Disclaimer: *This is my personal preparation guide, but remember, the best path to success is the one that works for you. Explore and tailor your approach based on your learning style and goals. I'm simply sharing resources that I found valuable, along with others that could be a great fit for your journey to cracking interviews at top tech companies.* 🤗

Linkedin Series:

Interview Preparation for Software and Test Engineer: Interview Prep Series

Exploring Different Types of System Design: **Exploring System Design**

Mid-Career Professionals: Learnings for Mid-Career

Problem Solving Best Practices: Problem Solving Best practices

Overcome Fear of Rejection and Journey to Success: Overcome Rejection and Path to success

DSA resources: ➕ **DSA + System Design Learning Resources**

System Design Resources: ➕ **DSA + System Design Learning Resources**

Platforms:

LeetCode: This platform is a goldmine of coding problems, perfect for honing your problem-solving skills and algorithm knowledge.

GeeksforGeeks: This comprehensive website offers in-depth tutorials and articles on a wide range of computer science topics, including data structures, algorithms, and system design.

YouTube: Explore the numerous educational channels that offer comprehensive courses and tutorials on coding interview preparation.

What To Learn?

Fundamental Computer Science Knowledge 💻**:**

- **Data Structures and Algorithms** 💻**:** Understanding fundamental data structures (arrays, linked lists, trees, graphs, etc.) and algorithms (sorting, searching, dynamic programming, etc.) is crucial for designing efficient solutions.
- **Object-Oriented Programming (OOP)** 📦**:** Proficiency in OOP concepts like encapsulation, inheritance, and polymorphism is essential for building modular and maintainable code.

- **Design Patterns** ✍️**:** Familiarity with common design patterns (e.g., Singleton, Factory, Observer) helps in creating scalable and reusable software architectures.
- **Operating Systems** 🖥️**:** Understanding operating system concepts like processes, threads, memory management, and file systems is important for optimizing performance and debugging issues.
- **Networking** 📶**:** Knowledge of networking protocols (TCP/IP, HTTP, etc.) and concepts like routing, load balancing, and security is valuable for developing distributed systems.
- **Database Systems** 🛢️**:** Understanding relational and NoSQL databases, including their design principles, query languages, and performance optimization techniques, is essential for data-driven applications.

Programming Languages and Frameworks 🔤✍️**:**

- **Proficiency in one or more languages** ✍️**:** Learn languages like C++, Java, Python, Go, or JavaScript. Expertise in at least one of these languages is crucial.
- **Web Technologies** 🌐**:** Understanding HTML, CSS, and JavaScript is essential for front-end development, while knowledge of backend frameworks like Django, Flask, or Node.js is beneficial for building web applications.
- **Cloud Computing** 💭**:** Familiarity with cloud platforms like Google Cloud Platform (GCP) or Amazon Web Services (AWS) is becoming increasingly important for building scalable and reliable systems.

Problem-Solving and Critical Thinking 🤓🤔**:**

- **Analytical Skills** 🙋**:** The ability to break down complex problems into smaller components and devise effective solutions is essential.
- **Debugging Skills** 🧐**:** Proficiency in identifying and fixing errors in code is crucial for ensuring software quality.
- **Algorithm Design** 🤓**:** The ability to design efficient algorithms to solve real-world problems is a key differentiator.
- **Testing and Validation** 💡 **:** Understanding different testing methodologies and writing robust unit, integration, and system tests are essential for ensuring software reliability.

Additional Skills 🆕➕**:**

- **Machine Learning and AI** 💻✨**:** Knowledge of machine learning algorithms and tools like TensorFlow can be advantageous for building intelligent applications.

- **System Design** 💻 ✍️**:** Understanding large-scale system design principles and tradeoffs is valuable for designing scalable and resilient systems.
- **Distributed Systems** 🛢️ **:** Familiarity with concepts like concurrency, consistency, and fault tolerance is essential for building distributed applications.
- **Security** 🔐 **:** Awareness of security best practices and common vulnerabilities is important for protecting systems and data.

Soft Skills 🗣️👋🙋‍♂️**:**

- **Communication** 🗣️**:** Effective communication skills are essential for collaborating with colleagues, understanding requirements, and presenting ideas clearly.
- **Teamwork** 🤝🧑‍🤝‍🧑**:** The ability to work well in a team, collaborate effectively, and share knowledge with others is crucial.
- **Adaptability** 💫**:** Being able to adapt to new technologies and changing project requirements is essential in a fast-paced environment.
- **Time Management** ⏳⏱️**:** Prioritizing tasks, managing your time effectively, and meeting deadlines are crucial for project success

Software Engineer Interview Guide

To empower your journey towards becoming a Software Engineer at FAANG/MAANG companies, I'm thrilled to share some invaluable resources that have been instrumental in my own preparation:

Here are some invaluable resources I recommend:

✅ **Coding Interview**
👩‍💻Supercharge your DSA and Algorithms review from here Learn Data Structures and Algorithms.
👩‍💻Access a curated list of top LeetCode Questions compiled in a convenient list.

✅ **System Design Interview**

👩‍💻 Follow Gaurav Sen's System Design for Beginners.

👩‍💻 Dive deep System Design Interview from Grokking the System Design Interview for expert insights and real-world case studies!
👩‍💻 Subscribe the ByteByteGo Newsletter or enroll in this System Design course.
👩‍💻 Read System Design Interview Questions and Answers and get interview ready.

✅ **Low-Level Design Interview**
👩‍💻 Sharpen your design skills with practical examples from real-world examples from Grokking the Object Oriented Design!
👩‍💻Prepare for Low-Level Design(LLD) Interview from GeeksForGeeks..
👩‍💻 Level up with UML diagram practice on Diagrams.net!

✅ **API Design Interview**
👩‍💻 Master API Design: Practical Tips, Best Practices, and Guidelines from here: API-Design.
👩‍💻Here is your go-to-resource for excelling in API design principles with examples from: Swagger.io.

✅ **Database Design Interview**
👩‍💻 Understand Object Oriented Design with relational mapping from here: Grokking the Object Oriented Design Interview.
👩‍💻Learn designing databases from Database Design in DBMS.

✅ **Behavioral Interview**
👩‍💻Revolutionize your interview approach with Grokking Modern Behavioral Interview, a comprehensive guide to cutting-edge strategies.
👩‍💻Prepare for a behavioral interview by following this Behavioral Interview guide.

Test Engineer Interview Guide

To help you on your path to becoming a Test Engineer at a top tech company (FAANG/MAANG), I'm excited to share some of the resources that were instrumental in my own success:

Here are some invaluable resources I recommend:

✅ **Data Structure and Coding Interview**

👩‍💻Supercharge your DSA and Algorithms review from here Learn Data Structures and Algorithms.
👩‍💻Access a curated list of top LeetCode Questions compiled in a convenient list. Simple to medium problem is a good fit here.

✅ **Testing Interview**
👩‍💻 To kickstart your journey start with Guru99 to build a solid foundation in the testing skills required for this role.
👩‍💻Learn Test Design Coverage from GeeksForGeeks.
👩‍💻Learn Software Testing basics and become a master of this field.
👩‍💻 For a comprehensive course that covers all aspects of test engineering interviews, including coding, system design, and behavioral questions follow this Test Engineering Interview Masterclass (Interview Kickstart).
👩‍💻 Go through the 15 best Software testing course in 2024 which can give you insights about what to learn.
👩‍💻 Subscribe the ByteByteGo Newsletter.

✅ **Behavioral Interview**
👩‍💻Revolutionize your interview approach with Grokking Modern Behavioral Interview, a comprehensive guide to cutting-edge strategies.
👩‍💻Prepare for a behavioral interview by following this Behavioral Interview guide.