



For  Google

The Google interview, *decoded.*

What Google asks freshers, and a 6-month plan to get ready.

Made with an engineer who **works at Google** and **teaches at NYC.**





Before you follow a plan, you should know who made it.

Made with an engineer
who works at Google *and*
teaches at NYC.



Abhishek

Engineering Analyst, Google

Earlier a senior data scientist at Optum. Teaches Data Science at NYC.

He speaks for himself, not for Google.



Our students are already *doing this plan.*

Live classes, five days a week.



MONTHS 1-2

DSA and your first web app

MONTHS 3-4

Projects you can explain

MONTHS 5-6

Timed tests and mock interviews



Where our students *work* *now.*

Students taught by the team behind NYC, at Sheryians Coding School. 1,200+ are placed.

HIGHEST PACKAGE



Kushagra Patidar

SDE 1, Amazon

₹47 LPA

[LinkedIn](#) ↗



Anuj Parashar

SDE, Care Health Insurance

₹22 LPA



Rishika Jain

Cisco

₹17 LPA



Mansi Patel

Airtel

₹15 LPA



Now, let's start *the plan*.

Eight short sections. Read them in order.

- 01 **How Google hires**
- 02 **The 6-month plan**
- 03 **Months 1–2: the basics**
- 04 **Months 3–4: the hardest part**
- 05 **Months 5–6: apply and interview**
- 06 **Your week and 15 problem types**
- 07 **Interview day**
- 08 **7 mistakes to avoid**

3 ways into Google *as a student.*

NOW



You are here

Start the plan in this guide.

1ST-2ND
YEAR

1

STEP

Student Training in Engineering Program. A 10–12 week internship for 1st- and 2nd-years with little experience.

Applications usually open Oct–Dec

Online coding test, then 2 interviews

Best way in from a non-IIT college

3RD
YEAR

2

SWE internship

Most offers

A summer software internship. Do well in your review, and you get the full-time offer before final year.

careers.google.com/students

Usually Aug–Oct

Online coding test, then 2–3 interviews

FINAL
YEAR

3

Full-time job

A new-graduate software engineer role. Referrals and a strong GitHub matter more than your college name.

A referral from any Googler

Deep questions on your projects

Same rounds, higher bar

Every way in has *the same rounds.*

1

Online coding test 90 min

2 coding problems, medium DSA. Your code must be clean and tested.

2

Coding interviews Rounds 2–3 45 min each

One or two problems in a shared doc, not a code editor. Think out loud the whole time.

3

Googleyness Round 4

Questions about you: how you handle unclear problems, feedback and teamwork. Real stories, not slogans.



Offer

Every interviewer scores all 4 things



Problem solving



Coding



Communication



Googleyness



At NYC: tests run without AI, like the online coding test. Working engineers run your mock interviews.

Process details change by year and location. Verify dates on careers.google.com/students. This guide is by the NYC team. NYC is not affiliated with Google.

Your 6-month *plan*.

2 hours a day on weekdays. 4 hours on weekends.
Every month ends with something you can show.

1

Basics

One language. Arrays, strings, hashing. Your first app with AI.

2

Core DSA

Recursion, linked lists, stacks, binary search.

3

Trees and graphs

BFS, DFS, heaps. Timed practice tests begin.

4

Dynamic programming

DP, greedy, backtracking. Put a project online.

5

CS basics, then apply

Design an app out loud. OS and DBMS basics. Résumé, referrals, STEP.

6

Interviews

Weekly mock interviews. Stories about you. Questions on your projects.

By the end of month 6

300

problems, in order.
Not 1,000 at random.

2

projects live online

12

timed practice tests
before the real one

Months 1–2: *learn the basics.*

1

MONTH 1

Pick one language and stay with it



LEARN

Arrays

Strings

Hashing

Two pointers

Sliding window

Use C++ or Java for interviews. Python only if you already know it well.



SOLVE

60 problems, easy to medium. Write the time complexity every time.



BUILD

Your first web app with AI. Put it online in week 2. Yes, week 2.



EVERY DAY

Learn Git. Make one public commit on GitHub.

2

MONTH 2

Core DSA



LEARN

Recursion

Linked lists

Stacks

Queues

Binary search, also on the answer

Sorting

Prefix sums



SOLVE

80 problems. Start a mistakes notebook: why you got each one wrong.



BUILD

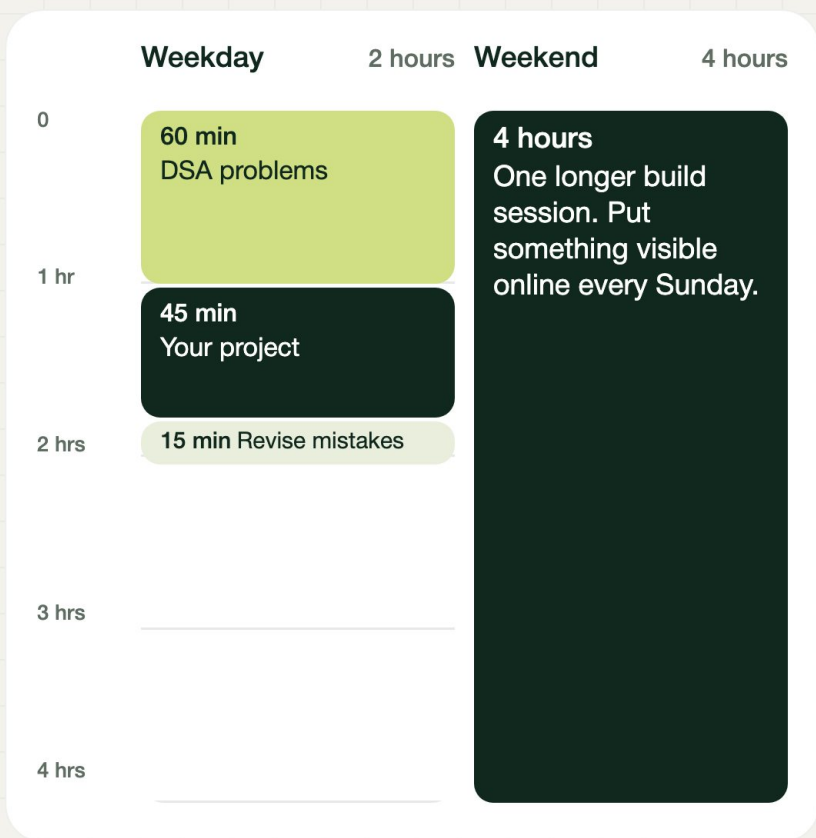
Add a backend, a database and login to your app. Put it online again.



TRY

Join one hackathon. Finish it, even badly.

Your daily routine *for* *months 1-2.*



By the end of month 2, you have



140 problems solved

140 of the 300 in this plan



1 full-stack app, live online

A GitHub that shows work every day



8 weeks, a commit every day



At NYC: projects are built in class on Fridays and reviewed by an engineer, so you can explain every line.

Months 3–4: *the hardest part.*

3

MONTH 3

Trees, graphs and heaps



LEARN

Binary trees

BST

Traversals

Lowest common ancestor

BFS

DFS

Topological sort

Shortest path

Heaps

Priority queues

Top-K



SOLVE

80 problems.



TEST

Your first timed practice test: 3 problems, 90 minutes, no other tabs.



BUILD

Add an AI feature to your app, like search over your own data.

4

MONTH 4

Dynamic programming



LEARN

1D DP

2D DP

Knapsack

DP on strings

Greedy

Backtracking

Tries

Bit tricks



SOLVE

80 problems.



TEST

A timed test every week from now on.



BUILD

A second project. Use a different stack, or build it for a real user. Put it online.



RÉSUMÉ

Version 1. One page, projects first, a number in every line.

By the end of month 4

300

problems

2

live projects

4

timed tests

1

résumé you are
not
embarrassed by

Months 5–6: *apply and interview.*

5

MONTH 5

CS basics, then apply



DESIGN

Design a parking lot, a library or an elevator. Out loud. (OOP and low-level design.)



CS BASICS

OS, DBMS and networks: the 40 questions everyone gets asked.



APPLY

STEP (1st–2nd year) or the SWE internship on careers.google.com/students. Set reminders for the opening dates.



REFERRALS



Any Googler can refer you. Message 10 on LinkedIn with your live project link. Short, specific, polite.



PROFILE

LinkedIn and GitHub tell the same story. Pin your 2 projects.

6

MONTH 6

Interview practice



MOCK INTERVIEW

Every week, with a real engineer. Written feedback. Fix one thing each week.



THINK OUT LOUD

Say the simple, slow answer first, then improve it. Silence loses marks.



YOUR PROJECT

Why this database. What broke. What you would change.



6 STORIES FOR GOOGLEYNESS

An unclear problem

Feedback you acted on

Conflict

Ownership

Teamwork

Pressure



SLEEP

The online coding test is a speed test.

When to *apply*.

J | F | M | A | M | J | J | A | S | O | N | D

STEP

SWE intern

■ STEP: usually Oct-Dec

For 1st- and 2nd-year students. Miss it, and you wait a year.

■ SWE internship: Aug-Oct

For next summer. Google rarely comes to campus, so apply on the site.

What to expect



■ No reply ■ Interview

Expect **20 applications** for **1 interview**. Keep a tracker, and keep going.

Your week, *day by day*.

MON
DSA**TUE**
DSA**WED**
Hard**THU**
CS**FRI**
Test**SAT**
Build**SUN**
Review

MON **2 new DSA problems**
45 min project

TUE **2 new problems**
Revise Monday's mistakes

WED **1 hard problem, 1 hour max**
45 min project

THU **2 new problems**
CS basics, 30 min

FRI **Timed test: 3 problems, 90 min**
No tabs, no AI

SAT **4-hour build session**
Put it online before you stop

SUN **Mock interview or review**
Plan next week, 20 min

The 15 problem types Google keeps asking

1 Two pointers

2 Sliding window

3 Hashing / frequency

4 Binary search on the answer

5 Linked list reversal

6 Stack: next greater

7 Tree traversals + LCA

8 BFS / DFS on grids

9 Topological sort

10 Heaps: top-K, merge K

11 Intervals: merge, overlap

12 1D / 2D DP

13 Knapsack family

14 Backtracking: subsets, permutations

15 Trie + string matching



At NYC: DSA runs as its own track with Akshay, one problem type at a time.

5 rules for *the interview*.

1 Repeat the question. Ask two questions back.

Input size, edge cases, duplicates. Interviewers score this.

2 Say the simple answer first. Then improve it.

A slow $O(n^2)$ answer you explained beats a fast one you could not explain.

3 Write clean code. Use clear names. Test it with one example, out loud.

Google interviews happen in a plain doc, with no autocomplete. Practise typing code in a text file.

4 Know your project better than your résumé does.

Why this stack. What broke. What you would change with 10,000 users.

5 Googleyness is a real round. Have stories for it.

An unclear problem you sorted out, feedback you acted on, a teammate you helped. Specific, short and true.

THEY WILL ASK

Tell me about yourself.

60 seconds, projects first

Tell me about a time you failed.

Tell me about a disagreement in a team.

Why Google?

What did you learn last month?

YOU SHOULD ASK

What does an intern own in the first month?

How is code reviewed on the team?

What made the last intern succeed?

7 mistakes that *get first-years rejected.*

☐ 1. ~~1,000 random problems~~

☒ Solve 300 in order, with a mistakes notebook. Problem types beat volume.

☐ 2. ~~Projects copied from tutorials~~

☒ A to-do app from YouTube is not a project. Build for one real user.

☐ 3. ~~No timed practice~~

☒ The online coding test is 90 minutes. If you have never raced the clock, you lose to it.

☐ 4. ~~Applying late~~

☒ STEP and internship windows open months early. Set the reminder today.

☐ 5. ~~Copying AI code you cannot explain~~

☒ Use AI to build faster. Then read every line. Interviewers will.

☐ 6. ~~Doing everything alone~~

☒ Hackathons and team projects teach you to explain code. Do both.

☐ 7. ~~Waiting for the placement cell~~

☒ Off-campus roles and referrals get more first-years in than campus drives. Send 10 polite messages with a live project link, every month.

We teach this whole plan, *live.*

● ● ● nyc · course

Covers this plan



WE TEACH THIS

Software Engineering + AI

Months 1–6 of this PDF, live. Engineers review every project. Internship prep takes you to the Google interview.

 Live classes Engineer reviews For: **Software engineer intern (SDE)**

Talk to an advisor

See the course → notyourcollege.com



Guide by the NYC team, 2026. Based on Google's public student programmes and interview patterns reported by candidates. NYC is not affiliated with Google and does not guarantee employment. Verify dates and process on careers.google.com/students.