In BDD (Behavior-Driven Development) using Cucumber, we use Gherkin syntax to write test cases in plain language. Gherkin defines several keywords that structure the scenarios. These keywords make your tests readable and act as a bridge between business and technical people.

Primary focus of BDD is collaboration between stakeholders like - Product Owner, Business Analyst, QA and Developers.

✅ Core Cucumber Keywords

Keyword	Purpose
Feature	Describes the functionality being tested
Scenario	Describes a specific example of a feature
Given	Describes the initial context or preconditions
When	Describes an action or event
Then	Describes the expected outcome or result
And	Used to add more steps (can be used with Given, When, Then)
But	Used to introduce a negative or contrasting condition
Background	Defines steps common to all scenarios in a feature
Scenario Outline	Defines a template for multiple scenarios with different values
Examples	Provides the values for Scenario Outline placeholders

---

Feature: User Login

  Background:
    Given the user is on the login page

  Scenario: Successful login
    When the user enters valid credentials
    And clicks on the login button
    Then the user should be redirected to the dashboard

  Scenario Outline: Unsuccessful login
    When the user enters "<username>" and "<password>"
    And clicks on the login button
    Then an error message should be displayed

    Examples:
      | username | password |
      | user1 | wrong123 |
      | test2 | 12345 |

What is the language used for expressing scenario in feature file?
The language used for expressing scenarios in a feature file is called **Gherkin**.

What is Gherkin?
- **Gherkin** is a plain-text, structured language designed to describe software behaviors in a simple, readable way.
- It uses keywords like **Feature**, **Scenario**, **Given**, **When**, **Then**, **And**, and **But** to define test cases.
- It allows both technical and non-technical people to understand the test scenarios easily.

✅ What is a Feature File in BDD?

A Feature file is a plain text file written in Gherkin language used in Behavior Driven Development (BDD). It describes the expected behavior of the application in simple English using a structured format that both technical and non-technical stakeholders can understand.

What is a Step Definition file in BDD?

A Step Definition file is a Java class that maps Gherkin steps written in your .feature file to actual code implementation that performs the actions (like clicking a button, entering text, etc.).
Cucumber finds the Step Definition file with the help of the Glue code in Cucumber Options.

---

what is Test Runner file? Defination

A Test Runner file is a Java class used in Behavior Driven Development (BDD) frameworks (like Cucumber) that is responsible for:
📃 Initiating the execution of Cucumber feature files and linking them to their corresponding step definitions and configuration settings.

```
//code
@CucumberOptions(
    features = "src/test/resources/features",
    glue = "com.amazon.tests.stepdefinitions",
    plugin = {
        "pretty",
        "html:target/cucumber-reports/cucumber.html",
        "json:target/cucumber-reports/cucumber.json"
    },
    monochrome = true,
    dryRun = false,
    tags = "@Smoke"
)
```

Option	Description
features = "src/test/resources/features"	Tells Cucumber where to find the .feature files (written in Gherkin).
glue = "com.amazon.tests.stepdefinitions"	Specifies the package where step definition files (Java methods for steps) are located.
plugin = {"pretty", "html:target/cucumber-reports/cucumber.html"}	Adds plugins for output:• pretty – readable console output• html – generates a detailed HTML report in target/cucumber-reports.
monochrome = true	Removes unreadable characters from console output (makes it cleaner and readable).
dryRun = false	If set to true, Cucumber will **check** if all steps have matching definitions, but **won't run tests**.Set to false to actually **run the tests**.
Tag	Running only scenarios tagged with @Smoke

> Adding JSON report generation alongside the HTML report

Who Creates Different BDD Files?

File Type	Who Usually Creates It	Why/Role
Feature Files	Business Owners, Business Analysts, Product Owners	Written in plain English (Gherkin) to define behavior and acceptance criteria clearly for all stakeholders. They describe what the system should do.
Step Definitions	QA Automation Engineers, SDETs	Translate feature file steps into automated test code that interacts with the application. They implement how to perform the actions.
Test Runners	QA Automation Engineers, SDETs	Configure and execute the automated tests, linking feature files and step definitions, and manage test execution settings and reports.

---

difference between a **Feature** and a **Scenario** in BDD:

📝 Feature vs Scenario in BDD

Aspect	Feature	Scenario
Definition	A high-level description of a functionality or a business requirement. It represents **what** part of the system you are describing or testing.	A specific example or test case within a feature that describes a particular behavior or flow. It represents **how** the feature should work in a particular situation.
Purpose	To group related scenarios that together describe the overall	To detail a concrete behavior or user story example within that feature.

Aspect	Feature	Scenario
	behavior of a particular feature of the application.	
Level of Detail	Broad, general description of functionality.	Detailed, specific example or use case.
Gherkin Keyword	Feature:	Scenario: or Scenario Outline:
Example	Feature: Login functionality Describes the login feature in general.	Scenario: Successful login with valid credentials Describes one specific behavior of the login feature.

Example:
Feature: Login functionality

  Scenario: Successful login with valid credentials
    Given the user is on the login page
    When the user enters valid username and password
    And clicks the login button
    Then the user should be redirected to the homepage

  Scenario: Unsuccessful login with invalid password
    Given the user is on the login page
    When the user enters valid username and wrong password
    Then an error message should be displayed

Summary:
- **Feature** groups multiple related **scenarios**.
- **Scenario** is an individual test case or example under that feature.

How do you handle data-driven testing in BDD scenarios?
Data-driven testing in BDD is a technique where a single scenario is executed multiple times with different sets of input data and expected results. This is achieved using the **Scenario Outline** and **Examples** keywords in a Gherkin feature file.

📌 **Purpose:**
To **avoid duplicating scenarios** and make tests **more scalable, readable, and maintainable** by using a table of inputs and expected outputs.

🔑 **Key Elements:**
- **Scenario Outline**: Defines a scenario template with placeholders.
- **Examples**: Provides a table of data sets to substitute in the placeholders.

Scenario Outline: Successful login with different users

Given user is on login page
When user enters username "<username>" and password "<password>"
Then user should see message "<message>"

Examples:
```
| username | password | message      |
| user1    | pass123  | Welcome user1! |
| user2    | pass456  | Welcome user2! |
```
Each row in the Examples table will run the same scenario with different data.

What is a Hook in Cucumber?
A Hook in Cucumber is a block of code that runs before or after each scenario, or before/after each step or tagged scenario.
Hooks help manage setup and teardown actions — like opening a browser, clearing cookies, or taking screenshots.

What is the Purpose of the Behaviour Driven Development (BDD) Methodology in the Real World?

Behaviour Driven Development (BDD) is a collaborative software development approach that helps teams build the right product by defining how the software should behave in plain language that both technical and non-technical stakeholders can understand.

What is the Use of the Background Keyword in Cucumber?
The Background keyword in Cucumber is used to define common steps that are repeated before every scenario in a feature file. It helps avoid duplication and keeps your scenarios clean, readable.

What Are Before, After, BeforeStep and AfterStep Hooks?
Methods annotated with @Before will execute before every scenario. Methods annotated with @BeforeStep execute before every step. Methods annotated with @After execute after every scenario. Methods annotated with @AfterStep execute after every step.

What Are Tags in Cucumber?
Tags in Cucumber are used to categorize and control the execution of scenarios or feature files. They are annotations (starting with @) placed above scenarios or features.

What Is Cucumber Dry Run?
Dry Run in Cucumber is a mode of execution used to verify that every step in your feature files has a corresponding step definition, without actually executing the tests.
🔍 Purpose of Dry Run:
- To check mapping between Gherkin steps and Java methods.
- To ensure that all steps are implemented.
- Useful during development to identify missing step definitions early.
- Helps in faster feedback without launching the browser or executing actual tests.

What is the meaning of Steps in Cucumber tool?
Each step starts with Given, When, Then, And. Cucumber executes each step in a scenario one at a time, in the given sequence. When Cucumber tries to execute a step, it looks for a matching step definition to execute.

What are the programming languages supported by Cucumber?
Cucumber supports many programming languages like Perl, PHP, Python, .Net and Java

How Many Times Will a Scenario Outline Be Executed in Cucumber?
A Scenario Outline in Cucumber is executed once for each set of data provided in the Examples section.

What Is Background in Cucumber and When Is It Executed?
🏷 What is Background?
In Cucumber, the Background keyword is used to define common steps that are executed before each scenario in a feature file.

Types of Hooks in Cucumber
In **Cucumber**, **hooks** are special blocks of code that run **before or after** certain events during test execution. They are used for **setup** and **teardown** activities.
Hooks are defined in step definition classes using annotations from io.cucumber.java (for Java).

Hook Type	When It Executes	Common Use
@Before	Before each scenario	Browser launch, config setup
@After	After each scenario	Closing browser, cleanup, reports
@BeforeStep	Before each step	Logging, dynamic data setup
@AfterStep	After each step	Screenshot, logging, reporting

What Is the Pattern of Writing Given, When, Then, And in Cucumber?
In Cucumber, test scenarios are written in **Gherkin syntax**, which uses the keywords:
- Given – to describe the **initial state or setup**
- When – to describe the **action or event**
- Then – to describe the **expected result**
- And / But – to **extend** the previous step (used with any of the above)

What Is the Use of glue Property in @CucumberOptions?

In Cucumber (Java), the glue property inside the @CucumberOptions annotation tells Cucumber where to find the step definitions and hooks related to the Gherkin steps in the feature files.

```
@CucumberOptions(
    features = "src/test/resources/features",
    glue = "com.amazon.tests.stepdefinitions",
```

```
    plugin = {"pretty", "html:target/cucumber-reports"},
    monochrome = true
)
```

How to Comment a Line in a Cucumber Feature File?
In Cucumber (Gherkin syntax), you can **comment a line** by starting it with a # symbol.

Cucumber Tags are case sensitive. True or False?
Tag names are case-sensitive.

Name any two testing frameworks that can be integrated with Cucumber?
TestNG and Junit can be easily integrated with Cucumber.

Name any two build management tools that can be integrated with Cucumber?

Maven and Ant are two build management tools which can be integrated with cucumber

---

Name any advanced framework design that can be used with Cucumber?
Page Object Model can be used with Cucumber

Can we write cucumber tags (@smoke , @Run etc) above feature keyword in feature file?

Yes, you can write Cucumber tags above the Feature keyword, but it depends on the version you're using. It's always safe to place them above Scenario

@smoke @login
Feature: Login functionality for Amazon

  @positive
  Scenario: Login with valid credentials
    Given the user is on the login page
    When the user enters correct username and password
    Then the user should be redirected to the homepage

What is the real time use of Cucumber?
Cucumber is used to write acceptance tests for a web application.

(**Acceptance testing** is a final check to make sure the application is doing what the business wants it to do, before delivering it to users. In BDD, these tests are often written in plain English using tools like **Cucumber**.)

If Before hook is available and background is available for a scenario, in which order they will be executed?

Before hook will be executed first followed by the background steps for a scenario.

How to run multiple feature files in Cucumber?
We can mention all the different feature file paths or the folder path in Runner file.

To run **all features**, just give the **folder path**.
To run **specific features**, list the **feature file paths** explicitly.
This flexibility helps you run selective tests or all tests based on your need.

How to create feature file in Cucumber?
Create a new file with .feature extension under src/test/resources/features folder.

How to run Cucumber tests in parallel?
Cucumber can be executed in parallel using TestNG and Maven test execution plugins by setting the DataProvider parallel option to true. Using the scenario outline we can execute multiple scenarios with TestNG. One can use either Maven SureFire plugin for executing the tests in parallel.

Method	How It Helps
TestNG Parallelism	Runs multiple tests/scenarios simultaneously using threads.
Maven Surefire	Controls parallelism during build/test run.
Scenario Outline	Runs multiple test cases with different data.

Is Cucumber open source?
 Yes, Cucumber is an open source BDD framework.

What is the starting point of execution for feature files?
TestRunner is the starting point of execution.

Should any code be written within TestRunner class?
In a default TestRunner class, no code is required to execute it.

Can we use same step definition in different scenarios?
Yes. We can use same step definition in multiple scenarios.

What is the maximum number of steps that are to be written within a scenario?
There is no fixed limit of steps within a scenario and it is completely dependent on the application.

What software do you need to run a Cucumber Tests?
We need any IDE like Eclipse or IntelliJ and setup a Maven Java Project. Then we need to add maven dependencies for cucumber-JVM, cucumber-core and cucumber-Junit. After that we need to add Feature Files, Step Definition Files and Test Runner file to run a cucumber test.

On what places you can write tags in feature file?
Tags can be placed above the following elements:
Feature
 Scenario
 Scenario Outline
 Examples

What are the prerequisites for building a Selenium Cucumber automation framework?
We need to include the relevant Maven dependencies for Selenium and Cucumber.

What are Data Tables in Cucumber?
- **Data Tables** are a way to pass multiple sets of data to a step in a scenario.
- Data Tables are used in Examples for Scenario Outline. Scenario will be executed for every row in the data table

What are Cucumber Assertions?
- **Cucumber itself does not provide its own assertion library.**
- Instead, you use assertions from testing frameworks like **TestNG**, or others in your **step definition code**.
- Assertions are used to **verify** that the application behaves as expected after a step is executed.

What is Cucumber Report?
Mention the benefits of Cucumber Report? Cucumber Report contains details of the test execution like which scenarios passed or failed, what are the steps within the scenarios, and other environment details. This report can be shared with the project stakeholders for reporting purpose.

Can we or How to integrate TestNG with Cucumber?
- In your **Test Runner class**, extend the class AbstractTestNGCucumberTests provided by the Cucumber TestNG integration.
- This allows you to run your Cucumber feature files using TestNG.
- You can also leverage TestNG features like parallel execution, test configuration, and reporting.

hat is the main difference between Scenario and Scenario outline?
-Scenario can be executed only once but Scenario outline can be used to execute scenario multiple times with different sets of data.
Scenario outline contains example with data table, but scenario doesn't contain example

Mention the main reasons behind using a simple programming language such as Gherkin?
Gherkin is used in Cucumber to make it easy for non-technical people to understand the scenarios as it is quite similar to English language. Business Analysts can easily convert the business requirements into- Features/Scenarios/Steps using Gherkin .