Phase – 1 | CSE, ISE, AI-ML, AI-DS, ECE

DSA → Java

1st → C
2nd → Python → DSA → C
3rd → Java → ECE ✗
4th → SQL
5th →
6th → Dev Ops + JDBC

C & C++ + Java (Collections)

60 Hours 5 days + 5 days

Introduction + Searching + Sorting + Time Complexity

Linear DS → S, S, L, L
Non Linear DS → Trees, Graphs,
Back Tracking + Greedy + Bit manipulation

* Introduction to Data Structures & Algorithms: → "Unlearn"

Why Data Structures:
↳ Logic Building

Study Table: ① Access
Store manage access
Money ⟷ Time

Arrays: [Pointers]

Address
int a = 10; → 100x0
keyword, identifier, value
&a = 100x0

int * ptr = &a;
printf("%d", *ptr); 100x0
↳ De-referencing

a = 10 → 1000
int * ptr = &a → 1000
xptr → value (10)
variable

&ptr → 2000
pointer to another pointer → Address of a pointer in another var → Double Pointer

Arrays: → 1D, 2D
C, C++, Java → Same data type
A collection of "homogeneous" data.

Python: → List → [ ]
heterogeneous [2, 4.5, "GSS"]
↳ mixed data types

int arr[ ] = { 2, 8, 9, 6, 5 };
indexes: 0 1 2 3 4
positions: 1 2 3 4 5
Zero-based indexing → 0 to n-1

index = position - 1
position = index + 1

Array Pointers: arr = {2, 4, 1$}
arr → pointer → 1000
The name of the array points to the address of the 1st element of the array.

For-each Loop / Enhanced for loop

for (data-type var : arr)
{
statements.
}

* Pointers → Revised
* Arrays → Revised + Array Pointers
* DMA Revised (new & delete)
* Standard Template Library
MST
priority queue
↳ stack
↳ queue
↳ list → forward_list, list
↳ map → ordered, unordered
↳ vector
↳ map with list of strings