
✅Punishment Rule

If you miss a day:
Punishment Task: Solve 2 extra DSA problems + Write a blog post summarizing yesterday's missed topic.

□ Phase 1: Strong CS Foundations (Days 1–30)

Day	Topic
1	Time & Space Complexity
2	Arrays: Basics + Problems
3	Strings: Basics + Problems
4	Recursion: Concepts
5	Recursion: Problems
6	Searching Algorithms
7	Sorting Algorithms
8	Linked Lists
9	Stacks and Queues
10	Trees

Day	Topic
11	Binary Trees & BSTs
12	Graphs
13	Graph Algorithms (BFS, DFS)
14	Hashing
15	Dynamic Programming Basics
16	OS: Processes, Threads
17	OS: Scheduling Algorithms
18	OS: Memory Management
19	OS: Deadlock + Virtual Memory
20	DBMS: Relational Models
21	DBMS: Normalization & SQL Joins
22	DBMS: Transactions, Indexes
23	DBMS: ER Diagrams + Query Practice
24	CN: OSI Model
25	CN: TCP/IP, Routing, DNS

Day	Topic
26	CN: HTTP, FTP, Networking Commands
27	CN: Protocols + Practical Tools
28	SE: SDLC, Agile, Waterfall
29	SE: UML, Requirement Analysis
30	SE: Design Patterns Overview

□ Phase 2: Competitive + Low-Level Coding (Days 31–60)

Day	Topic
31	C++ Basics: Syntax, Variables
32	C++: Functions, Scope, Recursion
33	C++: Arrays, Vectors
34	C++: Pointers
35	C++: Memory Management
36	C++: OOP Basics
37	C++: Inheritance, Polymorphism

Day	Topic
38	STL Containers
39	STL Algorithms
40	CP: Greedy Algorithms
41	CP: Backtracking
42	CP: Two Pointers, Sliding Window
43	CP: Binary Search on Answer
44	CP: Graph Problems
45	CP: Dynamic Programming
46	CP: Bit Manipulation
47	CP: Segment Trees
48	CP: Fenwick Trees
49	CP: Practice Contest
50	CP: Practice Contest
51	CP: Codeforces A/B Problems
52	CP: Codeforces C Problems

Day	Topic
53	CP: CodeChef Starters
54	CP: Leetcode Biweekly
55	CP: Practice Problems
56	CP: Weekly Contest Review
57	CP: Practice Contest
58	CP: Contest Analysis
59	CP: Solve Missed Problems
60	CP: Summary + Revision

□ Phase 3: Full Stack Development (Days 61–120)

Day	Topic
61	HTML Basics
62	CSS Basics
63	CSS Flexbox & Grid
64	JS Basics: Variables, Loops
65	JS: Functions, Scope

Day	Topic
66	JS: Arrays & Objects
67	JS: DOM Manipulation
68	JS: Events, Forms
69	JS: Async/Await, Fetch API
70	JS: ES6+, Callbacks, Promises
71	React Basics
72	React Hooks (useState, useEffect)
73	React Routing
74	React Forms + Validation
75	React Context API
76	Project: Todo App
77	Project: Blog App
78	Node.js Basics
79	Express.js Basics
80	MongoDB Basics

Day	Topic
81	Mongoose ODM
82	Authentication (JWT, bcrypt)
83	RESTful APIs
84	Project: Auth System
85	Project: Blog Backend
86	CORS, Error Handling
87	File Upload, Multer
88	Email & Notification APIs
89	Pagination & Filtering
90	Fullstack: Connect Front + Back
91	Fullstack Project Setup
92–120	Build a Capstone Full Stack App

□ Phase 4: DevOps Engineering (Days 121–180)

Day	Topic
121	Git & GitHub Basics

Day	Topic
122	Git: Branching & Merging
123	Git: Conflict Resolution
124	Linux CLI Basics
125	Bash Scripting
126	Docker: Basics
127	Docker: Images, Containers
128	Docker: Volumes, Networking
129	Docker Compose
130	Kubernetes: Architecture
131	K8s: Pods, Deployments
132	K8s: Services, Ingress
133	K8s: ConfigMaps, Secrets
134	K8s: Helm Basics
135	Terraform: Basics
136	Terraform: AWS Integration

Day	Topic
137	Terraform: Modules
138	AWS: EC2, S3
139	AWS: IAM, RDS
140	AWS: Lambda, CloudWatch
141	CI/CD Basics
142	GitHub Actions
143	Jenkins Overview
144	Docker + Jenkins Pipeline
145	Kubernetes Deployment
146–180	DevOps Capstone Project with CI/CD Pipeline

□ Phase 5: Projects + Open Source (Days 181–240)

Day	Topic
181–210	Build 3 Major Capstone Projects
211–220	Refactor + Write Tests
221–230	Deploy All Projects (Netlify, Vercel, AWS)

Day	Topic
231–235	Create GitHub Repositories with README
236–240	Contribute to Open Source (PRs + Issues)

□ Phase 6: Interview + Real World Prep (Days 241–300)

Day	Topic
241–260	DSA Revise + Leetcode Daily
261–270	System Design: Basics to Advanced
271–280	Resume, LinkedIn, GitHub Optimization
281–290	Mock Interviews, HR Rounds
291–295	Participate in Hiring Challenges
296–300	Apply to Jobs + Reflect & Improve

Would you like me to create a downloadable version (PDF/Notion) or generate this into a Trello/Google Sheet for tracking?