1. **Acronym of ANT and POI?**

**POI** stands For “**Poor Obfuscation Implementation**”. Apache POI is an API provided by Apache foundation which is a collection of different java libraries. This libraries gives the facility to read, write and manipulate different Microsoft files such as excel sheet, power-point, and word files.

**Apache Ant** is a software tool for automating software build processes, which originated from the Apache Tomcat project in early 2000. It was a replacement for the Make build tool of Unix, and was created due to a number of problems with Unix's make.[2] It is similar to Make but is implemented using the Java language, requires the Java platform, and is best suited to building Java projects.

The most immediately noticeable difference between Ant and Make is that Ant uses XML to describe the code build process and its dependencies, whereas Make uses the Makefile format. By default, the XML file is named build.xml.

**Explain ANT/MAVEN and what are its pros and cons?**

Con Maven

-------------

\* You need to know maven command line or use an ide that has maven integration, such as netbeans or eclispe.

\* Learning Curve (there are something like 20-26 different build phases, plus packaging types).

\* Your project pretty much has to be laid out the correct way, you can work around this but you shouldn't.

\* Its verbose and complex.

\* If you have a dependent jar that isn't mavenized, you might lose your mind before you figure out how to integrate it.

Pro Maven

------------

\* All dependencies are downloaded automatically;

\* All compile/build/dependency info is bundled with your maven spec, and therefore forced to be in source control with your code. (this is a huge plus)

\* Drastically simplifies the way your build will work, because every build follows a generally standard procedure.

\* Every member of your team will be building/deploying in the same way with every compile.

\* Turning on new features (such as junit) across the board for things like CI/CD can be as easy as one line in your maven config file.

Pro Ant

=====

\* You control everything, because you have to largely build it yourself (though you can copy and paste).

\* Easy to ad support for legacy builds.

\* Ant is a programming language (almost) so you can do crazy things, but it requires much more debugging

Con Ant

======

\* IDE integration doesn't seem to be as well done. Running your ant scripts with \*every build\* requires more setup and configuration, and generally on every machine.

\* Its no longer the standard, and many projects which are mavenized are much easier to use via maven than downloading the jar and getting it into your lib. then doing wire up via ant.

Ant is just a build tools , while Maven is much more then that.

With maven you can manage dependencies and do a lot of other stuff with the help of plugins.

**1438.Is JVM platform independent?**

**JVM translates bytecode into machine language**

Every Java program is compiled into an intermediate language called Java bytecode. The JVM is used to both translate the bytecode into the machine language for a particular computer, and actually execute the corresponding machine-language instructions as well. The JVM and bytecode combined give Java its status as a "portable" language.

**Machine language is OS dependent**

Given the previous information, it should be easier to deduce an answer to the question. Since the JVM must translate the bytecode into machine language, and since the machine language depends on the operating system being used, it is clear that the JVM is platform (operating system) dependent. This fact can be verified by trying to download the JVM – you will be given a list of JVM’s corresponding to different operating systems, and you will obviously pick whichever JVM is targeted for the operating system that you are running.

**1437.How do u say JAVA is platform independent?**

Java is a platform independent programming language, Because when you install jdk software on your system then automatically JVM are installed on your system. For every operating system separate JVM is available which is capable to read the .class file or byte code. When we compile your Java code then .class file is generated by javac compiler these codes are readable by JVM and every operating system have its own JVM so JVM is platform dependent but due to JVM java language is become platform independent.

1. **Explain the features of JAVA?**

Robust(Reliable),Secure,Platform Independent,Famliar,Simple,Multi Threaded environment

1. **What is Cucumber?**

Cucumber is a tool that supports Behavior Driven Development (BDD). It offers a way to write tests that anybody can understand, regardless of their technical knowledge.

In BDD, users (business analysts, product owners) first write scenarios or acceptance tests that describes the behavior of the system from the customer's perspective, for review and sign-off by the product owners before developers write their codes.

Cucumber use Ruby programming language.

Advantages of Cucumber

It is helpful to involve business stakeholders who can't easily read code

Cucumber Testing focuses on end-user experience

Style of writing tests allow for easier reuse of code in the tests

Quick and easy set up and execution

Efficient tool for testing

**1442.What are the 2 files required to execute a Cucumber test scenario?**

Two files required to execute a Cucumber test scenario are

Features

Step Definition

**1443.What language is used by Cucumber?**

**Gherkin** is a Business Readable, Domain Specific **Language** created especially for behavior descriptions. It gives you the ability to remove logic details from behavior tests. **Gherkin** serves two purposes: serving as your project's documentation and automated tests.

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

A **Feature File** is an entry point to the **Cucumber** tests. This is a **file** where you will describe your tests in Descriptive language (Like English). ... A **feature file** can contain a **scenario** or can contain many scenarios in a single **feature file** but it usually contains a list of scenarios

**1446 . What are the various keywords that are used in Cucumber for writing a scenario?**

The primary keywords are:

Feature

Rule (as of Gherkin 6)

Example (or Scenario)

Given, When, Then, And, But (steps)

Background

Scenario Outline (or Scenario Template)

Examples

There are a few secondary keywords as well:

""" (Doc Strings)

| (Data Tables)

@ (Tags)

# (Comments)

**What is Scenario Outline in Cucumber and its purpose?**

**Scenario outline** basically replaces variable/keywords with the value from the table. Each row in the table is considered to be a scenario.

**Feature:** Free CRM Create Contacts

**Scenario Outline:** Free CRM Create a new conatct Scenario

*Given* User is already on login Page

*When* User title of login page is Free CRM

*Then* User enters "*<username>*" and "*<password>*"

*Then* User clicks on login button

*Then* User is on Home Page

*Then* User moves to new Contact Page

*Then* User provides "*<firstname>*" and "*<lastname>*" and "*<position>*"

*Then* Close the browser

**Examples:**

|username|password|firstname|lastname|position|

|preethik|Liveall@456|Tom|Peter|Manager|

|preethik|Liveall@456|David|DSouza|Director|

**What programming language is used by Cucumber?**

Cucumber was originally written in the Ruby programming language. and was originally used exclusively for Ruby testing as a complement to the RSpec BDD framework. Cucumber now supports a variety of different programming languages through various implementations, including Java. and JavaScript.

**What is the purpose of Step Definition file in Cucumber?**

Step definition maps the Test Case Steps in the feature files(introduced by Given/When/Then) to code. It executes the steps on Application Under Test and checks the outcomes against expected results. For a step definition to be executed, it must match the given component in a feature

**Provide an example of a feature file using the Cucumber framework.**

Feature: Home Page

In order to test Home Page of application

As a Registered user

I want to specify the features of home page

Scenario: Home Page Default content

Given user is on Github home page

Then user gets a GitHub bootcamp section

And username is also displayed on right corner

Scenario: GitHub Bootcamp Section

Given user is on GitHub home page

When user focuses on GitHub Bootcamp Section

Then user gets an option to setup git

And user gets an option to create repository

And user gets an option to Fork Repository

And user gets an option to work together

Scenario: Top Banner content

Given user is on GitHub home page

When user focuses on Top Banner

Then user gets an option of home page

And user gets an option to search

And user gets settings options

And user gets an option to logout

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

The primary **purpose** of **BDD methodology** is to improve communication amongst the stakeholders of the project so that each feature is correctly understood by all members of the team before **development** process starts. This helps to identify key scenarios for each story and also to eradicate ambiguities from requirements.

**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

**What is the use of Background keyword in Cucumber?**

****Background in Cucumber****is 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.

Feature: Test Background Feature

Description: The purpose of this feature is to test the Background keyword

Background: User is Logged In

Given I navigate to the login page

When I submit username and password

Then I should be logged in

Scenario: Search a product and add the first product to the User basket

Given User search for Lenovo Laptop

When Add the first laptop that appears in the search result to the basket

Then User basket should display with added item

Scenario: Navigate to a product and add the same to the User basket

Given User navigate for Lenovo Laptop

When Add the laptop to the basket

Then User basket should display with added item

**What symbol is used for parameterization in Cucumber?**

Pipe **symbol** (|) is **used** to specify one or more parameter values in a feature file

**What is the purpose of Examples keyword in Cucumber?**

It is with these **keywords** that **Cucumber** allows for easy Data Driven testing to be completed where no changes need to be made to the Java file. ... **Example keyword** can only be used with the Scenario Outline **Keyword**. Scenario Outline – This is used to run the same scenario for 2 or more different set of test data

**Provide an example of step definition file in Cucumber.**

****Feature:**** Visit ****career guide**** page in career.guru99.com

****Scenario:**** Visit career.guru99.com

****Given:**** I am on career.guru99.com

****When:**** I click on career guide menu

****Then:**** I should see career guide page

Step 1:

Given (^ I am on career.guru99.com$)

Public void I\_am\_on\_career.guru99.com(){

selenium code

}

Step 2:

When (^ click on career guide menu$)

selenium code

Step 3:

Then (^ I should see career guide page$)

selenium code

**What is the purpose of 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.

**How can Cucumber be integrated with Selenium WebDriver?**

**USe the below dependencie, create a .feature file,step definition file and TestRunner file**

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.11</version>

<scope>test</scope>

</dependency>

<dependency>

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

<artifactId>selenium-java</artifactId>

<version>3.7.0</version>

</dependency>

<dependency>

<groupId>info.cukes</groupId>

<artifactId>cucumber-java</artifactId>

<version>1.2.0</version>

</dependency>

<dependency>

<groupId>info.cukes</groupId>

<artifactId>cucumber-jvm-deps</artifactId>

<version>1.0.5</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>info.cukes</groupId>

<artifactId>cucumber-junit</artifactId>

<version>1.2.0</version>

<scope>test</scope>

</dependency>

</dependencies>

**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

Acceptance Criteria is written before implementation

**What is the use of Behavior Driven Development in Agile methodology?**

In Agile environments, BDD plays a vital role because it strongly encourages the use of Agile methodologies during the development and testing. BDD brings customers, end-users, BAs, QAs, and SEs of the software product into one table for effective sharing of knowledge on the system and its testing requirements.

BDD consists of cycles of a set of steps to follow.

Identify business feature.

Identify scenarios under the selected feature.

Define steps for each scenario.

Run feature and fail.

Write code to make steps pass.

Refactor code, Create reusable automation library.

Run feature and pass.

Generate test reports.

**Explain the purpose of keywords that are used for writing a scenario in Cucumber?**

The primary keywords are:

Feature

Rule (as of Gherkin 6)

Example (or Scenario)

Given, When, Then, And, But (steps)

Background

Scenario Outline (or Scenario Template)

Examples

There are a few secondary keywords as well:

""" (Doc Strings)

| (Data Tables)

@ (Tags)

# (Comments)

Feature

The purpose of the Feature keyword is to provide a high-level description of a software feature, and to group related scenarios.

Descriptions 🔗︎

Free-form descriptions (as described above for Feature) can also be placed underneath Example/Scenario, Background, Scenario Outline and Rule.

Rule 🔗︎

The (optional) Rule keyword has been added in Gherkin v6. (Note that Gherkin 6 has not yet been incorporated into all implementation of Cucumber!) The purpose of the Rule keyword is to represent one business rule that should be implemented. It provides additional information for a feature. A Rule is used to group together several scenarios that belong to this business rule. A Rule should contain one or more scenarios that illustrate the particular rule.

Example 🔗︎

This is a concrete example that illustrates a business rule. It consists of a list of steps.

The keyword Scenario is a synonym of the keyword Example.

You can have as many steps as you like, but we recommend you keep the number at 3-5 per example. If they become longer than that, they lose their expressive power as specification and documentation.

Steps 🔗︎

Each step starts with Given, When, Then, And, or But.

Given 🔗︎

Given steps are used to describe the initial context of the system - the scene of the scenario. It is typically something that happened in the past.

When 🔗︎

When steps are used to describe an event, or an action. This can be a person interacting with the system, or it can be an event triggered by another system.

Then 🔗︎

Then steps are used to describe an expected outcome, or result.

The step definition of a Then step should use an assertion to compare the actual outcome (what the system actually does) to the expected outcome (what the step says the system is supposed to do).

Background 🔗︎

Occasionally you’ll find yourself repeating the same Given steps in all of the scenarios in a feature.

A Background allows you to add some context to the scenarios in the feature. It can contain one or more Given steps.

A Background is run before each scenario, but after any Before hooks. In your feature file, put the Background before the first Scenario.

You can only have one set of Background steps per feature. If you need different Background steps for different scenarios, you’ll need to split them into different feature files.

Scenario Outline 🔗︎

The Scenario Outline keyword can be used to run the same Scenario multiple times, with different combinations of values.

The keyword Scenario Template is a synonym of the keyword Scenario Outline.

Copying and pasting scenarios to use different values quickly becomes tedious and repetitive:

Scenario: eat 5 out of 12

Given there are 12 cucumbers

When I eat 5 cucumbers

Then I should have 7 cucumbers

Scenario: eat 5 out of 20

Given there are 20 cucumbers

When I eat 5 cucumbers

Then I should have 15 cucumbers

We can collapse these two similar scenarios into a Scenario Outline.

Scenario outlines allow us to more concisely express these scenarios through the use of a template with < >-delimited parameters:

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 |

A Scenario Outline must contain an Examples (or Scenarios) section. Its steps are interpreted as a template which is never directly run. Instead, the Scenario Outline is run once for each row in the Examples section beneath it (not counting the first header row).

The steps can use <> delimited parameters that reference headers in the examples table. Cucumber will replace these parameters with values from the table before it tries to match the step against a step definition.

What is the meaning of TestRunner class in Cucumber?

@RunWith(Cucumber.class)

@CucumberOptions(features="Features",glue={"StepDefinition"})

public class Runner

{

}

@RunWith() annotation tells about the test runner class to start executing our tests.

@CucmberOptions() annotation is used to set some properties for our cucumber test like feature file, step definition, etc.

First import statement ‘org.junit.runner.RunWith‘ imports @RunWith annotation from the Junit class. @RunWith annotation tells JUnit that tests should run using Cucumber class present in ‘Cucumber.api.junit‘ package.

Second import statement ‘cucumber.api.CucumberOptions‘ imports the @CucumberOptions annotation. This annotation tells Cucumber a lot of things like where to look for feature files, what reporting system to use and some other things also. But as of now in the above test we have just told it for the Feature file folder.

What is the starting point of execution for feature files?

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

Should any code be written within the TestRunner class?

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

What is the use of features ,glue property under the Cucumber Options tag?

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

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

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

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

Ans: The maximum number of steps to be written in a scenario is 3-4 steps.

Can we use same step definition in different scenarios?

Yes

What is the execution priority in cucumber?

Cucumber features/scenarios are run in Alphabetical order by feature file name.

However, if you specifically specify features, they should be run in the order as declared. For example:

@Cucumber.Options(features={"automatedTestingServices.feature", "smoketest.feature"})

What is Test Runner in cucumber?

Cucumber test runner class is one of the many mechanisms using which you can run Cucumber feature file. The test runner class that will use in this article is a JUnit runner class. Below are some of its salient features:

In addition to running a cucumber feature file, the test runner class also acts as an interlink between feature files and step definition classes. It is in test runner class, that you provide the path for both feature file and step defs class

There are multiple types of test runners such as JUnit runner, CLI runner, Android runner etc, that you can use to run Cucumber feature file. In this article, we will be using the JUnit runner

With a test runner class, you have the option to run either a single feature file, or multiple feature files as well. For now, we will focus on running a single feature file

How will you execute specific scenario using line number?

If you want to run a specific scenario using cucumber you need to provide the line number the scenario starts on like:

cucumber features/test.feature:7

How does the Cucumber execution start?

The TestRunner.java file is executed as a jUnit Test.

It recognizes the feature files present by the extension as login.feature

It executes the step\_definition files, by reading the Given, When & Then statements, by matching using RegEx from the Feature files

The tests are executed using a headless browser - PhantomJS.

As most of these test cases are planned to be executed on a scheduled time, the need for a browser to be launched and viewing of execution is not required

However, if the need arises of viewing the execution, even that can be changed by just replacing the driver of the browser to the best suited one

The report of the test gets created in the target/site file and gets saved in an index.html file.

Can we use TestNG with cucumber?

Yes

What are Data Tables in Cucumber?

Difference between Scenario Outline & Data Table

Scenario Outline:

This uses Example keyword to define the test data for the Scenario

This works for the whole test

Cucumber automatically run the complete test the number of times equal to the number of data in the Test Set

Data Table:

No keyword is used to define the test data

This works only for the single step, below which it is defined

A separate code needs to understand the test data and then it can be run single or multiple times but again just for the single step, not for the complete test

## **Data Tables in Cucumber**

Scenario: Successful Login with Valid Credentials

Given User **is** on Home Page

When User Navigate **to** LogIn Page

**And** User enters Credentials **to** LogIn

    | testuser\_1 | Test@153 |

**Then** Message displayed Login Successfully

The implementation of the above step will belike this:

@When("^User enters Credentials to LogIn$")

public void user\_enters\_testuser\_\_and\_Test(DataTable usercredentials) throws Throwable {

//Write the code to handle Data Table

List<List<String>> data = usercredentials.raw();

//This is to get the first data of the set (First Row + First Column)

driver.findElement(By.id("log")).sendKeys(data.get(0).get(0));

//This is to get the first data of the set (First Row + Second Column)

driver.findElement(By.id("pwd")).sendKeys(data.get(0).get(1));

driver.findElement(By.id("login")).click();

}

What is Tag Inheritance in cucumber?

Tags are inherited by child elements. Tags that are placed above a Feature will be inherited by Scenario , Scenario Outline , or Examples . Tags that are placed above a Scenario Outline will be inherited by Examples .

On what places you can write tags in feature file?

It is not possible to place **tags** above Background or **steps** ( **Given** , When , Then , And and But ).

Tags are defined in the feature files

Tags are defined in the runner class as below :

1. tags = {“@tag”, “@tag1”} : means AND condition. –It says that scenarios matching both these tag needs to be executed.

2. tags = {“@tag1, @tag2”} : means OR condition. — It says that scenarios matching any of this tag needs to be executed.