ZOHO

Software developer role - Cheat Sheet

Hi everyone! ??

I hope this sheet will help you easily clear the interview rounds for a software developer role. I'll explain all the rounds and provide some preparation materials. At the end, I'll also share my own preparation resources too. This documentation will clear all your doubts, I guess. Thank me later.

L1 - Aptitude + Programming Questions
L2 - Programming Round
L3 - Advanced Programming Round
L4 - Technical Interview
L5 - HR Interview

Before diving into the materials, it's important to mention that knowing C++ or Java is a big advantage. If you know Java, that's even better. However, keep in mind that the preferred language can vary depending on the team you'll be working with. Ideally, aim for a 10/10 proficiency in Java and a 9/10 in C++.

Round 1:
In L1, there are no multiple-choice options. Each question is like a fill-in-the-blank. Sometimes, the programming questions are in C++ or Java.

Tips:
For the aptitude section, focus on the following topics:

- Speed and Distance
- Time and Work
- Boats and Streams
- Card, Coin, and Ball-based Questions
- Permutation-based Questions
- Profit and Loss
- Ratio-based Questions

For the programming section, focus on the following topics:

You will often encounter questions based on recursion, string manipulation, and array operations. These questions can consume a lot of time during your L1 round.

One important tip: **focus on understanding the logic.** Don’t waste time tracing the program line by line. Instead, try to grasp the underlying logic and the objective of the problem.

For example, if you see a question involving a while loop that runs nearly 100 times or more, think about how much time it will take you to solve it. This is why it's crucial to quickly identify the logic and purpose of the program—you'll be able to solve it more efficiently.

Round 2:Programming Round

Welcome to the best part of your interview rounds! Problem-solving isn't just a hurdle to clear—it's a skill that will serve you throughout your entire career.

So, instead of just preparing for the problem-solving rounds, make it a habit to solve problems regularly. When you join a company, you'll be solving real-world problems through code based on specific requirements. Those solutions will be known as features.

Materials:  https://www.linkedin.com/in/chandru-s-5a60a4275/

For preparation, just log in to LeetCode. You'll find curated lists, like "150 Questions for Interviews" or the "75 Problems" card. These two sets are enough to clear Round 2. Also, try participating in CodeChef contests to help you manage your time effectively during Round 2.

If you're new to problem-solving, don't worry about the time it takes at first. I spent 5 hours on easy-level problems when I was starting out. Focus on grasping the logic and take your time initially. The only issue is if, after a month, you're still taking 5 hours to solve easy problems—that's where improvement is needed.

There are a lot of algorithms and data structures that can help you solve problems, but remember, you don't need to memorize everything. The key is to explore different approaches so that during the interview, you can quickly identify the logic within the

---

Round 3 : Advanced Programming Round

Welcome to another exciting round! Guess what? If you perform well in this round, you won’t need to speak much during the technical interview. Why? Because this is an application

development round. Your performance here can speak volumes about your skills. However, this round can vary based on the team:

- Some teams may ask for challenging programming problems.
- Others might focus on logical console applications.
- Some might expect you to work with files, JSON, and data structures/algorithms (DSA).

How to Prepare:

Why did I say that performing well in this round means you won't need to prove your skills again in L4? Let me explain. Suppose you're asked to develop a shopping cart application. How would you approach it? Many might create 8-10 classes, store values in every class, and use a lot of static variables. But imagine if you did the following instead:

- Follow design patterns.
- Use package separation.
- Implement singleton classes.
- Write clean, well-structured code.

If you do these things, the interviewer will be impressed with your work in L3.

What Should You Learn?

- **Design Patterns:** In L3, always use the MVVM (Model-View-ViewModel) pattern. It's efficient and less time-consuming compared to other patterns.
- **Singleton Classes:** A singleton class is like a security guard at a mall with five entrances, where one guard handles all the entrances. If you want to access any properties in that class from other classes, you must go through that one "security guard." Explore more about singletons to understand their importance.
- **Write Reusable Code:** Always aim to write code that can be reused across different parts of your application.
- **Think Before You Create Variables:** Before creating a variable, think 3-5 times about whether you can handle the flow without it.
- **Centralized Data Management:** Don't store data in every class. Have a dedicated database class to manage all the necessary data for your entire application.
- **Use the Right Data Structures:** This is crucial. Be aware of DSA and choose the appropriate data structures for your application.

Example Questions:

- Shopping Cart Application
- Lift System
- Taxi Booking System
- ATM Process
- Snake and Ladder Game
- Sudoku Game
- Tetris Game
- Mail Server
- Invoice Management
- Scheduler-Based Application

Now, do you have time to develop all these console applications? No, and that's not the point. I'm not suggesting you develop all these specific applications. What I'm trying to convey is that you should be well-prepared for application-based questions.

If I mention something as simple as a "broomstick," you should be able to list out the features for that and think about how to develop it as a console app. For example, a broomstick could be made with plastic or iron, have different heights, weights, weather resistance, handgrips, and colors.

We even have automatic cleaning robots now! With these considerations, you could develop an application where you receive user requirements, showcase categories, and allow for custom modifications.

Just imagine—if you can come up with an application based on a simple broomstick, think about the questions you might face in L3. That's why I emphasize sharpening your mind to handle application development challenges.

Use all the tips I mentioned earlier. When you're developing a console application, remember that it's not just code for a simple console app; it's code that's meant to solve a real-world problem.

Round 4 : Technical round

For this round, I can’t teach you everything in detail, but I might create a YouTube playlist in the future. In the meantime, you already have plenty of resources available online. I highly recommend checking out Kunal Kushwaha's channel, particularly his playlist "JAVA + DSA + Interview Preparation," which I followed during my own preparation. Of course, there are many resources out there, so feel free to choose what works best for you

Java is like an ocean—there’s so much to learn that you can never truly master it all. Instead, focus on learning what you need. Prepare core concepts thoroughly and have a deep understanding of the basics. For example, creating a simple int CONST = 0 could lead to nearly 30 questions about variable creation and related concepts, such as memory management.

Focus on the following areas:

- **Core Concepts:** Learn deeply about classes, objects, encapsulation, inheritance, constructors, polymorphism, abstract classes and methods, interfaces, packages, exception handling, arrays, strings, Java I/O, multithreading, collections, lambda expressions, JDBC, etc.
- **Additional Topics:**
  - JDBC
  - Servlets
  - Struts 6
  - Front-end technologies like HTML, CSS, JavaScript
  - Networking
  - Database languages: SQL

Depth of Knowledge: Understand that even a single concept, like a variable, can lead to numerous questions. So, the more you delve into each concept, the better prepared you’ll be.

Think of Java as your "girlfriend." Give high priority to it, invest time in learning, and get to know it deeply. Once you’re committed, start your preparation with enthusiasm!

Round 5: HR Interview

Congratulations! If you've reached L5, you're almost there. However, how you handle this round is crucial. The interviewer will assess your mindset and fit for the company culture.

In this round:

- **Behavior and Attitude:** The interviewer will gauge your personality and how well you align with the company's values. Initially, they might ask about what type of person you are and then pose questions to see if your answers match your self-description.
- **Be Aware and Genuine:** Pay attention to what you say and how you respond. Aim to be humble, keep a smile on your face, and provide realistic answers.
- **Avoid Prepared Responses:** For L5, don't rely on pre-prepared answers. Be authentic and speak from the heart about why you want the job and how you navigated through the interview rounds.

Thats it. Let me explain how I prepared :

1. Start with Java:

- **Learn the Basics:** Begin with core Java concepts.
- **Deep Dive:** Once you're comfortable, study advanced topics thoroughly.
- **Full-Stack Development:** Aim to become proficient in full-stack development using Java.

2. Front-End Development:

- **Learn HTML, CSS, and JavaScript:** Start with these technologies and create good projects to build your front-end skills.

3. Java and DSA:

- **Java + DSA Playlist:** Check out the "Java + DSA + Interview Preparation" playlist by Kunal Kushwaha on YouTube (approximately 65 videos). Cover all the content to build a strong foundation.
- **Create Console Applications:** Practice by building console applications in Java, and use file storage systems to manage data.

4. Database Concepts:

- **Learn MySQL:** Understand the basics of MySQL for database management.
- **Integrate MySQL:** Create Java projects using MySQL for data storage.
- **Explore Redis:** Learn about Redis, experiment with it, and understand how it works.

5. Server-Side Programming:

- **Servlets and Struts:** Learn about servlets and the Struts framework. If possible, study Struts with the Tiles concept.
- **Dynamic Web Projects:** Learn to create dynamic web projects using Java.

6. Full-Stack Project:

- **Develop a Full-Stack Application:** Combine your front-end skills, Java for back-end development, and MySQL for data storage. Connect the front end and back end using server-side programming.

7. Preferred IDE:

- **Eclipse Java EE (Web Version):** Use Eclipse for web development projects.

8. Advanced Java:

- **Collections:** Study Java collections deeply. Implement your own versions of collections by referring to Java docs and libraries (e.g., ArrayList).
- **Design Patterns:** Learn and apply design patterns, such as Singleton, while developing applications.

9. Interview Preparation:

- **Aptitude:** Practice aptitude questions. Although specific websites aren't mentioned, focus on becoming proficient in this area.
- **Problem Solving:** Master problem-solving skills by practicing on LeetCode. Start with easy-level problems and solve 2-3 problems daily.
- **Concepts Review:** Ensure you remember all key concepts to clear technical rounds.
- **Design Patterns:** Continue learning and applying design patterns in your projects.

Additional Technologies:

- **Redis**
- **Tiles**

Summary:

- **HTML, CSS, JS**
- **Java**
- **Servlets**
- **Struts**
- **MySQL**
- **DSA**
- **Problem Solving**
- **Design Patterns**
- **Git**

Congratulations! You've made it to the 8th page—an impressive achievement. Your patience and dedication in reading through all this material demonstrate that you have what it takes to achieve your goals.

Key Takeaways:

- **Be Proud:** Your perseverance is a strong indicator of your potential.
- **Don't Rush:** Take your time to thoroughly prepare and understand each concept.
- **Stand Out:** Remember, in interviews, you might be one among thousands of candidates. To stand out, you need to work hard and differentiate y≠ourself.

Effort Over Luck : Trust in your efforts. Your hard work will pave the way for what might seem like luck.

Keep pushing forward, stay dedicated, and success will follow!

✨With loves this is **CHANDRU**. Wishing you to become the beast version of yourself

Linkedin Profile : https://www.linkedin.com/in/chandru-s-5a60a4275/

@THZISCHAN