Interview Prep Cheatsheet for Software Jobs

Subtitle: A practical, no-nonsense guide to crack developer interviews as a beginner.

Website Name: haas.dev
Website Link: https://dev-roast-app.vercel.app

Introduction

Most CS students don’t fail interviews because they “don’t know enough.” They fail because they prepare randomly. This cheatsheet gives you a **focused, interview-first strategy** so you know exactly what to study, practice, and revise before applying for software roles.

Section 1: Interview Structure (Know the Game First)

Most software interviews test four things:

1. **Coding & Problem Solving**
2. **CS Fundamentals**
3. **Projects & Practical Skills**
4. **Communication & Thinking Process**

If you prepare only coding questions, you’re already behind.

Section 2: Coding & DSA – What Actually Matters

Must-Know Topics

- Arrays & Strings
- Hash Maps / Dictionaries
- Stacks & Queues
- Basic Recursion
- Sorting & Searching
- Time & Space Complexity (Big-O)

What NOT to Overdo

- Advanced graph theory
- Extremely hard LeetCode problems

- Extremely hard LeetCode problems
- Rare algorithms you can't explain

Rule:
👉 Solve **easy → medium** problems and explain your logic clearly.

Daily Practice Plan

- 1 Easy problem
- 1 Medium problem
- Explain solution out loud (very important)

Section 3: CS Fundamentals (Beginner Level)

You don't need deep theory, but you must understand basics.

Core Topics

- What is an OS?
- How memory works (stack vs heap)
- What is an API?
- HTTP methods (GET, POST, PUT, DELETE)
- Difference between SQL & NoSQL

Mini Exercise:
Explain "How the internet works" in **5 simple sentences**.

If you can't explain it simply, you don't understand it yet.

Section 4: Projects – The Make-or-Break Factor

Interviewers care more about **how you built** a project than how big it is.

Be Ready to Explain:

- Why you built this project
- Tech stack choice
- Challenges you faced
- How you would improve it

Ideal Beginner Projects

- To-Do App with authentication
- Blog platform (CRUD)
- Expense tracker
- Simple REST API

Rule:
👉 2–3 well-explained projects > 10 shallow ones.

Section 5: Common Interview Questions (Prepare These)

- “Explain your project”
- “What happens when you type a URL in the browser?”
- “What is Git and why do we use it?”
- “Difference between == and ===”
- “How do you handle errors?”

Mini Exercise:
Write answers to **10 common questions** in your own words.

Section 6: Behavioral & Communication Skills

Most beginners ignore this — big mistake.

Interviewers Look For:

- Clear thinking
- Honesty (not fake confidence)
- Problem-solving approach

Use This Structure:

1. Understand the problem
2. Explain your approach
3. Write code
4. Optimize

4. Optimize
5. Test with examples

Tip:
If stuck, **talk through your thinking**. Silence kills interviews.

Section 7: 7-Day Final Revision Plan

Day 1–2:

- Revise arrays, strings, hash maps

Day 3:

- Revise CS fundamentals

Day 4:

- Revise projects + README

Day 5:

- Mock interviews

Day 6:

- Behavioral questions

Day 7:

- Light revision + confidence reset

Key Takeaways

- Interviews are a **skill**, not luck
- Prepare **strategically**, not randomly
- Clear explanations beat complex answers
- Projects + communication matter as much as coding
- Consistency > intensity

Visit **haas.dev** **https://dev-roast-app.vercel.app** for more interview-focused guides, roadmaps, and beginner-friendly resources.

Website Name: haas.dev
Website Link: https://dev-roast-app.vercel.app

Website Link: https://dev-roast-app.vercel.app