimum Spanning Tree  – Boruvka’s Minimum Spanning Tree  – Dijkstra’s Shortest Path Algorithm  – Minimum cost to connect all cities |

|  |  |
| --- | --- |
| 5 | Introduction to Dynamic Programming  – Longest Common Subsequence   * Longest Palindromic Subsequence   – 0/1 Knapsack Problem  – Subset Sum Problem  – Minimum Cost Path  – Coin Change Problem |
| 6 | **A green text on a white background  Description automatically generated**Coin Change Problem  Boyer Moore Algorithm for Pattern Searching  -Bad Character Heuristic  Boyer Moore Algorithm, Good Suffix heuristic  Floyd-Warshall Algorithm  Longest Common Subsequence  Dutch National Flag  Branch & Bound  Branch and bound vs backtracking  Traveling Salesperson problem using branch and bound |

**NOTE: Algorithm Topics allocated each day may vary based on Students level and no of days allocated.**