A. Common Software Engineer behavioral interview questions

1. What is the most challenging aspect of your current project?
2. What was the most difficult bug that you fixed in the past 6 months?
3. How do you tackle challenges? Name a difficult challenge you faced while working on a project, how you overcame it, and what you learned.
4. What are you excited about?
5. What frustrates you?
6. Imagine it is your first day here at the company. What do you want to work on? What features would you improve on?
7. What are the most interesting projects you have worked on and how might they be relevant to this company's environment?
8. Tell me about a time you had a disagreement with your manager.
9. Talk about a project you are most passionate about, or one where you did your best work.
10. What does your best day of work look like?
11. What is something that you had to push for in your previous projects?
12. What is the most constructive feedback you have received in your career?
13. What is something you had to persevere at for multiple months?
14. Tell me about a time you met a tight deadline.
15. Time management has become a necessary factor in productivity. Give an example of a time-management skill you've learned and applied at work.
16. Tell me about a problem you've had getting along with a work associate.
17. What aspects of your work are most often criticized?
18. How have you handled criticism of your work?
19. What strengths do you think are most important for *your job position*?
20. What words would your colleagues use to describe you?
21. What would you hope to achieve in the first six months after being hired?

22. Tell me why you will be a good fit for the position.
23.
24.

B. How to prepare for behavioral interviews

1. Learn the STAR answer format
2. Prepare your answers to commonly asked questions
3. Prepare experiences to showcase fit to the company's culture / core values
4. Try out mock behavioral interviews

1. Learn the STAR answer format

The STAR format is a framework to help you organize answers to behavioral questions - especially ones requiring you to discuss previous experiences

- Situation - The interviewer wants you to present a recent challenge and situation which you found yourself in
- Task - What were you required to achieve? The interviewer will be looking to see what you were trying to achieve from the situation. Some performance development methods use "Target" rather than "Task". Job interview candidates who describe a "Target" they set themselves instead of an externally imposed "Task" emphasize their own intrinsic motivation to perform and to develop their performance
- Action - What did you do? The interviewer will be looking for information on what you did, why you did it, and what the alternatives were
- Results - What was the outcome of your actions? What did you achieve through your actions and what did you learn? What steps did you take to improve after the experience?

Example of how to apply STAR to a behavioral interview question

Here's an example of how the STAR format can be used to answer the question: "Tell me about a time in which you had a conflict and needed to influence somebody else".

Situation

"I was the team lead of a school project about building a social network mobile web app. Our designer's midterms were approaching and didn't have time to produce the mockups. Our front-end person was rushing him for the mockups so that he could proceed with his work, and that was stressing the designer out. The atmosphere in the team was tense."

Task

"As the team lead, I had to resolve the tension between the front-end developer and the designer so that the team could work together peacefully and complete the project on time."

Action

"I spoke to the front-end developer to ask him why he was rushing the designer for the designs. He said that he wanted the designs early because it would be a waste of time rebuilding if the designer designed something different eventually. I explained to him that the midterm dates were out of the designer's control and we had to be more understanding about each other's schedules. I spoke to the designer to get a rough idea of what he had in mind and asked him when he could commit to producing the high-fidelity designs. He replied that he could start on them as soon as his midterms were over. I explained to him why the front-end developer was pushing him for the mockups, and that the front-end developer had no ill intentions and simply wanted the project to succeed. As someone with some experience in UI/UX design, I came up with wireframe mocks, ran them by the designer for approval, then passed them to the front-end developer to start building. I encouraged the front-end developer to use placeholders and not be too concerned about the details for now. We could build the non-UI parts first (authentication, hook up with APIs) and tweak pixels and add polish later on. The front-end developer agreed and went ahead with the approach. I explained to the front-end developer that the designer will pass us the mockups after his midterm, by <DATE>."

Result

"When our designer ended midterms, he came back with beautiful mockups that fit well into the wireframes. Our front-end developer implemented them with great care to detail. We ended up scoring top marks for the project and became a great team."

2. Prepare your answers to commonly asked questions

The next natural step is to start preparing your answers for commonly asked behavioral interview questions. You may refer to my list of 30 questions which were collated across top tech companies for this.

While most people might be inclined towards memorization, it's much better to pen down bullet points to each question and practice verbalizing them near to the interviews, so that your answers will come out more naturally.

3. Prepare experiences to showcase fit to the company's culture / core values

As aforementioned, most top tech companies use their company values to evaluate candidates in behavioral interviews. As such, you should do your research to find out what those values are and ensure you have prepared experiences that showcase fit.

4. Try out mock behavioral interviews

If you would like to practice behavioral interviews with professional interviewers from top tech companies,

How behavioral interviews are evaluated

Unlike technical interviews, behavioral interviews have a lot more variance in terms of evaluation criteria. However, most top tech companies use their company values to evaluate candidates. Interviewers typically have to fill in a section evaluating how a candidate has displayed behaviors in line with company values. It is for this reason that you should search up a company's culture and values and ensure that you embody them within your answers. You'd also find it useful to speak with one of your connections currently working at the company you are applying for to find out more about which values are typically valued in the team.

Some examples of common values evaluated are:

- Motivation - What drives you? Ideal candidates are self-motivated, passionate about technologies and products that have a real impact.
- Ability to be Proactive - Are you able to take initiative? Given a difficult problem, are you able to figure out how to get it done and execute on it?
- Ability to work in an unstructured environment - How well are you able to take ownership in ambiguous situations? Or do you rely on others to be told what to do?
- Perseverance - Are you able to push through difficult problems or blockers?
- Conflict Resolution - How well are you able to handle and work through challenging relationships?
- Empathy - How well are you able to see things from the perspective of others and understand your motivations?
- Growth - How well do you understand your strengths, weaknesses and growth areas? Are you making a continued effort to grow?
- Communication - Are you able to clearly communicate your stories during the interview?

To assess these focus areas, interviewers ask questions on your work history and dig into the details of how you handled various situations. To assess your seniority, for each focus area they determine if the scope of the situation is something they would expect for a junior, senior, or staff engineer (more details on this below).

Example questions and responses

Below are questions and answers illustrating how interviewers collect signals on the candidate for each focus area. In a typical interview, they'll ask the you five or six questions and dive deep into the details of each situation. Each question may provide signals on more than one focus area.

Motivation

Example Questions:

- What project are you most proud of and why?
- Tell me about a recent day working that was really great and/or fun.

Example Responses:

- Junior: A story about a project they are proud of that had an impact on their team.
- Senior: A story about a project they are proud of that had a large impact on their team.
- Staff: A story about a project they are proud of that had a large impact on their org.

Ability to be proactive

Example Questions:

- Tell me about a time when you wanted to change something that was outside of your regular scope of work.
- Tell me about a time you had to make a fast decision and live with the results.

Example Responses:

- Junior: A story about a change they proactively suggested and drove that had an impact on their team's focus area. Usually only requiring the candidate themselves to work on.
- Senior: A story about a change they proactively suggested and drove that had an impact on their entire team. Usually requiring three or more people to work on.
- Staff: A story about a change they proactively suggested and drove that had an impact on their entire org. Usually requiring two or more teams to work on.

Ability to work in an unstructured environment

Example Questions:

- How do you decide what to work on next?
- Tell me about a project or task that was ambiguous or underspecified.

Example Responses:

- Junior: A story about an ambiguous task that the candidate took ownership of and was able to drive consensus from a few stakeholders in their team. Usually only requiring the candidate themselves to work on.
- Senior: A story about an ambiguous project that the candidate took ownership of and was able to drive consensus from stakeholders in their team or org. Usually requiring three or more people to work on.
- Staff: A story about an ambiguous project that the candidate took ownership of and was able to drive consensus from stakeholders in their org. Usually requiring two or more teams to work on.

Perseverance

Example Questions:

- Tell me about a time when you needed to overcome external obstacles to complete a task or project.
- Tell me about a time a project took longer as expected

Example Responses:

- Junior: A story about a task with many technical difficulties and how they overcame each blocker.
- Senior: A story about a project with many technical difficulties that were blocking their team and how they overcame each blocker.
- Staff: A story about a project with many technical difficulties that were blocking many teams and how they overcame each blocker.

Conflict resolution and empathy

Example Questions:

- Tell me about a person or team who you found the most challenging to work with.
- Tell me about a time you disagreed with a coworker.
- Tell me about a situation where two teams couldn't agree on a path forward.

Example Responses:

- Junior: A story about how they were able to work through a disagreement with a coworker on an implementation detail of a larger project.
- Senior: A story about how they were able to work through a disagreement with a few coworkers or team leads on the direction of a larger project.
- Staff: A story about how they were able to work through a disagreement with two or more teams on the direction of a large project.

Growth

Example Questions:

- Describe a situation when you made a mistake, and what you learned from it.
- Tell me about some constructive feedback you received from a manager or a peer
- Tell me about a skill set that you observed in a peer or mentor that you want to develop in the next six months.

Example Responses:

- Junior: A story about a new technology they want to learn and the progress they have made to learn it.
- Senior: A story about a soft skill or technical skill they want to develop and the progress they have made to learn it. Usually a skill that will have the potential to affect the entire team.

- Staff: A story about a soft skill or technical skill they want to develop and the progress they have made to learn it. Usually a skill that will have the potential to affect two or more teams.

Communication

- Generally covered during the interview as to how clearly they are explaining the stories. There is also some overlap with Empathy and how they communicate with others.

C. Software Engineer interviews: Crafting the perfect self introduction

"Tell me about yourself" or "give me a quick introduction of your profile" is almost always the first question encountered in your software engineer interviews. This guide teaches you how to maximize this chance to impress the interviewer by crafting the perfect self introduction.

Interviewers want to work with candidates they like. Leave a good/deep impression and it will increase your chances of success. Most of us are not strangers to self introductions as we meet new people now and then and have to introduce ourselves every once in a while. However, self introductions in interviews are slightly different from real life - you need to tweak it to your advantage - tailor the self introduction to the role and company you are applying for! Your self introduction evolves as you grow and are at a different stage of your career.

When answering "tell me about yourself", you can rephrase the question into: "Tell me about your journey into tech. How did you get interested in coding, and why was web development (or replace with other job-specific skills) a good fit for you? How is that applicable to our role or company goals?" It is probably not a good idea to spend valuable time talking about things which aren't relevant to the job!

Make an elevator pitch

An "elevator pitch" originates from a journalist trying to pitch an idea to an editor. The only time to catch the editor was in the elevator and she had only around 30 seconds to do so. The key elements of elevator pitches include:

- Short - You have limited time!
- Direct - As you only have limited time, you should get to the point
- Attention-grabbing - Present your most attractive ideas

Whether you're at a job fair with hundreds of other candidates and you have limited time or you are simply explaining who you are to a potential connection or client, it is important to be able to clearly and accurately describe your knowledge and skills quickly and succinctly. Your self introduction is an elevator pitch for yourself!

Here are some tips to develop a good elevator pitch for your software engineer self introduction:

1. Start with basic background information

Include who you are, who you work for (or school and major), and what you do.

- Internships - You should mention the following: name, school and major, focus areas, past internships and/or noteworthy projects
- Full-time - You should mention the following: name, past companies, noteworthy projects (best if it's a public consumer product that they would have heard of)

Does this look familiar? It should be, because it is similar to your resume! Your resume is a condensed version of your knowledge and experiences and your self introduction is essentially a condensed version of your resume. As you grow older, professional experience becomes more important and school background becomes less important. Hence your self introduction changes as you become more senior.

2. KISS (Keep It Simple and Sweet)

Tell them some highlights from your favorite/most impressive projects and including some numbers if they're impressive or challenges that you've overcome. Do not delve into the depths of how you reverse engineered a game and decrypted a packet to predict when to use your DKP on a drop. Tell them the executive summary: "I reverse engineered X game by decrypting Y packet to predict Z." If this catches their interest, they might ask further questions on their own.

3. Why do they want you?

Tell the interviewer why you would make a good hire. Is your experience relevant to the company? Have you used a similar tech stack as the company or built relevant products? What unique talent(s) do you have that may give them confidence about your ability to contribute to the company?

4. Practice!

Lastly, you must practice your pitch! Having a great, succinct summary of your skills only helps if you can actually deliver it rapidly! You should practice keeping a quick but easy-to-follow pace that won't overwhelm them but won't bore them. It's a precarious balance, but can be ironed out with practice.

After coming up with your self introduction, keep it somewhere where you can refer/tweak in future. Memorize them and in future you can just use it when you need to but don't sound like you're recalling it from your memory when you're actually saying it out. Sound natural!

Having an elevator pitch on hand is a great way to create a network and chance upon new job opportunities. There will often be times when you can't prepare for an interview or meeting and it is incredibly handy to have a practiced pitch.

Good examples of software engineer self introductions

Example 1: Front End Engineer at Meta

Self introduction

"Hi I'm XXX and I graduated from National University of Singapore in 2015 with a degree in Computer Science. My interests are in Front End Engineering and I love to create beautiful and performant products with delightful user experiences.

Back in school, I designed and built a web application, NUSMods which solves a huge problem of class and timetable planning every semester. It receives over a million pageviews a month and is used by over 40,000 NUS students and even some professors. It is built using a modern web technology stack - React, Redux, Jest, Babel, Flow, webpack and is mobile-responsive."

I'm interested in the Front End Engineer role at Meta because I have been using Meta Open Source Front End technologies for a while now and am inspired by Meta's mission and Open Source culture.

Breakdown

"I love to create beautiful and performant products with delightful user experiences."

Qualities that a Front End engineer should possess.

"It receives over a million pageviews a month and is used by over 30,000 NUS undergraduates and even some professors."

Mention something about the project which stands out.

"It is built using a modern web technology stack - React, Redux, Jest, Babel, Flow, webpack and is mobile-responsive."

Meta tech stack! Also hints that you keep yourself updated with modern web technologies.

Example 2: Front End Engineer at Lyft

Self introduction

"Hi I'm XXX and I graduated from National University of Singapore in 2015 with a degree in Computer Science. My interests are in Front End Engineering and I love to create beautiful performant products with delightful user experiences.

I previously worked at Grab where I led the Grab for Work project. Grab for Work was a service for companies to make corporate transportation expenses convenient. Companies can create employee groups, set ride policies and share corporate payment methods with their employees. I built the project with another engineer over the period of 3 months on a React/Redux and Golang stack."

I'm interested in the Front End Engineer role at Lyft because I like working in this ridesharing space and creating products to improve the lives of users.

Breakdown

"I love to create beautiful and performant products with delightful user experiences."

Same as above, qualities that a Front End engineer should possess.

"I previously worked at Grab where I led the Grab for Work project."

Lyft was Grab's sister company! In fact they even had a partnership in the past. Most Lyft engineers would have heard of Grab before and mentioning this catches their attention.

"I built the project with another engineer over the period of 4 months on a React/Redux and Golang stack."

Acknowledge that you work with others. Building a non-trivial system with just 2 people in 3 months is quite good for a non-trivial system. Lyft also uses Golang for their high performance systems.

D. Software Engineer interviews: Best end of interview questions to ask

Something you can always count on to happen at the end of your Software Engineer interview - both technical and non-technical rounds - is for the interviewer to ask you if you "have any final questions?".

This question actually isn't a real question at all - candidates are generally expected to ask questions. As an interviewer myself, candidates who don't have any questions might come off as less interested in the role.

Besides that, the questions you ask reveal what you care about. If asked well, this can be a very predictable opportunity for you to leave a good impression while also knowing more about the role (including uncovering potential red flags).

Here, I present questions to ask at the end of your software engineer interviews, for every purpose. The ones in bold are the ones that tend to make the interviewer go "That's a good question" and pause and think for a bit.

Best questions to ask for knowing more about technical work

- What are the engineering challenges that the company/team is facing?
- What has been the worst technical blunder that has happened in the recent past? How did you guys deal with it? What changes were implemented afterwards to make sure it didn't happen again?
- What is the most costly technical decision made early on that the company is living with now?
- What is the most fulfilling/exciting/technically complex project that you've worked on here so far?
- I do/don't have experience in domain X. How important is this for me to be able to succeed?
- How do you evaluate new technologies? Who makes the final decisions?
- How do you know what to work on each day?
- How would you describe your engineering culture?
- How has your role changed since joining the company?
- What is your stack? What is the rationale for/story behind this specific stack?
- Do you tend to roll your own solutions more often or rely on third party tools? What's the rationale in a specific case?

- How does the engineering team balance resources between feature requests and engineering maintenance?
- What do you measure? What are your most important product metrics?
- How often have you moved teams? What made you join the team you're on right now? If you wanted to move teams, what would need to happen?
- What resources does the company have for new hires to study its product and processes? Are there specifications, requirements, documentation?
- How do you think my expertise would be relevant to this team? What unique value can I add?

Best questions to ask for knowing more about the role

- What qualities do you look out for when hiring for this role?
- What would be the most important problem you would want me to solve if I joined your team?
- What does a typical day look like in this role?
- What are the strengths and weaknesses of the current team? What is being done to improve upon the weaknesses?
- What resources does the company have for new hires to study its product and processes? Are there specifications, requirements, documentation?
- What would I work on if I joined this team and who would I work most closely with?

Best questions to ask for knowing more culture and welfare

- What is the most frustrating part about working here?
- What is unique about working at this company that you have not experienced elsewhere?
- What is something you wish were different about your job?

- How is individual performance measured?
- What do you like about working here?
- What is your policy on working from home/remotely?
- What does the company do to nurture and train its employees?
- Does the company culture encourage entrepreneurship and creativity? Could you give me any specific examples?

Best questions to ask to know more about team leadership or management

These questions are suitable for asking Engineering Managers or senior level management, such as CEO, CTO, VPs and are especially useful for the Team Matching phase of Google interviews or post-offer calls that your recruiters set up with the various team managers.

- How do you train/ramp up engineers who are new to the team?
- What does success look like for your team/project?
- What are the strengths and weaknesses of the current team? What is being done to improve upon the weaknesses?
- Can you tell me about a time you resolved an interpersonal conflict?
- How did you become a manager?
- How do your engineers know what to work on each day?
- What is your team's biggest challenge right now?
- How do you measure individual performance?
- How often are 1:1s conducted?
- What is the current team composition like?
- What opportunities are available to switch roles? How does this work?
- Two senior team members disagree over a technical issue. How do you handle it?
- Have you managed a poor performer at some point in your career before? What did you do and how did it work?
- Where do you spend more of your time, high performers or low performers?

- Sometimes there's a trade-off between what's best for one of your team members and what's best for the team. Give an example of how you handled this and why.
- Give an example of a time you faced a difficult mentoring/coaching challenge. What did you do and why?
- What is your management philosophy?
- What is the role of data and metrics in managing a team like ours?
- What role does the manager play in making technical decisions?
- What is an example of a change you have made in the team that improved the team?
- What would be the most important problem you would want me to solve if I joined your team?
- What opportunities for growth will your team provide?
- What would I work on if I joined this team and who would I work most closely with?

Best questions to ask to know more about company direction

- How does the company decide on what to work on next?
- What assurance do you have that this company will be successful?
- Which companies are your main competitors and what differentiates your company?
- What are your highest priorities right now? For example, new features, new products, solidifying existing code, reducing operations overhead?

E. Data Related Technical Interview

- What are SQL Databases and NoSQL Databases
- Difference between SQL and NoSQL Databases
- SQL v.s. NoSQL: Which is more scalable? (Scalability)
- SQL v.s. NoSQL: Which is faster? (Performance)
- SQL v.s. NoSQL: Which is more secure?
- Pros and Cons of SQL and NoSQL

- Use cases for SQL and NoSQL: When to Use Which? (with real-world examples)
- Can you explain A/B Testing cases
- What is z score? On which case you implement it

D. Engineer Related Technical Interview

- What happens when you click on a URL in your browser