**Python**

1. **What are Python decorators and how do you use them?**
   * **Answer:** Decorators are functions that modify the functionality of another function. They are often used for logging, enforcing access control, or instrumentation. You use them by defining a function that takes another function as an argument and returns a new function.
2. **Can you explain the difference between a list and a tuple?**
   * **Answer:** Lists are mutable, meaning you can change their contents, while tuples are immutable, meaning their contents cannot be changed after creation. Lists use square brackets [], while tuples use parentheses ().
3. **How do you manage dependencies in a Python project?**
   * **Answer:** Dependencies are managed using a requirements.txt file or with tools like pipenv or poetry. This allows you to specify package versions and ensure that all dependencies are installed in a consistent manner.

**Java**

1. **What are the main principles of Object-Oriented Programming in Java?**
   * **Answer:** The main principles are Encapsulation (bundling data with methods), Inheritance (creating new classes from existing ones), Polymorphism (methods behaving differently based on the object), and Abstraction (hiding complex implementation details).
2. **Can you explain the difference between == and equals() in Java?**
   * **Answer:** == checks for reference equality (whether two references point to the same object), while equals() checks for value equality (whether two objects are logically equivalent).
3. **What are Java Collections, and can you describe some of the most commonly used ones?**
   * **Answer:** Java Collections are data structures that store groups of objects. Commonly used ones include ArrayList (dynamic array), HashMap (key-value pairs), and HashSet (unordered collection of unique elements).

**SQL**

1. **What are the different types of JOIN operations in SQL?**
   * **Answer:** The main types are INNER JOIN (returns rows with matching values), LEFT JOIN (returns all rows from the left table and matched rows from the right), RIGHT JOIN (opposite of LEFT JOIN), and FULL OUTER JOIN (returns rows when there is a match in either table).
2. **How would you optimize a slow SQL query?**
   * **Answer:** Optimization techniques include using proper indexing, avoiding SELECT \*, analyzing query execution plans, and using joins instead of subqueries when appropriate.
3. **Can you explain the difference between DELETE, TRUNCATE, and DROP?**
   * **Answer:** DELETE removes specific rows from a table and can be rolled back. TRUNCATE removes all rows but cannot be rolled back and does not log individual row deletions. DROP removes the entire table or database structure.

**Postman**

1. **How do you use Postman to test APIs?**
   * **Answer:** You can create requests (GET, POST, etc.), set parameters, headers, and body, and then send them to the API. You can also validate responses using tests written in JavaScript.
2. **Can you explain how to create a collection in Postman?**
   * **Answer:** A collection can be created by clicking the “New” button, selecting “Collection,” and then adding requests to it. Collections help organize requests and share them with teams.
3. **What is the purpose of environment variables in Postman?**
   * **Answer:** Environment variables store values that can change between different environments (e.g., development, testing, production). They help manage configurations without hardcoding values in requests.

**Git**

1. **What is the difference between git pull and git fetch?**
   * **Answer:** git fetch retrieves changes from the remote repository without merging them into the current branch, while git pull fetches and automatically merges those changes.
2. **How do you resolve merge conflicts in Git?**
   * **Answer:** You identify the conflicting files, manually edit them to resolve the conflicts, then mark them as resolved using git add, and finally commit the changes.
3. **Can you explain the purpose of a .gitignore file?**
   * **Answer:** A .gitignore file specifies which files and directories should be ignored by Git. This is useful for excluding temporary files, build artifacts, and sensitive information.

**Jira**

1. **How do you prioritize tasks in Jira?**
   * **Answer:** Tasks can be prioritized by assigning them a priority level (e.g., High, Medium, Low) and using the backlog to arrange them based on importance and urgency.
2. **What is a sprint, and how do you manage one in Jira?**
   * **Answer:** A sprint is a time-boxed period during which specific tasks are completed. In Jira, you manage sprints by creating a sprint backlog, tracking progress through boards, and reviewing the sprint at its end.
3. **Can you explain the concept of user stories and epics in Agile?**
   * **Answer:** User stories are short descriptions of a feature from the user’s perspective, while epics are larger bodies of work that can be broken down into multiple user stories.

**Jenkins - CI/CD**

1. **What is Continuous Integration, and why is it important?**
   * **Answer:** Continuous Integration (CI) is the practice of automatically integrating code changes into a shared repository several times a day. It helps detect errors early, improves software quality, and reduces integration problems.
2. **How do you set up a Jenkins pipeline?**
   * **Answer:** A Jenkins pipeline can be set up by creating a Jenkinsfile that defines the stages of the build process. This can include build, test, and deploy stages, and can be configured in the Jenkins UI.
3. **What are some common plugins you have used in Jenkins?**
   * **Answer:** Common plugins include Git plugin (for version control), Pipeline plugin (for defining CI/CD workflows), and Jenkinsfile Runner plugin (for running pipelines).

**Selenium**

1. **What is the Page Object Model (POM), and why is it useful?**
   * **Answer:** POM is a design pattern that creates an object repository for web UI elements. It enhances test maintainability and reusability by separating test logic from the UI structure.
2. **Can you explain how to use XPath in Selenium?**
   * **Answer:** XPath is used to navigate through elements and attributes in an XML document. In Selenium, you can use driver.findElement(By.xpath("your\_xpath")) to locate elements using XPath expressions.
3. **How do you handle alerts and pop-ups in Selenium?**
   * **Answer:** You can use driver.switchTo().alert() to switch to an alert. You can then accept, dismiss, or retrieve text from the alert using methods like alert.accept(), alert.dismiss(), or alert.getText().

**Cucumber**

1. **What is BDD (Behavior-Driven Development), and how does Cucumber support it?**
   * **Answer:** BDD is a software development approach that encourages collaboration between developers, testers, and business stakeholders. Cucumber supports BDD by allowing you to write tests in plain language that describe the expected behavior of the application.
2. **Can you explain how to write a feature file in Cucumber?**
   * **Answer:** A feature file describes a feature using scenarios written in Gherkin syntax. Each scenario consists of Given (setup), When (action), and Then (expected outcome) steps.
3. **How do you integrate Cucumber with Selenium?**
   * **Answer:** You can integrate Cucumber with Selenium by creating step definitions in Java that link Gherkin steps to Selenium actions, allowing for automated browser testing based on feature files.

**TestNG**

1. **What are the main advantages of using TestNG over JUnit?**
   * **Answer:** TestNG provides advanced features like annotations, grouping, dependency testing, and parallel execution, making it more flexible and powerful compared to JUnit.
2. **How do you manage test dependencies in TestNG?**
   * **Answer:** You can manage test dependencies using the dependsOnMethods attribute in the @Test annotation to specify which tests must run before others.
3. **Can you explain how to group tests in TestNG?**
   * **Answer:** You can group tests by using the groups attribute in the @Test annotation and defining the groups in the TestNG XML configuration file, allowing you to run specific groups of tests together.

**Manual Testing**

1. **What are the key differences between manual and automated testing?**
   * **Answer:** Manual testing involves human testers executing test cases without automation tools, while automated testing uses scripts and tools to execute tests. Manual testing is useful for exploratory testing, while automated testing is better for repetitive tasks.
2. **How do you approach writing test cases?**
   * **Answer:** I start by understanding the requirements, then identify test scenarios, and write clear, concise test cases that include preconditions, steps, and expected results.
3. **What are some common testing techniques you use?**
   * **Answer:** Common techniques include boundary value analysis, equivalence partitioning, exploratory testing, and user acceptance testing (UAT).

**Agile Methodology**

1. **Can you explain the Agile Manifesto and its principles?**
   * **Answer:** The Agile Manifesto emphasizes individuals and interactions over processes, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan.
2. **What roles are involved in an Agile team, and what are their responsibilities?**
   * **Answer:** Key roles include the Product Owner (defines the product backlog), Scrum Master (facilitates the process), and Development Team (builds the product). Each member collaborates to deliver increments of value.
3. **How do you handle changes in requirements during a sprint?**
   * **Answer:** Changes should be discussed with the Product Owner and evaluated based on their impact on the sprint goals. If necessary, they can be added to the product backlog for future sprints.