**1. What is Cucumber?**

* Cucumber is a testing tool based on BDD(Behavior Driven Development) framework.
* Cucumber reads executable specifications written in plain text and validates if the software is able to follow those specifications.
* In simple terms, cucumber help Software tester to develop test cases in plain text to validate software behavior.

**2. What is Gherkin language?**

* Gherkin is a plain/simple English language used to define application/software behavior.
* It's a domain-specific language for defining specs tests in Cucumber format.
* Gherkin serves multiple purposes: Automated testing using Cucumber, Document how the system actually behaves.

**3. What do you meant by a feature file?**

* A feature file must provide a high-level description of an Application feature 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.
* For each feature under test, it is recommended that a separate feature file should be created. The feature file must have the extension ".feature."

**4. What is Step definition?**

* Step Definition is a method/function/block with an expression that links it to one or more Gherkin steps. When Cucumber executes a Gherkin step in a scenario, it will look for a matching step definition to execute.



**5. What do you mean by Scenario and Scenario Outline?**

* **Scenario**is a fundamental Gherkin structure. Every scenario begins with the keyword "Scenario:" and ends with a scenario title. Every feature file can have one or more scenarios, each of which has one or more steps.  
  ***Scenario***: Sunday isn't Friday  
      Given today is Sunday  
      When I ask whether it's Friday yet  
      Then I should be told "Nope"
* **Scenario Outline** is a way to parameterize the scenarios. This is ideally used when the same scenario needs to be executed for multiple sets of data. Scenario Outline must be followed by the keyword ‘Examples’, which specify the set of values for each parameter.  
  ***Scenario Outline*:** eating  
    Given there are <start> cucumbers  
    When I eat <eat> cucumbers  
    Then I should have <left> cucumbers  
    *Examples:*  
      | start | eat | left |  
      |    12 |   5 |    7 |  
      |    20 |   5 |   15 |

**6. Different keywords used in Cucumber to write a scenario?**

* Given: Pre Condition
* When: User Actions
* Then: Expected Output
* And: Multiple user actions with positive statements
* But: Multiple user actions with negative statements

**7. What is the meaning of Test Runner class file?**

* It is used to bind Feature and Step definition file. With the help of Test Runner file we can control and manipulate execution as per requirement.

**8. Which are the 2 annotations used within TestRunner class file?**

* @RunWith: It tells JUnit that tests should run using Cucumber class present in 'Cucumber.api
* @CucumberOptions: It can be used to provide additional configuration to the runner. This annotation tells Cucumber a lot of things like where to look for feature files, what reporting system to use etc.

**9. Different properties supported by CucumberOptions?**

-cucumber.ansi-colors.disabled  
-cucumber.execution.dry-run  
-cucumber.execution.limit  
-cucumber.execution.order  
-cucumber.execution.strict  
-cucumber.execution.wip  
-cucumber.features  
-cucumber.filter.name  
-cucumber.filter.tags  
-cucumber.glue  
-cucumber.plugin  
-cucumber.object-factory  
-cucumber.snippet-type

**10. What is Datatable in Cucumber?**

- Data Tables are handy for passing a list of values to a step definition. Cucumber provides a rich API for manipulating tables from within step definitions. Data Tables are also used to handle large amount of data. Tables in Cucumber feature files are represented by using the pipeline “|” sign.

We can use data table under scenario OR scenario Outline.

Possible ways to convert DataTable:  
asMaps()  
asLists()  
asList()  
asMap()

**11. What is the use 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 given statements are repeated in each scenario of the feature file. A Background is placed before the first Scenario/Example, at the same level of indentation.  
  
*Feature*: Multiple site support  
   Only blog owners can post to a blog, except administrators,  
   who can post to all blogs.

*Background:*  
    Given a global administrator named "Greg"  
    And a blog named "Greg's anti-tax rants"  
    And a customer named "Dr. Bill"  
    And a blog named "Expensive Therapy" owned by "Dr. Bill"

*Scenario*: Dr. Bill posts to his own blog  
            Given I am logged in as Dr. Bill  
            When I try to post to "Expensive Therapy"  
            Then I should see "Your article was published."

**12. What are cucumber tags and where we use them?**

-Tags are a great way to organize your features and scenarios. It also help in filtering the scenarios. We can group the scenarios using tags and then run them based on tags. It can be tagged with Feature, Scenario, Scenario outline and Examples using @ symbol.

@billing  
*Feature*: Verify billing

@important  
*Scenario*: Missing product description  
            Given hello  
  
@Compatibility  
*Scenario Outline:*Steps will run conditionally if tagged  
            Given user is logged in  
            When user clicks <link>  
            Then user will be logged out

@mobile  
*Examples*:  
| link                               |  
| logout link on mobile |  
  
@desktop  
*Examples*:  
| link                                 |  
| logout link on desktop |

**13. What are Hooks?**

-Hooks are blocks of code that can run at various points in the Cucumber execution cycle. They are typically used for setup and teardown of the environment before and after each scenario.

**14. Different Hooks in Cucumber.**

- *Scenario Hooks*: Run for every scenarios.  
Two types of Scenario hooks,  
*-Before Hook:*Run before the first step of each scenario  
*@Before*  
public void doSomethingBefore() {  
}  
*-After Hooks*: Run after the last step of each scenario, even when the step result is failed, undefined, pending, or skipped.

*@After*  
public void doSomethingAfter(Scenario scenario){  
}

            -Step Hooks: Run for every steps.

Two types of Step hooks,  
-BeforeStep Hook: Run before each step.  
@*BeforeStep*  
public void doSomethingBeforeStep(Scenario scenario){  
}  
-AfterStep Hook: Run after each step.  
@*AfterStep*  
public void doSomethingAfterStep(Scenario scenario){  
}

**15. What is Cucumber expression and different parameter types used with Cucumber expression?**

            -Cucumber Expressions is an alternative to Regular Expressions with a more intuitive syntax. Cucumber supports both Cucumber Expressions and Regular Expressions for defining Step Definitions, but you cannot mix Cucumber Expression syntax with Regular Expression syntax in the same expression.

*Parameter types*:  
{int}, {float}, {word}, {string}, {} anonymous  
{int}:  
I have 42 cucumbers in my belly  
I have {int} cucumbers in my belly  
  
{float}:  
I have 42.5 cucumbers in my belly  
I have {float} cucumbers in my belly  
  
{word}:  
I have a red ball  
I have a {color} ball

{string}: Matches single-quoted or double-quoted strings

"banana split" or 'banana split'

**16. What is dry run in Cucumber?**

            -Dry run is to verify whether all feature file steps have corresponding step definitions. The value of dry run can be either true or false. The default value of dry run is false and it is a part of the Test Runner Class file.  
  
package com.example;  
import io.cucumber.junit.Cucumber;  
import io.cucumber.junit.CucumberOptions;  
import org.junit.runner.RunWith;

@RunWith(Cucumber.class)  
@CucumberOptions(dryRun=true)  
public class RunCucumberTest {  
}

**17. What is monochrome option in Cucumber?**

            - It is used to display output in the console in readable format. It basically takes 2 values(true and false), dafulat value is false. if the monochrome is set to,  
true: then the console output for the Cucumber test are much more readable and remove any unreadable character.  
false: then the console output is not as readable as it should be.

**18. Reporting in Cucumber?**

         - Cucumber uses reporter plugins to produce reports that contain information about what scenarios have passed or failed. The easiest way to get started with reporting is to use the Cucumber Reports service. Publishing to the Cucumber Reports service is currently supported in:  
Cucumber-JVM 6.7.0 and above  
Cucumber-Ruby 5.1.1 and above  
Cucumber-JS 7.0.0 and above

**19. What is strict option in Cucumber?**

            - Strict is used to identify undefined steps from execution. By default it is set to true.  if the strict is set to,  
false: At execution time if cucumber encounters any undefined/pending steps then cucumber does not fail the execution and undefined steps are skipped and BUILD is SUCCESSFUL.  
true: At execution time if cucumber encounters any undefined/pending steps then cucumber does fails the execution and undefined steps are marked as fail and BUILD is FAILURE

**20. What are the advantages of using Cucumber?**

* Cucumber is an open-source and free-to-use tool.
* Cucumber supports a variety of programming languages, including Java.net and Ruby.
* It serves as a link between commercial and technical language. This can be done by writing a test case in plain English text
* You can involve business stakeholders who cannot code.
* Code reusability.

**21. What is the purpose of Glue parameter within CucumberOptions annotation?**

* The glue parameter is used to specify the path of the step definition file.

**22. What is BDD and how Cucumber complies with BDD?**

     -BDD is a Behavior Driven Development approach of the software development process where Cucumber works in BDD way of approach. The tests in BDD are written in a human-readable format and understandable which are easier to implement. The BDD tests are non-programming and easier to write. The Behavior Driven Development process is highly efficient and useful in resolving complex solutions and requirements in the project to handle smoothly.

**23. Explain the test harness?**

            -A test harness for Cucumber and RSpec enables a separate responsibility between the context setup and interacting with the browser cleaning up the step definition files.

**24. Name any 2 testing framework that can be integrated with Cucumber?**

* TestNG
* JUnit

**25. Name any 2 build management tools that can be integrated with Cucumber?**

* Gradle
* Maven

**26. How to run cucumber test using maven ?**

A. I can run cucumber TestRunner class using maven SureFire pluggin. TestRunner class must have the word Test then only maven will execute the test.

**27. how to run cucumber test on jenkins ?**

A. in jenkins set the source code path of pom.xml file path or git project url then set build periodically usin cron timer.

**28. how to generate extent report in cucumber with screeshot ?**

A. <https://www.toolsqa.com/extent-report/extent-report-for-cucumber-junit-project/>

B. <https://www.numpyninja.com/post/step-by-step-implementation-cucumber-bdd-framework-using-java-selenium-jenkins-git-extent-report>