Google Apprenticeship Complete Playbook

# 2️⃣ Core Coding Prep (Daily Practice)

🔹 Data Structures & Algorithms (must-know)

Arrays & Strings → sliding window, two pointers, prefix sums  
Linked Lists → reverse, detect cycle, merge  
Hash Tables / Sets → frequency maps, lookups  
Trees & Graphs → BFS, DFS, shortest path (BFS/Dijkstra basics), recursion  
Sorting & Searching → quicksort, mergesort, binary search  
Dynamic Programming → subsequence problems (LCS, LIS, knapsack)

📌 Action Plan:

- Solve 1-2 LeetCode Easy/Medium daily  
- Mix topics: do 5 arrays, then 5 strings, then 5 trees, etc.  
- Keep a notes doc for patterns → this will be your revision sheet.

Resources:

- LeetCode Patterns  
- HackerRank Interview Prep Kit  
- Neetcode.io (excellent curated roadmap)

# 3️⃣ System Design Basics (for interviews)

Not full-blown design like senior SWE, but you should know:  
- Client-server model  
- REST APIs (how they work, status codes)  
- Database design basics (tables, primary key, indexing)  
- Caching (e.g., Redis basics)  
- Scalability → load balancer, replication basics

📌 Action Plan:

- Read System Design Primer (intro sections only)  
- Practice explaining how you’d design a simple URL shortener or ToDo app

# 4️⃣ Programming Language Focus

Pick one primary language (Python or Java recommended).

Python Focus:

- Practice writing clean, readable functions  
- Learn collections module (Counter, defaultdict, heapq)  
- OOP basics: classes, inheritance  
- File handling, exceptions

Java Focus:

- Learn Collections API (ArrayList, HashMap, HashSet)  
- OOP principles (encapsulation, inheritance, polymorphism, abstraction)  
- Exception handling & streams

📌 Don’t try to master all 4 languages → focus on 1 solid + be aware of others.

# 5️⃣ Git & Collaboration Skills

Google values teamwork + collaboration.  
- Learn basic GitHub workflows: clone, commit, branch, PR  
- Try one small open-source contribution (even documentation counts!)  
- Document your code (docstrings, README.md)

# 6️⃣ Communication & “Googliness”

During interviews:  
- Talk out loud while coding → show how you’re thinking  
- Ask clarifying questions → shows problem-solving maturity  
- Be positive, curious, collaborative

📌 Practice mock interviews:

- Pramp (free peer interviews)  
- Interviewing.io (sometimes free mock interviews with real engineers)

# 7️⃣ Resume & Application Review

Since this is an apprenticeship (not a senior SWE role):  
- Highlight projects (web apps, small games, automation scripts, open source)  
- Show learning ability → online courses, certifications, hackathons  
- Mention teamwork/collaboration (college projects, GitHub collabs)

📅 8-Week Daily Prep Roadmap

## Week 1 – Arrays & Strings

* Day 1: Sliding Window basics
* Day 2: Two Pointers intro
* Day 3: Prefix Sums practice
* Day 4: Easy Array LC problems
* Day 5: String manipulation
* Day 6: Mix problems (Arrays + Strings)
* Day 7: Review + Notes

## Week 2 – Linked Lists

* Day 8: Reverse Linked List
* Day 9: Detect cycle
* Day 10: Merge two lists
* Day 11: Middle of list
* Day 12: Practice LC Easy/Medium
* Day 13: More Linked List drills
* Day 14: Review + Notes

## Week 3 – Hash Tables / Sets

* Day 15: Frequency maps
* Day 16: Lookup operations
* Day 17: Two-sum, anagrams
* Day 18: Set operations
* Day 19: Medium hash problems
* Day 20: Mix practice
* Day 21: Review + Notes

## Week 4 – Trees & Graphs

* Day 22: Tree traversals (DFS, BFS)
* Day 23: Binary Tree basics
* Day 24: Binary Search Tree problems
* Day 25: Graph BFS
* Day 26: Graph DFS
* Day 27: Shortest Path basics
* Day 28: Review + Notes

## Week 5 – Sorting & Searching

* Day 29: Quicksort basics
* Day 30: Mergesort
* Day 31: Binary Search
* Day 32: LC Easy sorting problems
* Day 33: LC Medium searching
* Day 34: Mix practice
* Day 35: Review + Notes

## Week 6 – Dynamic Programming

* Day 36: DP intro (Fibonacci, Climbing Stairs)
* Day 37: Knapsack basics
* Day 38: LIS problem
* Day 39: LCS problem
* Day 40: Subsequence problems
* Day 41: Mix practice
* Day 42: Review + Notes

## Week 7 – System Design Basics

* Day 43: Client-Server model
* Day 44: REST APIs
* Day 45: Database design basics
* Day 46: Indexing + Primary keys
* Day 47: Caching basics
* Day 48: Scalability concepts
* Day 49: Review + Notes

## Week 8 – Final Revision & Mock Interviews

* Day 50: Arrays & Strings recap
* Day 51: Linked Lists recap
* Day 52: Hash Tables recap
* Day 53: Trees & Graphs recap
* Day 54: Sorting & Searching recap
* Day 55: DP recap
* Day 56: Mock Interviews + Resume polish