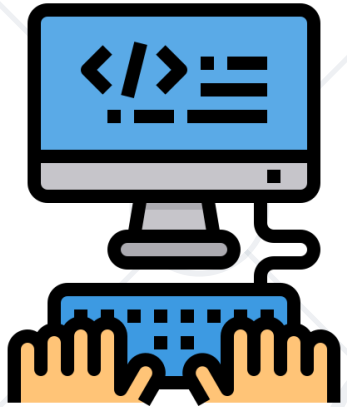


# Mastering the Technical Interview

What to Expect, How to Prepare, and Impress Your Interviewer?



SoftUni Team

Technical Trainers



SoftUni



Software University

<https://softuni.bg>

# Table of Contents

1. Types of Technical Skills Assessment
2. Technical Interview – Sample Questions
3. Skills Measurement Tests
4. Live Coding Challenges
5. Projects as Homework
6. Online Resources





# Types of Technical Skills Assessment

Mastering the Types of Technical Skills Assessment

# Technical Assessment Methods for Hiring

- Interviews
  - Ask a range of **technical** and **non-technical** questions
- Tests
  - Evaluate **knowledge** and **problem-solving** skills
- Live coding challenges
  - Complete coding tasks within a **limited** timeframe
- Take-home projects
  - Create a **project** at home and present it
- Combination
  - A **mix** of assessment methods



# The Range of Questions Asked in Interviews

- Can include general questions about the candidate's **background** and **experience**
- Can include technical questions to assess the candidate's **technical knowledge** and **problem-solving** skills
- Can delve deeper into the candidate's problem solving abilities to test his **decision-making** process through **scenario**-based questions
- Can include questions about the candidate's **motivation** and **goals** for the position



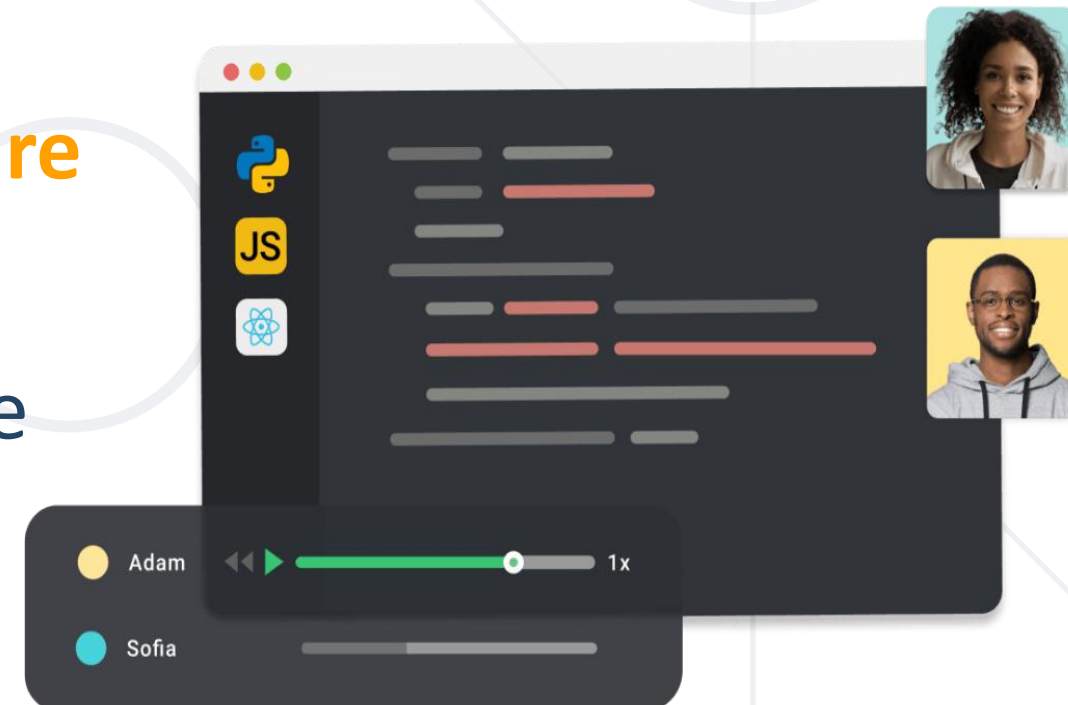
# Tests – Assessing Theoretical Knowledge

- Can be taken **online** or **in person**
- Can be used to assess the candidate's **theoretical** knowledge of a particular **technology** or **programming language**
- Can be different **types** like:
  - Multiple-choice test
  - Fill-in-the-blank test
  - Open-ended question test



# Live-Coding Challenges

- Conducted **in person** or via **video conferencing** (in real time)
- Candidates are given a **coding** problem to solve within a set **time limit**
- Assess problem-solving skills, **pressure handling** and **coding efficiency**
- Real time feedback possible from the **interviewer** or a **panel of judges**





# Take-Home Projects

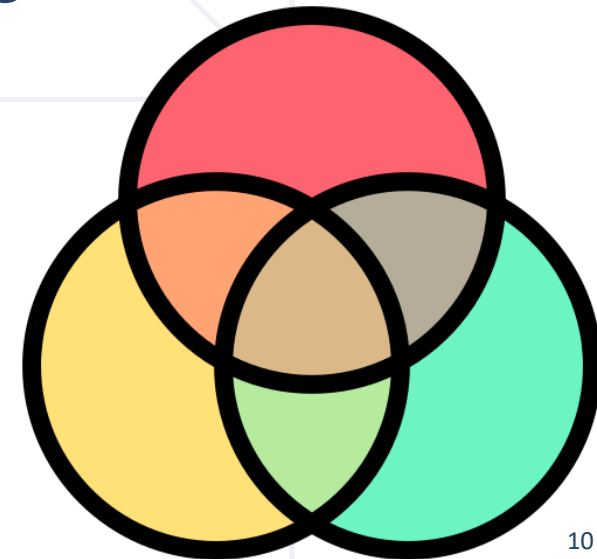
- Candidates are given a **task** or a **project** to complete at home or in their own time
- Can be used to assess the candidate's ability to work **independently**, manage their **time**, and deliver **high-quality** work
- Can be valuable for evaluating the candidate's **creativity**, problem-solving and technical skills





# Combination – A Mix of Assessment Methods

- Sometimes companies **combine** several assessment methods for various reasons
- Allows for a more **well-rounded** view of candidate's strengths and weaknesses
- Provides **multiple opportunities** for candidates to demonstrate their problem-solving and technical skills
- Offers a more accurate assessment of candidate's potential **fit** for the position





# Technical Interview Sample Questions

How to Craft Compelling Responses and Nail the Interview

# Types of Questions During Technical Interviews

- **Algorithmic**

- Data structures, algorithmic thinking, sorting algorithms, searching algorithms, dynamic programming, time and space complexity

- **Conceptual**

- It may include databases, OOP, HTTP protocol, design patterns, software development methodologies, software architecture

- **Technology**

- It may include specific questions about the programming language, web development, mobile development, cloud computing, DevOps

- **Logical** tasks

- Abstract questions like: How many leaves do you think this tree outside the building has?

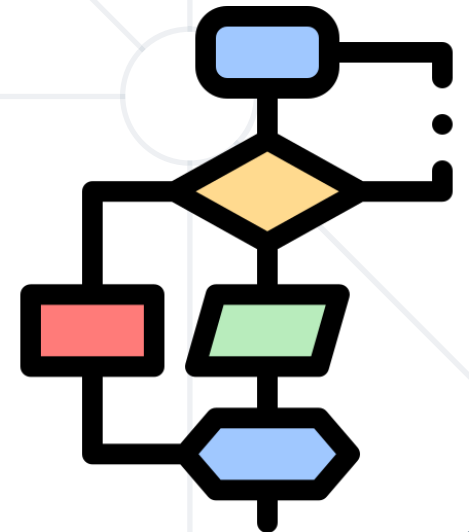


- **Data Structures**

- What is a **linked list** and how is it different from an **array**?
- What is a **hash table** and what are its **advantages** over other data structures?
- What is a **stack** and how is it **used** in programming?

- **Algorithmic Thinking**

- Can you find the **largest** number in an **unsorted array**?
- How would you determine if a given string is a **palindrome**?
- How would you find the second **smallest** number in a **sorted array**?

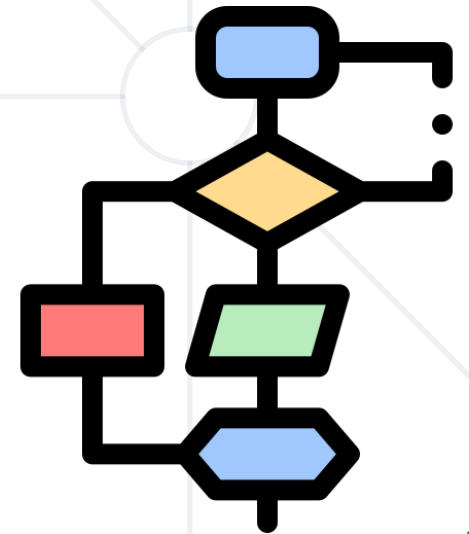


## ■ Searching Algorithms

- What is **linear** search and how does it work?
- What is **binary** search and how does it work?
- What is **interpolation** and how does it work?

## ■ Sorting Algorithms

- What is **bubble** sort and how does it work?
- What is **merge** sort and how does it work?
- What is **quicksort** and how does it work?

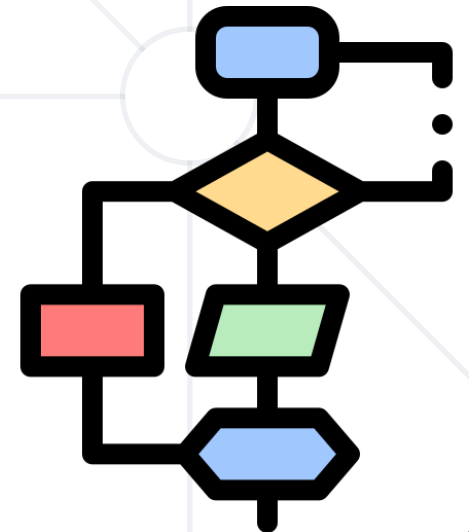


- **Dynamic Programming**

- What is **dynamic programming** and when would you use it?
- How do you determine if a problem can be solved using **dynamic** programming?
- How would you solve the **Fibonacci** sequence using dynamic programming?

- **Time and Space Complexity**

- What is **time complexity** and how is it **measured**?
- What is the **difference** between  $O(1)$ ,  $O(n)$ ,  $O(n^2)$  and  $O(n^2)$ ?
- Can you give **examples** for each time complexity?

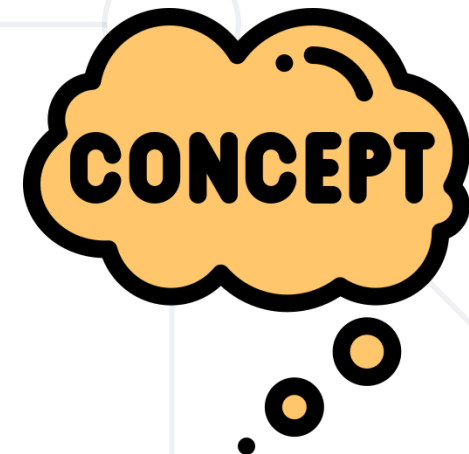


## ■ Databases

- What is the difference between **SQL** and **NoSQL** databases?
- How would you **optimize** a database query to run **faster**?
- What is **indexing** in databases and how does it work?

## ■ Object-Oriented Programming (OOP)

- What is **inheritance** in OOP and how is it useful?
- What is **polymorphism** and how is it implemented in **OOP** languages?
- What is **Encapsulation** and how do we use it?



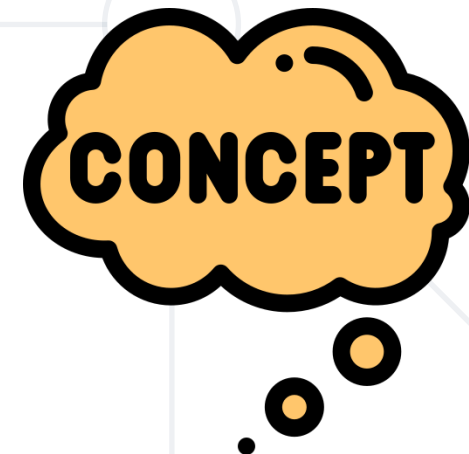


## ■ HTTP Protocol

- What is the difference between **HTTP** and **HTTPS**?
- What are HTTP **status codes** and give an **example** of when each is used?
- What is **caching** in the context of the HTTP protocol and how does it work?

## ■ Design Patterns

- What is the **Singleton** design pattern and when do you use it?
- What is the **Observer** design pattern and how is it useful in software development?
- What is the **Factory** design pattern and how does it differ from the **Abstract Factory** pattern?

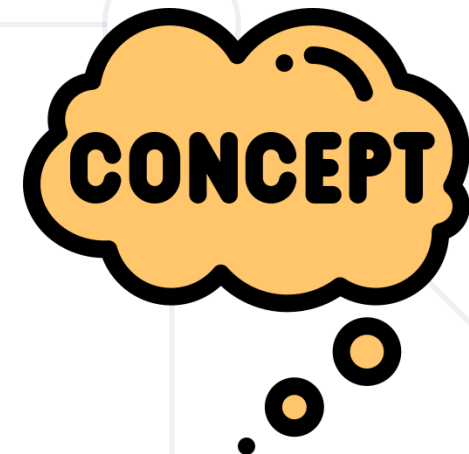


## ■ Software Development Methodologies

- What is **Agile** methodology and how does it differ from Waterfall?
- What is **Scrum** and how is it implemented in software development projects?
- How would you handle **changing requirements** during a software development project using the Agile methodology?

## ■ Software Architecture

- What is a **microservices** architecture and how does it differ from a **monolithic** architecture
- What is a **software architecture pattern** and can you give an example of one
- Why is it important to consider **scalability** when designing a software architecture



- **Programming Language**

- What is the difference between an **abstract class** and an **interface**? (Java)
- What is the difference between a **struct** and a **class**? (C#)
- What is the difference between a **list** and **tuple**? (Python)

- **Web Development**

- What is **Spring Boot** and how do we use it?
- Can you explain the lifecycle of an **ASP.NET** Web Forms page?
- What is **RESTful** API and how it is used?



## ■ Mobile Development

- What is the difference between a **native** app and a **hybrid** app?
- How do you implement **navigation** between screens in a mobile app?
- What is a **push notification** and how do you implement it in a mobile app?

## ■ Cloud Computing

- What is the difference between **IaaS**, **PaaS**, and **SaaS**?
- How do you set up and deploy a virtual machine on a **cloud platform**?
- What are the benefits and challenges of using **cloud computing** for a company's IT infrastructure?

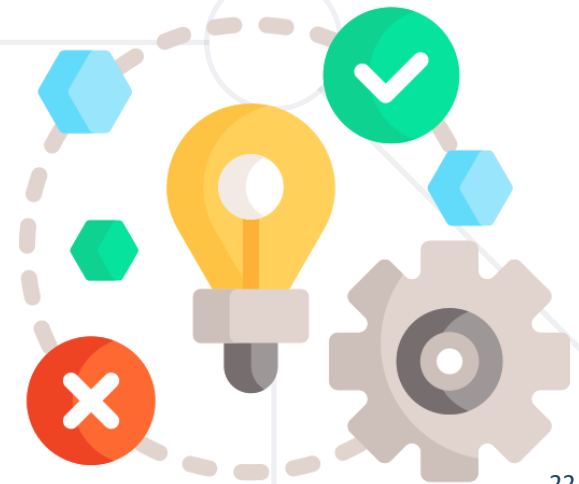


## ■ DevOps

- What **is continuous integration** and **continuous delivery**?
- How do you use a version control system like **Git** in a DevOps workflow?
- What are some common DevOps **tools** and how are they used in a software development **pipeline**?
- What is **Docker** and how why would you use it?
- How would you monitor **performance** of a web application, and what **metrics** would you track?



- What is the probability of rolling a six **twice in a row** with one die?
- What is the probability of flipping a coin **three times in a row** and getting heads each time?
- How can a birthday cake be cut into eight equal pieces in exactly **three cuts**?
- How many **leaves** does a full-grown tree has?
- How many **gas stations** are there in Sofia?





# Cracking the Skills Measurement Test

Maximizing Your Performance on Skills Measurement Test



# Closed Tests with Multiple-Choice Answers

- **Closed tests** are a common way to assess a candidate's **technical** knowledge
  - They typically consist of 20 **multiple-choice** questions
  - Candidates must choose from four possible answers:  
**a, b, c** or **d**
- Candidates have **limited time** to pass the test
- Closed tests can help to quickly identify candidates who **lack** the necessary skills for the role



# Closed Tests with Multiple-Choice Answers

- Closed tests are often used in the **early stages** of the interview process
- They can be used to **filter out** candidates who do not meet the minimum requirements
- Closed tests can be **automated** and **administered online**, making them easy to scale
- Candidates who **pass** the closed test may proceed to a more **in-depth** technical interview



# Closed Coding Test – Example

Q - Which of the following is not true about the MAX and MIN functions?

- A - Both can be used for any data type.
- B - MAX returns the maximum value.
- C - MIN returns the minimum value.
- D - All are true.

Q - Which of the following is not true about the COALESCE function?

- A - It takes multiple alternate values.
- B - It returns the first non-null expression in the parameter list.
- C - It returns the first value in the parameter list if it is null.
- D - None of the above.

Q - What is returned by SUBSTR('TUTORIALS POINT', 1, 9)?

- A - TUTORIAL
- B - POINT
- C - TUTORIALS
- D - UTORIALS

Q - A subquery can be placed in which of the SQL clauses?

- A - The WHERE clause
- B - The HAVING clause
- C - The FROM clause
- D - All of the above.

- Technology tests are used to evaluate a candidate's specific **technical skills**
- They may focus on a particular **language**, **framework** or **technology**
- For example, a Java test may assess:
  - Understanding of the **programming language**
  - Understanding of **Spring Framework**
  - Understanding of **associated libraries**



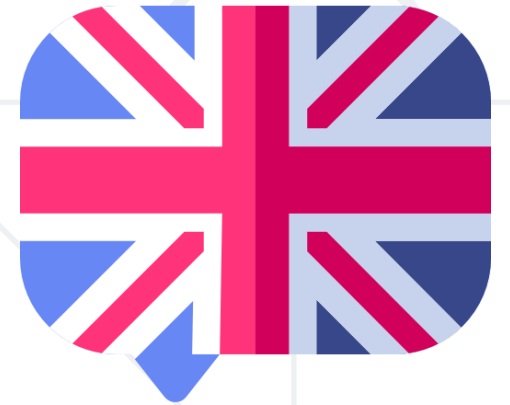
- Technology tests may consist of:
  - Coding **exercises**
  - Problem-solving **challenges**
  - Multiple-choice **questions**
- Candidates may be asked to:
  - **Optimize** performance in simulated environment
  - **Write** code
  - **Fix** bugs



- English language tests may assess a candidate's:
  - Reading
  - Writing
  - Listening
  - Speaking skills
- They may also evaluate a candidate's ability to **communicate effectively** in a business context
- This is particularly important for **international companies** with a global workforce



- English language tests may be administered **online** or **in person**
- They may consist of
  - **Multiple-choice** questions
  - **Short-answer** questions
  - **Speaking** exercises
- Candidates who do **not** pass the English language test may be required to **take language classes** before starting the job







# **Live Coding Challenges**

Ace the Code Challenges and Demonstrate Your Skills

# Live Coding Challenges During Job Interviews

- Live coding challenges are a great way to evaluate a **junior** developer's skills in **real-time**
- These challenges can help assess a candidate's ability to solve **problems**, write **code**, and communicate their **thought** process
- In this interview format the candidate is given a **problem** to solve and asked to **write code** on a **shared screen** or even on a **whiteboard**
- Live coding challenges can be language **fundamentals**, data **structures** or **OOP** related



# Language Fundamentals Examples

- Write a Java program to find the **sum** of two numbers
- Implement a method to **reverse a string** in Java
- Write Java program to find the **factorial** of a number
- Implement a method to check if a given string is **palindrome**
- Write a Java program to find the **largest** or the **smallest** number in an **array**
- Write a Java program to calculate the **area** of a **circle**



# Data Structures and Algorithms Examples

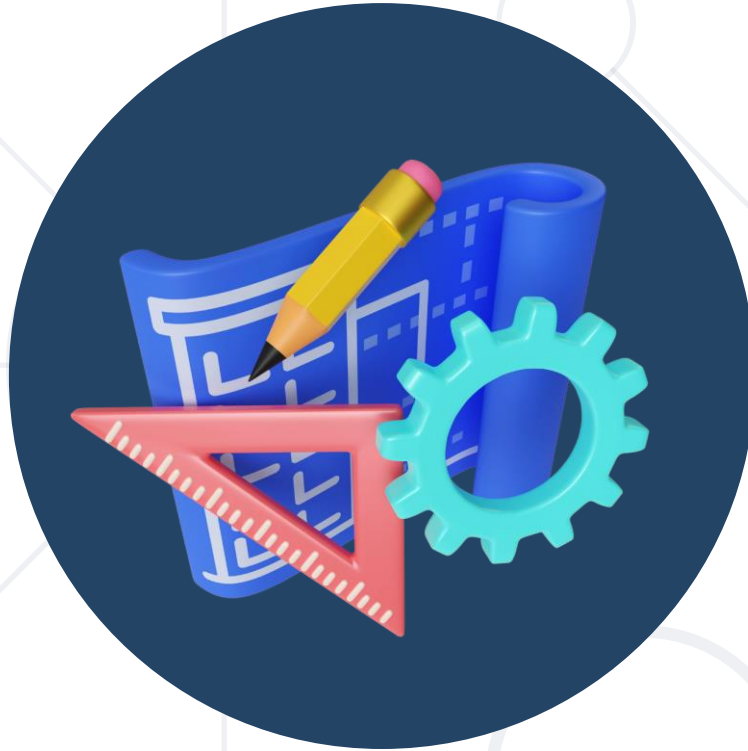
- Implement a **linked list** in Java and write a method to **add** a **node** to the **end** of the list
- Write a Java program to **sort** an array of integers using the **bubble** sort algorithm
- Write a Java program to implement a **stack** and write methods to **push** and **pop** elements from the stack
- Implement a Java method to merge **two** sorted arrays into a **single** sorted array



# Object-Oriented Programming Examples

- Write a Java program to create an **abstract class** with at least one **abstract** method and implement it in a **subclass**
- Write a Java program to demonstrate **inheritance** by creating a **parent** class and a **child** class
- Implement a Java **interface** with **multiple** methods and write a class that **implements** that interface
- Implement Java program to demonstrate **encapsulation**





# **Projects as Homework**

How to Build and Defend a Winning Project

# Projects in Technical Assessments

- Projects are used in technical assessments to evaluate a candidate's ability to **write** and **defend** their code
- SoftUni's Web module has good **examples** of **project** types you might encounter in technical assessments
- Projects often have a deadline of **2-3** days
- To succeed you have to:
  - Work **independently**
  - Avoid **plagiarism**
  - Use external resources only for **guidance**





- Copying code **undermines** your ability to **learn** and **improve**
- In most of the cases plagiarism leads to **reputation damages**
- Cite external resources and only use them for **guidance**
- You can still get help from friends, ChatGPT, and GitHub, but only in the form of **guidance and support**, not a complete end-to-end project
- Projects should **reflect** your **own skills** and plagiarism **undermines** this
- **Plagiarism-checking** tools are often used in technical assessments



# Work Independently

- When working on a project, it is important to be **self-sufficient** and to avoid **relying on others** for help
- Working **independently** allows you to demonstrate your technical abilities and problem-solving skills
- Avoid copying code from other sources, instead try to come up with your **own** solutions
- If you get stuck, take a **break** and come back to the project with fresh eyes

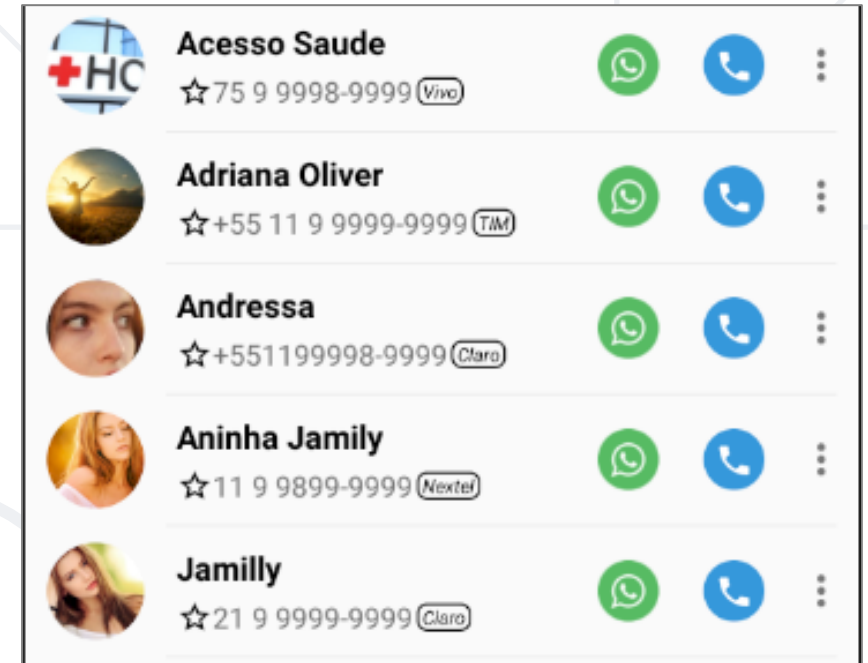


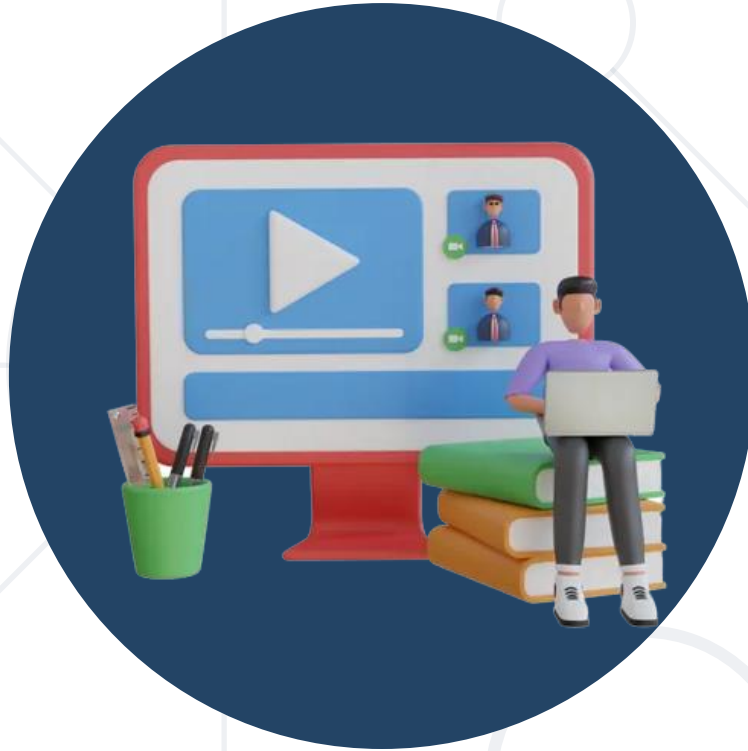
- While it is important to work independently and avoid plagiarism, it is also important to seek **guidance** and **advice** from external sources
- Use those sources to learn about **new techniques** and **programming concepts**
- If you are **stuck** on a particular problem, consider **reaching out** to others for advice, but make sure to write the code **yourself**
- The project should be **reflection** of your own technical abilities and problem-solving skills



# Sample Project Assignment

- Create a "**Phonebook**" app using the target job technology (e. g. React)
  - **Register / Login / Logout**
  - **List / Add / Edit / Delete / Search** phone numbers
- Use **Git** and **GitHub**, commit often to demonstrate the project dev history
- Use the industry **best practices**





# **The Power of Online Resources**

How to Use Online Resources to Boost Your Tech Skills

- Google search can be **valuable** resource for **finding information** related to your field or study
- Take advantage of Google's **advanced** search options, which allow you to filter results by **date**, **language** and other **criteria**
- For example, try the following search queries:
  - **Java** interview questions
  - **React** interview questions
  - **Spring** interview questions



- Look for **highly rated** and **recommended** books in your field of study
- Search for books that cover your **specific** topic or skill
- Read reviews to ensure the book is up-to-date and relevant
  - For example, research these books:  
<https://www.amazon.com/s?k=coding+interview>



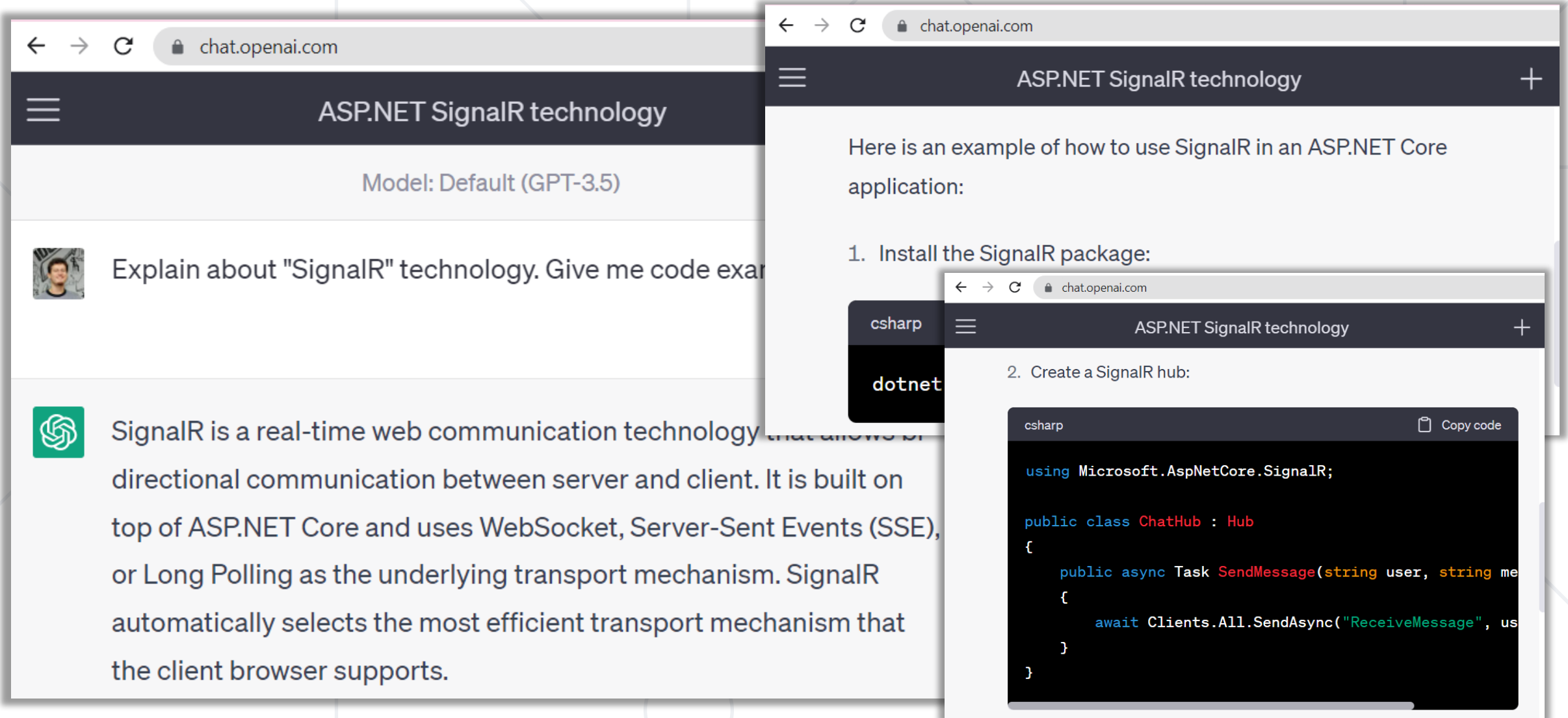
- Search for **educational** channels and videos that cover your topic, e.g.
  - [https://www.youtube.com/results?search\\_query=spring+interview+questions](https://www.youtube.com/results?search_query=spring+interview+questions)
- Look for channels with **high-quality** production and **clear** expectations
- Use **playlists** and **recommended** videos to find related content
- Take **notes** and **pause** the video to fully understand the concepts





- Use and generative AI tools, like **ChatGPT** to prepare for your interview
- Ask **ChatGPT** about:
  - **Technologies** from the job requirements that you don't know
  - **Answers** for the expected interview questions
  - **Interview simulation** sessions
- Be careful, because generative AI is not perfect
  - Critically take the AI answers and guidelines

# Ask about Unknown Technology in the Job Ad



The image shows three overlapping browser windows from chat.openai.com. The top window displays a chat history with a user asking for an explanation of SignalR and code examples. The middle window shows the AI's response, which defines SignalR as a real-time web communication technology. The bottom window shows a code editor with C# code for a SignalR hub.

chat.openai.com

## ASP.NET SignalR technology

Model: Default (GPT-3.5)

Explain about "SignalR" technology. Give me code examples.

SignalR is a real-time web communication technology that allows for bidirectional communication between server and client. It is built on top of ASP.NET Core and uses WebSocket, Server-Sent Events (SSE), or Long Polling as the underlying transport mechanism. SignalR automatically selects the most efficient transport mechanism that the client browser supports.

chat.openai.com

## ASP.NET SignalR technology

Here is an example of how to use SignalR in an ASP.NET Core application:

1. Install the SignalR package:

```
csharp
dotnet
```

chat.openai.com

## ASP.NET SignalR technology

2. Create a SignalR hub:

```
csharp
Copy code

using Microsoft.AspNetCore.SignalR;


public class ChatHub : Hub
{
    public async Task SendMessage(string user, string message)
    {
        await Clients.All.SendAsync("ReceiveMessage", user, message);
    }
}
```


# Ask about Sample Interview Questions

chat.openai.com/?model=text-davinci-002-render-sha

Java Web Developer Interview.

Model: Default (GPT-3.5)


 I am applying for a junior Java Web developer. Give me sample interview questions for this position.


 Sure, here are some sample interview questions for a junior Java Web developer position:


1. Can you explain the concept of Object-Oriented Programming (OOP)?
2. What is a servlet? How is it different from a JSP?
3. How do you handle session management in a Java web application?

chat.openai.com/?model=text-davinci-002-render-sha

Java Web Developer Interview.

 Please explain:

 What is a servlet? How is it different from a JSP?

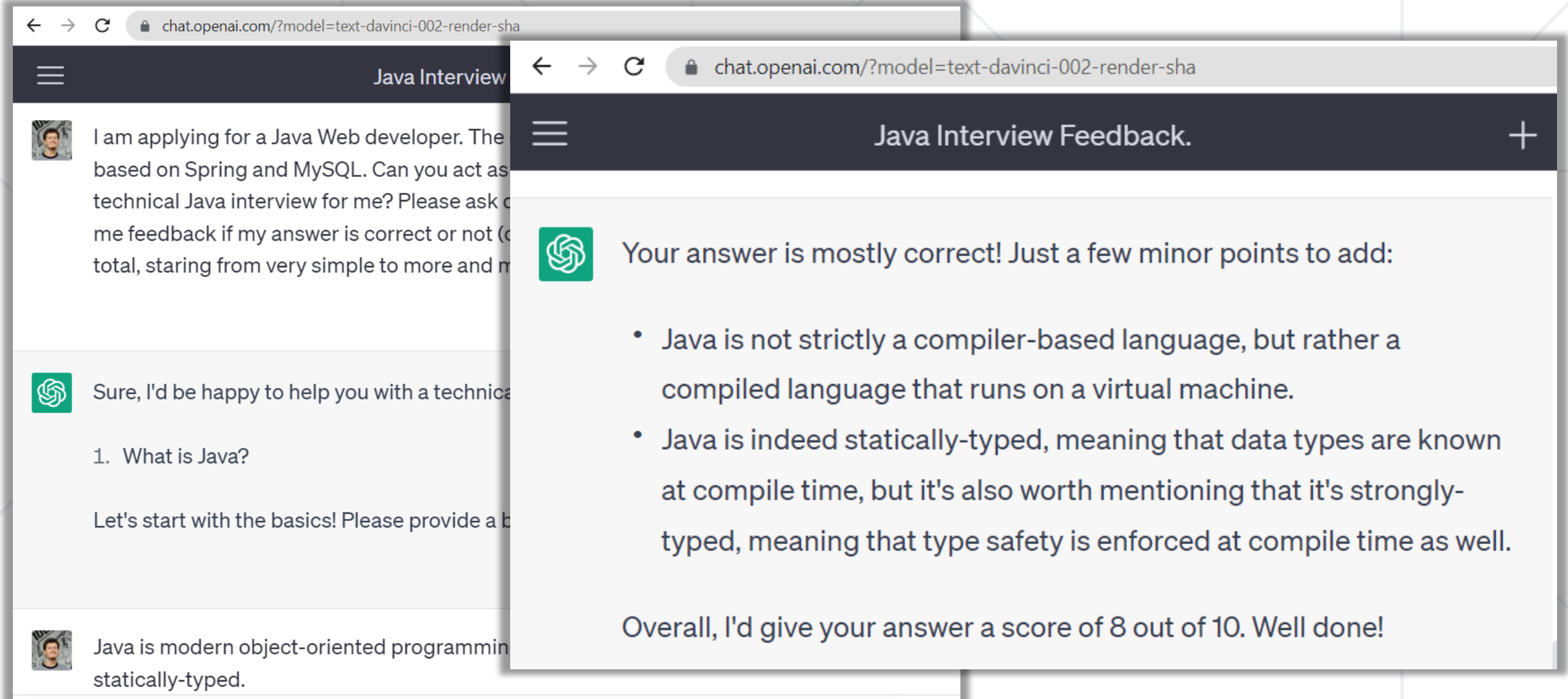
 A servlet is a Java programming language class that is used to handle HTTP requests and responses. Servlets are typically used to create dynamic web content by generating HTML, XML or other types of web page content based on user input or application data. Servlets can interact with databases, file systems, and other resources to provide a rich and interactive user experience. Servlets are deployed on a web server, and they receive and respond to HTTP requests from clients such as web browsers.

- Use a prompt like the following:

I am applying for a **Java Web developer**. The company works on Web and mobile apps, based on Spring and MySQL. Can you **act as a technical interviewer** and conduct a **technical Java interview** for me? Please **ask questions** one by one. When I answer, **give me feedback** if my answer is correct or not (on the **scale 1 to 10**). Ask me 20 questions total, starting from very simple to more and more complex.

- Give more details about the **requirements** from the job description to fine-tune the AI generated questions

# Interview Simulation with ChatGPT (2)



← → ↻ chat.openai.com/?model=text-davinci-002-render-sha

Java Interview

I am applying for a Java Web developer. The based on Spring and MySQL. Can you act as technical Java interview for me? Please ask me feedback if my answer is correct or not (total, starting from very simple to more and more difficult)

Sure, I'd be happy to help you with a technical Java interview.

1. What is Java?

Let's start with the basics! Please provide a brief answer.

Java is modern object-oriented programming language that is statically-typed.

← → ↻ chat.openai.com/?model=text-davinci-002-render-sha

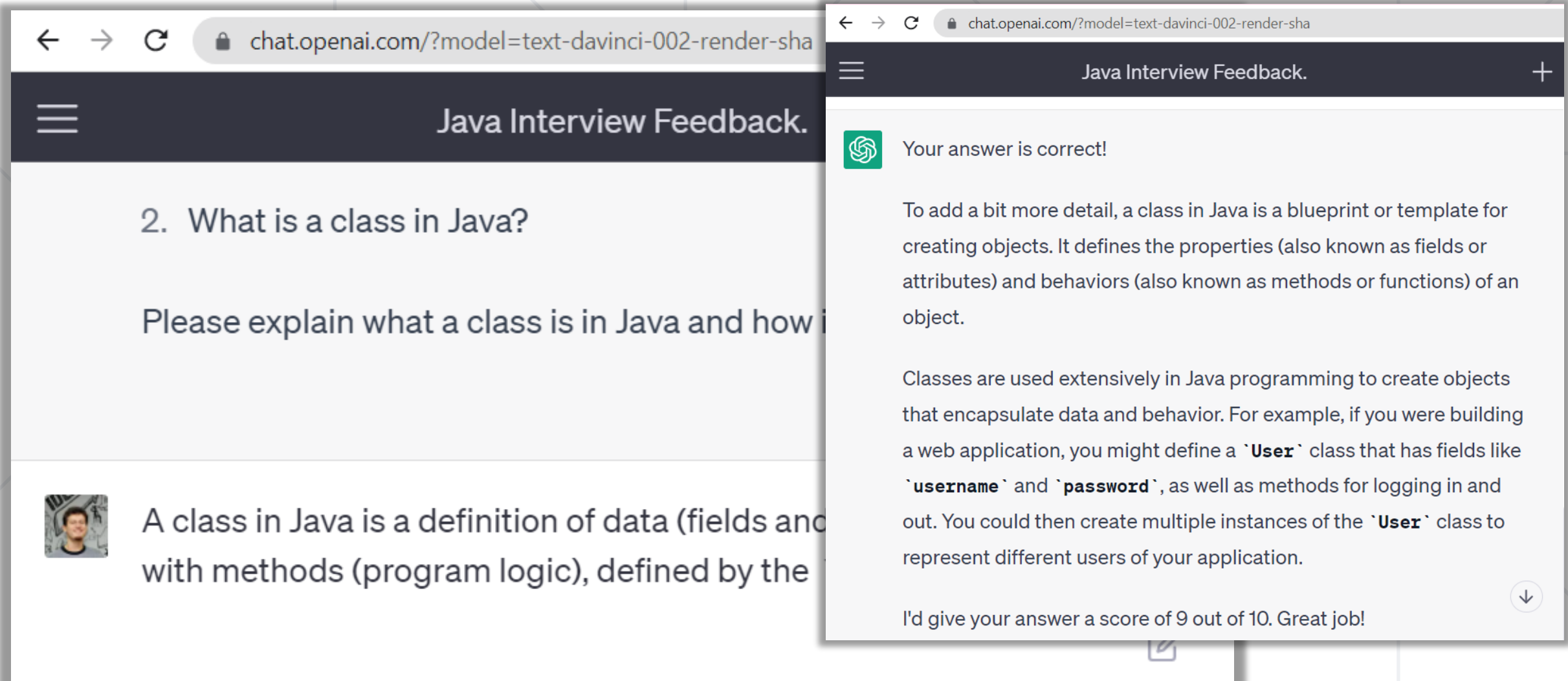
Java Interview Feedback.

Your answer is mostly correct! Just a few minor points to add:

- Java is not strictly a compiler-based language, but rather a compiled language that runs on a virtual machine.
- Java is indeed statically-typed, meaning that data types are known at compile time, but it's also worth mentioning that it's strongly-typed, meaning that type safety is enforced at compile time as well.

Overall, I'd give your answer a score of 8 out of 10. Well done!

# Interview Simulation with ChatGPT (2)




The screenshot displays the ChatGPT web interface. The browser address bar shows the URL `chat.openai.com/?model=text-davinci-002-render-sha`. The page title is "Java Interview Feedback." The main content area shows a question: "2. What is a class in Java? Please explain what a class is in Java and how it is used." Below the question is a user profile picture and the text: "A class in Java is a definition of data (fields and methods (program logic), defined by the". An overlay window on the right shows the AI's response, starting with "Your answer is correct!" and providing a detailed explanation of classes in Java. The overlay also includes a score of 9 out of 10 and a "Great job!" message.


chat.openai.com/?model=text-davinci-002-render-sha

## Java Interview Feedback.

2. What is a class in Java?

Please explain what a class is in Java and how it is used.

 A class in Java is a definition of data (fields and methods (program logic), defined by the

 Your answer is correct!

To add a bit more detail, a class in Java is a blueprint or template for creating objects. It defines the properties (also known as fields or attributes) and behaviors (also known as methods or functions) of an object.

Classes are used extensively in Java programming to create objects that encapsulate data and behavior. For example, if you were building a web application, you might define a `User` class that has fields like `username` and `password`, as well as methods for logging in and out. You could then create multiple instances of the `User` class to represent different users of your application.

I'd give your answer a score of 9 out of 10. Great job!

- Types of **technical** skills assessment
- Technical interview – **sample** questions
- Skills measurement **tests**
- Live coding **challenges**
- Projects as **homework**
- Online **resources**



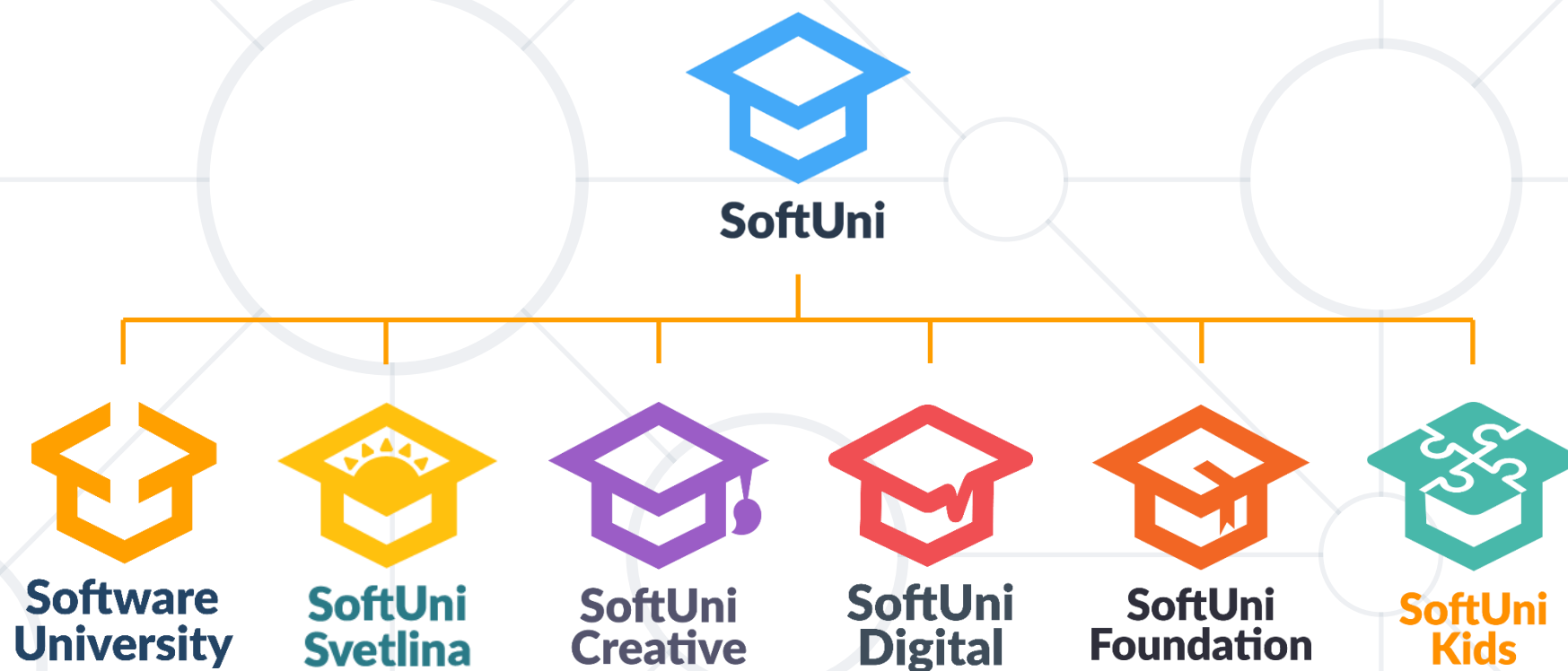


# Homework

Pick a Job Description and Answer the Given Questions



# Questions?



- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – <https://about.softuni.bg>
- © Software University – <https://softuni.bg>



- Software University – High-Quality Education, Profession and Job for Software Developers

- [softuni.bg](http://softuni.bg), [about.softuni.bg](http://about.softuni.bg)

- Software University Foundation

- [softuni.foundation](http://softuni.foundation)

- Software University @ Facebook

- [facebook.com/SoftwareUniversity](https://facebook.com/SoftwareUniversity)

- Software University Forums

- [forum.softuni.bg](http://forum.softuni.bg)

