

INFOSYS SPECIALIST PROGRAMMER COMPLETE ROADMAP

Package: ₹9-11 LPA | **Timeline:** 8-9 Months | **Level:** Hard DSA + Competitive Programming

SPECIALIST PROGRAMMER PACKAGE DETAILS

Specialist Programmer: ₹9-11 LPA (Can go up to ₹13 LPA for top performers)

- **Elite role** - Only 5-10% candidates selected
- Requires **strong DSA + CP** background
- Works on **high-impact projects**
- **Fast-track career growth**

⚠ SPECIALIST PROGRAMMER vs OTHER ROLES

Role	Package	DSA Level	Focus
Systems Engineer	₹3.5-4.5L	Easy	Aptitude 70%
Digital Specialist	₹6-7.5L	Medium	Web + DSA 60%
Specialist Programmer	₹9-11L	Hard	DSA 80% + CP

Key Difference:

- Hard DSA (DP, Graphs, Trees, Advanced Algorithms)
- Competitive Programming mindset
- Time/Space Optimization critical
- 3-4 coding problems (1 Medium + 2-3 Hard)
- Less focus on aptitude (only qualifying round)

COMPLETE ROADMAP (8-9 MONTHS)

MONTH 1-2: PROGRAMMING MASTERY + BASIC DSA

Week 1-2: Language Deep Dive

Choose: C++ (Highly Recommended) or Java

C++ Topics:

- STL (vector, set, map, stack, queue, priority_queue)
- Algorithms (sort, binary_search, lower_bound, upper_bound)
- Lambda functions
- Bit manipulation
- Fast I/O techniques
- **Resource:** C++ STL by Luv (YouTube)

Java Alternative:

- Collections Framework
- Comparator, Comparable
- Stream API basics
- **Resource:** Java Programming MOOC

Week 3-4: Basic DSA Foundation

- Arrays & Strings (Advanced)
- Two Pointers & Sliding Window
- Prefix Sum & Kadane's Algorithm
- Basic Recursion
- **Practice:** 50 Easy problems
- **Resource:** Striver's A2Z Sheet

Week 5-8: Core Data Structures

Topics:

1. **LinkedList:** Reversal, Cycle Detection, Merge, Clone
2. **Stacks:** NGE, Stock Span, Histogram Area
3. **Queues:** Deque, Circular Queue, Priority Queue
4. **Hashing:** Frequency Count, Subarray Sum, Longest Substring
5. **Recursion:** All patterns, Backtracking basics

Practice: 100 problems (50 Easy + 50 Medium)

MONTH 3: ADVANCED DATA STRUCTURES

Week 1-2: Trees (Most Important!)

Topics:

- Binary Tree (All traversals)
- Binary Search Tree (Insert, Delete, Search)
- Height, Diameter, LCA
- Views (Top, Bottom, Left, Right)
- Vertical/Diagonal traversal
- Serialize/Deserialize
- Morris Traversal
- Segment Trees (basics)
- Fenwick Tree (Binary Indexed Tree)

Practice: 60-70 tree problems **Resource:** Striver Tree Series, Aditya Verma

Week 3-4: Graphs (Critical!)

Topics:

- Graph Representation
- BFS, DFS (all patterns)
- Cycle Detection (Directed/Undirected)
- Topological Sort (DFS + Kahn's)
- Shortest Path: Dijkstra, Bellman-Ford
- MST: Prim's, Kruskal's
- Union-Find (Disjoint Set)
- Bridges & Articulation Points
- Strongly Connected Components (Kosaraju)
- Bipartite Graph Check

Practice: 50-60 graph problems **Resource:** Striver Graph Series, William Fiset

MONTH 4-5: DYNAMIC PROGRAMMING (DP)

Month 4: DP Fundamentals

Week 1-2: Basic DP Patterns

- Fibonacci, Climbing Stairs
- 0/1 Knapsack (all variations)
- Unbounded Knapsack
- Subset Sum, Target Sum
- Rod Cutting, Coin Change
- LCS (Longest Common Subsequence)
- LIS (Longest Increasing Subsequence)
- Edit Distance
- Matrix Chain Multiplication

Week 3-4: Intermediate DP

- DP on Grids (Unique Paths, Min Path Sum)
- DP on Strings (Palindrome problems)
- DP on Stocks (Buy-Sell problems)
- Partition DP
- Kadane's variants
- 2D DP optimization

Practice: 80-100 DP problems **Resource:** Aditya Verma DP Playlist, Striver DP Series

Month 5: Advanced DP

- DP with Bitmask
- Digit DP
- DP on Trees
- DP with State Compression
- Multi-dimensional DP
- Space Optimization techniques

Practice: 40-50 hard DP problems

MONTH 6: ADVANCED ALGORITHMS

Week 1: Binary Search (Advanced)

- BS on Answer
- BS on 2D Matrix
- Ternary Search
- Exponential Search
- Median of Two Sorted Arrays

Practice: 30-40 problems

Week 2: Greedy Algorithms

- Activity Selection
- Huffman Coding
- Job Sequencing
- Fractional Knapsack
- Interval problems
- Meeting Rooms

Practice: 25-30 problems

Week 3: Bit Manipulation

- XOR tricks
- Subset generation
- Count set bits

- Power of 2 checks
- Divide without operator
- Single Number problems

Practice: 20-25 problems

Week 4: Advanced Topics

- Trie (Prefix Tree)
- Sliding Window Maximum
- Monotonic Stack/Queue
- Binary Lifting
- Sparse Table
- KMP, Rabin-Karp (String matching)

Practice: 30-35 problems

MONTH 7: COMPETITIVE PROGRAMMING

Focus Areas:

- **Codeforces:** Solve Div 2 (A, B, C)
- **CodeChef:** Long Challenge + Starters
- **LeetCode:** Weekly Contests (Top 30%)
- **AtCoder:** Beginner Contests

Practice Strategy:

- Participate in **3-4 contests/week**
- Upsolve **all unsolved problems**
- Learn from **editorials**
- Time yourself: **15-20 mins per problem**

Contest Platforms:

- Codeforces: <https://codeforces.com/>
- CodeChef: <https://www.codechef.com/>
- LeetCode: <https://leetcode.com/contest/>
- AtCoder: <https://atcoder.jp/>

Target Rating:

- Codeforces: 1400+ (Specialist)
 - CodeChef: 4★+
 - LeetCode: 1800+ (Knight)
-

MONTH 8: CS FUNDAMENTALS (Quick Revision)

Week 1: OOP (Deep Understanding)

- 4 Pillars with complex examples
- Design Patterns (Singleton, Factory, Observer, Strategy)
- SOLID Principles
- **Practice:** Design class hierarchies (5-7 problems)
- **Resource:** Head First Design Patterns

Week 2: DBMS

- Advanced SQL (Window Functions, CTEs)
- Query Optimization
- Indexing strategies
- Transaction Isolation Levels
- **Practice:** 100+ SQL queries (Medium-Hard)
- **Resource:** LeetCode SQL Hard, HackerRank

Week 3: OS + Networks

- Process Synchronization (Semaphore, Mutex)
- Deadlock scenarios
- Memory Management (Paging, Segmentation)
- Network protocols deep dive
- **Resource:** Gate Smashers, NPTEL

Week 4: System Design Basics

- Design URL Shortener
- Design Rate Limiter
- Design LRU Cache
- Load Balancing concepts
- Database Sharding
- **Resource:** System Design Primer (GitHub)

MONTH 9: FINAL PREPARATION

Week 1-2: Problem Solving Marathon

- Solve **10-15 problems daily** (Mix of Medium + Hard)
- Revise all **DP patterns**
- Revise **Graph algorithms**
- Tree problems speed practice

Week 3: Mock Tests + Contests

- Take **15-20 mock tests** (Specialist Programmer level)
- **Contest simulation:** 4 problems in 120 mins
- Focus on **speed + accuracy**

Mock Test Platforms:

- Prepinsta Specialist Programmer
- InterviewBit Mock Interviews
- LeetCode Mock Assessments

Week 4: Interview Preparation

- **Technical:** Solve problems on whiteboard
- Explain **approach + complexity** clearly
- **HR:** Prepare answers (Why Specialist Programmer?)
- **Projects:** 1-2 advanced projects ready

PROBLEM PRACTICE BREAKDOWN

Total Problems to Solve: 600-700

Easy: 150 problems (Months 1-2) **Medium:** 300 problems (Months 2-7) **Hard:** 150-200 problems (Months 4-8)

Platform-wise Distribution:

- **LeetCode:** 400+ problems (Primary)
- **Codeforces:** 100+ problems (CP practice)
- **CodeChef:** 50+ problems
- **HackerRank:** 50+ problems (SQL + DSA)
- **GeeksforGeeks:** 50+ problems (Company-specific)

TOPIC-WISE PROBLEM COUNT

Topic	Problems	Difficulty
Arrays & Strings	80	Easy-Hard
LinkedList	40	Easy-Medium
Stack & Queue	40	Medium
Recursion & Backtracking	50	Medium-Hard
Trees	80	Medium-Hard
Graphs	70	Medium-Hard
Dynamic Programming	120	Medium-Hard

Topic	Problems	Difficulty
Binary Search	40	Medium-Hard
Greedy	30	Medium
Bit Manipulation	25	Medium
Trie + Advanced	25	Hard

DAILY SCHEDULE (6-8 HOURS)

Month 1-3 (DSA Building Phase):

- **Morning (3 hours):** Learn new topic + Theory
- **Afternoon (3 hours):** Solve 4-5 problems
- **Evening (1 hour):** Revise previous topics
- **Night (1 hour):** Aptitude basics

Month 4-6 (DP + Advanced Algorithms):

- **Morning (3 hours):** DP/Advanced topics
- **Afternoon (3 hours):** Solve 5-6 hard problems
- **Evening (1.5 hours):** Contest participation
- **Night (30 mins):** CS Fundamentals

Month 7-8 (CP + System Design):

- **Morning (2 hours):** Contest/Upsolving
- **Afternoon (3 hours):** Hard problem practice
- **Evening (2 hours):** System Design/Projects
- **Night (1 hour):** Mock interviews

Month 9 (Final Sprint):

- **Morning (3 hours):** 10-15 problem solving
 - **Afternoon (3 hours):** Mock tests
 - **Evening (1.5 hours):** Interview prep
 - **Night (30 mins):** Revision + Confidence building
-

INFOSYS SPECIALIST PROGRAMMER SELECTION

Round 1: Online Coding Test (150 mins)

Coding Problems: 3-4 Questions

- **Problem 1:** Medium (Array/String) - 30 mins
- **Problem 2:** Hard (DP/Graph) - 45 mins
- **Problem 3:** Hard (Tree/Graph/DP) - 45 mins
- **Problem 4 (Bonus):** Very Hard - 30 mins

Aptitude (Qualifying): 20 questions - 20 mins

- Quick aptitude (Just to filter)
- 60% accuracy needed

Passing Criteria:

- **Must solve:** 2-3 problems completely (all test cases)
 - **Preferred:** 3-4 problems with optimal solutions
 - **Complexity matters:** Brute force won't pass!
-

Round 2: Technical Interview - 1 (60-90 mins)

Coding Round (80% time):

- **3-4 hard problems** on whiteboard/online editor
- Explain approach before coding
- Optimize time & space complexity
- Handle all edge cases
- Clean, modular code

Topics Asked:

- DP (50% chance)
- Graphs (40% chance)
- Trees (40% chance)
- Advanced DS (Trie, Segment Tree)

CS Fundamentals (20% time):

- OOP Design Patterns
 - DBMS Query Optimization
 - OS Scheduling scenarios
-

Round 3: Technical Interview - 2 (45-60 mins)

Focus:

- **System Design:** Design Twitter, Design Cache
- **Project Discussion:** Architecture, scalability
- **Advanced DSA:** Explain complex algorithms

- **Problem Solving:** New hard problem on spot
-

Round 4: HR + Managerial (30-45 mins)

Discussion:

- Why Specialist Programmer? (Not Digital Specialist)
 - Competitive Programming profile
 - Problem-solving approach
 - Salary expectations (₹9-11 LPA)
 - Long-term career goals
 - Relocation flexibility
-

FINAL CHECKLIST

DSA Mastery:

- 600-700 problems solved (LeetCode + CF + CC)
- Codeforces: 1400+ rating (Specialist)
- CodeChef: 4★+ rating
- LeetCode: 1800+ rating
- All DP patterns mastered (100+ problems)
- Graph algorithms crystal clear (70+ problems)
- Tree problems: Can solve in 15-20 mins
- Can solve Hard problems in 45-60 mins

Competitive Programming:

- Participated in 50+ contests
- Div 2 C level problems comfortable
- Fast typing speed (40+ WPM)
- Template ready (C++ STL macros)

CS Fundamentals:

- OOP Design Patterns clear
- 100+ advanced SQL queries
- System Design basics understood
- OS + Networks deep concepts clear

Profile:

- Codeforces/CodeChef profile active
- LeetCode 600+ problems badge
- GitHub: 1-2 advanced projects
- Resume: Highlight CP rating + problem count
- LinkedIn: "Specialist Programmer Aspirant"

Interview Prep:

- 20+ mock coding interviews
- Can explain complexity analysis clearly
- Whiteboard coding comfortable
- 10+ hard problems solved in timed conditions

BEST RESOURCES

DSA Learning:

- **Striver's A2Z Sheet:** <https://takeuforward.org/strivers-a2z-dsa-course/strivers-a2z-dsa-course-sheet-2>
- **NeetCode 150:** <https://neetcode.io/>
- **LeetCode Patterns:** <https://seanprashad.com/leetcode-patterns/>

YouTube Channels:

- **Striver (takeUforward):** Complete DSA + DP
- **Aditya Verma:** DP Playlist (Must watch!)
- **Abdul Bari:** Algorithms
- **William Fiset:** Graph algorithms
- **Errichto:** CP techniques
- **Code N Code:** Advanced topics

Practice Platforms:

- **LeetCode:** <https://leetcode.com/> (Primary)
- **Codeforces:** <https://codeforces.com/> (CP practice)
- **CodeChef:** <https://www.codechef.com/>
- **AtCoder:** <https://atcoder.jp/>
- **CSES Problem Set:** <https://cses.fi/problemset/>

Books (PDFs Available):

- **Introduction to Algorithms (CLRS)**
- **Competitive Programming 3 (Steven Halim)**
- **Cracking the Coding Interview**
- **Elements of Programming Interviews**

Sheets to Follow:

- **Striver's SDE Sheet:** 180+ problems
 - **Fraz's DSA Sheet:** 450+ problems
 - **Love Babbar's 450 DSA:** Comprehensive
 - **Blind 75:** Must-solve problems
-

PRO TIPS FOR SPECIALIST PROGRAMMER

1. Speed is Everything

- Practice **timed problem solving**
- **Template for C++:** Fast I/O, common snippets
- **Target:** Medium in 20 mins, Hard in 45 mins

2. Contest Performance Matters

- Participate in **3-4 contests weekly**
- Upsolve **every unsolved problem**
- Learn from **top-rated coders' solutions**

3. Complexity Analysis is Critical

- Always **explain time/space complexity**
- Know **Big-O for all algorithms**
- **Optimize before coding** in interviews

4. Pattern Recognition

- Identify problem patterns quickly
- **"This is a DP problem"** within 2 mins
- Build **mental map of patterns**

5. Code Quality

- Write **clean, modular code**
- Use **meaningful variable names**
- Add **comments for complex logic**

6. Don't Memorize - Understand

- Understand **why algorithm works**
- Can **modify for variations**
- Explain to others clearly

7. Mock Interviews

- Practice **whiteboard coding**
- Explain **thought process aloud**
- Handle **pressure situations**

8. Profile Building

- Maintain **Codeforces rating**
- LeetCode **contest participation**
- GitHub **quality projects** (not quantity)

SPECIALIST PROGRAMMER vs FAANG PREPARATION

Aspect	Specialist Programmer	FAANG
DSA Difficulty	Hard	Hard-Expert
Problem Count	600-700	800-1000
System Design	Basic	Advanced
Time	8-9 months	12-18 months
CP Rating	1400+ CF	1600+ CF
Focus	Infosys-specific	General tech giants

Good News: Specialist Programmer prep = 70% FAANG ready!

SUCCESS FORMULA

The 80-20 Rule:

- **80% DSA** (Arrays, DP, Graphs, Trees)
- **20% CS Fundamentals** (OOP, DBMS, OS)

Quality > Quantity:

- 600 problems **understood** > 1000 problems **rushed**
- 50 contests **participated** > 200 problems **passively solved**

Consistency > Intensity:

- **6-8 hours daily** for 8 months
- **NOT** 14 hours for 4 months (Burnout!)

Pattern-Based Learning:

- Master **20-25 patterns**
 - Apply to **600+ problems**
 - Faster problem recognition
-

FINAL MOTIVATION

Why Specialist Programmer?

- **₹9-11 LPA** (2.5x Systems Engineer)
- Work on **cutting-edge projects**
- **Fast-track to Tech Lead** (3-4 years)
- **International opportunities**
- Recognition as **elite coder**

The Reality Check:

- **NOT for everyone** - needs dedication
- **Hard work:** 6-8 hours daily for 8 months
- **700+ problems** to solve
- **50+ contests** to participate
- **Multiple rejections** before success

But If You Make It:

- You're in **Top 5-10%** of candidates
 - **Premium package** + **Better projects**
 - Strong foundation for **FAANG/Product companies**
 - **Problem-solving** becomes your superpower
-

THE ULTIMATE MINDSET

"Specialist Programmer is not a job title - it's a BADGE OF HONOR!"

You're not just preparing for Infosys. You're becoming a **PROBLEM SOLVER**.

- After this prep: **Google, Amazon, Microsoft** are within reach
- You'll crack **Codeforces Div 2 problems** easily
- You'll be the **go-to person** for DSA help
- Your **problem-solving mindset** stays forever

The Journey:

- Month 1-2: "DSA is tough!"
- Month 3-4: "I'm getting it!"

- Month 5-6: "DP is beautiful!"
 - Month 7-8: "I love contests!"
 - Month 9: "I'm ready to CRACK IT!"
-

FINAL WORDS

Timeline: 8-9 months of **intense** preparation **Daily Commitment:** 6-8 hours (No compromise!) **Success Rate:** Only **5-10%** make it, but **rewards are HUGE!**

Remember:

- **Specialist Programmer \neq Regular Programmer**
- You're competing for an **ELITE position**
- Your competition is **CodeChef 4★, Codeforces Specialist**
- **Aptitude won't save you** - DSA decides everything!

The Final Push: 700 problems solved 1400+ Codeforces rating DP mastery (100+ problems) Graphs fluency (70+ problems) Contest consistency (50+ contests)

YOU DON'T NEED TO BE A GENIUS. YOU NEED TO BE OBSESSED!

8-9 months of **obsession** = ₹9-11 LPA + **Elite Coder Status**

NOW GO SOLVE PROBLEMS!