

Contents

Overview	1
Interview Prep Game Plan	2
Coding Game Plan	3
Personal Brand Game Plan	4
Taking your game to the next level	5
Matt's Journey from Undergraduate to Google	6

Overview

I think everyone — whether it be you never coded / not cs major, are cs major but haven't done much outside of school or have interviewed with some companies already — can easily achieve the SWE title (100k sal) you desire in as little time possible taking the right strategy: a mix of learning to code applications and interview prep.

Learning how to code will help you put projects on your resume and land that first internship. Interview prep you want to have a strong exposure to break into Fortune 500 companies, banks and high paying salaries at FAANG. Spending like 60% on interview prep 35% on code / dev and 5% on resume / application / linkedin is a move.

If yk me I am big on hackathons. If you win, thats a plus; a cherry on top. Reach out to judges or sponsors that were representing the company during the hackathon, they company you are now applying to and ask them for internal referral or somehow let them k you applied and would appreciate if they flag your application. I do it every time I apply to Google and Microsoft. But the best thing about hackathon is your time to code a project and test what you know, what you can create, and how could you can pitch your work. First hack will be intimidating af. But the 5th and 10th will be much smoother. There's like a hack happening everyday. Check them out here: devpost.com/hackathons. One a month is a move.

Perks of SWE: high demand in every city in every firm, high pay relative to other jobs, great office furniture and perks, solve real life problems at scale, and good career movement...lots of opportunity to rise in seniority, impact and transition to other roles in business such as PM, Data

Check out this website for some pay perspective. levels.fyi/

Hope this helps - Yasin and Matt

Interview Prep Game Plan

- Pick up python basics
 - Codecademy has great tutorials
- Pick up what intv prep is all about
 - [a playlist by cs dojo](#) + anything
- Pick up data structures basics
 - A coursera / udemy / youtube playlist of choice (not more than 10 hrs)
- Pick up algorithms basics
 - A coursera / udemy / youtube playlist of choice (not more than 10 hrs)
- Pick up technical interview patterns ~ 10-15 topics
 - [Grokking the coding Interview...resource](#)
- Do problems 20-30 per topic
 - Leetcode / hackerrank
- Do mock interviewing 20-30
 - [Pramp live interviews](#) + 1v1 w intv buddies

Coding Game Plan

- Website basics
 - Start [here html](#) with Codecademy
- Full stack Website and software engineering overview
 - [The Complete 2020 Web Development Bootcamp Udemmy course](#)
- Get comfortable to how to push code from terminal to github using bash
- Build personal website
 - Buy domain (google domain recommend)
 - Do some seo optimization strategy (goal is to be first when someone google u)
 - Setup firebase analytics to see how many users are on the website (firebase!)
- Youtube firebase firestore backend fundamentals
 - Add Contact form to website
- How to work with multiple people on same website project using branches + pull request
- Explore other customer facing development such as ios apps or android apps or hybrid apps
 - [iOS 13 & Swift 5 - The Complete iOS App Development Bootcamp udemmy course](#)
- Apply what you learned at hackathons...tbh I do most of my learning in hackathons...the first few times I hardly accomplished anything but after 18 hackathons I feel some solid progress
- The best time you would learn the most is during an internship hence why companies value work experience so much

Personal Brand Game Plan

- No point in doing all this if we can't land interviews. This is all about selling yourself.
- Interviews are given all just from resumes.
- Most of time a machine called ATS reads your resume and gives you a score
- Try to update cv every week; impromptu edits tend to be great
- Google recruiter says a strong resume is of the following format
 - Education
 - No more than one line for college and major
 - If your lacking on exp, feel free to add relevant coursework and self study courses like udemy stuff
 - Experience
 - Any job that shows responsibility is good. Programming roles are better.
 - "Accomplished x by doing y resulting in z" - more metrics better
 - Projects
 - Something that seems impossible to do but you pulled of - that how it's worded
 - Activities / awards
 - Anything you do outside the classroom preferably career related
 - Skills
 - Languages you are familiar with
 - Technologies you have used (put aws even if you haven't yet; watch quick youtube)
- Strong Linkedin profile with industry as "Intranet"
- Lots of green dots on Github...more the better
- Here's how my resume changed over last 3 years yasinehsan.com/url/past
 - Fall 2019 I interviewed with **Facebook, Apple, Google, Microsoft, Amazon**
 - Though, I didnt get these offers I knew one thing: my strategy to apply worked
 - What I have to improve was strong foundation in Technical prep
- For application process make a list of your top 30 companies to apply to
 - APPLY TO THEM ALL ON AUGUST 1st
 - After your top 30 done, keep applying 3-5 a day
 - I applied to 300+ fall 2019

Taking your game to the next level

- This is for high achievers if you wanna finesse this game
- I look at it like if I wanted to to be top 10 pick in NFL DRAFT
- This is goals if you are looking to land 150-250k as your entry offer
- Caution: you wanna be able to finesse everything above first

Interview Prep

- You are able to solve interview problems relatively easily and wanna step up
- Forget about space time complexity
- Talk about multi threading, parallelism, locks to run code faster better at OS level
- Talk about coding language quirks such as what happens bts during compile
- Give some real life example situations such as Uber runs like...
- Solve the same problem in multiple languages (swift, java) and opportunity costs for each language

Coding Prep

- You are able to code basic website / app with frontend and backend
- Look into cloud services in AWS such as s3, ec2, redis...all can also be done w Google
- Look into “middle-end” such as docker, kubernetes
- Look into unit testing
- Release apps that real users can search online or on app store and use
 - Perhaps this can be the next uber!

Personal Brand

- You are able

Matt's journey from Undergraduate to Google

Here I'll be sharing the steps I've specifically taken from 0 experience to landing my first full-time job at Google. Think of this more as a case study/application of the tips Yasin shared above. Everyone's journey will be different but hope this helps nonetheless.

Legend

Orange - Portfolio Builder

Green - Interview Prep

Yellow - Networking Efforts

Background

- School: CUNY Baruch College (Fall 2016 to Spring 2020)
- Major: BBA in Computer Information Systems
 - Sidenote: Baruch CIS has VERY MINIMAL programming requirements, after all its a BBA thats geared more towards business-side, project management etc.
 - **Key point: You don't need a great formal education background/CS degree to become a SWE.** GPA is also NOT IMPORTANT, see it more as a plus.
- Freshman Year
 - Did nothing
- Sophomore Year
 - Declared CIS major.
 - Fall 2017: Took 1 intro to python class.
 - Spring 2018: Didn't really do any programming. **Don't be like me. Start as early as possible.**
- Junior Year
 - Decided to start **seriously** pursuing SWE.
 - Fall 2018: **Started building side projects**
 - Fall 2018: **Enrolled in Codepath iOS Course at QC**
 - Fall 2018: **Picked up Udemy Course on Python Data Structures**
 - **Started to explore/learn about leetcode**
 - Fall 2018: **Started attending Hackathons.**
 - **Managed to win a Google NYC Tour out of HackCooper -> Networked with Brendan Collins (Google NYC University Recruiter)**
 - Spring 2019: **More Side Projects + Hackathons**
 - Spring 2019: **Led Codepath iOS Course at Baruch**
 - Spring 2019: **Attended QC Tech Festival at Google NYC**
 - **More facetime with Brendan Collins.**
 - Summer 2019: **First SWE Internship at SAS Institute.**
 - Summer 2019: **Took Codepath Interview Prep Course**
 - **More leetcode prep**

- Senior Year
 - Fall 2019: More side projects + Hackathons.
 - Side-note: [CodeForGood](#) Hackathon by JPM is a great place to get internships/FT offers. They run it annually every Fall semester. Applications start around july/august.
 - Fall 2019: Enrolled in CUNY Tech Prep. Great for networking with CUNY CS students.
 - Fall 2019: Enrolled in Facebook ABCS Program. Great place to network with FB engineers + CS students across NYC.
 - More Leetcode prep
 - Managed to develop relationships with FB recruiters
 - Managed to get FB interview (though failed).
 - Spring 2020: HEAVY LEETCODE GRIND (resources I used in detail below)
 - Received Offer at Google

Key thoughts

Three things are really essential to becoming a marketable SWE:

- 1) Building your **portfolio** -> For resume
- 2) **Networking** Efforts/Events -> To land interviews. Don't rely on the online black hole.
- 3) **Interview Prep** -> To get the *OFFER*.

Ways to build your portfolio

- **Build Side Projects.** ([Watch from 14:08 to 16:19](#))
 - Do individual projects, group projects, open source, whatever you're passionate about. Explore multiple fields such as mobile/backend/web full-stack or focus on one and go deep. Whatever it is, JUST DO SIDE PROJECTS.
- **Attend Hackathons.**
 - Best way to learn new tools/frameworks/projects in the shortest amount of time. ALSO GREAT FOR NETWORKING. HIGHLY RECOMMEND.
- **Internships**
 - Experiences above will help you land the first internship. If you don't have internships, KEEP BEING PROACTIVE OUTSIDE OF THE CLASSROOM.

How to land interviews ([Watch from 5:36 to 13:38](#))

- **Recruiters**
 - Attend hackathons/events to network and find recruiters.
 - HARNESS LINKEDIN to search for recruiters. Message them asking if they hire for SWE, if not ask to be redirected to the SWE recruiter/local recruiter/University Programs recruiter/School recruiter/you get the gist. Get in touch with the appropriate recruiter.
 - SAVE all your recruiter contacts from your previous job hunts. Make a doc/excel sheet of their emails. Keep in touch with them.
- **Referrals**
 - LinkedIn. Search “**company name** + Software Engineer + Queens College/CUNY/Your school”. CONNECT WITH THEM.
 - [rooftopslushie.com](https://www.rooftopslushie.com)
- **Key point:** Knowing an Internal connection is the **most effective way** to go job hunting. Beat the ineffective online blackhole. **Watch these 2 videos below, FULL OF NUGGETS OF WISDOM.**
 - [Software Engineering Job Tips From A Google Recruiter](#)
 - [Google Recruiter Tips On Offer Negotiation, Interviews, And More](#)

How I personally interview prepped/Resources I used

- 1) Learn/Implement Data Structures & Algorithms
 - [Udemy Course I used](#)
- 2) Practice on leetcode/hackerrank
 - (for reference, I solved ~200 problems on [LC](#), 93 easy, 98 medium, 18 hard)
 - These 2 links below cover SO MUCH together and should cover your bases pretty well.
 - First Link groups questions by **Algorithmic pattern**.
 - Second Link groups questions by **Data Structure**.
 - [Grokking Coding](#)
 - [Team Blind Top 75](#)
 - **Key point: By doing these 2 lists, you'll be better equipped to identify the right pattern and data structure to use in an actual interview**
 - Other resources I used to fill in gaps in knowledge/weaker areas
 - Cracking the Coding Interview
 - [GeeksForGeeks](#)
 - I used this mainly for studying Dynamic Programming
 - [Back To Back SWE](#)
 - Great for visualizing DP/Graph questions
 - [Tushar Roy](#)
 - Great for DP questions
 - And of course, general [leetcode](#)
- 3) Apply knowledge through mock interviews
 - [interviewing.io](#)
 - [pramp](#)
 - Whiteboard with friends
- 4) **KEY POINT:**
 - The number one factor (the deal breaker) that decides if someone gets hired or not is your ability to show your algorithmic ability. So go do some leetcode.
 - *"Interview feedback. It really is that simple. Like do you have the right signal, do you have the right expertise, did we prove it during the interview process. It is literally interview feedback. This person meets the requirements, passes the bar, or doesn't."* -[Amy Miller](#), Google Recruiter, ~28:53 onwards.

TL;DR (Too Long; Didn't Read)

- Build stuff.
- Get an internal connection (recruiter/referrals). Please watch this video.
 - [Software Engineering Job Tips From A Google Recruiter](#)
- Interview Prep with these 2 lists. These will get you pretty far.
 - [Grokking Coding](#)
 - [Team Blind Top 75](#)
- Get Offer.