



TAKE YOUR NEXT STEP  
WITH

PEPCODING's

'YOU'NIFIED  
LEVEL-UP

BUILD ONTO  
YOUR BASICS



# LEVEL-UP

TOPICS	LECTURES
✓ RECURSION	8
✓ DP + TIME AND SPACE	8
✓ GRAPH + TEXT PROCESSING	8
✓ BASIC DS	12
• Stacks and Queues	
• Linked Lists	
• Generic Trees	
• Binary Trees	
✓ ADVANCED DS	8
• Heap and Priority Queue	
• Segment Tree	
• BST and AVL	
• HashMap	

44  
CLASSES

COURSE FEE:  
₹15000 incl. of taxes

BATCH STARTS:  
21<sup>ST</sup> JUNE  
6 DAYS/WEEK  
(4:30 PM- 8:30 PM)

---

# YOUNIFIED LEVEL-UP

---

## OVERVIEW

The course will involve rigorous practice of questions based on Sorting, Searching, Greedy Algorithms, Text processing, Backtracking, Dynamic Programming along with comprehensive revision of Data Structures like Stacks, Queues, Linked-Lists, Trees, Graphs, Heaps, Range Queries etc. using both JAVA and C++.

It is ideal for all those students who find it difficult to deal with advanced topics and desire to gain expertise in them.

## MODULES

### Recursion & Backtracking

(Lecture 1 - 8)

- Basics and Control Flow in recursion
- Recursion with arrays
- Recursion with ArrayLists
- Recursion on the way up
- Recursion and combinatorics
- Recursion with queens and knights
- Recursion and memory maps
- Advanced recursion – sudoku, crossword, bit masking

**Outcome:** This part lays a strong foundation for trees and graphs. “Peppers” invariably perform better than peers at recursive codes.



## Complexity Analysis

(Lecture 9-10)

- Time Complexity analysis
- Space Complexity
- Optimised Searching, Sorting and Hashing

**Outcome:** This part enables the student in analysing complexity of their codes and optimising them.

## DP & Greedy

(Lecture 11-16)

- Recursion to DP
- Memoization and Tabulation
- 1d DP
- 2d DP
- Classical DP Problems
- DP with Bit Masking
- Greedy vs DP

**Outcome:** This part prepares the student for competitive programming contests by laying a fairly strong foundation in dynamic programming.



## Data Structures

(Lecture 17-28)

Lesson Titles	Topics
<b>Stack and Queues</b>	<ul style="list-style-type: none"><li>➤ Introduction and Implementation of Stack and Queue</li><li>➤ Adapter Design Pattern</li><li>➤ Application of Stack and Queue</li><li>➤ Interview Questions</li></ul>
<b>Linked Lists</b>	<ul style="list-style-type: none"><li>➤ Implementation of Linked List Data Structure</li><li>➤ Application of Linked List</li><li>➤ Iterator Design Pattern</li><li>➤ Interview Questions</li></ul>
<b>Generic Tree</b>	<ul style="list-style-type: none"><li>➤ Introduction of hierarchical data structure</li><li>➤ Implementation of Generic Tree</li><li>➤ Application of Generic Tree</li><li>➤ Interview Questions</li></ul>
<b>Binary Tree</b>	<ul style="list-style-type: none"><li>➤ Implementation of Binary Tree</li><li>➤ Application of Binary Tree</li><li>➤ Interview Questions</li></ul>

Outcome: This part emphasises on subtleties of linear and hierarchical data structures. Linked Lists and Trees feature very prominently in interviews.



## Advanced Data Structure

(Lecture 29-36)

Lesson Title	Topics
<b>BST &amp; AVL</b>	<ul style="list-style-type: none"><li>➤ Balancing Property</li><li>➤ Implementation of BST &amp; AVL</li><li>➤ Application of BST &amp; AVL</li><li>➤ Interview Questions</li></ul>
<b>HashMap</b>	<ul style="list-style-type: none"><li>➤ Applications of HashMap</li><li>➤ Iterable vs Iterator</li><li>➤ Interview Questions</li></ul>
<b>Priority Queue</b>	<ul style="list-style-type: none"><li>➤ Applications of Priority Queue</li><li>➤ Adapter Design Pattern</li><li>➤ Array.sort &amp; Collection.sort</li><li>➤ Comparable vs Comparator</li><li>➤ Interview Questions</li></ul>
<b>Generics</b>	<ul style="list-style-type: none"><li>➤ OOPs – generics, exceptions, interfaces</li><li>➤ Generic Priority Queue creation</li><li>➤ Generics Linked List creation</li><li>➤ Generics HashMap creation</li></ul>
<b>Range Queries</b>	<ul style="list-style-type: none"><li>➤ Segment Tree</li><li>➤ Modular Segment Tree</li><li>➤ Lazy Segment Tree</li><li>➤ Square Root Decomposition</li><li>➤ Sparse Table</li></ul>



Outcome: Besides preparing the student via a lot of interview questions, this part makes the student industry ready by retouching on key OOPs concepts.

## Graphs & Text Processing

(Lecture 37-44)

- Adj. matrix implementation
- DFS and it's Applications
- BFS and its Applications
- Dijkstra, Prims, Kruskals, Bipartite
- DAG implementation, Topological sort
- Bellman ford & Floyd Warshall
- Trie and Huffman Encoder
- Rabin Karp, KMP, Z, Manachers

Outcome: This part adds the "x-factor" in student by doing a lot of practice on graphs and key text processing algorithms.



## “PEP” EFFECT

1. **Videos:** Online videos for important topics to help revise and cover missed classes.
2. **Assignment:** Practice hundreds of high-quality questions given as Hacker rank Assignment.
3. **Doubt Support:** Online TA help in prompt doubt support. **1-12TA to student ratio in classes.**
4. **Revision Option:** We provide our students with unlimited revision option so they can revise as many times they want that to free of cost.
5. **Back-up Class:** We provide catch up classes via doubts teacher in case you miss one.

## PATH AHEAD

This course covers a lot of ground for a starter. Now you are ready to explore more.

- ✓ You may like to compete online for bagging placements and internships. We have a “Interview Prep” course for that purpose.
- ✓ You might like to make real world websites, Web Apps, Mobile Apps, deploying them on cloud or automating your boring task. “DEV-101” will be the perfect pick for your choices.





## LET'S TALK

1. Call us on **011-4019-4461**, or
2. Walk in our centre at **PepCoding, 3rd Floor, 15 Vaishali, Pitampura, Opposite Metro Pillar 347, Above Karur Vysya Bank, New Delhi-110034, Nearest Metro Station - Kohat Enclave** or
3. Our FB page – **[facebook.com/pepcoding](https://www.facebook.com/pepcoding)**
4. Visit [www.pepcoding.com](http://www.pepcoding.com).

