Phase – 0

- Python Environment
- Git basics
- Leet code/code studio setup

Daily routine fix

Phase – 1 60 Hrs (2 weeks)

- Variables, Loops, conditions
- Functions
- Lists, strings, dictionaries
- classes (Basic)

Write code (5 - 10 mini programs)

→ Python Fundamentals ⟶

Phase – 2 60 Hrs (2 weeks)

Advanced Python + OOPs

- OOPs (class, object, Inheritance, Polymorphism)
- Encapsulation
- Exception handling
- Clean code

take real world examples (User, Order, Payment)
Explain out loud (interview styles)

Phase – 3 220 hrs (8 - 10 weeks)

DSA Core

★ Data structures –
- Arrays, strings
- Linked lists
- Stack, Queue
- Hash map
- Trees (BST, traversal)
- Graphs (BFS, DFS)
- Heaps

★ Algorithms –
- Sorting
- Binary search
- Recursion
- Backtracking
- Greedy
- DP (Dynamic Programming)

Pattern-based, first brute then optimize
Dry run + Big O (every solution) 2 - 3 Ques/day.

Phase – 4 20 hrs (Parallel with Phase-3)

- Big - O, Big - Theta
- space analysis
- Comparing Solutions

write the complexity below every code

Phase – 5 100 hrs (4 - 5 weeks)

CS core (OS, CN, DBMS)

★ OS –
- Process vs Thread
- Deadlock
- Memory Management

★ CN –
- HTTP/HTTPS
- DNS
- TCP/IP basics

★ DBMS-
- SQL queries (Joins, Subqueries)
- Normalization
- ACID
- Indexing

Short notes and Focus on "why" Interview Q & A style

Phase - 6 60 hrs (3 weeks)

System Design (LLD)

- Class diagrams
- Relationships
- Design Patterns —
  - singleton
  - Factory
  - observer
- Practice —
  - Parking Lot
  - Snake & Ladder
  - Chat System

Pen-paper design and explain Trade-offs

Phase - 7 80 hrs (4 weeks)

Maths for AI foundation → solid

- Linear Algebra (vectors, eigenvalues)
- Probability (Bayes, RVs)
- Calculus (gradients)
- Optimization basics

Lectures + assignments (YT.com) and connect ML with intution...

Phase - 8 90 hrs (4-5 weeks)

Machine Learning

- Regression
- Classification
- Clustering
- Evaluation metrics
- Ensemble methods
- Cross-validation

concepts → code → use-case

(Interview explanation Practice)

(self projects)

Phase-9 70 hrs (3-4 weeks)

Deep Learning

- Neural Networks
- CNN
- RNN / LSTM
- Transformer (Basic Idea)

Intution > heavy Math with Real Exaples

Phase - 10 80 hrs (4-5 weeks)

Gen AI + NLP (Advanced EDGE)

- LLMs
- OpenAI API
- RAG
- Lang Chain / Lang Graph
- CV (Computer vision) (basics)

End to End Project and Prompt with API + retrieval

Phase - 11 50 hrs (2-3 weeks)

Deployment + Full Stack Awareness

- Fast API / Flask
- Rest APIs
- Basic Frontend (HTML, CSS, JS, React)
- Node JS

for awareness and learn the flow of front-end and Back-end

Phase – 12 100 hrs (6-8 weeks)

Projects Selection Makes

Projects — 1) Backend / system project

2) AI / Gen AI flagship project- Like TRINETRA

Architecture, trade-offs and improvement at every steps

Phase – 13 40 hrs (2-3 weeks)

Behavioral + Mock Interviews

Ownership Failure stories Team conflict

Why Amazon STAR Method

write and Read loud out Practice along with Mock interviews

Total study hours ≈ 780 – 850 hrs

Python (basic — advance) : ~ 120 hrs

AI / ML / DL / Gen AI : ~ 280 hrs

CS core + LLD : ~ 160 hrs

DSA + Algo : ~ 220 hrs

Behavioral + Mocks : ~ 40 hrs

With 35-40 hrs / week avg → ~ 24-26 weeks → ~ 6 to 6.5 months

ToDo - 1

Phase - 1

ToDo - 2 (Parallel start)

2A → Phase 2

2B → DSA level 1 • Arrays • Strings • Linked List • Stack Queue

ToDo - 3 (main Parallel Engine) Daily

3A → DSA level 2 & 3 • Trees • Graphs • Heaps • Recursion • DP • Greedy

3B → CS core • OS • CN • DBMS

3C → Maths for AI (IIT Mandi)

• Linear Algebra • Probability • Calculus

ToDo-4 (System thinking + ML)

4A → System Design LLD • Class Digrams • Design Patterns

4B → (Weak) Machine Learning • Regression • classification • Matrics • cross-validation

ToDo - 5 (AI Advanced + Gen AI)

5A → Deep learning • NN • CNN • NLP • RNN / LSTM

5B → Gen AI / RAG, CV • LLMs • open AI API • Lang chain / graph • CV (Basic)

ToDo-6

6A → Deployment + APIs • FastAPI / Flask • Rest (HTML, CSS, JS, React, NODE JS)

↳ Awareness

6B → Projects • Back end / system Project • Gen AI flag ship Project

ToDo - 7

Behavioral + Mock interviews (STAR Methods)

Resume walkthrough (Build up + live codes)

Total → 800 hrs
Duration → 7 Months
Style → (sequential + parallel).

Python → DSA early → DSA runs longest → CS + AI Parallel
→ Projects + Behavioral at end.

A) DSA + Basic OOPS :-

1) Rate Limiter
   Q) Design a class that counts requests in the last 5 minutes)
   Logic : use Queue / HashMap
2) File System Logic
   Q) Calculate total size of files in a directory structure.
   logic : Recursion / Stack (DFS) Uses.
3) Snake & Ladder / Game Logic :
   Q) Find minimum dice ~~throughs~~ throws to reach the end.
   Logic : BFS (Graph)
4) Badge Access System:
   Q) Employee enter room but not ~~exist~~ exit. List invalid Entries..
   logic : Hashmap tracking.

B) OOPS Round (Machine coding) (Basically you have to run the code for given question)
Eg → Q) Design a voting system / Library mangement system.

Expectation : Class Book, Class Member, Class Library make these correctly.
inheritance is important along with Clean code.

+1 yr. exp.

A) Rate limiter, Badge Access but expectation high↑
- Variable Naming : x, y, temp use Kiya to fail. use requestTime, userId ☆
- Edge case : if input is empty code should not crash. must Error handling
- Complexity : Discussion on O(n) Vs O(Logn)

B) LLD
   Q) Design a parking lot or design a Router (API handling)
   must use design patterns (like :- singleton pattern for the database connection, factory pattern for Vehicle types).
   Extensibility : Agar kal ko ev add karna ho to kya poor code change karna hoga? (If yes to rejection).
C) Project Deep Dive
   Eg → why you use this tool/~~lib~~ Why not this one ?
   Explaination on Previous company work / Project.
   did you fix any toughest bug if yes how? (story type Ans:)

The "Atlassian Special" (For Both) ★
Non-negotiable round.
1) tell me about a time you disagreed with your management / Team.
   Ans: (STAR) → Situation, Action, Task, Result). Solid valuable Points
2) Tell me about a time you failed
   Accept your mistakes, learn from mistakes, Process Fix
3) Why ATLASSIAN ?