Created By - @Kushal Parikh

https://www.linkedin.com/in/kushalparikh11/

30 Most Asked Manual Testing Interview Questions

1. What is Software Testing?

2. What is the difference between Verification and Validation?

3. What is the SDLC and what is its role in testing?

4. What is STLC (Software Testing Life Cycle)?

5. What are the different levels of testing?

6. What is the difference between Functional and Non-Functional Testing?

7. What is the difference between Regression and Retesting?

8. What are Positive and Negative Testing?

9. What is the difference between Severity and Priority?

10. What is a Test Case? What are its components?

11. What is a Test Plan and what does it include?

12. What are the different types of testing? (e.g., Smoke, Sanity, Ad-hoc, Exploratory)

13. What is Boundary Value Analysis (BVA)?

14. What is Equivalence Partitioning?

15. What is the difference between Static and Dynamic Testing?

16. What is the Defect Life Cycle?

17. What is a Traceability Matrix?

18. How do you prioritize test cases?

19. What is Risk-Based Testing?

20. What are Entry and Exit Criteria in testing?

21. What would you do if you find a bug in the production environment?

22. How do you handle missing requirements or unclear documentation?

23. If the developer is not agreeing with your bug, what will you do?

24. How do you report a bug effectively?

25. Have you ever participated in a requirement review meeting? What was your role?

https://topmate.io/kushalparikh11/

26. What will you do if you have limited time to test?

27. How do you perform exploratory testing?

28. How do you ensure 100% test coverage?

29. How do you test without requirements or documentation?

30. What are some common challenges you face during testing?

30 Most Asked Java Interview Questions

1. What are the main features of Java?

2. What is the difference between JDK, JRE, and JVM?

3. What are the OOP principles in Java?

4. What is the difference between `==` and `.equals()` in Java?

5. What is the difference between `ArrayList` and `LinkedList`?

6. What is the difference between `HashMap`, `LinkedHashMap`, and `TreeMap`?

7. What is a constructor in Java? How is it different from a method?

8. What is method overloading and method overriding?

9. What is the difference between abstract class and interface?

10. What is the difference between `final`, `finally`, and `finalize()`?

11. What is a static variable and static method in Java?

12. What are access modifiers in Java?

13. What is the difference between `String`, `StringBuilder`, and `StringBuffer`?

14. What is exception handling in Java? Explain try-catch-finally.

15. What is the difference between checked and unchecked exceptions?

16. What is the use of `throw` and `throws` keywords?

17. What is a collection in Java?

18. What is the difference between `Set`, `List`, and `Map` interfaces?

19. What are Generics in Java?

20. What is multithreading in Java?

https://topmate.io/kushalparikh11/

21. What is the difference between `synchronized` block and method?

22. What is the purpose of the `volatile` keyword?

23. What is a lambda expression in Java 8?

24. What are functional interfaces?

25. What is the Stream API in Java 8?

26. What is the difference between Iterator and ListIterator?

27. How does garbage collection work in Java?

28. How do you handle file operations in Java?

29. How do you use Java to send an HTTP request?

30. How is Java used in Selenium or any automation framework?

30 Core Java OOPs Concept Interview Questions

1. What is Object-Oriented Programming (OOP)?

2. What are the main principles of OOP in Java?

3. What is a class and how is it different from an object?

4. What is encapsulation in Java and how is it achieved?

5. What is abstraction and how do you implement it in Java?

6. What is the difference between abstraction and encapsulation?

7. What is inheritance in Java?

8. What is the difference between single and multiple inheritance?

9. Why does Java not support multiple inheritance with classes?

10. How does Java achieve multiple inheritance using interfaces?

11. What is polymorphism in Java?

12. What is the difference between compile-time and runtime polymorphism?

13. What is method overloading?

14. What is method overriding?

15. Can you override a private or static method in Java?

https://topmate.io/kushalparikh11/

16. What is the use of the `super` keyword?

17. What is the `this` keyword in Java?

18. What is an abstract class and when would you use it?

19. What is an interface and how is it different from an abstract class?

20. Can a class implement multiple interfaces?

21. What is the diamond problem in inheritance? How does Java handle it?

22. What is the difference between `==` and `.equals()` in object comparison?

23. What is constructor overloading?

24. Can a constructor be inherited in Java?

25. What is object slicing in Java?

26. What are access modifiers and how do they relate to encapsulation?

27. What are the key differences between class and structure (though Java doesn't have struct)?

28. How are interfaces used in real-world OOP design?

29. How do you achieve loose coupling in OOP?

30. What is a design pattern and how is it related to OOP principles?

30 Top Most Asked Java Coding Questions for SDET Interviews

1. Reverse a String without using reverse() method

2. Check if a String is a palindrome

3. Find duplicate characters in a String

4. Count the occurrence of each character in a String

5. Check if two Strings are anagrams

6. Find the first non-repeated character in a String

7. Reverse each word in a sentence

8. Remove all whitespaces from a String

9. Swap two numbers without using a third variable

https://topmate.io/kushalparikh11/

10. Find the factorial of a number using recursion

11. Check if a number is prime

12. Print Fibonacci series up to N terms

13. Check if a number is an Armstrong number

14. Find the largest and smallest number in an array

15. Sort an array without using sort() method

16. Find duplicate elements in an array

17. Check if two arrays are equal

18. Find the second highest number in an array

19. Remove duplicates from an array

20. Reverse an array

21. Implement a custom logic to convert String to Integer (like `Integer.parseInt`)

22. Check for balanced parentheses using Stack

23. Implement a basic LinkedList (add, delete, print)

24. Use HashMap to count word frequency in a paragraph

25. Find common elements in two arrays/lists

26. Sort a Map by values

27. Use Java 8 Stream to filter even numbers from a list

28. Find the first repeating element in an array

29. Find missing number in a sequence (e.g., 1 to 100)

30. Detect a cycle in a LinkedList

30 Most Asked Selenium WebDriver Interview Questions

1. What is Selenium and what are its components?

2. What is the difference between Selenium WebDriver and Selenium RC?

3. What are the limitations of Selenium WebDriver?

4. What programming languages are supported by Selenium WebDriver?

https://topmate.io/kushalparikh11/

5. How do you launch a browser using Selenium WebDriver?

6. What is the difference between `findElement()` and `findElements()`?

7. What are different types of waits in Selenium WebDriver?

8. How do you handle dynamic web elements in Selenium?

9. What is the difference between `get()` and `navigate().to()` methods?

10. How do you handle dropdowns in Selenium?

11. How do you handle alerts and pop-ups in Selenium WebDriver?

12. How do you perform mouse hover or right-click actions?

13. How can you perform drag-and-drop in Selenium WebDriver?

14. How do you handle multiple browser windows or tabs?

15. How do you take a screenshot using Selenium WebDriver?

16. How do you upload or download a file using Selenium?

17. How do you handle frames and iframes in Selenium?

18. How do you scroll down to an element using WebDriver?

19. What are XPath and CSS Selectors? Which one is better and why?

20. What is the difference between absolute and relative XPath?

21. What is StaleElementReferenceException and how do you handle it?

22. What is Page Object Model (POM)?

23. What is PageFactory in Selenium?

24. What is TestNG and how is it used with Selenium?

25. How do you prioritize test cases in TestNG?

26. How do you run tests in parallel using TestNG?

27. How do you manage browser drivers like ChromeDriver or GeckoDriver?

28. What is headless browser testing and how is it configured?

29. What are the best practices you follow in Selenium Automation?

30. How do you generate reports in Selenium (ExtentReports/Allure)?

Created By - @Kushal Parikh

https://www.linkedin.com/in/kushalparikh11/

30 Maven Interview Questions for SDET

1. What is Apache Maven and why is it used?

2. How does Maven differ from Ant?

3. What is the POM file in Maven?

4. What are the key elements of a POM file?

5. How do you define dependencies in Maven?

6. What is the Maven lifecycle?

7. Can you explain the different build phases in Maven?

8. How do you create a Maven project from the command line?

9. What are Maven repositories?

10. What is the difference between local, central, and remote repositories?

11. How does Maven handle dependency management?

12. What is a Maven plugin?

13. How do you add plugins to a Maven project?

14. What is the use of the `mvn clean` command?

15. How do you skip tests while building a Maven project?

16. What is the `mvn install` command used for?

17. How can you run a specific test or test suite using Maven?

18. How do you manage different environments in Maven?

19. What is the `settings.xml` file?

20. How do you configure proxy settings in Maven?

21. What is the difference between `compile` and `provided` scopes in Maven dependencies?

22. How do you handle version conflicts in Maven dependencies?

23. What is the use of the `mvn dependency:tree` command?

24. How do you create a Maven archetype?

25. How can you customize the build process in Maven?

30 Maven Interview Questions for SDET

https://topmate.io/kushalparikh11/

26. What is the difference between SNAPSHOT and release versions?

27. How do you integrate Maven with Jenkins for CI/CD?

28. Can Maven be used for non-Java projects?

29. What is the difference between Maven goals and phases?

30. How do you generate reports using Maven?

30 Most Asked TestNG Interview Questions

1. What is TestNG and why is it used in Selenium automation?

2. How is TestNG different from JUnit?

3. What are the advantages of using TestNG?

4. How do you install and configure TestNG in Eclipse?

5. How do you run a TestNG test?

6. What are the commonly used TestNG annotations?

7. What is the execution order of TestNG annotations?

8. What is the difference between `@BeforeMethod` and `@BeforeClass`?

9. What is the use of `@DataProvider` in TestNG?

10. How do you use `@Test(enabled = false)`?

11. How do you group tests in TestNG?

12. How do you run tests in parallel using TestNG?

13. What is the purpose of `testng.xml` file?

14. How do you create and configure `testng.xml`?

15. What is dependency testing in TestNG?

16. How do you make one test method dependent on another?

17. What is the use of `priority` attribute in TestNG?

18. How do you handle timeouts in TestNG?

19. How do you skip tests in TestNG?

https://topmate.io/kushalparikh11/

20. What is the difference between `assert` and `soft assert` in TestNG?

21. How do you implement parameterization in TestNG?

22. What is a listener in TestNG?

23. How do you use listeners in TestNG?

24. What is the difference between `@BeforeSuite` and `@BeforeTest`?

25. How do you generate reports in TestNG?

26. How can you run tests from the command line using TestNG?

27. How do you handle exceptions in TestNG tests?

28. What are the different types of assertions in TestNG?

29. Can you explain how TestNG integrates with Maven?

30. How do you debug TestNG tests?

30 Most Asked Cucumber BDD Interview Questions

1. What is Cucumber and what is BDD?

2. How does Cucumber support BDD?

3. What are the main components of Cucumber?

4. What is a feature file in Cucumber?

5. What is the syntax of a feature file?

6. What are Gherkin keywords?

7. What is the purpose of the `Given`, `When`, `Then` keywords?

8. How do you write scenarios in Cucumber?

9. What is a Scenario Outline?

10. How do you use Examples in Cucumber?

11. How are step definitions written in Cucumber?

12. What is the role of glue code in Cucumber?

13. How do you map steps in the feature file to code?

14. How do you handle parameterization in Cucumber steps?

https://topmate.io/kushalparikh11/

15. What are hooks in Cucumber?

16. How do `@Before` and `@After` hooks work?

17. How do you handle tags in Cucumber?

18. How do you run scenarios with specific tags?

19. What is the difference between Scenario and Scenario Outline?

20. How do you share data between steps?

21. How do you integrate Cucumber with Selenium?

22. How do you generate reports in Cucumber?

23. What is the purpose of the `cucumber.options` in the runner class?

24. What are Data Tables in Cucumber?

25. How do you handle exceptions in Cucumber steps?

26. Can Cucumber tests be run in parallel? How?

27. What is the difference between Background and Before hooks?

28. How do you organize feature files in a project?

29. What is the role of the runner class in Cucumber?

30. How do you integrate Cucumber with Maven or Gradle?

30 Most Asked Page Object Model (POM) Framework Interview Questions

1. What is the Page Object Model (POM) in test automation?

2. What are the advantages of using the POM framework?

3. How do you implement POM in Selenium?

4. What is the structure of a typical POM framework?

5. How do you manage locators in the POM design pattern?

6. What is the difference between Page Object and Page Factory?

7. How do you initialize web elements in POM?

8. What are the best practices to follow when using POM?

9. How do you handle dynamic elements in POM?

https://topmate.io/kushalparikh11/

10. How does POM improve test maintenance?

11. Can you explain how to handle multiple page objects in a test?

12. How do you handle synchronization issues in POM?

13. How do you manage test data in POM framework?

14. What is the role of base classes in POM?

15. How do you implement reusable methods in POM?

16. How do you integrate POM with test frameworks like TestNG or JUnit?

17. How do you handle exceptions in POM classes?

18. How do you structure your project using POM?

19. How does POM support scalability in automation projects?

20. Can you explain the use of the `PageFactory.initElements()` method?

21. How do you handle pop-ups and alerts in POM?

22. How do you implement logging in a POM framework?

23. How do you manage configuration files in POM?

24. How do you integrate POM with reporting tools?

25. How do you design tests for complex web applications using POM?

26. What are the limitations of the POM framework?

27. How do you handle cross-browser testing in a POM framework?

28. How do you implement parallel test execution in POM?

29. What design patterns can be combined with POM?

30. How do you keep your POM framework maintainable over time?

30 Most Real-time Selenium Coding Challenges

1. Write a Selenium script to handle dynamic web elements whose locators change frequently.

2. Automate a scenario to select multiple options from a multi-select dropdown.

3. Write code to upload a file using Selenium without using AutoIT or external tools.

https://topmate.io/kushalparikh11/

4. Handle a JavaScript alert popup and verify its message.

5. Automate drag and drop functionality on a web page.

6. Scroll to a specific element on a page and verify its visibility.

7. Switch between multiple browser windows and verify the page titles.

8. Write a method to wait explicitly for an element to be clickable.

9. Handle a frame or iframe and interact with elements inside it.

10. Automate mouse hover and perform click action on a submenu.

11. Capture and save a screenshot after a test failure automatically.

12. Write a reusable method to handle calendar date selection.

13. Automate pagination controls and verify data on multiple pages.

14. Handle dynamic tables and extract specific row/column data.

15. Write a script to verify broken links on a web page.

16. Automate a login test with data-driven inputs using Excel or CSV.

17. Implement a retry logic for flaky tests that fail intermittently.

18. Write code to simulate keyboard actions like Ctrl+C, Ctrl+V, etc.

19. Automate drag-and-drop where the target location is calculated dynamically.

20. Verify tooltip text on hover of a web element.

21. Handle file download and verify the file in the local system.

22. Write code to interact with shadow DOM elements.

23. Handle AJAX-based loading elements and verify content changes.

24. Automate login tests with CAPTCHA handling (explain approach).

25. Write a script to perform right-click and select context menu options.

26. Automate tests for responsive design by changing browser size.

27. Write a custom wait method for element visibility without using WebDriverWait.

28. Implement a method to handle stale element exceptions dynamically.

29. Automate interaction with web components built using React or Angular.

30. Create a data-driven test framework that reads input from JSON files.