



For



Microsoft

Crack Microsoft. *From first year.*

The 6-month plan, for students who
haven't started yet.

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





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

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



Swapnil

Forward Deployed Engineer, Microsoft

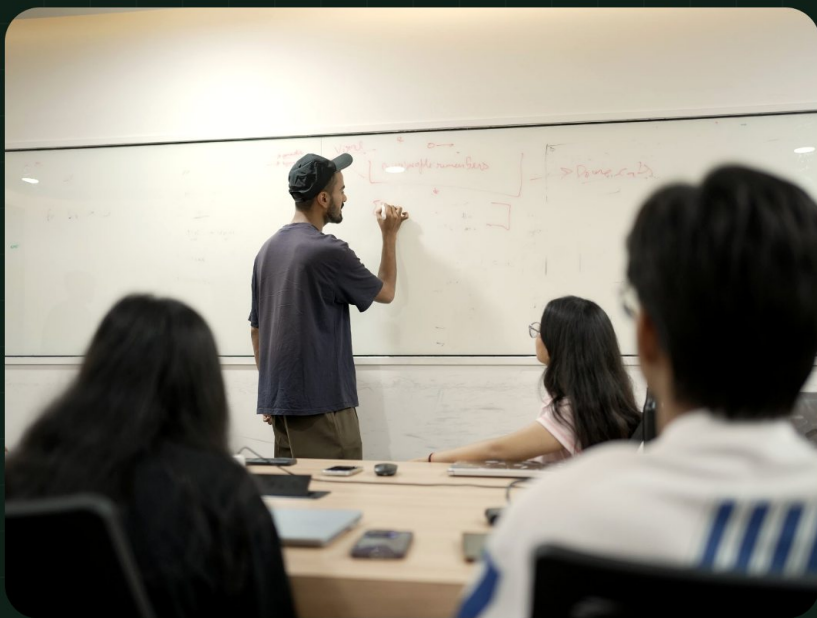
Works on Microsoft Teams. Teaches Full Stack Engineering at NYC.

He speaks for himself, not for Microsoft.



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, reviewed by an engineer

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 Microsoft 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 Microsoft *as a student.*

1ST
YEAR



You are here

Start the plan in this guide.

2ND
YEAR

1

Engage

A Microsoft program. You build a project with a mentor, then interview for an internship.

Apply Feb–Apr

Starts with a coding test

Good way in from a non-IIT college

3RD
YEAR

2

Internship

Most offers

A summer internship. Do well, and you get a job offer before final year.

On campus or the careers site

Coding test, then 2–3 interviews

FINAL
YEAR

3

Full-time job

Hiring through your college and referrals. The hardest way in if you skip the first two.

A referral helps

Deep questions on your projects

Same steps, tougher

Every way in has *the same 4 steps.*

1

Online coding test 60–90 min

2–3 problems, medium level. Your code must be clean and must work.

2

Technical interviews 2–3 rounds

Solve problems while you explain out loud. One design question. Questions on your projects, line by line.

3

Manager interview 1 round

Questions about you: why Microsoft, a time you failed, how you learn.



Offer

In every step, they check 4 things



Problem solving



Clean code



Explaining clearly



Wanting to learn



At NYC: you practise every step in class. Timed tests run without AI. Working engineers run your mock interviews.

Steps can change by year and city. Check dates on careers.microsoft.com. This guide is by the NYC team. NYC is not linked to Microsoft.

Your 6-month *plan.*

2 hours a day on weekdays. 4 hours on weekends.

1

Basics

Learn one language. 60 problems. Build your first app with AI.

2

Core coding

80 problems. Add login and a database to your app.

3

Trees and graphs

80 problems. Your first timed test.

4

Harder problems

80 problems. A second project. Your résumé.

5

Apply

CS basics. Apply to Engage or internships. Ask for referrals.

6

Interviews

A mock interview every week. Practise your stories.

By the end of month 6

300

coding problems, in order

2

projects live online

12

timed practice tests

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

One public commit on GitHub.

2

MONTH 2

The core topics



LEARN

Recursion

Linked lists

Stacks

Queues

Binary search

Sorting

Prefix sums



SOLVE

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



BUILD

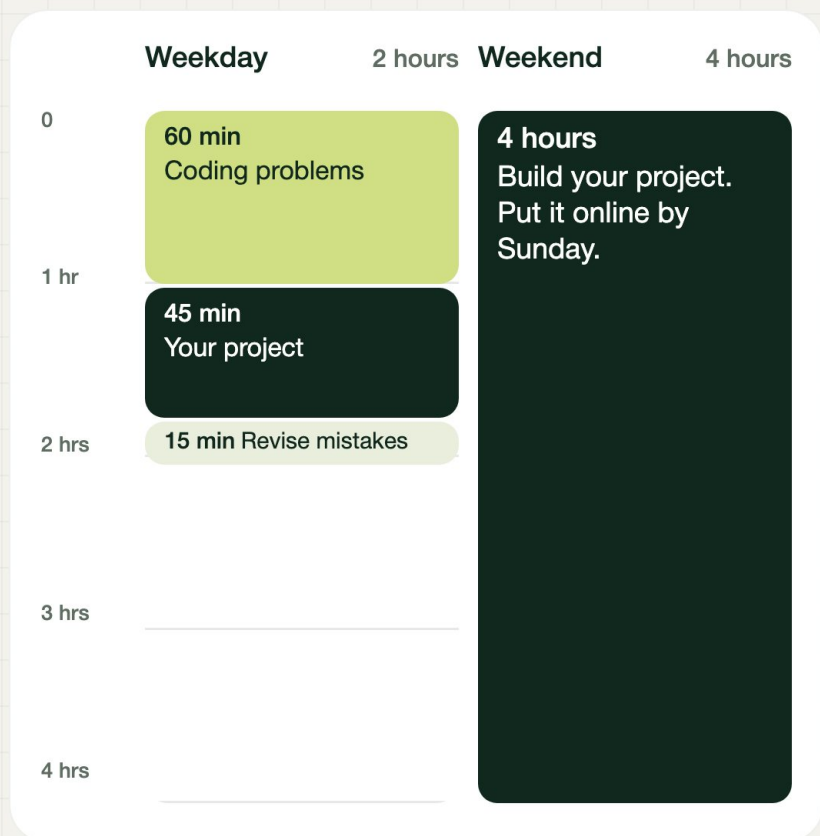
Add a backend, a database and login to your app. Put it online again.



TRY

Join one hackathon. Finish it, even badly.

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



By the end of month 2, you have



140 coding 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: Fridays are build days. You build your project in class, and an engineer reviews it.

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

Top-K



SOLVE

80 problems.



TEST

Your first timed 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

Harder problems



LEARN

Dynamic programming

Knapsack

Greedy

Backtracking

Tries

Bit tricks



SOLVE

80 problems.



TEST

One 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

Engage (2nd year) or internships on careers.microsoft.com. Set reminders for the opening dates.



REFERRALS



Message 10 Microsoft engineers 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 answer first, then improve it. Silence loses marks.



YOUR PROJECT

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



6 STORIES ABOUT YOU

Failure

Conflict

Learning fast

Ownership

Teamwork

Pressure



SLEEP

The online test is a speed test.

When to *apply*.

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

Engage

Internships

■ Engage: Feb–Apr

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

■ Internships: Aug–Oct

For next summer. Off-campus roles open earlier than your college tells you.

What to expect



■ No reply ■ Interview

About 20 applications get you **1 interview**. That is normal. Track every application 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 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 Microsoft asks most

1 Two pointers

2 Sliding window

3 Hashing / counting

4 Binary search on the answer

5 Reverse a linked list

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 make it faster.

A slow 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.

Microsoft looks at code quality, not just the answer.

4 Know your project better than your résumé does.

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

5 Have a true story about learning from a mistake.

They check this. "Here is what I got wrong, and what I did next week." 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 Microsoft?

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 test is 90 minutes. Practise against the clock, or the clock wins.

☐ ~~4. Applying late~~

☒ Engage 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 Microsoft 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 Microsoft's public hiring programmes and interview patterns reported by candidates. NYC is not affiliated with Microsoft and does not guarantee employment. Verify dates and process on careers.microsoft.com.