Technical Interview Prep Guide

 Lambda

This document contains resources to help you prepare for technical interviews. In addition to more in-depth technical preparation, you can also find a recap of behavioral interview prep. Behavioral interviewing techniques are important in any interview setting, since you will never be evaluated solely on your technical skills for a job!

This guide has a lot of information, so don't try to use it all in one sitting. Go through the resources in this and practice even when you don't have an interview scheduled. This will help you break your preparation up into bite-sized pieces. We recommend spending 30 minutes a day reviewing technical challenges when you're getting ready to start applying to jobs, that way you're ready even if an interview is scheduled with minimal time to prepare!

Prepare to talk about your projects:
1. List out every project you've worked on at Lambda School, plus any side projects.
2. For each project, complete the following information (note you only have to do this once, but put it in writing and save it somewhere easily accessed):
   - What was the objective or goal of your project in 2-3 sentences?
   - Did you accomplish your agreed-upon objective or goal? If not, what was the barrier to successful completion?
   - What did you specifically work on within the project?
   - How many people did you collaborate with?
   - What did you enjoy most about the project?
   - What did you find most technically challenging?
   - What is one example of positive collaboration within the project?
   - What is one example of challenging collaboration within the project?
   - What is functional about your project, and what is still a work in progress?
   - What would v.2 look like? What features are incomplete or don't meet your standards?
   - If you were to do this project again, what would you do differently?
3. Then, any time you have an interview, do the following (note you should do this for every interview):
   - Go through each line of the job description- both the "responsibilities" section and the "requirements" section.

Technical Interview Prep Guide

Λ Lambda

- For each responsibility or requirement, jot down a relatable project or experience that you can use as an example in an interview. It's ok to think outside the box here- you likely have more relatable experience than it initially may seem!
- If you do truly do not have experience with a specific responsibility or requirement, research it so you can at least talk about it. Think about how you would have approached one of your existing projects differently using this tool, software, process, etc.

Start with: (Zety)
- ❏ [Common Job Interview Questions & Answers (Top 35 Samples for 2019)](#)

Next, move on to behavioral questions: (Zety)
- ❏ [STAR Method for Acing Behavioral Interview Questions](#)

Finally, move on to specific answers: (Zety)
- ❏ *[How to Describe Your Current Job Responsibilities](#)*
- ❏ *[Tell Me About Yourself](#)*
- ❏ *[What Are Your Strengths?](#)*
- ❏ *[What Is Your Greatest Weakness?](#)*
- ❏ *[What Is Your Management Style?](#)*
- ❏ *[Where Do You See Yourself in 5 Years?](#)*
- ❏ *[Why Did You Leave Your Last Job?](#)*
- ❏ *[Why Do You Want to Work Here?](#)*
- ❏ *[Why Should We Hire You?](#)*

Question Sets (Counter-Interview)
- ❏ [10 Questions You Should Ask In a Web Dev Interview](#)
- ❏ [The Joel Test - 12 Steps to Better Code](#)
- ❏ [8 Question, to fit candidate into the team.](#)
- ❏ [9 Questions from my past 4-5 interviews.](#)
- ❏ [Some questions I like to ask interviewers.](#)

[The Ultimate Guide to Acing Your Technical Interview](#)
- ❏ [What is a Technical Interview?](#)
  - ❏ [Tech Interview Basics](#)

Technical Interview Prep Guide

Λ Lambda

YouTube Resources: Whiteboarding
- ❏ [How to Ace the Whiteboard Coding Interview](#)
- ❏ [50 Practice Questions for your Coding Interview](#)
- ❏ [Death of a Whiteboard: How to crush any technical interview](#)
- ❏ [Everything you need to know to rock your next whiteboard test](#)
- ❏ [The Best Whiteboard Interview Advice I Ever Received](#)
- ❏ [The ultimate guide to preparing for the coding interview](#)
    - ❏ Task #1: Buy Gayle Laakmann McDowell's "Cracking the Code Interview"
    - ❏ Task #2: Read the section VII of the Introduction chapter of the "Cracking the Coding Interview" book: Technical Questions.
    - ❏ Task #3: As a refresher of data structures and algorithms, complete ONLY the first coding challenge in each section (i.e. Array, Dynamic Programming, Geometry, Graph, Hash, etc.) of [this page](#) using your preferred language.
    - ❏ Task #4: Find resources to test yourself against the most common conceptual questions asked about the different technologies that you know. If you don't know the answer to a question, go ahead and use Google to learn about it before trying to see the solution.
    - ❏ System Design Questions
        - ❏ Task #5: Read chapter 9 of the "Cracking the Coding Interview" book: System Design and Scalability.
        - ❏ Task #6: Donne Martin, engineering manager at Facebook, has created another awesome open source resource to help people with their interviews. Go ahead and read the section called "[How to approach a system design interview question](#)".
        - ❏ ***Task #7****: Read about the four different ways of scaling a system:*
            - ❏ *[Scalability for Dummies - Part 1: Clones](#)*
            - ❏ *[Scalability for Dummies - Part 2: Database](#)*
            - ❏ *[Scalability for Dummies - Part 3: Cache](#)*

❏ *Scalability for Dummies - Part 4: Asynchronism*
❏ **Task #8**: *Read about the different components of every software system (you don't need to understand everything in detail, but you should be aware of the different components and concepts):*
- ❏ *Domain Name System (DNS)*
- ❏ *Content Delivery Network (CDN)*
- ❏ *Load Balancer*
- ❏ *Reverse Proxy*
- ❏ *Application layer*
- ❏ *Database*
- ❏ *Cache*
- ❏ *Asynchronism*
- ❏ *Communication*
❏ **Task #9**: *Pick 2 - 3 companies that you like from this list and read about their architectures.*
❏ **Task #10**: *Look at the answers to the following two design questions:*
❏ *URL Shortener*
❏ *Collaborative Editor*
❏ **Task #11**: *Finally, pick any 2 questions from this list of design questions and, using pen and paper, or any text editor, try to answer to them as if you were in a real interview. I recommend that you try doing this with a friend or colleague so you can help each other.*
❏ *Behavioral Questions*
- ❏ Task #12: Read section V of the Introduction of the "Cracking the Coding Interview" book: Behavioral Questions.
- ❏ Task #13: Using a Google Docs document, complete the "Interview Preparation Grid" described in the book. It's ideal if you can use mostly software projects, but if there is any behavioral situation (e.g. Leadership) where you cannot think of a software project, use a non-software project to fill in the grid.
- ❏ *Task #14: Following the S.A.R. (Situation, Action, Result) methodology and the general tips from the book, write down and add the answers to questions from the book (i.e. "weaknesses" and "tell me about yourself"), and to each one of the question in the "General" section of this list to the Google Doc from the previous task.*
❏ *Questions to Ask*

Technical Interview Prep Guide

Λ Lambda

❏ *Task #15: Take a look at the questions from [this list](#) and use them as a reference for your future interviews. There is no need to memorize them now.*

More
   ❏ [The Best Resources to Ace your Full Stack JavaScript Interview](#)
   ❏ [8 Must Use Resources for Crushing Your Coding Interview](#)
   ❏ [Top 50 React Interview Questions](#)
   ❏ [A Collective List of APIs, Go Build Something](#)

Lambda WhiteBoarding Recordings
   ❏ [Product of All Other Numbers](#)
   ❏ [Magic Squares](#)
   ❏ [Allocating Budget Cuts](#)
   ❏ [Advent of Code](#)
   ❏ [Merging M Sorted Lists](#)
   ❏ [City Distances](#)
   ❏ [Tree Visibility](#)
   ❏ [Soldier Ranks](#)
   ❏ [Bigger is Greater](#)
   ❏ [River Sizes](#)