**Manual Testing**

What is a test plan?

Is a document describing the scope, approach, resources and schedule of intended test activities. It identifies amongst others test items, the features to be tested, the testing tasks, who will do each task, degree of tester independence, the test environment, the test design techniques and entry and exit criteria to be used, and the rationale for their choice, and any risks requiring contingency planning. It is a record of the test planning process.

This document is derived from the SRS developed by the Team Lead/Project Manager.

**master test plan**: A test plan that typically addresses multiple test levels.

**phase test plan**: A test plan that typically addresses one test phase.

What are test cases?

A **TEST CASE** is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly.

The process of developing test cases can also help find problems in the requirements or design of an application.

3. Can you give sample test cases for ATM

We can write many test cases considering the different scenarios as a end user accessing ATM and Tell some features, we can use excel sheet to write manual test case by following step by step process using Gherkins language will can we can write the keywords Scenario,Given,When,Then Scenario Outline. In feature files.

4. What is importance of test data and how do you prepare it?

Test data is the input given to a software program. It represents data that affects or is affected by the execution of the specific module. Some data may be used for positive testing, typically to verify that a given set of input to a given function produces an expected result. Other data may be used for negative testing to test the ability of the program to handle unusual, extreme, exceptional, or unexpected input. Poorly designed testing data may not test all possible test scenarios which will hamper the quality of the software.

5. What will you do if you find defect?

Once the bug/defect is found, I will report and log it as new bug to the development team, (occasionally we report it our Team lead and he will then report it to development team and it will be assigned as new bug). We will provide as much as details in a bug report, unique ID, Tile, Description, Expected Result, Actual Result, details of the project, Environment details, severity and priority of the bug, steps to reproduce, attachments and screenshot, Test data, Developer assigned.

6. What are different tests you will perform on a UI rich webpage like BBC

Usability test, performance test, functional testing, regression test.

7. What is Smoke and Sanity tests ?

We do Smoke test to confirm the basic functionality of a build, whether the build is stable or unstable. it also confirms to test further tests or not. The Smoke test is carried out in the initial development phase.

Sanity test is similar to Smoke Test but it is done on the production environment before the end user start using the product.

8. What will happened if you don’t have time for testing

We do Ad hoc Testing – testing the application’s basic and main functions randomly.

9. What will happened if an issue is found in Production.

If bug/issue found at the production, It depends upon the impact of the bug/defect on the whole application which is developed. if the impact is critical and effects the clients business. Then BA and whole team will decide whether to go release or not. If the defect can easily fix and tested again before we release it will then it will be fixed promptly. If not, BA will mitigate with the client or stakeholders and decision will be made.

10. What will happened if your developer don’t accept your bug?

Before approaching the Developer, I will double check the SRS whether there was misunderstanding of the requirement/user story. if everything is fine then I will approach the developer with all the facts and evidence. If still the Developer don not accept then I will involve BA for clarification and his conclusion.

11. Tell me the best bug you found

Tell about the crucial or important defect you have found.

12. Tell me how do you decide priority and Severity

If the defect has impact on the system or application crashing or unable to test further, I will report it as critical and priority will be high.

The major difference between Severity and priority is, Severity defines the impact on the software application and priority defines the order in which the bug has to be fixed.

There are five categories to assess the severity 1) Critical 2) Major 3) Moderate 4) Minor 5) Cosmetic whereas priority of three types 1) High 2) Medium 3)Low

13. Tell me what it test strategy?

Test Strategy is a high-level document which defines the approach for software testing. It is basically derived from the BRS. It is developed by the BA. It is a static document which sets the the standards for testing, so not often updated.

14. How do to plan a testing if your company don’t have any testing process.

Fristly, I will check the software application being developed and understand the requirements. Given the chance to explore, I will discuss with the team leader and research the open sources tools which are freely available in the market/internet. I will design the framework/prototype which is best suitable for testing the application. I will demonstrate to the whole team about the framework and benefits and efficiency of using the framework.

15. If you are only tester in your team what would you do to ensure quality

The answer can be same as above with some enhancement.

**Java**

What are different data types and differences between primitive and nonprimitive

There are two types Data Types in Java

**i) Primitive data types**

**ii) Non-Primitive data types**

**i) Primitive data types**

byte, short, int, long data types are used for storing whole numbers

float and double data types are used for storing fractional number

char data type is used for storing characters(letters)

Boolean data type is used for variables that holds either true or false.

**ii) Non-Primitive data types**

Arrays, String, etc.

2. Is String object or Datatypes

String is an immutable object which means it is constant and cannot be changed once it is created.

There are two ways to create a String in Java

a) String literal - String str = ”hello”; // here complier creates the object of String

b) Using new Keyword - String str = new String(“hello”);

3. What are major String Functions and its examples

There are more than 40 methods available to Java,

the String major functions are .length(), .equals() , .equalsIgnoreCase(), .compateTo(str), . stratsWith().isEmpty,. charAt(), .toUperCase, toLowerCase, .trim(),replace(), . replaceAll(), .contains();

4. What is object and class and differences

Class is a blue print of an object using which we create as many as objects.

Object is an instance of a class, object is an entity which has two characteristics **State (Variables)** and **Behaviour (Methods)**

5. How to create an object in different ways

By using new keyword – we create an object instantly.

Eg: ClassName objectName = new ClassName();

And by declaring and assigning

Eg: ClassName objectName;

objectName= new ClassName();

6. What are different type of Functions and differences (Constructor,Method,Function,Static)

**Constructor** is a block of code that initialize the newly created object. A constructor resembles an instance of method in Java but it is not a method as it doesn’t have return type.

Constructor has same name as the Class.

Constructor overloading is a concept of having more that one constructor with different parameters.

Constructor cannot be inherited.

**Method**

A method is a set of code which is referred to by name and can be called (invoked) at any point in a program simply by utilizing the method's name.

7. What is return variable, arguments/parameters,

Question is not clear will refer to later

8. What is main method and its importance

public static void main(String args[])

{ }

**public** : public method that means that we can call the method from outside the class.

**Static:** We do not need to create object for static methods to run. They can run itself.

**void :** It does not return anything.

**main:** It is the method name. This is the entry point method from which the JVM can run your program.

**(String[] args):** Used for command line arguments that are passed as strings.

9. How to send data from command line

By using System.in to get the data from console or Scanner class not sure, let you once I go through them

10. What are different loops and its differences

While loop

do-while loop

for- loop

enhanced for-each loop

11. what is switch case statement

When we have number of options and may need to perform a different task for each choice.

12. What are oops and its importance

**OOPS** is a programming paradigm based on the concept of **“objects”** that contain data and methods. The primary purpose of the oops is to increase the flexibility and maintainability of programs

**Oops features:**

1. **Abstraction**
2. **Encapsulation**
3. **Inheritance**
4. **Polymorphism**

**Abstraction:** The process where we show only “relevant” data and “hide” unnecessary details of an object to the user.

**Encapsulation:** binding object state(variables) and behaviour(methods) together. For example if we are creating a class we are doing encapsulation.

**Inheritance:** The process by which one class can acquire the properties of and functions of another class. Inheritance provides the idea of reusability of code and each sub class defines only features that are unique to it, rest of the features can be inherited from the parent class.

**Polymorphism:** Polymorphism is a object oriented programming feature that allows us to perform a single action in different ways.

13. Examples of How oops can help reduce duplication of code

Answer: explain about Inheritance

14. What are different access specifiers(public/Protected/default/Private)

**Public:** the members ,methods and classes that are declared public can be accessed from anywhere. The modifier do not put any restriction on an the access

**Private:** private member and methods can be accessed within the class.

**Default:** when we do not mention any access modifier, it is called default access modifier – it is limited to package.

**Protected:** can be accessed by the classes of the same package.

15. Method Over loading and Riding

**Method Overloading:** Method overloading is a feature that allows a class to have more than one method having same name, if their arguments lists are different.

**Method overriding:** Declaring a method in sub class which is already present in parent class is known as method overriding. Overring is done so that a child class can give its own implementation to a method which is already provided by the parent class.

The advantage of method overriding is that the class can give its own specific implementation to a inherited method even modifying the parent class code.

16. What is abstract, Interface and its real time uses with examples

**Abstract Class:** An abstract class outlines the methods but not necessarily implements all the methods. Cannot be instantiated.

**Abstract method:** A method that is declared but not defined. Only method signature no body

**Interface :** An Interface is a blue print of a class, which can be declared by using interface keyword. Interface can contain only constants and abstract methods (methods with only signature no body),

They can only be implemented by classes or extended by other interface.

Interface is a common way to achieve full abstraction in Java.

17. What are different type of exceptions checked and unchecked and its differences (which is used where)

An Exception is an unwanted event that interrupts the normal flow of the program and program execution get terminated. Exceptions occurs in the code.

Exception handling is one of the most important feature of java programming language that allow us to handle the runtime errors caused by exceptions.

**types of exceptions**

Checked Exception

Unchecked Exception

1.Checked Exceptions: All exceptions other than Runtime Exceptions are known as Checked Exceptions as the compiler checks them during compilations to see whether the programmer has handled them or not.

2.Unchecked Exceptions: The exceptions are not checked at compile time so complier does not check whether the programmer has handled them or not.

18. What to do if you found and exception

In Java we handle exceptions in two ways try-catch block or throws keyword on method level.

19. How to debug a java program

**jdb** helps you find and fix bugs in Java language programs.

The Java Debugger, **jdb**, is a simple command-line debugger for Java classes. It is a demonstration of the [Java Platform Debugger Architecture](https://docs.oracle.com/javase/7/docs/technotes/guides/jpda/index.html) that provides inspection and debugging of a local or remote Java Virtual Machine.

**jdb** [ options ] [ class ] [ arguments ]

[**options**](https://docs.oracle.com/javase/7/docs/technotes/tools/windows/jdb.html#options)

Command-line options, as specified below.

**class**

Name of the class to begin debugging.

**arguments**

Arguments passed to the main() method of class.

20. What are scopes in a class of variable and method

Scope is often referred to as Visibility. This is important because if you understand the concept of Visibility in OO programming,

For example: If you specify a variable at the top of a program, meaning not in a method, then its scope is universal within that program. You can use it, and change it, anywhere.

if you create a method and declare a variable within that method, then we only specify and use it within that method. This means the variables have "local" scope, not "global".

21. What is a class and its real time uses

A *class* is the blueprint from which individual objects are created. Then explain about variable and methods in the class and its various uses by creating objects in various ways.

22. How to use 3rd party library like Junit

By importing the jar file or packages in the program.

**Selenium IDE**

1. What are locators

Selenium Webdriver have different types of locators to identify a web elements uniquely within the webpage. Locator is termed as an address that identifies a web element.

8 Types of locators:

**ID:** Unique for every web element

**Name:** Same as ID although it is not unique

**CSS Selector:** Works on element tags and attributes

**XPath:** Searches elements in the DOM, Reliable but slow

**Class name:** Uses the class name attribute

**TagName:** Uses HTML tags to locate web elements

**LinkText:** Uses anchor text to locate web elements

**Partial Link Text:** Uses partial link text to find web elements

2. What is xpath and its uses

XPath is one of the locator strategies Selenium uses to find the web elements.

It works by navigating through the DOM elements and attributes to locate the target object. For

example – a text box or a button or checkboxes.

Although, it guarantees to give you the element you are looking after. But it is slower than as compared to other locators like ID, name or CSS selectors.

XPath is the most powerful way of identifying the locations of the web elements on web page. It’s a best locator strategy when we have to find objects which are highly dynamic and fragile in nature.

XPath is a perfect technique for walking through the DOM structure of the web page. We can classify XPath in two groups, Absolute XPath, Relative XPath.

**Absolute XPath.**

It starts from the root element within the web page or part of the page and goes to identify the target element.

Absolute XPath Example:

HTML/head/body/table/tr/td

To use locators like the XPath is easy as you give the direct element path. But the XPath would break when the element structure changes.

**Relative XPath.**

The relative XPath are easy to manage as they are short and concise. It is also better than the previous XPath style as it may survive the changes in the Page HTML to a certain degree. Though, building a relative XPath is time-consuming and quite difficult as you need to check all the nodes to form the path.

Relative XPath Example:

//table/tr/td

**Single (forward) Slash “/”:** It represents the absolute path. In this case, the XPath engine navigates the DOM right from the first node.

**Double (forward) Slash “//”:** It represents the relative path. In this case, the XPath engine searches for the matching element anywhere in the DOM.

An absolute XPath will always search from the root node until it reaches the target. Such an XPath expression includes the single forward slash (/) as the prefix.

/html/body/div[1]/div[5]/form/table/tbody/tr[3]/td/input

A relative XPath doesn’t have a specific point to start. It can begin navigation from any node inside the DOM and continues. Such an XPath expression includes the double forward slash (//) as given below.

//input[@id='username']

3. What if id is changing dynamically

We choose Relative XPath to find web elements if the web element id is changing dynamically.

4. What are common interaction commands in Selenium

**1) get()**

**2) getCurrentUrl()**

**3) findElement(By, by).click()**

**4) isEnabled()**

**5) findElement(By, by).sendKeys()**

**6) findElement(By, by).getText()**

**7) Submit()**

**8) findElements(By, by)**

**9) findElements(By, by).size()**

**10) driver.manager().pageLoadTimeout(time,unit**

**11) driver.manager().timouts().implicitlyWait()**

**12) getTitle()**

**13) driver.navigate().to ("url") and driver.navigate().back() and driver.naviagte.forward()**

**14) getScreenshotAs()**

**15) moveToElement() - of Actions class**

**16) dragAndDrop() - of Action class**

**17) 20) switchTo() and accept(), dismiss() and sendKeys() - of Alert class**

**18) getWindowHandle() and getWindowHandles() - to handle multiple windows**

**19) assertEquals(),assertNotEquals(), assertTrue() and assertFalse()**

**20) close() and quit() - quit is used to quit driver instance**.

5. What is wait for page to load and wait for element present

Implicit Wait: It is a wait timeout which applies to a Webdriver instance. It implies that all actions of this instance will timeout only after waiting for a duration specified by the implicit wait.

WebDriver driver = new ChromeDriver();

driver.manage().timeouts().implicitlyWait(15, TimeUnit.SECONDS);

Explicit Wait: It is an exclusive timeout method that works by adding code to delay the execution until a specific condition arises. It is more customizable in terms that we can set it up to wait for any suitable situation. Usually, we use a few of the pre-built Expected Conditions to wait for elements to become clickable, visible, invisible, etc.

WebDriver driver = new ChromeDriver();

driver.get("http://target\_page\_url");

WebElement dynamicElement = (new WebDriverWait(driver, 15))

.until(ExpectedConditions.presenceOfElementLocated(By.id("dynamicElement")));

6. Why waits are used

We use waits to handle synchronization.

Waits help the user to troubleshoot issues observed while re-directing to different web pages and during loading of new web elements. After refreshing the page, a time lag appears during reloading of the web pages and the web elements.

Webdriver Wait commands help here to overcome the issues that occur due to the variation in time lag. Webdriver does this by checking if the element is present or visible or enabled or clickable etc. WebDriver provides two types of waits for handling the recurring page loads, invocation of pop-ups, alert messages, and the reflection of HTML objects on the web page.

**1-**[**Implicit Wait**](https://www.techbeamers.com/webdriver-wait-commands-tutorial-examples/#webdriver-implicit-wait)  
**2-**[**Explicit Wait**](https://www.techbeamers.com/webdriver-wait-commands-tutorial-examples/#webdriver-explicit-wait)

**3- Fluent Wait**

Implicit WebDriver Wait Commands.

Implicit waits are used to provide a default waiting time between each consecutive test step/command across the entire test script. Once you’d defined the implicit wait for X seconds, then the next step would only run after waiting for the X seconds.

The drawback with implicit wait is that it increases the test script execution time. It makes each command wait for a stipulated amount of time before resuming the execution.

Thus, implicit wait may slow down execution of your test scripts if your application responds normally. driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

**Explicit WebDriver Wait Commands.**

As the name signifies, by using an explicit wait, we explicitly instruct the WebDriver to wait. By using custom coding, we can tell Selenium WebDriver to pause till the expected condition occurs.

**What is the need of using the explicit wait when an implicit wait is in place and doing well already?** I there may be instances when some elements take more time to load. Setting implicit wait in such cases may not be wise because the browser will needlessly wait for the same amount of time for every element. All of this would delay the test execution. Hence, the WebDriver brings the concept of Explicit waits to handle such elements by passing the implicit wait. This wait command gives us the ability to apply wait where required. And avoids the force waiting while executing each of the test steps.

WebDriverWait wait = new WebDriverWait(driver,30);

7. what is verify and assert and its difference

**Assert:** It allows us to verify the result of an expression or an operation. If the “assert” fails, then it will abort the test execution and continues with the next case.

**Verify:** It also operates the same as the assert does. However, if the “verify” fails, then it won’t abort the test instead continues with the next step.

8. How to handle popups in selenium IDE

When we want to switch the control to the pop up then press the ok or cancel button. After that, turn back to the source page screen. By using Alert

Code Example:

String srcPage = driver.getWindowHandle();

Alert pop = driver.switchTo().alert(); // shift control to the alert popup.

pop.accept(); // click on ok button.

pop.dismiss(); // click on cancel button.

// Move the control back to source page.

driver.switchTo().window(srcPage); // move back to the source page.

Usually, we come across following two type of alerts while using the web.

Windows-based alerts

Web-based alerts

**Web-Based Alert Pop-Ups**

WebDriver exposes the following APIs to handle such popups.

**Dismiss():** It handles the alert by simulating the Cancel button.

**Accept():** It handles the alert window by simulating the Okay button.

**GetText():** You may call it to find out the text shown by the alert.

**SendKeys():** This method simulates keystrokes in the alert window.

**Windows-Based Alert Pop-Ups**

Handling a window based pop-up is not straight-forward. Selenium only supports web applications and doesn’t provide a way to automate Windows-based applications. However, the following approaches can help.

Use Robot class (Java-based) utility to simulate the keyboard and mouse actions. That’s how you can handle the window based pop.

The KeyPress and KeyRelease methods simulate the user pressing and releasing a specific key on the keyboard.

9. How to debug in Selenium IDE

In Action Menu we have Step and Execute this command options

**Step:** This option allows to step through the playing-back of the test case. This is useful for debugging purpose and works with breakpoints.

**Execute this command:** This allows user to execute a single test step within the entire test script without executing the entire test script. This can be used at times when we want to debug/see the behavior of a particular test step.

10. when is Xpath used in real time and its examples

Check answer for locators

11. What are different type of web elements in a normal webpage like demo.nopecommerce.com

Check the web page

12. How to read and check the date form the UI

13. What are different assertions

Assertions in selenium can be used in 3 modes which are explained below:

**1. assert:** If you use assert in your tests then the test will be aborted if the assert fails. Next test case will start executing in case you are running a test suite.

**2. verify:** If verify is used then the test case will not abort even if the verify fails. It will continue executing and log the failure for the failed conditions.

**3. waitFor:** waitFor command waits for the condition to become true. If the condition is true already the test case continues else it waits for the conditions to become true. If the condition doesn’t becomes true within specified time-out period test will fail and halt.

14. Why is selenium used at first place

Selenium is a free (open source) automated testing suite for web applications across different browsers and platforms

Selenium WebDriver is a collection of open source APIs which are used to automate the testing of a web application.

Selenium WebDriver tool is used to automate web application testing to verify that it works as expected. It supports many browsers such as Firefox, Chrome, IE, and Safari.

WebDriver is a public interface, we just define a reference variable(driver) whose type is interface.

Now any object we assign to it must be a instance of a class (fireFoxDriver)that implement the interface

15. What are the other important things you can do in Selenium IDE apart from record and play back.

Apart from record and play, the test cases written in IDE can be exported in many programming languages like Ruby, Java, C-sharp,Perl,Php

Edit and debug options are also available.

**Webdriver**

1. What is difference between web driver and selenium ide

Selenium IDE is a very popular Firefox browser extension (add-on to the Browser. It presents a graphical interface to record, play and save user actions.

Selenium WebDriver is a collection of open source APIs which are used to automate the testing of a web application.

Selenium WebDriver tool is used to automate web application testing to verify that it works as expected. It supports many browsers such as Firefox, Chrome, IE, and Safari.

WebDriver is a public interface, we just define a reference variable(driver) whose type is interface. Now any object we assign to it must be a instance of a class (fireFoxDriver)that implement the interface

Test automation beginners can find it convenient for web automation testing. After recording a workflow, we can also export the steps in various formats.

2. What is selenium as a whole

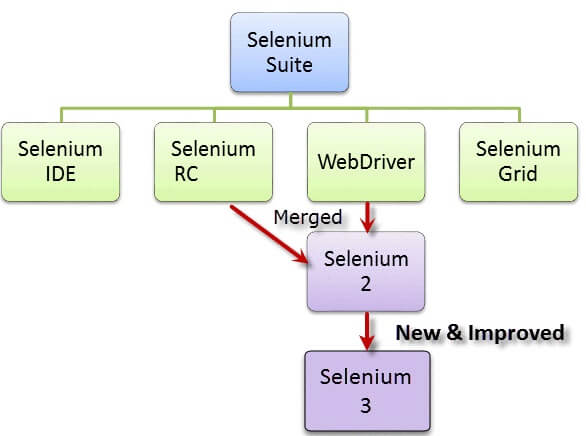
Selenium is not just a single tool but a **suite of software's,** each catering to different testing needs of an organization. **It has four components**.

1. Selenium Integrated Development Environment (IDE)

2. Selenium Remote Control (RC)

3. WebDriver

4. Selenium Grid



At the moment, Selenium RC and WebDriver are merged into a single framework to form Selenium 2. Selenium 1, by the way, refers to Selenium RC.

1. **Selenium Integrated Development Environment (IDE)** is the simplest framework in the Selenium suite and is the easiest one to learn. It is a Firefox plugin that you can install as easily as you can with other plugins. However, because of its simplicity, Selenium IDE should only be used as a prototyping tool. If you want to create more advanced test cases, you will need to use either Selenium RC or WebDriver.

2. **Selenium RC** was the flagship testing framework of the whole Selenium project for a long time. This is the first automated web testing tool that allowed users to use a programming language they prefer. As of version 2.25.0, RC can support the following programming languages: Java , C# ,PHP, Python,Perl,Ruby.

3. **WebDriver** proves itself to be better than both Selenium IDE and Selenium RC in many aspects. It implements a more modern and stable approach in automating the browser's actions. WebDriver, unlike Selenium RC, does not rely on JavaScript for Automation. It controls the browser by directly communicating with it. The supported languages are the same as those in Selenium RC. ,Java,C#,PHP,Python,Perl,Ruby.

4. **Selenium Grid** is a part of the Selenium Suite that specializes in running multiple tests across different browsers, operating systems, and machines in parallel.

(Or )

Selenium is an automation development kit which comprises the following components.

**Selenium IDE:**

A Firefox extension to record and play the user actions performed on a web page.

**Selenium RC:**

A Selenium server which exposes APIs for scripting tests in different languages and also runs them in browsers.

**Selenium Webdriver:**

These are native APIs that directly interact with the browser. They give more control and faster than the RC APIs.

**Selenium Grid:**

It provides concurrency. With its help, we can split testing and run a set of cases on one machine and some on another.

3. What are plus points for selenium

Selenium is a time-proven test framework which can dramatically reduce the testing efforts. It equips with many useful features to make a tester’s life easier. Moreover, it can align with other tools to bring in more power.

There are many benefits of using Selenium for automated testing.

**Open source:** Since it is an OSS, so we don’t have to bear any licensing cost for using it.

**Cross-browser:** It works on all standard browsers such as Chrome, FF, IE, and Safari. We can run same the test script in all browsers.

**Multi-language:** We can choose a familiar programming language from Java, Python, C#, Ruby to use with Selenium.

**Cross-platform:** It provides test compatibility across Window, Linux, and Mac OSX. We can run same the test script on all platforms.

**Concurrency:** With Selenium Grid, we can execute thousands of test in parallel.

**CLI support:** We can create a test suite with hundreds of tests and launch it with a single command.

**CI support:** Jenkins is the best CI tool. It provides a Selenium plugin to configure tests for nightly execution.

**Free help:** We can get quick technical help from its large community.

**Tester friendly:** A non-programmer can also automate using Selenium. It is not too complicated to be used only by a programmer.

**Active project:** Active development and bug fixes on the latest project.

4. What are negative points for selenium

Selenium is a perfect clinical tool to impersonate user actions in a browser. However, it also has a few limitations given below.

Doesn’t support automation of Windows applications

Can’t perform mobile automation on its own

Lacks a good built-in reporting

Not 100% perfect for handling dynamic web elements

Poses challenges while processing popups or frames

Not that efficient in coping with the page load

Can’t automate a captcha

5. What's is given when then

Gherkin is a structured natural language that is used by business analysts to specify how they want the system to behave for given scenarios. The Gherkin language is simple. It uses about 10 keywords (Given, When, Then, And, But, Scenario, Feature, Background, Scenario Outline, Examples) which allow the language to be read and parsed by an automation tool called Cucumber.

6. What are challenges of using selenium webdriver

Other Interview Questions

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

01)What are the challenges in Agile Testing?

02)What is risk base testing?

Practically, due to time and budget considerations, it is not possible to perform exhausting testing for each set of test data, especially when there is a large pool of input combinations.

We need an easy way or special techniques that can select test cases intelligently from the pool of test-case, such that all test scenarios are covered.

We use testing techniques - Equivalence Partitioning, Boundary Value Analysis testing, Error guessing techniques to achieve this

03)How automation in Agile Testing?

04)What is QA role in Retrospective meeting, Backlog Grooming,Sprint Planning?

Sprint planning: The whole team (Scrum master, BA, Product Owner, Agile Devs, Agile Testers) will gather to estimate the development effect an testing effect of each and every User Story (). (Sprint is the duration of the iteration in Agile usually 2 weeks 3 weeks depends upon the complexity of the requirements and velocity of resources)

Backlog grooming: is a one activity during the sprint, to discuss not yet started user stories, upcoming user stories and not completed (product backlog or sprint backlog).

Retrospective meeting: is all about what went right and what went wrong and that should not follow in the coming sprints

05)What if there is no time to test at the end of the sprint?

It will be discussed in Backlog grooming and it will assigned as product backlog and will be done in the upcoming string

06)What is different types of testing for every ticket?

07)How do you make sure that the test coverage is 100%?

08)Explain what concepts of java is being used in your framework?

Oops

09)Tell me how do you handle exception in your framework?

10)Tell me the use of selenium IDE in real world?

The most obvious way to use Selenium IDE is through its record and play feature. But there are certain actions, accessors, and assertions which you can use to edit the recorded test cases. So that you can speed up automation and increase test coverage.

**1- Selenium IDE Action Commands.**

You can perform many browser actions using the Selenese commands. Examples are clicking a hyperlink or filling out a form. Can be done by running the IDE in “record mode”.

**2- Selenium IDE Accessors Commands.**

You can use accessors commands to fetch and save the state of web elements.

**3- Selenium IDE Assertions Commands.**

like Webdriver, you can place assert commands in Selenium IDE test cases to check the state of web elements.

**4- Selenium IDE Test Case/Test Suite.**

In Selenium IDE, it’s easy to create test cases and convert then into test suites. Add them to version control system for use in nightly automation.

**4.1-** Start writing a test case.  
**4.2-** Save the test case.  
**4.3-** Prepare to save it as a test suite.  
**4.4-** Add it to version control.

Above sequence of operations works best for Selenium IDE

11)Tell me different kind of locators and which one is better than other?

Selenium names eight types of locators to find the elements on a web page. Check out the below list of locators approved by Selenium. We are explaining each of them one by one and with examples.

1- Id To Select The Element With A Specified @Id Attribute.

It is a unique reference for a web object that the developer sets while writing the code. Ideally, the ID should not repeat on a page, but the browsers do allow exceptions to this rule. The ID is no doubt the best locator to use in Selenium. Still, if it belongs to an HTML table, then it’s possible that it would change or disappear from the list. Hence, you need to put in a more advanced locator technique.

<input id="user" class="required" type="text">

WebElement item = driver.findElement(By.id("user"));

2- Name To Select The First Element With The Specified @Name Attribute.

Every form has input fields with unique names. Names are unique most of the times, but it’s not a restriction. However, a field name locator is the best choice for testing a login form. But when you have multiple login types on the same page then you should use locators with a different scheme. Let’s see the example where you can use either the id or the field name.

<input id="user" name="admin" class="required" type="text">

WebElement locator = driver.findElement(By.name("admin"));

3- Link Text To Select The Link Element Which Contains The Matching Text.

It is a perfect way to find the links on a page.

<a href="http://www.google.com">How to use locators?</a>

WebElement item = driver.findElement(By.linkText("How to use locators?"));

4- Partial Link Text To Select Link (Anchor Tag) Element Which Contains Text Matching The Specified Partial Link Text.

It is almost similar to the previous locator. It differs in the way you use it to find the element.

<a href="http://www.google.com">How to use locators?</a>

WebElement item = driver.findElement(By.PartialLinkText("How to use locators?"));

5- Tag Name To Find The Element Using Its HTML Tag.

You can better understand to use this locator from the below example.

List<WebElement> linkElements = driver.findElements(By.tagName("results"));

String[] linkTexts = new String[linkElements.size()];

6- CSS Class Name To Access The Elements.

The CSS class locator uses a specific class attribute to get to the first element on a web page. It is useful for items that own a unique style.

CSS class locator example:

WebElement element =driver.findElement(By.className(“sample”));

7- CSS Selector To Access The Elements.

CSS Selectors are no different than the XPaths. But they are resilient and powerful. Unlike the XPath, they aren’t dependent on the DOM structure. They can help you perform actions which are difficult to do with XPath.

CSS Selector example:

WebElement CheckElements = driver.findElements(By.cssSelector("input[id=email']"));

8- XPath To Track An Element Using The XPath Expression.

XPath is a perfect technique for walking through the DOM structure of the web page. XPath locators are robust and reliable. It is one method which guarantees to locate any element on the page using the XPath expression.

We can classify XPaths in the following two groups.

**I- Absolute XPath.**

It starts from the root element within the web page or part of the page and goes to identify the target element.

Absolute XPath Example:

HTML/head/body/table/tr/td

To use locators like the XPath is easy as you give the direct element path. But the XPath would break when the element structure changes.

**II- Relative XPath.**

The relative XPath are easy to manage as they are short and concise. It is also better than the previous XPath style as it may survive the changes in the Page HTML to a certain degree. Though, building a relative XPath is time-consuming and quite difficult as you need to check all the nodes to form the path.

Relative XPath Example:

//table/tr/td

Guarantees to find accurate locators.

12)Tell me what are Xpath functions and how do you use in real life?

13)What is the different between Junit and TestNG?

14)Tell me the usage of maven in your project including continues Integration?

Apache Maven is a very powerful and widely used Java project management and build management tool. Here is a point-wise summary of its features.

**1-** It provides support for managing the entire lifecycle of a Java project.

**2-** Defining the project structure, dependencies, build, and test management are some of its magical traits.

**3-** Maven as a build tool allow setting up the execution environment for the project code to run independently.

**4-** It enables a unified platform where you can check out the source code from GIT/SVN, compile and package it into a JAR/WAR file.

**5-** You want a project management tool, Maven will do it for you. It has the project object model(POM) file to manage project’s build, dependency, and documentation.

**6-** Manage all project related dependencies using the POM.xml. It helps setting up all the configurations.

**7-** Last but not the least is its ability to download the project dependency jars automatically from the central repository.

**Why To Use Maven In Selenium Project?**

**1-** Since Maven is a build automation tool,  so it can manage the Selenium Webdriver test project’s build compilation and documentation. It eases up the task of creating right project structure, adding and managing jar files in the projects build path.

**2-** Most important feature of Maven is managing the project dependencies using the POM.xml. Let’s take a simple example of upgrading a single jar file. Say, we were using Selenium version 2.53.1 which we later updated to a newer version. Such type of scenario is easy to handle using Maven as it requires updating the version in the POM file.

**3-** Now imagine the situation, when we have a large number of jar files with updates available. And if we have to update them manually, then it’ll turn out to be a cumbersome task. Also, there is a high probability of errors even if we’ve them updated manually.

**4-** Maven comes to the rescue in a situation like above. All we have to do is change the version of target jar files in the POM file. Maven will download the newer versions of all jar files automatically and store them in a local repository.

15)What are the important classes in selenium webdriver and their usages?

16)How to take screen shots in webdriver and attached to the reports?

Sometimes, an image than a trace log can help us identify the right reason for an error. The code in the below example will capture the image and store in a file.

Import org.apache.commons.io.FileUtils;

WebDriver ins = new ChromeDriver();

ins.get("http://www.google.com/");

File screen = ((TakesScreenshot)ins).getScreenshotAs(OutputType.FILE);

// Now you can do whatever you need to do with it, for example copy somewhere

FileUtils.copyFile(screen, new File("c:\\tmp\\myscreen.png"));

17)How do I get number of similar objects in webdriver and click on every objects

and prints its texts and comeback?

18)How to drag and drop in webdriver?

Yes, we can use the Advanced User Interactions API to perform drag and drop operations in a Selenium Webdriver project.

Code example:

Actions act = new Actions(driver);

act.dragAndDrop(source\_locator, target\_locator).build().perform();

//Or you can use the below code style.

(new Actions(driver)).dragAndDrop(source\_locator, target\_locator).perform();

19)How to handle frames,multiple browser,pop up boxes in webdriver?

getWindowHandle() and getWindowHandles() - to handle multiple windows

for pop up – by using Alert Class

20)Tell me the important points in Gherkin language and its significance?

Gherkin is a structured natural language that is used by business analysts to specify how they want the system to behave for given scenarios. The Gherkin language is simple. It uses about 10 keywords (Given, When, Then, And, But, Scenario, Feature, Background, Scenario Outline, Examples) which allow the language to be read and parsed by an automation tool called Cucumber.

21)How to send list of parameters into stepdefinations,data table?

22)What is the difference between page object and page factory?

**Page Object Model.**

It is a design pattern which refers to treat web pages as classes. Each member of the class points to an element that exists on the page. It offers an elegant way of creating test routines that have greater readability, less maintenance, as well as easy to extend.

**Page Factory.**

Page factory enables initialization of each element with a counterpart element on the target web page. It opens up the use of Annotations like @FindBy to define strategies for selecting elements.

23)How to use before class in cucumber and what are hooks in cucumber?

**1- Before:** executes before the feature file execution.

**2- After:** executes after the feature file execution.

**3- BeforeStep:** executes before the each step execution.

**4- AfterStep:** executes after the each step execution.

24)Tell me the difference between all automation frame works including pros & cons?

25)How to implement continues integration?

**Jenkins** is a continuous integration suite that checks your code out of a repository, builds and packages it, and dumps it out to a server so you can test it - all hands-off. It can use Maven or Ant as its build tool.

In summary, Jenkins can use Maven as its build tool for continuous integration. You can use Maven without Jenkins if you choose not to do CI.

26)Difference between grid and sauce lab and pros & cons?