Selenium with Cucumber Framework Interview Questions & Answers

1) Explain Cucumber shortly.
Cucumber is a tool that is based on Behavior Driven Development (BDD) methodology. The main aim of the Behavior Driven Development framework is to make various project roles such as Business Analysts, Quality Assurance, Developers, etc., understand the application without diving deep into the technical aspects.

2) What language is used by Cucumber?
Gherkin is the language that is used by the Cucumber tool. It is a simple English representation of the application behavior. Gherkin language uses several keywords to describe the behavior of applications such as Feature, Scenario, Scenario Outline, Given, When, Then, etc.

3) What is meant by a feature file?
A feature file must provide a high-level description of an Application Under Test (AUT). The first line of the feature file must start with the keyword 'Feature' followed by the description of the application under test.
A feature file may include multiple scenarios within the same file. A feature file has the extension .feature.

4) What are the various keywords that are used in Cucumber for writing a scenario?
Mentioned below are the keywords that are used for writing a scenario:

- Given
- When
- Then
- And

5) What is the purpose of a Scenario Outline in Cucumber?
Scenario outline is a way of parameterization of scenarios. This is ideally used when the same scenario needs to be executed for multiple sets of data, however, the test steps remain the same. Scenario Outline must be followed by the keyword 'Examples', which specify the set of values for each parameter.

6) What programming language is used by Cucumber?
Cucumber tool provides support for multiple programming languages such as Java, .Net, Ruby etc. It can also be integrated with multiple tools such as Selenium, Capybara, etc.

7) What is the purpose of the Step Definition file in Cucumber?
A step definition file in Cucumber is used to segregate the feature files from the underlying code. Each step of the feature file can be mapped to a corresponding method on the Step Definition file.

While feature files are written in an easily understandable language like, Gherkin, Step Definition files are written in programming languages such as Java, .Net, Ruby, etc.

8) What are the major advantages of the Cucumber framework?
Given below are the advantages of the Cucumber Gherkin framework that make Cucumber an ideal choice for rapidly evolving Agile methodology in today's corporate world.

1. Cucumber is an open-source tool.
2. Plain Text representation makes it easier for non-technical users to understand the scenarios.
3. It bridges the communication gap between various project stakeholders such as Business Analysts, Developers, and Quality Assurance personnel.
4. Automation test cases developed using the Cucumber tool are easier to maintain and understand as well.
5. Easy to integrate with other tools such as Selenium and Capybara.

9) Provide an example of a feature file using the Cucumber framework.
Following is an **example** of a feature file for the scenario 'Login into the application:

Feature: Login to the application under test.
Scenario:

1. Login to the application.
2. Open the Chrome browser and launch the application.
3. When the user enters the username onto the UserName field.
4. And User enters the password into the Password field.
5. When the user clicks on the Login button.
6. Then validate if the user login is successful.

10) Provide an example of a Scenario Outline using the Cucumber framework.
The following is an **example** of a Scenario Outline keyword for the scenario 'Upload a file'. The number of parameter values to be included in the feature file is based on the tester's choice.
Scenario Outline:

1. Upload a file
2. Given that the user is on upload file screen.
3. When a user clicks on the Browse button.
4. And user enters <filename> onto the upload textbox.
5. And user clicks on the enter button.
6. Then verify that the file upload is successful.

Example:
|filename|
|file1|
|file2|

11) What is the purpose of the Behaviour Driven Development (BDD) methodology in the real world?

BDD is a methodology to understand the functionality of an applicatio the simple plain text representation.

The main aim of the Behavior Driven Development framework is to make various project roles such as Business Analysts, Quality Assurance, Developers, and Support Teams understand the application without diving deep into the technical aspects.

12) What is the limit for the maximum number of scenarios that can be included in the feature file?
A feature file can contain a maximum of 10 scenarios, but the number can vary from project to project and from one organization to another. But it is generally advisable to limit the number of scenarios included in the feature file.

13) What is the use the of Background keyword in Cucumber?
Background keyword is used to group multiple given statements into a single group. This is generally used when the same set of the given statement is repeated in each scenario of the feature file.

14) What symbol is used for parameterization in Cucumber?
Pipe symbol (|) is used to specify one or more parameter values in a feature file.

15) What is the purpose of Examples keyword in Cucumber?
Examples keyword is used to specify values for each parameter used in the scenario. Scenario Outline keyword must always be followed by the keyword Examples.

16) What is the file extension for a feature file?
File Extension for a feature file is .feature. A feature file is ideally written in a notepad file and is saved with the extension feature.

17) Provide an example of a step definition file in Cucumber.
Step definition corresponding to the step “Open Chrome browser and launch the application” may look like the code mentioned below:

```
@Given("^Open Chrome browser and launch the application$")
public void openBrowser()

{
driver = new ChromeDriver();
driver.manage().window().maximize();
driver.get("www.facebook.com");
}
```

18) What is the purpose of the Cucumber Options tag?
Cucumber Options tag is used to provide a link between the feature files and step definition files. Each step of the feature file is mapped to a corresponding method on the step definition file.

Below is the syntax of Cucumber Options tag:
@CucumberOptions(features="Features",glue={"StepDefinition"})

19) How can Cucumber be integrated with Selenium WebDriver?
Cucumber can be integrated with the Selenium Webdriver by downloading the necessary JAR files.
Given below are the list of JAR files that are to be downloaded for using Cucumber with Selenium web driver:

- cucumber-core-1.2.2.jar
- cucumber-java-1.2.2.jar
- cucumber-junit-1.2.2.jar
- cucumber-jvm-deps-1.0.3.jar
- cucumber-reporting-0.1.0.jar
- gherkin-2.12.2.jar

20) When is Cucumber used in real-time?
Cucumber tool is generally used in real-time to write acceptance tests for an application. It is generally used by non-technical people such as Business Analysts, Functional Testers, etc.

21) Provide an example of Background keyword in Cucumber.
Background: Given the user is on the application login page.

22) What is the use of Behavior Driven Development in Agile methodology?
The advantages of Behavior Driven Development are best realized when non-technical users such as Business Analysts use BDD to draft requirements and provide the same to the developers for implementation.

In Agile methodology, user stories can be written in the format of feature file and the same can be taken up for implementation by the developers.

23) Explain the purpose of keywords that are used for writing a scenario in Cucumber.

- **“Given”** keyword is used to specify a precondition for the scenario.
- **“When”** keyword is used to specify an operation to be performed.
- **“Then”** keyword is used to specify the expected result of a performed action.
- **“And”** keyword is used to join one or more statements together into a single statement.

24) What is the name of the plugin that is used to integrate Eclipse with Cucumber?
Cucumber Natural Plugin is the plugin that is used to integrate Eclipse with Cucumber.

25) What is the meaning of the TestRunner class in Cucumber?

TestRunner class is used to provide the link between the feature file and the step definition file. The next question provides a sample representation of how the TestRunner class will look like. A TestRunner class is generally an empty class with no class definition.

26) Provide an example of the TestRunner class in Cucumber.

```
Package com.sample.TestRunner

importorg.junit.runner.RunWith;
importcucumber.api.CucumberOptions;
importcucumber.api.junit.Cucumber;

@RunWith(Cucumber.class)
@CucumberOptions(features="Features",glue={"StepDefinition"})
public class Runner
{
}
```

27) What is the starting point of execution for feature files?

When integrated with Selenium, the starting point of execution must be from the TestRunner class.

28) Should any code be written within the TestRunner class?

No code should be written under the TestRunner class. It should include the tags @RunWith and @CucumberOptions.

29) What is the use of features property under the Cucumber Options tag?

Features property is used to let the Cucumber framework identify the location of the feature files.

30) What is the use of glue property under the Cucumber Options tag?

Glue property is used to let the Cucumber framework identify the location of step definition files.

31) What is the maximum number of steps that are to be written within a scenario?

3-4 steps.

32) What is Cucumber? Why is it used?

Cucumber is a testing tool based on Behavior Driven Development (BDD) framework. It is used to run functional tests written in plain text and develop test cases for software functionality. It plays a supporting role in automated testing.

In other words, we can say that "Cucumber is a software tool used by the testers to develop test cases for the testing of behavior of the software."

33) What is the main aim of the Behavior Driven Development (BDD) framework?

The main aim of the Behavior Driven Development framework is to make various project roles such as Business Analysts, Quality Assurance, Developers, etc., understand the application without diving deep into the technical aspects.

34) What are the two files required to execute a Cucumber test scenario?
Following are the two files required to execute a Cucumber test scenario:

- Features
- Step Definition

35) What do you understand by a feature file?
A feature file is used to provide a high-level description of an Application Under Test (AUT). The first line of the feature file must start with the keyword 'Feature' followed by the description of the application under test. A feature file may include multiple scenarios within the same file, and the extension of the feature file must be ".feature."

36) What are the differences between Jbehave and Cucumber?
Although Cucumber and Jbehave are designed for the same purpose, the most distinctive difference between them is that Jbehave is based on stories while Cucumber is based on features.

37) What do you understand by regular expressions?
A regular expression is a pattern used to describe a certain amount of text. The most basic regular expression consists of a single literal character.

38) What software is used to run a Cucumber Web Test case?
Following software is used to run a Cucumber Web Test case:

- Cucumber
- Ruby and its Development Kit
- IDE like ActiveState
- Watir (To simulate browser)
- Ansicon and RSpec (if required)

39) What do you understand by test harness in Cucumber?
In Cucumber, the test harness allows for separating responsibility between setting up the context and interacting with the browser, and cleaning up the step definition files. It collects stubs, drivers, and other supporting tools required to automate test execution in testing.

40) What is the difference between RSpec and Cucumber? When should we use RSpec and when to use Cucumber?
RSpec and Cucumber both are the example of testing frameworks. RSpec uses traditional Unit Testing. It means it uses testing a class or part of the application in isolation from the rest of the application. So your model does what your model is supposed to do, the controller does what it is supposed to do, etc. RSpec and Cucumber both are used for Acceptance Testing, also called ATDD, BDD, etc.

Difference between RSpec and Cucumber

The main difference between RSpec and Cucumber is the business readability factor. RSpec is mainly used for Unit Testing. On the other hand, Cucumber is mainly used for Behavior-driven development. We can also use it for System and Integration Testing.

In Cucumber, the specifications or features are separate from the test code, so the product owners can provide or review the specification without going through the code. These are the .feature files that you make in Cucumber.

RSpec also has a similar mechanism, but instead of describing a step with a Describe or Context, it uses the business specification to execute that statement. This approach is a little easier for developers to work with but a bit harder for non-technical guys.

Which should we use?

For a core developer, it is the best choice to use RSpec. It is easier to understand for a technical person and offers a few advantages in keeping things scoped and under control because you don't have to mess up with RegExs for test steps.

If you are building this for a client, you should choose Cucumber for Acceptance Testing and use RSpec for Unit Testing.

41) What is Selenium?

Selenium is a web browser automation tool widely used for Functional Testing of web-based applications. Selenium supports different programming languages such as Java, Python, Ruby, C#, etc.

42) What is the difference between Selenium and Cucumber?

Selenium and Cucumber are both open-source testing tools, and both are used for functional testing. But there are some differences between them.

Following are some critical differences between Selenium and Cucumber:

- Selenium is a web browser automation tool for web apps, while Cucumber is an automation tool for behavior-driven development that can be used with Selenium (or Appium).
- Selenium is used for automated UI testing, while Cucumber is used for acceptance testing.
- Selenium is preferred by technical teams (SDETs/programmers), while Cucumber is typically preferred by non-technical teams (business stakeholders and testers).
- Selenium can work independently of Cucumber. Cucumber depends on Selenium or Appium for step-definition implementation.
- In Selenium, the script creation is complex, while Cucumber is simpler than Selenium.

43) Why we have to use Cucumber with Selenium?

Cucumber and Selenium are both testing frameworks and prevalent technologies. Many organizations use Selenium for functional testing. Along with Selenium, these organizations integrate Cucumber with Selenium as Cucumber makes it easy to read and understand the application flow. The most significant benefit of using Cucumber with Selenium is that it facilitates developers to write test cases in simple feature files easily understood by managers, non-technical stakeholders, and business analysts. It provides the facility to write tests in a human-readable language called Gherkin. The Selenium-Cucumber framework supports programming languages such as Java, .NET, PHP, Python, Perl, etc.

44) What do you understand by TDD, and what are the different processes used in TDD?

TDD is an acronym that stands for Test-Driven Development. This is a software development technique used to create the test cases first and then write the code underlying those test cases. Although TDD is a development technique, it can also be used for automation testing development. TDD takes more time for development because it tends to find very few defects. The result provided by the TDD development technique has improved the quality of code, and that can be more reusable and flexible. TDD also helps developers to achieve high test coverage of about 90-100%. The only disadvantage for developers following TDD is to write their test cases before writing the code.

Following is the list of simple 6 step process used by TDD methodology:

First, write the test case: You have to write an automated test case according to your requirements.

Run all the test cases: Now, run these automated test cases on the currently developed code.

Develop the code for that test case: In this process, you must write the code to make that test case work as expected if the test case fails.

Run test cases again: Now, you have to rerun the test cases and check if all the test cases developed so far are implemented.

Refactor your code: This is an optional step. But, it is advised to refactor your code to make it more readable and reusable. That's why it is essential.

Repeat steps 1- 5 for new test cases: This is the last step. Here, you have to repeat the cycle for the other test cases until all the test cases are implemented.

45) What are the similarities between BDD and TDD?

- TDD stands for Test-Driven Development, and BDD stands for Behavior Driven Development. Both are two software development techniques.
- BDD and TDD are both very similar as they are both testing strategies for a software application. In both cases, the developers have to write the test before writing the code to pass the test. The second main similarity between them is in both cases; the tests can be used as part of an automated testing framework to prevent bugs.

46) What are the main differences between TDD and BDD?

Following is the list of main differences between TDD and BDD:

TDD	BDD
TDD stands for Test-Driven Development. It is a test-centered development process. This	BDD stands for Behavior Driven Development. It is a Behavior centered development process.

means the developers have first to write the test cases then code.	
In TDD, writing a test fails because the specified functionality doesn't exist, then writing the most straightforward code that can make the test pass, then refactoring to remove duplication, etc.	In BDD, creating an executable specification that fails because the feature doesn't exist, then writing the most straightforward code that can make the spec pass. You repeat this until a release candidate is ready to ship.
TDD tests are written using programming languages such as Java, .Net, Python, Ruby, etc.	BDD tests are written in a human-readable format using Given-When-Then steps. These tests are readable and understandable by non-technical persons also.
TDD tests are difficult to read by non-programmers as they are written in specific programming languages.	BDD tests are readable by non-programmers also as they are written in a human-readable format.
The critical difference between TDD and BDD is the scope. TDD is a development practice.	On the other hand, BDD is a team methodology.
In TDD, the developers write the test cases.	In BDD, the automated specifications are created by users or testers then the developers wiring them to the code under test.

47) What do you understand by cucumber dry run?

Cucumber dry run is used to compile cucumber features files and step definitions. It is run to find any compilation errors. If it finds anyone, it will show when we use dry run.

48) How can you use the Options tag in the Cucumber framework?

In the Cucumber framework, the Options tag is a part of the TestRunner file and comes in the form of an annotation called @CucumberOptions. It contains two parameters feature and glue.

- **Feature parameter:** The feature parameter is used to specify the path of the feature file.
- **Glue parameter:** The glue parameter is used to specify the path of the step definition file.

See the code implementation of TestRunner file with Option tag:

```
import org.junit.runner.RunWith;
import cucumber.api.CucumberOptions;
import cucumber.api.junit.Cucumber;
@RunWith (Cucumber.class)
@CucumberOptions (
features = "src/test/java/features ",
glue = {"stepDefinitions"}
)
public class TestRunner {
}
```

We have to import org.junit.runner.RunWith for the @RunWith annotation and cucumber.api.CucumberOptions for the @CucumberOptions annotation.

48) What are the two build management tools that can be integrated with Cucumber?

Following are the two build management tools that can be integrated with Cucumber:

- Gradle
- Maven

49) What is the use of hooks in the Cucumber framework?

In the Cucumber framework, the hooks are used to control the flow of the program and optimize lines of code. A block of code tagged with hooks in Cucumber can run before or post a scenario with the help of @Before and @After annotations.

Some scenarios may require certain preconditions for execution, such as launching the application, establishing a database connection, configuring the test data, and so on. Also, certain postconditions should be executed, such as terminating database connection, closing the browser, refreshing test data, application log out, and so on. All these conditions are handled in Cucumber with the help of the hooks. The @Before hook executes before the actual scenario, and the @After hook executes after the actual scenario even if the test corresponding to the actual scenario fails.
For the @Before annotation, we have to import cucumber.api.java.en.Before and for the @After annotation, we have to import cucumber.api.java.en.After.

See the code implementation with hooks:
The feature file: In this feature file, we have included two scenarios:
Feature: Invoice Generation.
Scenario 1: Verify the invoice generates in pdf format.
Given User navigates to Invoice Page.
Then User should be able to open the invoice in pdf format.
Scenario2: Verify the invoice fields.
Given User opens the invoice.
Then User should be able to see the payment amount in invoice.
See the step definition file:

```
import cucumber.api.java.en.Given;
import cucumber.api.java.en.Then;
import cucumber.api.java.en.Before;
import cucumber.api.java.en.After;
public class InvoiceValidation {
@Given ("^User navigates to Invoice Page$")
public void navigate_to_invoice () {
     System.out.println ("Navigation to Invoice Page");
  }
@Then ("^User should be able to open the invoice in pdf format$")
public void open_invoice_pdf () {
      System.out.println ("Invoice opened in pdf format ");
      }
@Given ("^User opens the invoice$")
public void open_invoice () {
      System.out.println ("Invoice is opened");
    }
@Then ("^User should be able to see the payment amount in invoice$")
public void verify_payment_invoice () {
      System.out.println ("Invoice contains payment amount");
}
@Before
    public void precondition () {
        System.out.println ("The method runs before all scenarios");
    }
@After
    public void postcondition () {
        System.out.println ("The method runs after all scenarios ");
    }
}
```

In the above example, you can see that we have used two scenarios, the test method precondition with @Before annotation and the test method postcondition with @After annotation.

50) What Is A Profile In Cucumber?

You can create Cucumber profiles to run a set of features and step definitions. Use the following command to execute a cucumber profile.

```
cucumber features -p <profile_name>
#Example: cucumber features -p acceptance
```

51) What Are Before, After, Beforestep And Afterstep Hooks?

Before: executes before the feature file execution.
After: executes after the feature file execution.
BeforeStep: executes before the each step execution.
AfterStep: executes after the each step execution.

52) What Are Cucumber Tags? And Why Do We Use Them?

Cucumber tags help in filtering the scenarios. We can tag the scenarios and then run them based on tags.

We can add tags to scenarios with <@> symbol.

We can use the following command to run a cucumber tagged scenario.

```
cucumber features -t @<tag_name>
Example: cucumber features -t @test
```

53) What Is The Purpose Of Cucumber Dry-run?

We use to compile the cucumber feature files and step definitions. If there occur any compilation errors, then it shows them on the console.

54) What If You Don't Use The Cucumber Keywords In Test Steps?

Please note that it's not mandatory to write keywords in test steps.
For example, we can build a test step like the one shown in the next line.
e.g.- We are testing using cucumber.

55) When Would You Use Rspec And When To Use Cucumber?

RSpec is more successful in doing unit testing.
As you know that Cucumber is a behaviour-driven development tool. So you can use it for System and Integration testing.

56) What Are The Steps To Generate A Report In Cucumber?

We run the following command to produce HTML reports.

```
cucumber <featurename>.feature --format html --out report.html --format pretty
```

57) What Is The Right Way To Execute A Specific Scenario From The Feature File?

We can select the target scenario from a feature file by providing its line number.

```
cucumber features/test.feature:10 --format html > testfeature.html
```