■ 3-Month GATE CS Study Plan

■ Overview

Month 1: Syllabus Coverage + Basics + Notes
Month 2: Core Subjects + Previous Year Papers
Month 3: Revision + Full Mocks + Weak Topics

■ Subjects to Cover

General Aptitude (15 marks)
Engineering Mathematics (Discrete Math, Linear Algebra, Probability)
Core CS Subjects:
Programming & Data Structures
Algorithms
Computer Organization & Architecture
Digital Logic
Operating Systems
Databases
Theory of Computation
Compiler Design
Computer Networks
Software Engineering / Web Technologies (Low Priority)

■ Month 1: Core Understanding

Week 1: Discrete Mathematics + Set Theory + Graph Theory
Week 2: Programming in C + Data Structures (Stacks, Queues, Trees, Graphs)
Week 3: Algorithms (Sorting, Searching, Greedy, Dynamic Programming)
Week 4: Digital Logic + Computer Organization (Basics, Number Systems, Memory)
Tasks:
- Make handwritten notes.
- Use YouTube/NPTEL: Gate Smashers, Neso Academy.
- Solve topic-wise MCQs (GateOverflow, MadeEasy books).

■ Month 2: Advance Concepts + PYQs

Week 1: OS (Scheduling, Deadlocks, Paging) + DBMS (ER Model, SQL, Normalization)
Week 2: TOC (Automata, CFGs, Turing Machines) + Compiler Design (Lexing, Parsing)
Week 3: Computer Networks (OSI Model, TCP/IP, Routing, IP, UDP, DNS)
Week 4: Linear Algebra + Probability + General Aptitude
Tasks:
- Solve GATE previous year questions (last 10 years).
- Daily solve 20–30 MCQs.
- Weekly revision of Month 1 notes.

■ Month 3: Revision + Mocks + Analysis

Week 1: Revise entire syllabus + Identify weak areas
Week 2: Full-length mock tests (2–3 per week)
Week 3: Analyze mock test results + Revise weak/tricky concepts
Week 4: Final mocks (4–5) + Final PYQ review
Tasks:
- Use GateOverflow, Testbook, or MadeEasy Test Series.
- Focus on accuracy + time management.
- Create 1-page revision sheet per subject.

■ Tips for Success

- Study 6–8 hours/day with a proper schedule.
- Use Pomodoro technique (25 min study, 5 min break).
- Don't ignore General Aptitude (12–14 marks).
- Reserve Saturday for revision and Sunday for mock tests.
- Analyze each mock test deeply — focus on mistakes and timing.