1. **What is Cucumber?**

Cucumber is a framework which can automate BDD (Behavior Driven Development) scenarios. Cucumber understands a language called Gherkin. Initially cucumber is written in Ruby programming language. Later it supported different language implementations like Java, PHP, Python, Dot Net etc.

1. **What are the advantages of cucumber?**

As cucumber is written in simple user understandable language called Gherkin we have many advantages.

1. We can involve business stakeholders who cannot code.
2. Here the priority is, End user experience.
3. Code reusability.
4. **What is BDD?**

Behavior Driven Development is a process of developing software based on TDD (Test Driven Development) which focusses on behavior rather than implementation.

1. **What are BDD supported tools?**

There are many tools that support BDD. Most famous tools are:

1. Cucumber
2. Jbehave
3. Nbehave
4. SpecFlow
5. **What is Gherkin language and its syntax?**

Gherkin is a format language for cucumber. It is a business readable language which can be understood easily by everyone as it is written in plain text. Gherkin has some spaces and indentation to define structure.

Gherkin Syntax:

1. Feature
2. Background
3. Scenario
4. Given
5. When
6. Then
7. And
8. But
9. Scenario outline
10. Examples
11. Scenario Templates
12. **Difference between BDD and Traditional automation?**

Gherkin is a format language for cucumber. It is a business readable language which can be understood easily by everyone as it is written in plain text. Gherkin has some spaces and indentation to define structure.

|  |  |
| --- | --- |
| BDD | Traditional Automation |
| 1. It is written in plain text and easy to understand. | 1. It is written in code format and hard to understand. |
| 2. It can be understood by any level of employers in a project like BA, QA, Automation Test engineer etc. | 2. Code can be understood only by automation engineer.(Sometimes developers can) |
| 3. Since it’s a plain text format, BDD can be shared to stakeholders | 3. It’s impossible |
| 4. Easy to implement. | 4. More knowledge is required while desingning. |

1. **What are installation pre requisites for cucumber (in JAVA implementation)?**

Intellij IDE

Maven

Jar files-

* Cucumber core
* Cucumber java
* Cucumber java deps
* Cucumber jvm
* Gherkin
* Selenium related jars

1. **What are the 2 files required to execute a Cucumber test scenario?**

* Features
* Step Definition

1. **What is a feature file?**

Feature file is more like a plain text file but with ***.feature*** extension.

**10.  What does feature file consist of?**

Feature file in cucumber consist of conditions required for executing code, they are

* Feature
* Scenario
* Scenario Outline
* Given
* When
* Then

**11. Explain what is Scenario Outline in feature file?**

There are situations where we may need to run same scenario another time with different data. Using scenario outline, same scenario can be executed for multiple sets of data. The data is provided by a tabular structure separated by ( | | ).

**12. What is the language used for expressing scenario in feature file?**

Gherkin language is used to express scenario in feature file.

**13. What is step definition?**

Step definition files are java class files which holds all the steps written with feature file.

**14. Give an example of behavior driven test in plain text?**

1. **Feature:**Login Feature
2. **Scenario:**Login with correct credentials
3. **Given:**  I navigate to the login page (Prerequisites)
4. **When:**I give username and password and Click on login (Action)
5. **Then:  Login** Result will be shown. (Validation)

**15. What are the differences between Jbehave and Cucumber?**

Although Cucumber and Jbehave are meant for the same purpose, acceptance tests are completely different frameworks

* Jbehave is Java based and Cucumber is Ruby based
* Jbehave are based on stories while Cucumber is based on features

**16. Explain what is test harness?**

A test harness for cucumber and rspec allows for separating responsibility between setting up the context and interacting with the browser and cleaning up the step definition files.

**17. Explain when to use Rspec and when to use Cucumber?**

* Rspec is used for Unit Testing.
* Cucumber is used for behavior driven development. Cucumber can be used for System and Integration Tests.

**18. Explain what is regular expressions?**

A regular expression is a pattern describing a certain amount of text.  The most basic regular expression consists of a single literal character.

**19. What software do you need to run a Cucumber Web Test?**

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

**20. How to work with multiple data?**

Instead of passing hard coded values sometimes we need to supply multiple data. This can be done using ***DataTable*** class available in cucumber.

**21. What are cucumber hooks?**

In cucumber we have 2 hooks:

@Before

@After

**22. Is Dependency/context Injections already installed in Java? If not how?**

In cucumber for java, dependency injection is not built out-of-box in cucumber-jvm, rather we need to add dependency called ***cucumber-picocontainer.***

This is how it looks:

Private Base page;

Pubic LoginStep(Base base) {

Page = base; }

**23. What is Argument Transformation?**

A step argument transformation can simplest be described as a way to get typed object for a certain pattern in a Given/When/Then. It allows transformation of a step definition argument to a custom type giving us full control over how that type is instantiated.

**24. How to achieve Argument Transformation?**

In cucumber java argument transformation comes out of box and we will achieve it using Transformer class (Transformer<String>).

a. Extend Transformer<> class for the type to be transformed.

b. Override transform method.

c. Call @ Transform annotation as parameter in calling method param.

**25. What is the new feature of Cucumber-Java8?**

Cucumber-java8 is another branch of cucumber-java in maven. They both share same versions. The major difference is in the step definition representation using lamda expression.

**26. How to add selenium dependency?**

Before starting to work with selenium, its important to add selenium jar files. We do it in ***pom.xml***.

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-jaava</artifactId>

<version>2.53.0</version>

</dependency

**27. How to execute application out of Intellij IDE?**

By using ***Maven*** we can execute application out of Intellij IDE rather running within IDE. To do this:

a. Download Maven

b. Set home path

c. Run Maven against project

We need to do following changes to the project as well, mainly in POM.xml

a. Add Maven compiler plugin

b. Add Maven surefire plugin

**28. What is Surefire plugin?**

The surefire plugin is used during the test phase of the build lifecycle to execute the unit tests of an application. It generates reports in two different file formats:

a. Plain text files (\*.txt)

b. XML files (\*.xml)

By default, these files are generated at ***${basedir}/target/surefire-reports.***

**29. What are reporting options supported by cucumber?**

a. HTML

b. json

c. Text

**30. How to write cucumber comments?**

Comments are not for execution. It is to make code understandable and readable. To place the comments just begin the statements with #.

**31. Why cucumber testing with Ruby?**

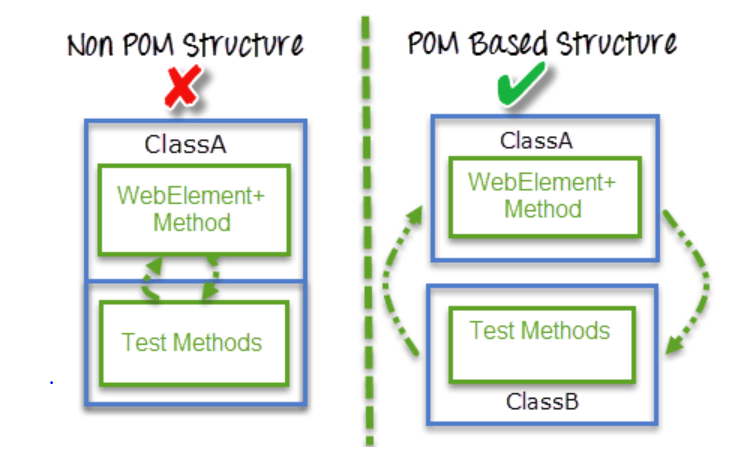
Ruby is an object-oriented language and easy to recognize. It is a powerful class library which has huge online support.

**32. What is background in cucumber?**

Background in Cucumberis used to define a step or series of steps which are common to all the tests in the feature file. It allows you to add some context to the scenarios for a feature where it is defined. A Background is much like a scenario containing a number of steps. But it runs before each and every scenario where for a feature in which it is defined.

**33. Why Page Object Model?**

There have been several problems with the traditional automation methods. Most common is the script maintenance which gets tougher with time as the test suite grows. Next, when some locator changes, we need to go over the whole source code to adjust locators. Another problem is the duplicate code which emanates from the repeated functionality**.** The latter also leads to the duplicity of locators which makes the test code inefficient. Finally, the unnecessary code increases the cost of maintenance for the entire project.



**34. What Is Page Object Model (POM)?**

Page Object Model is a design pattern to define container classes for web elements that behave as object repositories. In this model, each page has its private page object class to access UI locators.

The Page class maps all elements of the corresponding web page as its private members. Also, it defines public methods to manage operations on these objects. To increase readability – it allows the method name to set as per the task they are performing. For example, we can use a method name like waitForLoginSuccess() which waits until it gets a success message for the login operation.

**35. What are the advantages of POM?**

1.  You can segregate the UI operations from functional flows. This makes the code clean and easy to manage.  
2. It lets the test cases to be written independently of the object repository. So the same object repository can be re-used for a different purpose. For example, you can use the POM-based repository for functional testing with TestNG/JUnit and for acceptance testing with JBehave/Cucumber.  
3. Here, we have reusable page methods in the POM classes so that you can focus on the code optimization.  
4. You can use more sensible method names which can be easily mapped to the UI actions. For example, if you click the login button and moves to the dashboard page. Then you can give the method name as ‘gotoDashboard()’.

**36. What is Page Factory?**

Page Factory is an inbuilt Page Object Model concept for Selenium WebDriver but it is very optimized. Here as well, we follow the concept of separation of Page Object Repository and Test Methods. Additionally, with the help of Page Factory class, we use annotations @FindBy to find WebElement. We use initElements method to initialize web elements.

@FindBy can accept tagName, partialLinkText, name, linkText, id, css, className, xpath as attributes.

**37. How to implement POM?**

**Simple POM:**

It's the basic structure of Page object model (POM) where all Web Elements of the **AUT** and the method that operate on these Web Elements are maintained inside a class file. A task like **verification** should be **separate**as part of Test methods.