



For  Uber

Crack Uber. *Build it live.*

How Uber India hires students, including
the live build round.

Made with engineers who work at
Atlassian and Microsoft and teach at
NYC.





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

Made with engineers who
work at Atlassian and
Microsoft *and teach at NYC.*



Akshay

Senior Software Engineer, Atlassian

Worked on UPI payments at Jio Platforms. Teaches DSA at NYC.



Swapnil

Forward Deployed Engineer, Microsoft

Works on Microsoft Teams. Teaches Full Stack Engineering at NYC.

They speak for themselves, not for their employers.



ABOUT NYC · 2 OF 3

Our students are already *doing this plan.*

Live classes, five days a week.



MONTHS 1-2

Graphs and core DSA

EVERY
FRIDAY

Working features, built in class

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 Uber hires**
- 02 **The 6-month plan**
- 03 **Graphs: what Uber tests most**
- 04 **Machine coding, step by step**
- 05 **8 practice builds**
- 06 **Your week and 12 problem types**
- 07 **Interview day**
- 08 **7 mistakes to avoid**

3 ways into Uber *as a student.*

1ST-2ND
YEAR



You are here

Start the plan in this guide.

ANY
YEAR

1

Hiring contests

Uber has run coding contests for students in India in past years, some for women only. Top ranks got interviews.

Watch Unstop

Watch the careers page

3RD
YEAR

2

SDE intern

Intern → offer

A summer internship. A good review at the end leads to the job offer.

Mostly campus drives at selected colleges

Some off-campus listings

uber.com/careers

FINAL
YEAR

3

SDE-1

Hiring for new graduates, through campus and referrals.

Referrals matter

Same rounds, higher bar

Where the jobs are



Uber India hires students for engineering in Bangalore and Hyderabad.

Uber has 4 rounds. *2 of them are hard.*

1

Online coding test

60–90 min

Hard

3–4 problems, medium to hard. Speed matters.

2

DSA interview

1–2 problems

Find the best solution, and write clean code.

3

Machine coding

60–90 min

Hard

Build a small working program, live. Section 04 shows you how.

4

Hiring manager

1 round

Your projects, what you owned, and why Uber.

**Offer**

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

Rounds as reported by candidates. They vary by team and year. Verify on uber.com/careers. This guide is by the NYC team. NYC is not affiliated with Uber.

Your 6-month *plan*.

2 hours a day on weekdays. 4 hours on weekends.
Graphs and machine coding get extra time.

1

Basics

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

2

Core DSA

Recursion, stacks, binary search, sorting.

3

Graphs, in depth

BFS, DFS, Dijkstra, union-find. Uber runs on maps.

4

DP and harder problems

DP, heaps, intervals. A timed test every week.

5

Machine coding

OOP and design patterns. One 90-minute build a week.

6

Interviews

Mock rounds: one DSA round, one machine-coding round.

By the end of month 6

4

problems in 90 minutes. Practise at that speed.

8

machine-coding builds before the real one

1

real-time project: a map, live updates or a queue

Graphs: *what Uber tests most.*

Riders, drivers, roads, deliveries. Most of Uber's problems are graph and scheduling problems. Candidates report the same in the test.



MUST KNOW: GRAPHS

BFS on grids

Shortest path in unweighted graphs

Dijkstra with a heap

Topological sort

Union-find

Multi-source BFS (nearest driver)



MUST KNOW: HEAPS AND INTERVALS

Top-K

K closest points

Merge intervals

Meeting rooms

Sliding window maximum

A rate limiter with a queue

Practise these 3 problems



Nearest driver

A grid of drivers and riders. Find the nearest free driver for each rider.

Uses: multi-source BFS



Busiest zones

Ride requests per area per minute. Find the 5 busiest areas.

Uses: a heap, top-K



Travel time

A road graph with travel times. Find the fastest route from A to B. Then one road closes.

Uses: Dijkstra

Machine coding: *your 90 minutes, step by step.*

You get one problem and write a working, readable program. Students who only practised DSA fail here.



60–90 minutes



No framework



No internet



No AI

1

Ask questions first 5 min

Clarify what to build before you write any code.

2

Write the classes first

Rider, Driver, Trip, Matcher. Names on paper before code.

3

Get one flow working By minute 30

End to end. A working request-to-trip flow beats five half-written features.

4

Keep it easy to extend

Leave a clear place for the next feature, like new pricing or a new vehicle type.

5

Prove it runs

Two tests, or a `main()` that runs the flow. Run it in front of them.



Say what you would change

With more time: concurrency, storage, scale. Say it. Do not build it.



At NYC: every Friday you build working features in class, and an engineer reviews the code.

8 practice builds, *one a week.*



WEEK 1

Ride matching

Nearest driver



WEEK 2

Parking lot

WEEK 3

Expense sharing

Like Splitwise



WEEK 4

Rate limiter

WEEK 5

Food-order tracker

WEEK 6

Elevator system

WEEK 7

Snake and ladder

WEEK 8

Key-value store

With expiry

Every build, the same routine

90 min

Build it. Timer on, no AI.

30 min

Compare with a good solution.

Next day

Rewrite your worst class.

Also build one real-time app



Live location on a map, updating every few seconds



Deployed, with a link



10,000 drivers: you will be asked how your app handles them

Machine-coding builds are the fastest way to get better at code structure. Do them even if your target is not Uber.

Your week, *day by day*.

MON	TUE	WED	THU	FRI	SAT	SUN
DSA	Graphs	Hard	OOP	Test	Build	Review
MON	2 DSA problems 45 min project					
TUE	2 graph problems Revise Monday's mistakes					
WED	1 hard problem, 1 hour max OOP, 30 min					
THU	2 problems One design pattern					
FRI	Timed test: 4 problems, 90 min No tabs, no AI					
SAT	Machine coding: 90 min Then compare and fix					
SUN	Mock interview or review Plan next week, 20 min					

The 12 problem types Uber keeps asking

1	BFS on grids	2	Dijkstra
3	Union-find	4	Topological sort
5	Heaps: top-K	6	Intervals
7	Sliding window	8	Binary search on the answer
9	Hashing	10	1D / 2D DP
11	Trie	12	LRU cache



At NYC: DSA runs as its own track with Akshay, one pattern at a time, with practice sets.

5 rules for *the interview*.

1 Ask about scale before you code.

How many drivers? Real time or batch? It changes the answer.

2 Say the simple answer first. Then improve it, out loud.

Silence loses marks. A plan said out loud earns them.

3 In machine coding, run it early and often.

A program that runs at minute 30 beats a perfect one at minute 95.

4 Know your real-time project end to end.

How location updates flow. What happens when a phone goes offline.

5 Be specific about why Uber.

A problem you noticed as a rider, and how you would approach it.

THEY WILL ASK

Walk me through your real-time project.

Find the nearest free driver. Then make it faster.

Extend your design: add scheduled rides.

Tell me about a time you owned something end to end.

Why Uber?

YOU SHOULD ASK

What does an intern build in the first month?

How is code reviewed on the team?

What made the last intern get the job offer?

7 mistakes that *get students rejected.*

✕ ~~1. Only easy problems~~

✓ The online coding test is harder than most. Practise medium problems against the clock.

✕ ~~2. Skipping machine coding~~

✓ It is a full round. Do 8 timed builds before you interview.

✕ ~~3. Weak on graphs~~

✓ BFS, Dijkstra and union-find show up again and again.

✕ ~~4. One giant class~~

✓ A 500-line class fails machine coding, even if it works. Plan small classes first.

✕ ~~5. No timed practice~~

✓ 4 problems in 90 minutes needs practice at that speed.

✕ ~~6. Copying AI code you cannot explain~~

✓ The build round is live, with no AI. Read every line you write before then.

✕ ~~7. Only applying on campus~~

✓ Uber visits few colleges. Apply on the careers page, enter contests, and ask engineers for referrals with a live project link.

We teach this whole plan, *live.*

● ● ● nyc · course

Covers this plan



WE TEACH THIS

Software Engineering + AI

Months 1–6 of this PDF, live: DSA, OOP and machine-coding practice with engineer reviews. Internship prep runs timed mock rounds.

 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 Uber's public careers information and interview patterns reported by candidates. NYC is not affiliated with Uber. NYC does not guarantee employment or selection. Verify dates and process on uber.com/careers.