

Module 1

Selenium General + Core Java Mix Interview Questions

In this Module, We will go over high level Automation(Selenium) questions including Core Java Most important questions

RahulShettyAcademy.com – QA Platform



1. What are the Challenges with Selenium Automation?

- No Support for Non-Web Automation
- Timeout or Sync Issues
- Test Execution Slowness in Internet Explorer
- Limited Reporting

2. What are new Selenium 4 features?

- WebDriver is developed completely by W3C Standardization [🔗](#)
- The Selenium IDE support for Chrome is available now
- Selenium Grid - The hubs and nodes are smooth to setup and handle now. Once a Selenium server is started, the grid will act both as a hub and node.
- Taking screenshots at the Element level, Section level and Page Level is possible now.
- Support of Relative Locators

3. What is difference between WebDriver.findElement vs WebElement.findElement ?

driver.findElement():

It is for finding the element from the entire page using the given selector.

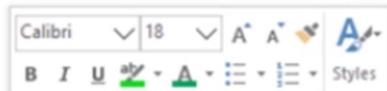
WebElement.findElement():

First, it generates the WebElement, and then the child elements of the given element are searched based on the given selector.

4. What is the difference between Page Object Model and Page Factory?

Page Object Model (POM):

POM is a Selenium design pattern; we can see it as a repository where we store all the WebElements. This has become very popular in industry these days, because it is very easy to manage, reusability of code and eliminates duplication of code.



The key benefit if UI changes in the future, then we can update WebElements to Page Classes in POM or Object Repository accordingly

```
public class Guru99Login { Guru99Login 1 Page class in object repository  
    WebDriver driver;  
    By user99GuruName = By.name("uid");  
    By password99Guru = By.name("password");  
    By titleText = By.className("barone");  
    By login = By.name("btnLogin");  
  
    public Guru99Login(WebDriver driver){  
        this.driver = driver;  
    }  
    //Set user name in textbox  
    public void setUserName(String strUserName){  
        driver.findElement(user99GuruName).sendKeys(strUserName); 3  
    } 2  
}
```

524 x 236

Find Web Element

Performing operation on web element

Page Factory:

Page Factory in Selenium WebDriver is an integrated concept or API. Here we follow again the same principle of keeping repository objects or page classes separate from test classes.

Here we use @FindBy to find elements and to initialize WebElements using initElements process.

WebElements are identify by
@FindBy Annotation

static initElements method of
PageFactory class for
initializing WebElement

```
@FindBy(xpath="/table//tr[@class='heading3']")
WebElement homePageUserName;

public Guru99HomePage(WebDriver driver){
    this.driver = driver;
    //This initElements method will create all WebElements
    PageFactory.initElements(driver, this);
}
```

616 x 252

6. How To Overcome StaleElementReferenceException in Selenium

The reference to an element is now "stale"; the element will no longer appear on the page's DOM. In simple words, when you started interacting with it, the element you located using the findElement method disappeared.

I

Adding exception handling to your action and, if the exception is stale, try to locate the element after a simple wait for 500 milliseconds and repeat these actions until the action or max iterations have been successful.



```
public class Sample {  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
  
        System.setProperty("webdriver.chrome.driver", "C:\\\\chromedriver.exe");  
        WebDriver driver = new ChromeDriver();  
        driver.get("https://rahulshettyacademy.com/angularpractice/");  
        WebElement nameBox=driver.findElement(By.name("name"));  
  
        driver.findElement(By.linkText("Shop")).click();  
        driver.findElement(By.linkText("Home")).click();  
        WebElement nameBox1=driver.findElement(By.name("name"))  
        nameBox1.sendKeys("hello");
```

7. Different Between XPath and CSS Selector?

- Xpath is slower than CSS, whereas CSS Selector is faster than XPath.
- XPath supports text though CSS Selector does not allow Text.
- XPath can move in both forward and backward directions whereas CSS Selector can only move forward

8. How to access the CSS selector using the nth element?

8. How to access the CSS selector using the nth element?

Here is a syntax for using the CSS selector to access the nth attribute: <type>:nth-child(n)

Ex: tr:nth-child(2)

9. How to handle alerts in Selenium WebDriver?

WebDriver provides an API to handle alert dialogs. Alerts cannot able to inspect if there is no Alert in the screen, you will get ‘NoAlertPresentException’

The Alert interface contains a number of APIs to execute different actions.

The following list:

`Alert alert = driver.switchTo().alert();`

`alert.accept();` This is equivalent to the OK button action on the dialog

`alert.dismiss();` This is equivalent to clicking on the CANCEL action button

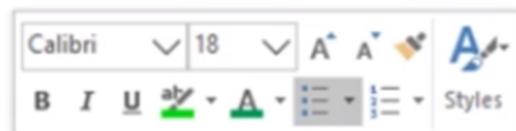
`alert.sendKeys("String");` This will allow to type in some text into the alert

`alert.getText();` This will return the text that appears on the dialog

10. What are the different exceptions you faced in Selenium WebDriver?

- WebDriverException
- NoSuchElementException
- NoSuchFrameException

- NoAlertPresentException
- NoSuchElementException
- ElementNotVisibleException
- ElementNotInteractableException
- SessionNotCreatedException
- TimeoutException
- InvalidSelectorException
- IllegalStateException
- StaleElementReferenceException



11. What is a framework? What are the different types of frameworks available?

A framework is a charter of rules and best practices for the systemic resolution of a problem.

There are different kinds of automation frameworks:

- Data-Driven Testing Framework
- Keyword Driven Testing Framework
- Hybrid Testing Framework
- Behavioural Driven Framework

12. How to run Tests in Headless Mode with Chrome?

```
ChromeOptions chromeOptions = new ChromeOptions();
```

```
chromeOptions.addArguments("--no-sandbox");
chromeOptions.addArguments("--headless");
driver = new ChromeDriver(chromeOptions)
```

13. How to handle windows-based alerts/popups in selenium?

Selenium only supports web applications and does not provide a way to automate Windows-based applications. However, the following approaches can help.

- Use the `Robot` class (Java-based) utility to simulate the keyboard and mouse actions. That is how you can handle the window based pop.
- AutoIt Integration with Selenium help to automate Window Based Popups

14. What are Listeners in Selenium?

Listeners is an interface that modifies the behavior of the system. Listeners allow customization of reports and logs.

Listeners mainly comprise of two types, namely

- WebDriver listeners

- TestNG listeners

15. What are the differences between StringBuffer and StringBuilder?

StringBuffer is *synchronized*, i.e., thread safe. It means two threads can't call the methods of StringBuffer simultaneously.

StringBuilder is *non-synchronized*, i.e., not thread safe. It means two threads can call the methods of StringBuilder simultaneously.

16. What are the advantages of selenium in automation testing world?

- It is an open source platform free to use. This method does not need to be allotted budget
- It works on systems such as Windows, Linux and Mac, as it is compatible across systems
- It works on almost all common browsers including Chrome, Firefox, Edge, Internet Explorer and Safari, since it is compatible with cross-browser testing
- It supports Java, Python, Perl, PHP, C #, Ruby programming languages
- Selenium Grid concepts allow parallel execution

- Continuous integration With Jenkins and Hudson we will achieve daily execution
- It provide a wide base of users and support communities because this is an open platform

17. What is soft and hard assertion in selenium?

Soft Assertion: Soft Assert will not throw an exception when an assert fails, and after the assert assertion will continue with the next step.

Hard Assertion: Hard Assert throws an Assert Exception immediately when an assert statement fails and test suite continues with next @Test.

18. What is the purpose of static methods and variables?

The methods or variables defined as static are shared among all the objects of the class. The static is the part of the class and not of the object. The static variables are stored in the class area, and we do not need to create the object to access such variables. Therefore, static is used in the case, where we need to define variables or methods which are common to all the objects of the class.

<https://www.javatpoint.com/static-keyword-in-java>

19. Usage of This and Super Keywords in Java?

This:

- This keyword is used to initialize class level variables in the constructor using local variables
- This keyword can only be used for constructor
- We can use more than one **This** Keyword within the Constructor at a time.

```
base.java validateTitl... LandingPage.... HomePage.java > demo1.java Sample.java >_  
17  
18 public class HomePage extends base{  
19  
20     public static Logger Log = LogManager.getLogger(base.class.getName());  
21 @BeforeTest  
22     public void initialize() throws IOException  
23     {  
24         driver = initializeDriver();  
25     }  
26  
27 }  
28 @Test(dataProvider="getData")  
29  
30     public void basePageNavigation(String Username, String Password, String text) throws IOException  
31     {  
32         //one is inheritance  
33         // creating object to that class and invokes methods of it  
34     }
```

The screenshot shows a Java code editor with several tabs at the top: base.java, validateTit..., LandingPage...., HomePage.java, demo1.java, Sample.java, and »1. The main pane displays the following code:

```
9 public class LandingPage {  
10  
11  
12     public WebDriver driver;  
13  
14     By signin=By.cssSelector("a[href*='sign_in']");  
15     By title=By.cssSelector(".text-center>h2");  
16     By NavBar=By.cssSelector(".nav.navbar-nav.navbar-right>li>a");  
17     By popup=By.xpath("//button[text()='NO THANKS']");  
18  
19  
20  
21  
22     public LandingPage(WebDriver driver) {  
23         // TODO Auto-generated constructor stub  
24  
25         this.driver=driver;  
26     }  
27 }
```

Super:

We can use super keyword to access the data member or field of parent class. It is used if parent class and child class have same fields.

The super keyword can also be used to invoke parent class method

super is used to invoke parent class constructor

<https://www.javatpoint.com/super-keyword>

3) super is used to invoke parent class constructor.

The super keyword can also be used to invoke the parent class constructor. Let's see a simple example:

```
class Animal{  
Animal(){System.out.println("animal is created");}  
}  
class Dog extends Animal{  
Dog(){  
super();  
System.out.println("dog is created");  
}  
}  
class TestSuper3{  
public static void main(String args[]){  
Dog d=new Dog();  
}}}
```

20. Difference Between Array and ArrayList?

Array is a fixed length data structure whereas ArrayList is a variable length Collection class. We cannot change length of array once created in Java but ArrayList can be changed.

```
int[] integerList = new int[10];  
ArrayList<Integer> integerList= new ArrayList<Integer>();  
integerList.add(1);
```

21. Difference between abstract class and interface?

Interface:

- 1) Interface contains only abstract methods
- 2) Access Specifiers for methods in interface must be public
- 3) Variables defined must be public, static, final
- 4) To implement an interface we use implements keyword

Abstract Class:

- 1) Abstract class can contain abstract methods, concrete methods or both
- 2) Except private we can have any access specifier for methods in abstract class.
- 3) Except **private variables** can have any access specifiers
- 5) To implement an interface we use implements keyword

22. Difference between HashMap and Hashtable

HashMap is **non synchronized**. It is not-thread safe and can't be shared between many threads without proper synchronization code.

Hashtable is **synchronized**. It is thread-safe and can be shared with many threads.

HashMap allows one null key and multiple null values.
Hashtable doesn't allow any null key or value.

```
base.java  LandingPage....  HomePage.java  demo1.java  Sample.java  listeners.java  »_2
1 package Academy;
2
3 import java.io.IOException;
4
5 public class listeners implements ITestListener {
6     base b=new base();
7
8     public void onFinish(ITestContext arg0) {
9         // TODO Auto-generated method stub
10        //hey i am done
11    }
12
13    public void onStart(ITestContext arg0) {
14        // TODO Auto-generated method stub
15        //
16    }
17
18    public void onTestFailedButWithinSuccessPercentage(ITestResult arg0) {
19        // TODO Auto-generated method stub
20
21    }
22
23}
```

23. Difference between final, finally and finalize

<https://www.javatpoint.com/difference-between-final-finally-and-finalize>

Difference between final, finally and finalize

← prev

next →

There are many differences between final, finally and finalize. A list of differences between final, finally and finalize are given below:

No.	final	finally	finalize
1)	Final is used to apply restrictions on class, method and variable. Final class can't be inherited, final method can't be overridden and final variable value can't be changed.	Finally is used to place important code, it will be executed whether exception is handled or not.	Finalize is used to perform clean up processing just before object is garbage collected.
2)	Final is a keyword.	Finally is a block.	Finalize is a method.

Java final example

```
class FinalExample{  
    public static void main(String[] args){  
        final int x=100;  
        x=200;//Compile Time Error  
    }}  
}
```

Java finally example

```
class FinallyExample{  
    public static void main(String[] args){  
        try{  
            int x=300;  
        }catch(Exception e){System.out.println(e);}  
        finally{System.out.println("finally block is executed");}   
    }}  
}
```

Java finalize example

```
class FinalizeExample{  
    public void finalize(){System.out.println("finalize called");}  
    public static void main(String[] args){  
        FinalizeExample f1=new FinalizeExample();  
        FinalizeExample f2=new FinalizeExample();  
        f1=null;  
        f2=null;  
        System.gc();  
    }  
}
```

24. When should I use Selenium Grid?

Selenium Grid can be used to execute same or different test scripts on multiple platforms and browsers concurrently so as to achieve distributed test execution, testing under different environments and saving execution time remarkably.

25. How many objects will be created in the following code?

1. String s1="Welcome";

[

2. String s2="Welcome";

3. String s3="Welcome";

Only one object will be created using the above code because strings in Java are immutable.

Each time you create a string literal, the JVM checks the "string constant pool" first. If the string already exists in the pool, a reference to the pooled instance is returned. If the string doesn't exist in the pool, a new string instance is created and placed in the pool. String objects are stored in a special memory area known as the **string constant pool**

Module 2

Automation Framework Interview Ques

In this Module, We will go over most frequently asked Automation Framework Interview Questions with answer Versions for both TestNG and Cucumber tools

1 - Automation Framework Interview Questions	
<p>1. What are the main file components of the Automation framework built for Selenium?</p> <p>Framework Built in TestNG</p> <ul style="list-style-type: none">1. TestNG file (.xml extension)2. Tests with proper TestNG Annotations driven by BaseDriver class3. Page object Package4. Maven Driven Project template with pom.xml5. Data driven Mechanism6. Reporting , Logfiles7. Custom Utilities/Helper Methods8. Jenkins file for CI Pipeline	<p>Framework Built in Cucumber</p> <ul style="list-style-type: none">1. Feature file (.feature extension)2. Step Definition file (.java extension)3. Test Runner file (.java extension)4. Hooks File for Pre and Post condition setup5. Page object Package6. Maven Driven Project template with pom.xml7. Reporting , Logfiles8. Custom Utilities/Helper Methods9. Jenkins file for CI Pipeline

2.

How do you handle data parameterization globally in Automation Frameworks ?

Framework Built in TestNG

testNG File

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Test Cycle">
<parameter name = "Url" value="http://qa.clickacademy.com" />
<test name = "Regression">
    <classes>
        <class name="test.cycle"/>
    </classes>
</test>
</suite>
```

Here the testNG xml file contains the site Url

Class File

```
@Parameter({"Url"})
@Test
public void Login(String url){
    driver.get(url);
}
```

The corresponding test method in the class file

Framework Built in Cucumber

Feature File

```
Feature: QAClick Login
Scenario: QAClick Login Scenario
Given Navigate to site "http://qa.clickacademy.com"
```

Here the Feature file contains the site Url.

Step Definition File

```
@Given("^Navigate to site \"([^\"]*)\"$")
public void navigate_to_site(String url) throws Throwable {
    driver.get(url);
}
```

The corresponding mapping of the feature file to the step definition file.

Automation Framework Interview Questions

3.

How do you handle data parametrizations specific to Tests in Automation Frameworks ?

Framework Built in TestNG

Class File

```
@DataProvider(name="ProvideSearch")
public Object[][] getDataFromprovider(){
    return new Object[][]
    {
        { "test", "test1" },
        { "qa.click", "test2" },
    };
}

@Test(dataProvider="ProvideSearch")
public void Userenters(String usrn, String pwd)
    System.out.println(usrn);
    System.out.println(pwd);
}
```

DataProvider attribute helper used for parameterizing.

Framework Built in Cucumber

Feature File

```
_Feature: Login Feature
Scenario Outline: Login Test
Given User is on Login Page
When title of login page is QAClick
Then User enters "<username>" and "<password>"
```

Examples:

username	password
qa.click	test#123
test	t123

Step Definition File

```
@Then("^User enters \"(.*)\" and \"(.*)\"$")
public void User_enters(String usrn, String pwd) throws Throwable {
    System.out.println(usrn);
    System.out.println(pwd);
}
```

The corresponding mapping of the parametrized **Then** in feature file.



4.

How to run selected tests from the set of test cases in Automation Frameworks ?

Framework Built in TestNG

testNG File

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Loan Department">
<test name="Regression">
  <groups>
    <run>
      <include name = "Sanity"/>
    </run>
  </groups>
  <classes>
    <class name="test.day1"/>
  </classes>
</test>
</suite>
```

Here the testNG xml contains **group Sanity** to be **included** in run.

Class File

```
@Test(groups={"Sanity"})
public void Login(){
    System.out.println("Login is successful");
}
```

Only the test methods having groups as **Sanity** will be executed, out of the full regression suite.

Framework Built in Cucumber

Feature File

```
@Regression Test
Feature: Login Feature Testing

@Sanity
Scenario: Login Test
Given User is on Login Page

@Regression
Scenario: Payment Test
Given User is on Payment Page
```

Here the Feature file contains the scenarios with tags **Sanity** and **Regression**.

Test Runner File

```
@RunWith(Cucumber.class)
@CucumberOptions(
    features = "src/test/java/features",
    glue="stepDefinations"
    tags = {@Sanity}
)
```

Test Runner File contains the tag **@Sanity** hence only the scenarios with **Sanity** tag will executed.

RahulShettyAcademy.com

5.

How to skip selected test method from the set of test cases in Automation Frameworks ?

Framework Built in TestNG

Class File

```
@Test(enabled=false)
public void validateDate()
{
    System.out.println("Validate date is successful");
}
```

Here the helper attribute **enabled set to false** is used to skip a particular test method from execution.

RahulShettyAcademy

Framework Built in Cucumber

Feature File

```
@Regression Test
Feature: Login Feature Testing

@Sanity
Scenario: Login Test
Given User is on Login Page

@Regression
Scenario: Payment Test
Given User is on Payment Page
```

Here the Feature file contains the scenarios with tags **Sanity** and **Regression**.

Test Runner File

```
@RunWith(Cucumber.class)
@CucumberOptions(
    features = "src/test/java/features",
    glue="stepDefinations"
    tags = {"~@Sanity"}
)
```

In order to **skip the scenarios with @Sanity** , **~** is placed before **@Sanity** tag.

7.

How to include and exclude test methods from set of test cases in Automation Framework ?

Framework Built in TestNG

testNG File

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Loan Department">
<test name="Cycle">
  <groups>
    <run>
      <include name = "Sanity"/>
      <exclude name = "Regression"/>
    </run>
  </groups>
  <classes>
    <class name="test.day1"/>
  </classes>
</test>
</suite>
```

Here the testNG xml contains group **Sanity to be included** and **Regression to be excluded** in run.

Class File

```
@Test(groups={"Sanity"})
public void login()
{
    System.out.println("Login successful");
}
@Test(groups={"Regression"})
public void verifyPayment()
{
    System.out.println("Payment successful");
}
```

Framework Built in Cucumber

Feature File

```
@Regression
Feature: Login Feature Testing

@Sanity
Scenario: Login Test
Given User is on Login Page

@Regression
Scenario: Payment Test
Given User is on Payment Page
```

Here the Feature file contains the scenarios with tags **Sanity** and **Regression**.

Test Runner File

```
@RunWith(Cucumber.class)
@CucumberOptions(
    features = "src/test/java/features",
    glue="stepDefinitions"
    tags = {"@Sanity" , "~@Regression"})
)
```

Here the test runner file contains tag **Sanity to be included** and **Regression to be excluded** in run.

RahulShettyAcademy.com

8.

How to execute pre conditions and post conditions for each test method in Automation Framework?

Framework Built in TestNG

Class File

```
@BeforeMethod
public void beforeevery()
{
    System.out.println("I will execute before every test method");
}

@AfterMethod
public void afterevery()
{
    System.out.println("I will execute after every test method");
}
@Test
public void Login()
{
    System.out.println("Login successful");
}
```

First the method with tags **@BeforeMethod** will be executed (pre condition), followed by the test method (`Login()`) and finally the **@AfterMethod** will be executed(post condition).

Framework Built in Cucumber

Feature File

```
Feature: Daily Time Table
Scenario: Morning and Evening Time Table
Given I go to office in the morning
```

Feature file contains the above scenario.

Hooks File

```
@Before
public void prerequisite()
{
    System.out.println("I will execute first");
}
@After
public void postcondition()
{
    System.out.println("I will execute last");
}
@Given("^I go to office in the morning$")
public void goTo()
{
    System.out.println("This is the scenario");
}
```

First the method with hook **@Before** will be executed (pre condition), followed by the test method (`goTo()` method) and finally the **@After** will be executed(post condition).

RahulShettyAcademy.com

8.

How to execute pre conditions and post conditions for each test method in Automation Framework?

Framework Built in TestNG

```
@BeforeMethod
public void beforeevery()
{
    System.out.println("I will execute before every test method");
}

@BeforeMethod
public void afterevery()
{
    System.out.println("I will execute after every test method");
}
@Test
public void Login()
{
    System.out.println("Login successful");
}
```

First the method with tags **@BeforeMethod** will be executed (pre condition), followed by the test method (Login()) and finally the **@AfterMethod** will be executed(post condition).

Framework Built in Cucumber

Feature File

```
Feature: Daily Time Table
Scenario: Morning and Evening Time Table
Given I go to office in the morning
```

Feature file contains the above scenario.

Hooks File

```
@Before
public void prerequisite()
{
    System.out.println("I will execute first");
}

@After
public void postcondition()
{
    System.out.println("I will execute last");
}

@Given("I go to office in the morning$")
public void goTo()
{
    System.out.println("This is the scenario");
}
```

First the method with hook **@Before** will be executed (pre condition), followed by the test method (goTo() method) and finally the **@After** will be executed(post condition).

10.

How to set priority for execution in Automation Framework ?

Framework Built in TestNG

```
@Test(priority = 1)
public void verifyLogin()
{
    System.out.println("Login successful");
}

@Test(priority = 2)
public void verifyPayment()
{
    System.out.println("Payment successful");
}
```

Here each test method is assigned with **priority**. Test method with **lower number** will be executed first followed by the test method having **higher priority**. verifyLogin() will be executed first then verifyPayment().

Framework Built in Cucumber

Step Definition File

```
@Before(order=1)
public void prerequisite()
{
    System.out.println("I will execute first");
}

@Before(order=2)
public void prerequisite2()
{
    System.out.println("I will execute second");
}

@Given("I go to office in the morning$")
public void goTo()
{
    System.out.println("This is the scenario");
}
```

Here each test method is assigned with **order**. Test method with **lower order** (prerequisite()) will executed first . Then prerequisite2() test method will be executed which is having a higher order.

Once these pre conditions get executed successfully test method goTo() will be executed.

11. What are the different annotations available in TestNG?

1. @Test
2. @BeforeSuite
3. @AfterSuite
4. @BeforeClass
5. @AfterClass
6. @BeforeTest
7. @AfterTest
8. @BeforeGroups
9. @AfterGroups
10. @BeforeMethod
11. @AfterMethod

12. What is invocation count in TestNG?

If we are required to execute a test case N number of times , then **invocationCount helper attribute** is used.

```
@Test(invocationCount=5)
public void goTo()
{
    System.out.println("This scenario will execute 5 times");
}
```

In the above example , **This scenario will execute 5 times** will print on the console five times.

13.

What is timeOut in TestNG and Cucumber?

If there any test method that consumes a **lot of time for execution**, we can **terminate** that method with the help of **timeOut** helper attribute in TestNG.

```
@Test(timeOut = 6000)
public void ValidatePayment()
{
    System.out.println("Payment Successful");
}
```

After 6000ms , the test method will be terminated and will be marked as **Failed**.

Cucumber :

```
@Then(value = "^verify (\d+) events sent$", timeout = 5000)
```

14.

How to achieve parallel execution in TestNG?

We use **parallel** attribute in testng.xml to achieve parallel execution in TestNG. It also has a parameter called as **thread-count**. The parallel can have the following values:

1. Tests
2. Classes
3. Methods
4. Instances

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Regression" parallel="tests" thread-count="6">
    <test name="Cycle1">
        <classes>
            <class name="test.day1"/>
        </classes>
    </test>
</suite>
```

In the above example , execution will happen in parallel mode for **tests** , having **thread- count** of 6.

15.

Explain the advantages of Cucumber.

Some of the advantages of cucumber are as the followings:

1. Involves the business stake holders who are not having coding knowledge.
2. Its is an open source tool.
3. It reduces communication gaps among Business Analyst, Developers and QA in an Agile environment.
4. Plain text representation makes it easy to understand for non technical individuals.
5. It is easy to maintain and work with.

16. What is Scenario Outline in Cucumber ?

Scenario Outline is the same scenario being executed for multiple data in multiple combinations. The data set is represented in form of table separated by (||). Each row constitute a set of data.

```
Feature: Login into Application
Scenario Outline: Positive test validating login
Given Initialize the browser with chrome
And Navigate to "http://qaclickacademy.com" Sit
When User enters <username> and <password> and logs in

Examples:
|username          |password|
|test99@gmail.com |123456   |
|test123@gmail.com|12345    |
```

In the above example , **When** having <username> and <password> will be executed with three combinations of data defined under **Examples** separated by (||).

17. What is the use of glue in Cucumber Options tag ?

Cucumber Identifies the **location of the step definition files** from the glue property in Cucumber Options tags.

```
@RunWith(Cucumber.class)
@CucumberOptions(
    features = "src/test/java/features",
    glue="stepDefinitions")
public class TestRunner {
}
```

18.

How will we achieve encapsulation in our Automation Framework ?

Encapsulation means enclosing anything in a container . It is done to hide details and it is achieved by the help of access modifiers like **public , protected and private**.

```
public class Loginpage {  
    public WebDriver driver;  
    private By username=By.xpath(".//*[@id='login1']");  
    private By Password=By.name("password");  
  
    public Loginpage(WebDriver driver) {  
        // TODO Auto-generated constructor stub  
        this.driver=driver;  
    }  
    public WebElement EmailId()  
    {  
        return driver.findElement(username);  
    }  
  
    public WebElement Password()  
    {  
        return driver.findElement(Password);  
    }  
}
```

```
@Test  
public void LoginTC(){  
  
    Loginpage rd=new Loginpage(driver);  
    rd.EmailId().sendKeys("hello");  
    rd.Password().sendKeys("hello");  
}
```

Here we have declared the variables as private and methods accessing them as public. Thus , while we are creating the objects , we are not handling them directly but with the help of the methods , thus keeping variables intact.

19. How will we declare global variables in Automation Framework ?

We can declare global variables in Automation Framework with the help of the **Properties** class and a **properties** file . The properties file contains global variables in the form of **key – value pairs**. We can read and write values in the .properties file.

Class File

```
public void login() throws IOException {
    Properties prop = new Properties();
    //Reading values from property file
    FileInputStream ips = new FileInputStream(
        "C:\\Users\\ghs6kor\\eclipse-workspace\\Inheritance\\config.properties");
    prop.load(ips);
    System.setProperty("webdriver.gecko.driver", "C:\\Users\\ghs6kor\\Desktop\\geckodriver.exe");
    WebDriver driver = new FirefoxDriver();
    driver.get(prop.getProperty("url"));
    //Writing values from property file
    FileOutputStream ops = new FileOutputStream(
        "C:\\Users\\ghs6kor\\eclipse-workspace\\Inheritance\\config.properties");
    String urlnm = driver.getTitle();
    prop.setProperty("title", urlnm);
    prop.store(ops, null);
}
```

Properties File

```
config.property facepathjava
1#Sun Jan 26 15:35:30 EST 2020
2url=https://www.google.com/
3browser=firefox
4title=Google
```

The **title** key writes the value **Google** in the properties file. It reads the keys **url**.

RahulShettyAcademy.com

20. How do you deal with reusable components in Automation Framework ?

We can deal with the reusable components in Automation Framework with the help of **inheritance** concept . It's a **parent child relationship** where the child inherits the properties of the parent class.

Parent Class

```
public class Baseclass {
    public void login() throws IOException {
        Properties prop = new Properties();
        //Reading values from property file
        FileInputStream ips = new FileInputStream(
            "C:\\Users\\ghs6kor\\eclipse-workspace\\Inheritance\\config.properties");
        prop.load(ips);
        System.setProperty("webdriver.gecko.driver", "C:\\Users\\ghs6kor\\Desktop\\geckodriver.exe");
        WebDriver driver = new FirefoxDriver();
        driver.get(prop.getProperty("url"));
    }
}
```

Child Class

```
public class Child extends Baseclass {
    public void testinheriance() throws IOException {
        login();
    }
}
```

Here the Child class **extends** the Baseclass and calls the **login()** method from it thereby achieving reusability of code.

20.

How do you deal with reusable components in Automation Framework ?

We can deal with the reusable components in Automation Framework with the help of **inheritance** concept . It's a **parent child relationship** where the child inherits the properties of the parent class.

Parent Class

```
public class Baseclass {  
    public void login() throws IOException {  
        Properties prop = new Properties();  
        //Reading values from property file  
        FileInputStream ips = new FileInputStream(  
            "C:\\\\Users\\\\ghs6kor\\\\eclipse-workspace\\\\Inheritance\\\\config.properties");  
        prop.load(ips);  
        System.setProperty("webdriver.gecko.driver", "C:\\\\Users\\\\ghs6kor\\\\Desktop\\\\geckodriver.exe");  
        WebDriver driver = new FirefoxDriver();  
        driver.get(prop.getProperty("url"));  
    }  
}
```

Child Class

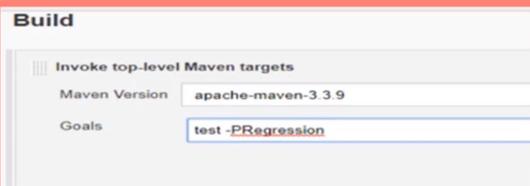
```
public class Child extends Baseclass {  
  
    public void testinheriance() throws IOException {  
        login();  
    }  
}
```

Here the Child class **extends** the Baseclass and calls the login() method from it thereby achieving reusability of code.

21.

How to run Automation Framework from Jenkins ?

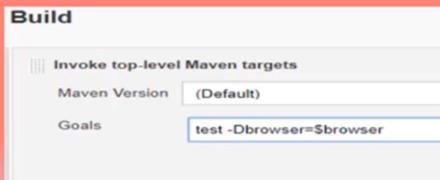
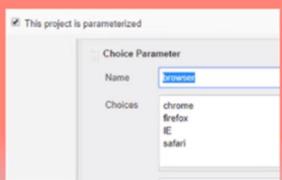
We can run Automation Framework from Jenkins with the help of Maven commands. We need to first configure Java and Maven paths in Jenkins. Then while creating the Jenkins jobs , the select **Invoke top-level Maven targets** from the Build section.



Here in the Goals , we need to provided the Maven commands.

test -PRegression is the specific command to trigger build without parameters.

Parameterize Jenkins Build



Here first the parameters are defined in Jenkin jobs. Then the Maven Goals is set with **test -Dbrowser=\$browser**, with parameter **browser**.

22.

How do we arrange locators in Automation Framework ?

We can arrange locators in Automation Framework in Page Object Model in the following way:

- Segregate locators for a particular screen in separate Java class. For example , LoginPage java class will contain objects with respect to that screen. While the PaymentPage java class will contain objects with respect to that screen.
- The test cases are maintain in different Java classes where we will use the locators by calling the objects from that particular page (for example , LoginPage) to the test case.
- In case of any locator needs to be updated/deleted , we need to modify them to that particular java class for that specific screen and not every where.
- Easy to use and maintain.

23.

Which is the Maven command to achieve profiling?

We can do profiling in Maven using the command : **mvn test -PSmoke**

Pom file

```
<profiles>
<profile>
<id>Regression</id>
<build>
<pluginManagement>
<plugins>
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-surefire-plugin</artifactId>
<version>2.20.1</version>
<configuration>
<suiteXmlFiles>
<suiteXmlFile>testng2.xml</suiteXmlFile>
</suiteXmlFiles>
</configuration>
</plugin>
</plugins>
</pluginManagement>
</build>
</profile>
<profile>
<id>Smoke</id>
<build>
<pluginManagement>
<plugins>
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-surefire-plugin</artifactId>
<version>2.20.1</version>
<configuration>
<suiteXmlFiles>
<suiteXmlFile>testng.xml</suiteXmlFile>
</suiteXmlFiles>
</configuration>
</plugin>
</plugins>
</pluginManagement>
</build>
</profile>
</profiles>
```

Here we have defined two profiles with Id **Regression** and **Smoke**. With the command , given above we will be able to trigger the test cases with profile id **Smoke** pointing to the **testng.xml** file. In this way , we can have N number of TestNG xml files driven by a single pom file.

Similarly , to trigger the execution with profile Id **Regression** , Maven command is :
mvn test -PRegression . In this case , **testng2.xml** file will be pointed.

24.

How to capture Screenshot automatically on Test Failure in the Framework

Using TestNG Listeners

```
• public void onTestFailure(ITestResult result) {  
    // Selenium Screenshot Method  
}
```

Cucumber do not have Listeners by default. We should Integrate it with TestNG to enjoy the features of Cucumber

```
@CucumberOptions(...)  
@Listeners({com.coveros.utilities.Listener.class})  
public class MyTestsRunner extends AbstractTestNGCucumberTests {  
}  
RahulShettyAcademy.com
```

25.

How do you manage code when multiple people in the team contributing for the Framework

Using Version Control Tools like

GIT

SVN

Module 3

API Testing Interview Questions

In this Module, We will go over most frequently asked API Testing Interview questions both Manual + Automation(Rest Assured)

Top 25 API Testing Interview Questions – Rahul Shetty

What all challenges are included under API testing?

API Documentation

Access to DB

Authorization overhead

1. What is the difference between PUT and POST methods?

POST request - creating a new object on the server

PUT request, Update the object in server with new value

2. What are commonly used HTTP Methods?

GET: It enables you to retrieve data from a server

POST: It enables you to add data to an existing file or resource in a server

PUT: It lets you replace an existing file or resource in a server

DELETE: It lets you delete data from a server

3. List out few Authentication Techniques used in API's?

Session/Cookie Based Authentication

Basic Authentication

Digest Authentication

OAuth

4. Why API testing is determined as the most suitable form for Automation testing?

[

5. What is Rest API?

REST stands for Representational State Transfer. It is a set of functions helping developers in performing requests and receive responses. Interaction is made through HTTP Protocol in REST API.

6. What exactly needs to verify in API testing?

In API testing, we send a request to API with the known data and then analysis the response.

1. We will verify the accuracy of the data.
2. Will see the HTTP status code.
3. We will see the response time.
4. Error codes in case API returns any errors.
5. Authorization would be checked.
6. Non-Functional testing such as performance testing, security testing.

7. What are Path Parameters and Query Parameters for below API request URL

<http://rahulshettyacademy.com/orders/112234?location=IND>

[]

8. What are the core components of an HTTP request?

1. HTTP Request methods like:, PUT, POST, DELETE.
2. Base Uniform Resource Identifier (URI)
3. Resources and Parameters
4. Request Header, which carries metadata (as key-value pairs) for the HTTP Request message.
5. Request Body, which indicates the message content or resource representation.

9. What could be the HTTP Method for below API Scenario? Answer if it is GET or POST

Scenario: An API which has Endpoint, Parameters, Headers, cookies and Payload

Ans :

10. What are the differences between API testing and UI testing?

UI (User Interface) testing means the testing of the graphical user interface. The focus of UI testing is on the look and feel of the application. In user interface testing the main focus is on how users can interact with app elements such as images, fonts, layout etc. are checked.

API testing allows the communication between two software systems. API testing works on backend also known as backend testing.

11. What protocol is used by the RESTFUL Web Services?

RESTFUL Web Services uses the HTTP protocol. They use the HTTP protocol as a medium of communication between the client and the server.

12. What are Soap WebServices?

SOAP stands for Simple Object Access Protocol. It is an XML based messaging protocol. It helps in exchanging information among computers.

13. How do we Represent A Resource In REST?

Using HTTP Methods.

14. Can you use GET request instead of PUT to create a resource?

No, GET request only allows read only rights. It enables you to retrieve data from a server but not create a resource. PUT or POST methods should be used to create a resource.

POST should be used when the client sends the page to the server and then the server lets the client know where it put it. PUT should be used when the client specifies the location of the page

15. Can you use POST request instead of PUT to create a resource?

Yes We can. Because POST is super set of all other HTTP methods except GET

16. What Do You Understand by Payload In Restful Web Service?

Ans. Payload/Body is the secured input data which is sent to API to process the request.
Payload is generally represented in Json format in Rest API's

17. How do we differentiate Path and Query Parameters in API Request Endpoint?

18. What is Rest Assured?

It is Java library which can automate Rest API's

19. How would we define API details in Rest Assured Automation?

We shall define all the request details and Send it to sever in GIVEN, WHEN, THEN methods

20. What is Json Serialization & Deserialization in Rest Assured?

Serialization in Rest Assured context is a process of converting a Java object into Request body (Payload)

Rest Assured also Supports **deserialization** by converting Response body back to Java object

20. What is Json Serialization & Deserialization in Rest Assured?

Serialization in Rest Assured context is a process of converting a Java object into Request body (Payload)

Rest Assured also Supports **deserialization** by converting Response body back to Java object

21. List out few common Json Parsing Techniques used in Rest Assured Automation?

Json Path

Deserialization of Json using POJO Classes

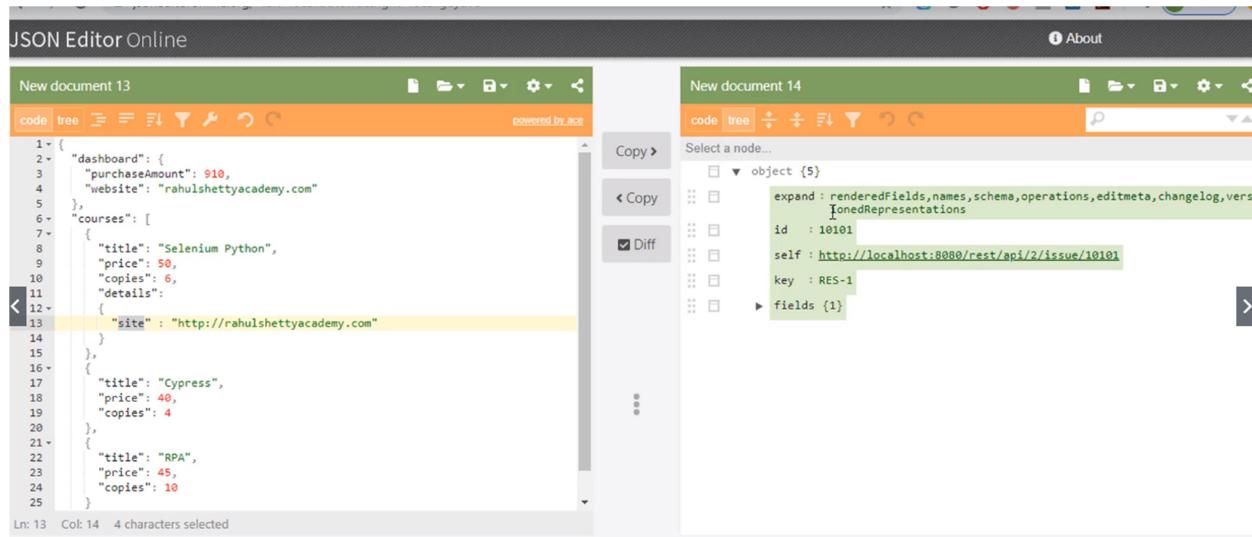
22. How would you send attachments to API using Rest Assured Test?

Using Multipart method

23. Different Status Codes and their descriptions

200	OK	The request was successfully completed.
201	Created	A new resource was successfully created.
400	Bad Request	The request was invalid.
401	Unauthorized	The request did not include an authentication token or the authentication token was expired.
403	Forbidden	The client did not have permission to access the requested resource.
404	Not Found	The requested resource was not found.
405	Method Not Allowed	The HTTP method in the request was not supported by the resource. For example, the DELETE method cannot be used to retrieve a resource.
409	Conflict	The request could not be completed due to a conflict. For example, POST ContentStore Folder API cannot complete if there is already a folder with the same name in the parent location.
500	Internal Server Error	The request was not completed due to an internal error on the server side. 
503	Service Unavailable	The server was unavailable.

24. Get the Json Path of "site" from below JSON



The screenshot shows two separate JSON Editor Online windows side-by-side.

Left Window (New document 13):

```

1: {
2:   "dashboard": {
3:     "purchaseAmount": 910,
4:     "website": "rahulshettyacademy.com"
5:   },
6:   "courses": [
7:     {
8:       "title": "Selenium Python",
9:       "price": 50,
10:      "copies": 6,
11:      "details": [
12:        {
13:          "site": "http://rahulshettyacademy.com"
14:        }
15:      ],
16:      {
17:        "title": "Cypress",
18:        "price": 40,
19:        "copies": 4
20:      },
21:      {
22:        "title": "RPA",
23:        "price": 45,
24:        "copies": 10
25:      }
26:    ]
27:  }
28: }
```

Right Window (New document 14):

JSON Path results:

```

object {5}
  id : 10101
  self : http://localhost:8080/rest/api/2/issue/10101
  key : RES-1
  fields {1}
```

A screenshot of a code editor interface. On the left, there is a code editor window displaying the following JSON code:

```
1+ {
2+   "dashboard": {
3+     "purchaseAmount": 910,
4+     "website": "rahulshettyacademy.com"
5+   },
6+   "courses": [
7+     {
8+       "title": "Selenium Python",
9+       "price": 50,
10+      "copies": 6,
11+      "details": {
12+        "site": "http://rahulshettyacademy.com"
13+      }
14+    },
15+    {
16+      "title": "Cypress",
17+      "price": 40,
18+      "copies": 4
19+    },
20+  ],
21+  [
22+    {
23+      "title": "RPA",
24+      "price": 45,
25+      "copies": 10
26+    }
27+  ]
28+}
```

On the right, there is a diff tool window comparing two versions of the same JSON structure. The left pane shows the original JSON, and the right pane shows the modified version where the title of the first course has been changed to "Selenium Python".

A screenshot of a code editor interface showing a detailed JSON tree view. The path selected is `object > courses > 0 > details > site`. The tree structure is as follows:

- object {2}
 - dashboard {2}
 - courses [3]
 - 0 {4}
 - title : Selenium Python
 - price : 50
 - copies : 6
 - details {1}
 - site : <http://rahulshettyacademy.com>
 - 1 {3}
 - 2 {3}

A screenshot of a Notepad window titled "Untitled - Notepad". The menu bar includes File, Edit, Format, View, Help. The content area contains the following text:

courses[0].title

courses[0].details.site

Module 4

Selenium WebDriver API Interview Questions

In this Module, We will go over the Interview Questions related to WebDriver API Methods with Code samples

Part 1:

1. Difference between get and navigate method in Selenium?
get("google.com")- click
navigate("google.com")-back ,forward or refresh

2. Difference between quit and close methods in Webdriver?
Close method- 3rd browser-
Quit- all the 3 browsers
3. What is implicit wait?
Beginning of test – 5 seconds-
3 seconds- `driver.manage().Timeout().implicitwait(TimeUnit.seconds,5)`
4. Difference between implicit and explicit wait?
2 minutes to load-(Explicit wait 2 minutes)
5. In how many ways we can handle frames in the application using webdriver methods?
Frames- `driver.switchTo().frame(id)- id, name, frame webelement`
6. Code to handle 3rd child window?
Selenium will be on first window- (Which it opens)
`Set<String> s=Driver.getWindowHandles();
It=s.iterator();
It.next()//0 index of set box
It.next()// 1st index of S
String windowid3rd=It.next();

Driver.switchTo().window(windowid3rd)`

7. How to handle https certifications?
`DesiredCapabilities cap=DesiredCapabilities.chrome();

// Set ACCEPT_SSL_CERTS variable to true
cap.setCapability(CapabilityType.ACCEPT_SSL_CERTS, true);`

8. Different type of locators present in webdriver?

Xpath, css, id classname, linkText, name

9. Write Syntax for xpath and CSS if id and tag are given

Xpath syntax = //tagname[@id='value']

Css= tagname[id=value]

10. How to use Contains regular expression to xpath? I

Id= u012 ua34 u0544

//tagname[contains(@id,'uo')]

11. How to use regular expression to CSS?

Tagname[id*='u']

12. What is the class available in selenium to handle drop downs?

Select

13. What is the method to check if checkbox is selected?

Driver.findElement("locator").isSelected() //true

14. How to validate if element is visible or hidden in webpages?

Driver.findElement("locator").isDisplayed(); //true

15. How to get the count of similar objects list in the web page?

Class name=abc

Driver.findElements(By.className("abc")).size()

T^E

16. Importance of desiredcapabilites Mechanism/ Selenium Grid

Distrubute tests across multiple machines/OS..

17. How to enter the text in caps lock?

Driver.findElement(By.xpath("//fdfd")).sendKeys(Keys.SHIFT,"hello");

HELLO

18. How to mouse over on the web element on page?

Actions a =new Actions(driver);

a.moveToElement().build().perform();

18.How to mouse over on the web element on page?

```
Actions a =new Actions(driver);
a.moveToElement(locator).build().perform();
```

19.Methods to handle Java Alert?

```
Driver.switchTo.Alert();
```

20.How to get links count in the page?

```
Driver.findElements(By.tagName("a")).size();
```

21.How to validate if we are navigated to child window successfully?

```
Driver.getTitle()
```

22.Difference between relative and absolute xpath?

Absolute xpath- html/body – to actual element

Relative-

23.Write down the sample xpath syntax to handle parent from child object?

```
//parent/child
```

24.What driver is must to run tests in Firefox driver?

```
Geckodriver-
```

25.What driver is must to run tests in Chrome Browser?
Chrome driver

26.How do you set driver in Firefox and chrome drivers through script?
System.setProperty("Webdriver.chrome.driver", path to chrome)

- 27.Difference between find Element and find Elements
- 28.List out any 2 methods available in explicit wait
- 29.How to take screenshots with selenium webdriver
- 30.How to hit enter from webdriver commands

Module 5 Software Testing + Behavioral Interview Questions

In this Module, We will go over the Manual Testing Project related questions and QA Behavioral questions which are frequently asked in Managerial rounds

Suppose you find a bug in production. how would you make sure that the same bug is not introduced again?

Add uncaught functionality to regression test cases
If you have Automated Regression Suite, then write a new Script which validates above functionality

What do you do when your developer denies that what you filed IS A BUG?

- Provide Business Documentation reference to support why the existing functionality is not as per design.
- Involve Product Owner / Business Analyst for Discussion.

IF Bug is not reproducible then

- Provide Screenshots of the Bug, Give Timestamp on when you reproduced this so that Developer can check in Application Logs.
- Provide Test Data you have used for replicating issue

What has been one of your greatest challenges while doing regression testing?

- Test Data issue
- Improper selection of regression test cases might skip a major regression defect to be found

Difference between functional and non-functional testing?

- Functional testing verifies that features/System working as expected according to requirements,
- Nonfunctional requirements: How Well the system does it within design & resource constraints

What has been one of your greatest challenges while doing regression testing?

- Test Data issue
- Improper selection of regression test cases might skip a major regression defect to be found

Enlist some of the key challenges that are faced while performing Software Testing

- Data Issues,
- Environment Available,
- Using right set of tools

What are the different levels of testing?

- Unit Testing
-  Integration Testing
-  System Testing
-  Acceptance Testing

What are the drawbacks of the Agile implementation/ methodology that you faced?

- Sprints are usually very deadline constrained.
- Documentation is not the priority
- Frequent change in requirements

What is your approach when you have a high priority release to be delivered in a very short time?

- Run Automation Suites
- . Run Unit tests.
- Manual testing on high level Priority Business test cases

Give an example for High Priority Low Severity Bug

- Wrong Logo Image in Application

What is your understanding regarding a Test plan?

- test plan is a document that consists of scope, approach, resources, and outline of the testing project as well as the activities for tracking the progress of the project.

Explain what will be your reaction if a project you had been working got Sudden change in its deadline?

- As a QA Engineer, Be open on your thoughts if you can deliver the project with QA Sign off covering all testcases
- If Pre release is must, then discuss the opportunities of increasing QA Resources or possibility of partial product delivery
- You have the power to hold QA Sign off if you are not satisfied with quality of product which eventually stops the release date

What is the difference between Smoke testing and Sanity Testing?

- Sanity testing is a kind of Software Testing performed after receiving a software build, with minor changes in code, or functionality, to ascertain that the bugs have been fixed and no further issues are introduced due to these changes.
- Smoke Testing is a special type of testing performed on Software build to check the critical functionalities of the program

Differentiate Ad-hoc Testing and Exploratory Testing?

- Adhoc testing Includes learning the application first and then proceeding with the testing process.
- Exploratory testing form of testing includes learning the application while testing.

What are your daily activities as a member of automation tester in your office?

- Running Smoke Suite on daily Basis and sending the mail to team on health of the application
- Verify Defects Assigned to you and take appropriate Actions
- Working on Manual and Automation testing for the stories in Current Sprint

How do you select regression test cases or form the regression test suite?

- Include the test cases that verify core features of the application:
- . Include the test cases for functionalities that have undergone recent changes:
- Include the test cases that have frequently yielded bugs:

Have you ever managed writing the test cases without having any documents?

In one of my previous project, we had to redevelop our internal tool with new Technology, But There are no testcases/Documentation for the old/existing product. As there is no documentation, Below are the steps I have followed

- Understand and exploring the existing Product to come up with Scenarios.
- Spending time with Product owner or Seniors to understand the Business of the tool.
- Going through production Bugs which found previously for product so that edge testcases are not missed in writing the tests for Upgraded product

What is the first action you perform as a tester when application throws any weird Errors?

- Open Developer tools-> Check in Console for any Java Script Errors
- Open Network Tab -> And see if any request Responses are failed
- Verify Application logs to understand the actual issue

The screenshot shows a website for 'RAHUL SHETTY' with a navigation bar including Home, Courses, Mentorship (which is highlighted in red), Practice Projects, Consulting, Jobs, Articles, About, and Contact.

The main content area displays a 'BRONZE' mentorship package for short-term use, listing benefits like direct access to Rahul Shetty on live call, 30-minute time box, email support for 3 weeks, and resume review/career advice. It costs \$39.00 and has a 'JOIN NOW' button.

To the right, there's a testimonial from Leigha Marsh, a Senior Quality Engineer at Wells Fargo, who discusses her transition from manual testing to automation. Below the testimonial is another quote from someone struggling with landing a job as a software automation engineer.

At the bottom, a developer tools console window is open, showing the Chrome DevTools interface with the 'Console' tab selected. It displays several error messages related to non-unique IDs for service type inputs, indicating a potential UI or server-side issue.

Error in above and will get to know we are getting proper request and response

Name	Status	Type	Initiator	Size	Time	Waterfall
Font	200	font	jquery.min.js:2694	(disk cache)	5 ms	
Font	200	font	jquery.min.js:2694	(disk cache)	20 ms	
Script	200	script	controltag.js:73ba...5	239 B	28 ms	
Script	200	script	controltag.js:73ba...5	(disk cache)	9 ms	
Image	204	text/plain	controltag.js:73ba...5	338 B	22 ms	

How do you solve if there is any conflict with your peer QA on any technical aspect?

- There should be argument only up to certain extent with your Peer on why you are correct
- If it is still conflict,
- Involve team and discuss the conflict issue with larger Audience. Open to take suggestions from your Team.
- Accept any decision made from Team meeting with Smile.

How do we decide if test case Is not ideal candidate for Automation?

- Tests which are highly dependent on Data with frequent changes every time
 - Tests which cannot be automated due to technical challenges
- Tests which are unstable due to complex nature of Application.

Difference between retesting and regression testing?

- Retesting is done to make sure that the tests cases which failed in last execution are passed after the defects are fixed.
- Regression testing is to ensure that changes have not affected unchanged part.
- Regression testing is not carried out for specific defect fixes

What's the most interesting bug that you've found recently?

In Agile Environments, in case you have any doubts regarding your project, how do you approach?

- As a tester, For Domain related queries, You should reach out to Business Analyst, Product Owner

As a tester, how do you differentiate Dummy login page which is exactly designed like your Working project login page

- Application Logs

How Will You Tell If Enough Test Cases Have Been Created To Test A Product?

- Each Business requirement is mapped to at least one test Scenario
- Each Test Scenario should have positive and negative test cases
- Using traceability Matrix to check whether these testcases are covering all the requirement specification

Core Java Coding Programs for Interview Preparation: RahulShettyAcademy.com

Create a Method which accepts Array and returns sum of all the elements in array

Print 5 Multiplication table without using Multiply operator

Max difference between any adjacent index in array

compare same indexes of 2 different arrays and create another array for matching values

Sort the Array in Ascending and Descending order

Swap the variables with and with out using Temp variable

Program for Printing Pyramid Pattern in Java

Print Fibanocci Series

Check if given number is Prime

```
| *SumofElements.java ✎
1
2 public class SumofElements {
3
4     //Create a Method which accepts Array and returns sum of all the elements in array
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7
8         int[] a = { 1,2,3,4,5};
9
10        int sum= sumArray(a);
11        System.out.println(sum);
12
13    }
14
15    public static int sumArray(int[] a)
16    {
17
18        int sum =0; [
19        //extract every value of array and sum each value with other
20        for(int i=0;i<a.length;i++)
21        {
22            System.out.println(a[i]);
23            sum = sum + a[i]; //10
24
25        }
26
27        return sum;
28    }
29
30
31    //Print 5 Multiplication table without using Multiply operator
32    public static void main(String[] args) {
33        // TODO Auto-generated method stub
34
35        int[] a = { 1,2,3,4,5};
36
37        int sum= sumArray(a);
38        System.out.println(sum);
39
40        int result= multiply(5,10); //50
41        System.out.println(result);
42
43    }
44
```

```
private static int multiply(int i, int j) {
    // TODO Auto-generated method stub
    //i has to sum itself j times to get result
    int k=1;
    int sum=0;
    while(k<=j)
    {
        sum = sum + i;
        k++;
    }
    return 0;
}
```

SumofElements.java *AdjacentArray.java

```
1
2 public class AdjacentArray {
3
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6         //Max difference between any adjacent index in array
7         int[] a = {1,4,8,15,17}; //3 4,7,2
8         int diff =0;
9         for(int i=0;i<a.length-1;i++)
10         {
11             if(a[i+1] - a[i]>diff)
12             {
13                 diff=a[i+1] - a[i];
14             }
15         }
16
17         System.out.println(diff);
18 }
```

```

//compare same indexes of 2 different arrays and create another array for matching values
public static void main(String[] args) {
    // TODO Auto-generated method stub
    int a[] = { 1,4,5,7};
    int b[] = {6,4,3,7};      // {4,7}
    int[] ab = new int[4];
    ArrayList<Integer> al = new ArrayList<Integer>();

    for(int i=0;i<a.length;i++)
    {
        if(a[i] == b[i])
        {
            //code to create another array
            al.add(a[i]);
        }
    }
    Object[] c=al.toArray();
    System.out.println(c);

    int a[] = { 1,4,5,7};
    int b[] = {6,4,3,7};      // {4,7}
    int[] ab = new int[4];
    ArrayList<Integer> al = new ArrayList<Integer>();

    for(int i=0;i<a.length;i++)
    {
        if(a[i] == b[i])
        {
            //code to create another array
            al.add(a[i]);
        }
    }
    Object[] c=al.toArray();

    for(Object obj : c)
    {
        System.out.println(obj);
    }
}

```

Sorting of arrays either in ascending or descending order

```

public class sortarraydemo {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int a[] = {2,6,1,4,9};
        int temp;

```

```
// 1,2,3,4,6, 9

for(int i=0;i<5;i++)
{
    for(int j=i+1;j<5;j++)
    {
        if(a[i] > a[j])
        {
            temp=a[i];
            a[i]=a[j];
            a[j]=temp;
        }
    }
}
```

Swapping two numbers either by using temp or without temp

The screenshot shows a Java code editor with a file named "sortarraydemo.java". The code is as follows:

```
1 package demopack;
2
3 public class sortarraydemo {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         // int a[] = {2,6,1,4,9};
8
9         int a = 4;
10        int b = 5;
11        int temp;
12        temp=a;
13        a=b;
14        b=temp;
15        System.out.println(a);
16        System.out.println(b);
17    }
18}
```

The code demonstrates swapping the values of variables `a` and `b` using a temporary variable `temp`. The output of the program will be the swapped values.

```
int a= 5;
int b =4;
/*int temp;
temp=a;
a=b;
b=temp;
System.out.println(a);
System.out.println(b);*/
// Swap with out variable

a=a+b; // a= 9
b=a-b; //b = 5
a=a-b; // a= 4
```

The screenshot shows a Java code editor with five tabs at the top: SumofElements.java, AdjacentArray.java, CompareArray.java, *TestPyramid.java (which is the active tab), and another unnamed tab. The code in the editor is for printing a pyramid pattern:

```
1
2 public class TestPyramid {
3
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6     // *
7     // * *
8     // * * *
9     // * * * *
10    // Program for Printing Pyramid Pattern in Java
11
12    System.out.println(" *");
13
14    for(int i=1 ; i<5;i++)
15    {
16        // take care the logic of printing right format
17        for(int j=1;j<=i;j++)
18        {
19            System.out.print(" *");
20            System.out.println("\t");
21        }
22        System.out.println();
23    }
24
25}
```

A blue rectangular highlight covers the code from line 17 to line 21, specifically the inner loop body where asterisks are printed and a tab character is added after each one.

Fibonacci Series

```
Fibonacci.java ✘
```

```
4④    public static void main(String[] args) {  
5        // TODO Auto-generated method stub  
6    //0, 1,1, 2 3 5 [8 13]21 34  
7        //0 -a  
8        // 1 -b /a  
9        //1 - sum/b  
10       //2  
11  
12       int a =0;  
13       int b= 1;  
14       int sum=0;  
15       int i =1;  
16       while (i<9)  
17       {  
18           sum=a+b;  
19           System.out.println(sum);  
20           a=b;  
21           b=sum;  
22  
23
```

To Check a number is prime or not

A screenshot of a Java IDE showing the PrimeNumber.java file. The code checks if a given number is prime or not. It uses a for loop to iterate from 2 to val/2, checking if val is divisible by i. If it is, it sets flag to true and breaks out of the loop. Finally, it prints whether the number is prime or not based on the value of flag.

```
1
2 public class PrimeNumber {
3
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6
7         // 11%1= 0
8         //31
9         boolean flag = false;
10        int val =31;
11
12        for(int i=2;i<=val/2;i++)
13        {
14
15            if(val%i==0)
16            {
17                flag=true;
18                break;
19            }
20        }
21        if(flag)
22            System.out.println("it is not prime");
23        else
24            System.out.println("it is prime");
25
26
```

Print Min number and Max number in Multi Dimensional
Array 3* 3 Matrix

A screenshot of a Notepad window titled "Untitled - Notepad". It contains a 3x3 matrix of integers:

2	4	5
3	4	7
1	2	9

Below the matrix, the text "Print minimum number from the above matrix" is visible.

```
public static void main(String[] args) {  
    // TODO Auto-generated method stub  
  
    int abc[][]={{2,4,5},{3,4,7},{1,2,9}};  
    int min=abc[0][0];  
  
    for(int i=0;i<3;i++)  
    {  
        for(int j=0;j<3;j++)  
        {  
            if(abc[i][j]<min)//2  
            {  
                min=abc[i][j];  
            }  
        }  
    }  
    System.out.println(min);  
}
```

Reverse the String with put using Predefined Reverse methods or Palindrome Verification

print Unique numbers in the Array

Count no of occurrence of characters in the String

Reverse the Number

```
collectiondemo.java ① *revresedemo.java ②
```

```
1 package demopack;
2
3 public class revresedemo {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         //
8
9         String s = "madam";
10        String t= "";
11        for(int i=s.length()-1; i>=0;i--)
12        {
13            t= t+ s.charAt(i);
14        }
15
16        System.out.println(t);
17    // t= madam
18}
```

Eliminate duplicate and print unique number in the array

```
public class collectiondemo {

②     public static void main(String[] args) {
         // TODO Auto-generated method stub

         int a[] ={ 4,5,5,5,4,6,6,9,4};
         // Print unique number from the list- Amazon

         ArrayList<Integer>ab =new ArrayList<Integer>();
         for(int i=0;i<a.length;i++)
         {
             int k=0;

             if(ab.contains(a[i]))
             {
                 ab.add(a[i]);
                 k++;

                 for(int j=i+1;j<a.length;j++)

```

```

        for(int j=i+1;j<a.length;j++)
        {
            if(a[i]==a[j])
            {
                k++;
            }
        }
    }

System.out.println(a[i]);
System.out.println(k);
}

```

Count number of occurrences of char in a string

The screenshot shows an IDE interface with three tabs at the top: 'Fibonacci.java', 'PrimeNumber.java', and 'CountStringOccurrence.java'. The 'CountStringOccurrence.java' tab is active. The code is as follows:

```

1
2 public class CountStringOccurrence {
3
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6
7
8
9         int count= countOccurrences("aaanjda",'a');
10
11    }
12
13    private static int countOccurrences(String word, char character) {
14        // TODO Auto-generated method stub
15        int count=0;
16        for(int i=0;i<word.length();i++)
17        {
18            if(word.charAt(i)==character)
19            {
20                count++;
21            }
22        }
23

```

Reverse any given number with the help of arithmetic operation

```
public class ReverseNum {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
  
        int a =543;  
        int reverse =0;  
        while(a!=0)  
        {  
            int digit= a%10;  
  
            reverse = digit + reverse*10;  
            a=a/10;  
  
        }  
        System.out.println(reverse);  
    }  
<
```

Core Java Miscellaneous Concepts

Brush Up Interview Point of View

What are Instance Variables?

what are Local Variables?

What are Class variables?

Use of Static keyword in Java

Importance of calling Static method and Static variables

How Static blocks are defined?

Why Strings are Immutable in Java?

How memory is allocated to String literal and String class object?

When to use == and equals() methods to compare Strings

Importance of String Builder and String Buffer classes and their Differences

Program in making the string mutable with String Buffer

What are Interfaces?

What is Runtime Polymorphism?

What are Abstract Classes?

What are differences between Abstract Classes and Interfaces?

Why Strings are Immutable in Java?

How memory is allocated to String literal and String class object?

I

When to use == and equals() methods to compare Strings

Importance of String Builder and String Buffer classes and their Differences

Program in making the string mutable with String Buffer

```
*StringDemo.java
```

```
10     String c=a.concat("World");
11     System.out.println(a);
12     String s=new String("hello");
13     String s1=new String("hello");
14
15     //StringBuffer and StringBuilder -Mutable
16     StringBuffer sb =new StringBuffer("hello");
17     sb.append("world");
18     System.out.println(sb);
19     sb.insert(2, "She");//heHelloworld
20     System.out.println(sb);
21     sb.replace(5, 7, "aa");//heSheaaoworld
22     System.out.println(sb);
23     sb.deleteCharAt(12);//heSheaaoworl
24     System.out.println(sb);
25     sb.reverse();//
26     System.out.println(sb);
27     //StringBuiler is not thread safe. It is Non SYNcrhonized, It is faster
28
29
30
```

```
Problems @ Javadoc Declaration Console Results of running class demo
<terminated> StringDemo [Java Application] C:\Program Files\Java\jdk1.8.0_211\bin\javaw.exe (Apr 30, 2020, 10:18:14 AM)
helloworld
heHelloworld
heSheaaoworld
heSheaaoworl
1rowoaahSeh
```

```
String a ="hello";//String literal -Mutable
String b ="hello";
String c=a.concat("World");
System.out.println(a);
String s=new String("hello");
String s1=new String("hello"); //String class creates new object every time in memory

System.out.println(a.equals(b));//true
System.out.println(a==b);//true
System.out.println(a.equalsIgnoreCase(s)); //equals operator checks for content true
System.out.println(a==s);//fail matching the references ==
System.out.println(s==s1); //fail references are different as they are defined with string class
```

Interface

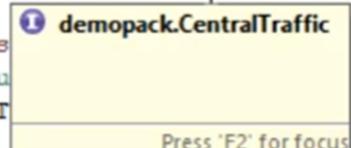
```
1 package demopack;
2
3 public interface CentralTraffic {
4
5     public void greenGo();
6     public void redStop();
7     public void FlashYellow();
8
9
10
11 }
12
```

```
1 package coreJava;
2
3 import demopack.CentralTraffic;
4
5 public class AustralianTrafic implements CentralTraffic {
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9
10    }
11
12    @Override
13    public void greenGo() {
14        // TODO Auto-generated method stub
15        //code
16    }
17
18    @Override
19    public void redStop() {
```

You need to implement all the methods

```
*CentralTraffic.java *AustralianTrafic.java X
6
7  public static void main(String[] args) {
8      // TODO Auto-generated method stub
9      CentralTraffic a= new AustralianTrafic();
10     a.redStop();
11     a.FlashYellow();
12     a.greenGo();
13
14
15
16 }
17
18
19 @Override
20     public void redStop() {
21         // TODO Auto-generated method stub
22         System.out.println(" redstop implementation");
23     }
24 public void walkonsymbol()
<
```

```
*CentralTraffic.java *AustralianTrafic.java X
1 package coreJava;
2
3 import demopack.CentralTraffic;
4
5 public class AustralianTrafic implements CentralTraffic {
6
7     public static void main(String[] args
8         // TODO Auto-generated method stu
9         CentralTraffic a= new AustralianT
0         a.redStop();
1         a.FlashYellow();
2         a.greenGo();
3
4         AustralianTrafic at=new AustralianTrafic();
5         at.walkonsymbol();|
```



In order to call method of this call we need to create object of this class type

One class may implement more than one interface.

```
1 package coreJava;
2
3 public interface ContinentalTraffic {
4
5     public void Trainsymbol();
6 }
7
```

```
1 package coreJava;
2
3 import demopack.CentralTraffic;
4
5 public class AustralianTrafic implements CentralTraffic,ContinentalTraffic
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9         CentralTraffic a= new AustralianTrafic();
10        a.redStop();
11        a.FlashYellow();
12        a.greenGo();
13
14        AustralianTrafic at=new AustralianTrafic();
15        ContinentalTraffic ct=new AustralianTrafic();
16        at.walkonsymbol();
17        ct.Trainsymbol();
18
19    }
20
```

What is Abstraction?

abstraction is a process of hiding the implementation details from the user, only the functionality will be provided to the user. In other words, the user will have the information on what the object does instead of how it does it.

Abstract Classes & Methods:

Class which is declared with the abstract keyword is known as an abstract class in Java

It can have abstract (method with out the body).and non-abstract methods (method with the body).

Abstraction Classes achieves Partial Abstraction

Interfaces on the other hand are used for 100% abstraction

Real Life Example



There is a parent class which defines Aircraft Body Construction rules

All Child classes (Different Aircraft companies) have to inherit parent class and use those methods to build their own Aircraft

But there is a method called Aircraft color which can be unique to every child class depending on their company logo

So, color method in parent class is not implemented with anybody. Instead only method is defined with out any body which we call as Abstract method.

If there is any method in class which is Abstract, then the class will be treated as Abstract class

Unlike Interfaces, this Abstract Class can have methods (Concrete) which have implementation as well. So, with Abstract class only partial Abstraction is achieved.

```
1
2 public abstract class ParentAirCraft {
3
4     public void engine()
5     {
6         System.out.println("Follow Engine Guidelines");
7     }
8
9     public void safetyGuidelines()
10    {
11        System.out.println("Follow Safety Guidelines");
12    }
13
14     public abstract void bodyColor();
15
16
17 }
```

```
1  CoreJava/src/ParentAirCraft.java
2  public class CHILDEMIRITES extends ParentAirCraft{
3
4  public static void main(String[] args) {
5      // TODO Auto-generated method stub
6
7      ChildEmirites c= new ChildEmirites();
8      c.engine();
9      c.safetyGuidelines();
10     c.bodyColor();|
11 }
12
13 @Override
14 public void bodyColor() {
15     // TODO Auto-generated method stub
16     System.out.println("Red color on the body");
17 }
18
19 }
20
```

Access modifier can be anything instead of Private as child class has to override it.

Differences between Abstract Classes & Interfaces

Interface:

- 1) Interface contains only abstract methods
- 2) Access Specifiers for methods in interface must be public
- 3) Variables defined must be public, static, final
- 4) To implement an interface we use implements keyword

Abstract Class:

- 1) Abstract class can contain abstract methods, concrete methods or both
 - 2) Except private we can have any access specifier for methods in abstract class.
 - 3) Except private variables can have any access specifiers
-

Automation Tester

automationtester@gmail.com

[linkedin.com/automationtester](https://www.linkedin.com/in/automationtester)

EXPERIENCE SUMMARY and KEY highlights

- 5 years of professional experience in software testing which includes **Functional Testing, Automation testing and Performance testing**
 - Hands on Experience of **preparation and execution of test cases using Selenium (Web Automation), Appium (Mobile) and API Testing.**
 - Excellent technical skills, attention to detail, strong issue identification and problem-solving skills.
 - Exposure to writing, reviewing and executing test scenarios and test cases
 - Exposure to root cause analysis of defects and failed test scripts
 - Good Analytical, Communication, interpersonal skills and an excellent team player.
 - Expertise in automation using Selenium WebDriver, with Java on TestNG and Junit library
 - Designed and implemented different automation frameworks from scratch like Page Good Exposure to work on Agile-Scrum and Waterfall Models.
 - Developed Page object Model Design Selenium framework on TestNG Platform
 - Focus on efficient Test Delivery with leveraging Test efforts to bring **Cost savings in Testing Projects.**
 - Good exposure to each of the phases of **Software Testing Life Cycle (STLC).**
-
- Hands-on experience in **Test strategy, Test Estimation, System Testing, System, Integration Testing, Mainframe Testing, HP QC 9.2, HP ALM 11.0, Selenium tool, SOAP UI.**
 - A skilled **Tester** with a flair for using new testing methodology such as working Release Based model for driving faster execution.
 - A key Team Player with thorough understanding of all aspects of the STLC from understanding client requirements through direct client interaction, translating them into Test Deliverable and driving test execution to closure.

EDUCATIONAL QUALIFICATIONS

TECHNICAL SKILLS and SOFTWARE PROFICIENCY

Operating Systems	Windows OS(XP), Windows 7, Unix,
Test Management Tools	ALM 11.0, JIRA
Software	MS Office
Databases	SQL, Access
Languages	C/C++, Java, VB.NET, JavaScript
Automation Tools	Selenium, SOAP UI, RPA, Cypress

PROJECT EXPERIENCE

Cognizant Technology Solutions

11'

Sep till date

Sep till date

1. Project: Payment processing Gate Way Implementation

Role: Senior Test Engineer

Description: This project aims at making smooth transaction through Visa/Master/a.

Responsibilities:

- Understand the business requirements from EAD.
- Designing RTM and mapping the requirement to related test cases.
- Executing the test cases by sending the request using SOAP UI and verifying the response
- Identifying Automation Scenarios from the set of Testcases/User stories identified in cycle
- Developing Web Automation Testcases for the identified Automation Scenarios
- Performed Cross Browser Automation using Selenium GRID with Sauce Labs Cloud Integration
- Reviewed the Code prepared by other team mates and suggested improvements if required.
- Development of test cases, test data creation, test environment, and execution of test cases.
- Execution of test cases using Selenium Tool and proper logging, tracking, retesting and closure of defects in ALM.
- Defect Management via HP ALM.
- Review of Business Requirements, Technical Requirements, and Interface and Design documents.

Sep till date

1. Project: Payment processing Gate Way Implementation

Role: Senior Test Engineer

Description: This project aims at making smooth transaction through Visa/Master/a.

Responsibilities:

- Understand the business requirements from EAD.
- Designing RTM and mapping the requirement to related test cases.
- Executing the test cases by sending the request using SOAP UI and verifying the response
- Identifying Automation Scenarios from the set of Testcases/User stories identified in cycle
- Developing Web Automation Testcases for the identified Automation Scenarios
- Performed Cross Browser Automation using Selenium GRID with Sauce Labs Cloud Integration
- Reviewed the Code prepared by other team mates and suggested improvements if required.
- Development of test cases, test data creation, test environment, and execution of test cases.
- Execution of test cases using Selenium Tool and proper logging, tracking, retesting and closure of defects in ALM.
- Defect Management via HP ALM.
- Review of Business Requirements, Technical Requirements, and Interface and Design documents.

Bonus Section - Discount Coupon to all my courses

Rahul Shetty Courses Discount Coupons

For Every course below, I have shared two Discount coupon links. You can enroll from the link which ever gives the lowest price for you out of two

Robotic Process Automation(RPA) using UIPATH + Live Projects

<https://www.udemy.com/course/rpa-uipath/?couponCode=RAHULSHETTYACADEMY6>

<https://www.udemy.com/course/rpa-uipath/?referralCode=18539ACE13CA10B3F913>

Selenium Python Automation Testing from Scratch + Frameworks

<https://www.udemy.com/course/learn-software-testing-in-practical-become-a-qa-expert/?couponCode=RAHULSHETTYACADEMY6>

<https://www.udemy.com/course/learn-selenium-automation-in-easy-python-language/?referralCode=907494E3C21AF19DF472>

Rest API Automation with Java + Frameworks– Rest Assured

<https://www.udemy.com/course/rest-api-automation-testing-rest-assured/?couponCode=RAHULSHETTYACADEMY6>

<https://www.udemy.com/course/rest-api-automation-testing-rest-assured/?referralCode=5DC20F3F9AEC680CF129>

Appium – Mobile Automation from Scratch + Frameworks

<https://www.udemy.com/course/mobile-automation-using-appiumselenium-3/?couponCode=RAHULSHETTYACADEMY6>

<https://www.udemy.com/course/mobile-automation-using-appiumselenium-3/?referralCode=C46BF551F5B9EAF08E10>

Learn Protractor(Angular Testing) from scratch +Framework

<https://www.udemy.com/course/protractor-tutorial/?referralCode=4BD09EB80BA2B9E8FFEA>

<https://www.udemy.com/course/protractor-tutorial/?couponCode=RAHULSHETTYACADEMY6>

Cypress-Modern Automation Testing from Scratch + Framework

<https://www.udemy.com/course/cypress-tutorial/?couponCode=RAHULSHETTYACADEMY6>

<https://www.udemy.com/course/cypress-tutorial/?referralCode=32996BF536115C960678>

Master Software Testing+Jira+Agile on Live App-Be a TeamLead

<https://www.udemy.com/course/learn-software-testing-in-practical-become-a-qa-expert/?couponCode=RAHULSHETTYACADEMY6>

<https://www.udemy.com/course/learn-software-testing-in-practical-become-a-qa-expert/?referralCode=B2901574D20A84F4F94F>

Learn JMETER from Scratch on Live Apps -Performance Testing

<https://www.udemy.com/course/learn-software-testing-in-practical-become-a-qa-expert/?couponCode=RAHULSHETTYACADEMY6>

<https://www.udemy.com/course/learn-jmeter-from-scratch-performance-load-testing-tool/?referralCode=E29109A7BEC228EED509>

WebServices/Rest API Testing with SoapUI +Real time Projects

<https://www.udemy.com/course/webservices-testing-with-soapui/?couponCode=RAHULSHETTYACADEMY6>

<https://www.udemy.com/course/webservices-testing-with-soapui/?referralCode=A6461A7438563696BB77>

SDET/Test Architect Essentials -Road to Full stack QA

<https://www.udemy.com/course/sdettraining-testarchitect-fullstackqa/?referralCode=E76A8BC1A22903C614BE>

<https://www.udemy.com/course/sdettraining-testarchitect-fullstackqa/?couponCode=RAHULSHETTYACADEMY6>

Cucumber with Java-Build Automation Framework in lesser code

<https://www.udemy.com/course/cucumber-tutorial/?couponCode=RAHULSHETTYACADEMY6>

<https://www.udemy.com/course/cucumber-tutorial/?referralCode=452C30D3B2EF720F0072>

Selenium WebDriver with Java -Basics to Advanced+Frameworks

<https://www.udemy.com/course/selenium-real-time-examplesinterview-questions/?couponCode=RAHULSHETTYACADEMY6>

<https://www.udemy.com/course/selenium-real-time-examplesinterview-questions/?referralCode=0D6919D2CF070539F2DA>

Learn SQL +Security(pen) testing from Scratch

<https://www.udemy.com/course/database-testing-tutorial/?couponCode=RAHULSHETTYACADEMY6>

<https://www.udemy.com/course/database-testing-tutorial/?referralCode=23FB742C2BD00EC7A977>

Core java for Automation Testers + Interview Programs

<https://www.udemy.com/course/core-java-tutorials/?referralCode=DC96B2014338907EA1C2>

<https://www.udemy.com/course/core-java-tutorials/?couponCode=RAHULSHETTYACADEMY6>