facebook

Interview Prep for Software Engineering Interns and University Grads

Write an Exceptional Resume

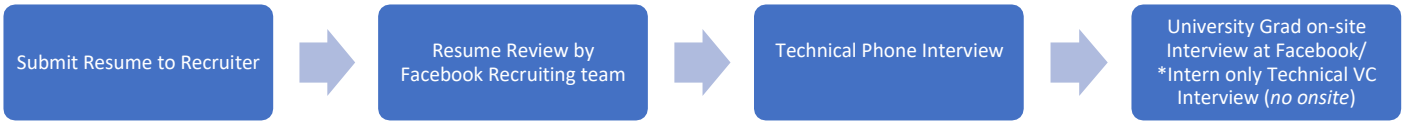
- Please keep your resume to a **1-page PDF**
- Header should include: Name, email address, contact number, links to personal website, LinkedIn, GitHub, StackOverflow, etc.
- Strong Bullet Points: 2-3 per role - focus on your accomplishments in the following approach: "Accomplished X by implementing Y which led to Z." Here's an example "Reduced object rendering time by 75% by implementing distributed caching, leading to 10% reduction in log-in time."
- A suggestion for the outline of your resume: **(1)** Education and Grad Year **(2)** Programming Languages and Software **(3)** Internship and/ or Industry Experience **(4)** Projects/ Hackathons Experience **(5)** Notable classes
- **No industry experience? No problem!** We're looking for a variety of experience – here are a few ideas: internships, hackathon participation, contribution to Open Source projects, class projects, and/or personal apps and projects built outside of the classroom. We also appreciate participation in CS-related student groups or experience as a TA of a CS class – particularly Data Structures, Algorithms or specific Programming-related coursework

Recruitment Process Overview:

- **Technical Video Conference (VC) Interview**: This will include a 45-minute interview with an engineer one-on-one and answer technical questions by writing code on Coderpad. Discussion is encouraged throughout the interview and you'll have an opportunity to ask questions. ***Interns who pass their First-Round interview will be invited to participate in a Second-Round Technical VC Interview.**

- **On-campus Interviews:** This will include a 45-minute interview on your college campus with an engineer one-on-one. You'll be asked technical questions and code out your answers on a whiteboard. Discussion is encouraged throughout the interview; and you'll have an opportunity to ask questions. Both Interns and University Grads are encouraged to participate in OCI's. OCI's can include a second-round interview on-site for both Interns and University Grad candidates.

- **On-site Interview for University Grads (Second-Round of Interviews)**: This will include at least two 45-minute coding interviews with an engineer one-on-one. You'll be asked technical questions and code out your answers on a whiteboard. Discussion is encouraged throughout the interview; and you'll have an opportunity to ask questions. It also includes a third technical interview which also has a coding component but is aimed to get a sense of who you are - independent from the specifics of engineering and code.



Recommended Preparation

Step #1: Watch this Crush Your Coding Interview at Facebook Workshop and read Preparing for a Facebook Software Engineering Interview at Facebook. Visit the following links to practice your coding skills: Career Cup, Top Coder, Project Euler. And, visit the Facebook Engineering page. Use the programming language you're best at - it's important to write your solution correctly and in time, so use the coding language you are most familiar with.

Step #2: Build an **Interview Prep Schedule** for yourself (*template below*). We recommend scheduling your interview **2-weeks** from when your recruiter reaches out to you regarding an interview unless you are up against a hiring deadline. Next, build in 1-2 hours per school night to practice whiteboarding code **without** a computer. Build 4-10 hours of interview prep into your weekend. Read these Glassdoor reviews regarding the interview process from an interviewee's perspective.

- Practice solving interview-style coding questions in a timed environment where you give yourself 10-15 minutes to write out the solution to the question. Think out loud if you are working through a solution as the engineer/interviewer will want to know how you approach the coding problem. Pro-tip: Learn how to compute time and space complexity of their code (O(n) and work the clock.

  Know how to work the clock: You will have **45 minutes** total during your initial interview
  - …give yourself **5 minutes** for introductions
  - …you have **40 minutes** remaining; or **20 minutes** for each question (2 possible coding questions)

- …spend **5 minutes** asking clarifying questions and thinking outload
- …spend **10 minutes** coding on Coderpad
- …spend **5 minutes** testing your code and repeat

- Once you have a solution, you should look at it and see if it's something that you would approve if it were submitted to you as a proposed part of your codebase. Make sure that it is correct, that you have considered the edge cases, that it is efficient, and that it clearly reflects the ideas that you're trying to express in your code. A common mistake is to look at interview problems, recognize them, and sort of understand them, but not to be able to code them
- Give your interviewer good signal: Be able to quickly whiteboard solutions to questions that involve describing an algorithm and translating that description into working code. Focus on the big picture and not on minor bugs or syntactical errors. Of course, if you're able to write clean code please do so.
- Communicating during the Interview**:** Give structured answers, start off with your "Nugget First". "Nugget First" means starting your response with a "nugget" that succinctly describes what your response will be about. Next, move into the S.A.R method. Start off with outlining the **Situation**, then explain the **Action** you took, and lastly, describe the **Result**.
- Mock Interview with a peer, TA, professor or manager – Mock Interview with someone who knows absolutely nothing about code and explain your solution. Be able to talk to your interviewer about your resume in detail regarding internship experience, industry experience, projects and apps you've worked on, hackathons you've participated in, classes you've been a TA in, etc.
- Ask clarifying questions to your interviewer regarding your technical question during your interview.
- Find and designate a quiet area to prepare and take your interview.
- Take a minute to visit Careers at Facebook

Interview Prep Schedule

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
Activity							
Learnings							
What did I accomplish this week?							
Mock Interviewer 1: **Whiteboard Interviewer/ Reviewer 1**:							
	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
Activity							
Learnings							
What did I accomplish this week?							
Mock Interviewer 2: **Whiteboard Interviewer/ Reviewer 2**:							

Step #3: Have a good understanding of Facebook Family of Products and Facebook's 5 Core Values

- **Know Your Technical Projects**: As part of your preparation, focus on two or three projects that you have deeply mastered. Select challenging projects that you played a key role in and that have technical depth.
- **Practice Talking About Your Experience**: Get comfortable talking about your internship or industry experience. Focus on the impact and/ outcome of your work. Sprinkle in examples of success.
- **Know Why You Want to Work at Facebook**: This question will be asked by one or more of your interviewers.

Coding Brain Teaser – Helpful steps in working through a solution

Array of integers and a target value: Determine the number of *distinct* pairs of elements in the array that sum to the target value. For instance, given the array [1, 2, 3, 4, 5, 5, 5, 5, 6, 7, 8, 9, 1, 9], and a target value of 10, the five pairs [1,9], [2,8], [3,7] and [4,6] and [5, 5] all sum to 10. **Note** that the pair [1, 9] at indexes 0 and 11 is indistinct from any other combination of 9 and 1 from indexes 0, 11, 12 and 13. Also, note that [5, 5] pairs from indexes 4, 5, 6 and 7 are indistinct. All permutations creating pairs from these four elements count as one pair as well.

1) Repeat the question back to your interviewer to ensure you fully understand what's being asked.
2) Ask questions about edge cases.
3) Take your time to think about a solution before coding.
4) Start coding while explaining your entire thought process. It's encouraged to talk out loud. Don't worry if you run into trouble! Feel free to ask your interviewer for a hint.
5) Run through your solution with test cases. Continue talking out loud.
6) Iterate – how can you do better with runtime/memory?