Infosys (38 min)

1. Tell me about different locators

2.Tell me about Xpath axes

3. Synchronization methods

4. Where you had used Encapsulation

5.WHat is difference between Stringbuffer and StringBuilder which is synchrinized

Why we use synchronization

6.Write code for Waits

7. Write code for taking screenshots

8.Write code for Scrollingwindows

9.Write code for handling alerts

10. Write code for Handling windows

11. Tell me about API methods

12. Tell me about API codes

13 Tell me about Feature file

14. Tell me about TestNG

**1.Tell me about yourself**

Hello everyone I am Shubhangi Sirsat and I am from Pune. Currently I am working as QA in Expleo solutions and I am having total 4 years of experience in which i performed AUtomation as well as manual Testing. Whenever I talk about manual testing my roles and responsibilities are like whenever I receive user stories I have to analyse that and write test cases for same and after that we need to review test cases by 4 ways ie Self review, peer review,internal review, and external review.After review we have to create traceability matrix in which we have to map user stories against the test cases and once the test review part is done and I receive my test cases I start for execution and before starting with execution we need to perform smoke testing on the latest built. then I have to do functional testing non functional testing,Retesting. Basically in my organization once done with retesting for regression testing we use automation testing. For automation we use Selenium with java. Also I worked on cucumber with BDD framework Maven, cucumber TestNG and POM concepts as well. Apart from this I worked on Agile methodologies and we need to attend different ceremonies like Sprint planning meeting daily standup call,review meetings and retrospective meetings.

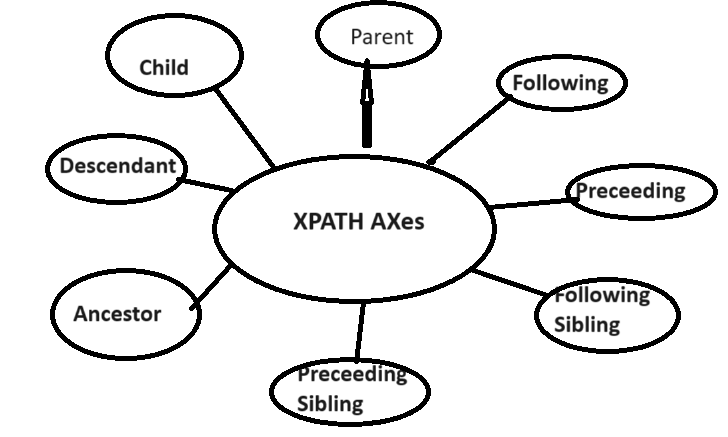
**2. Tell me about different locators**

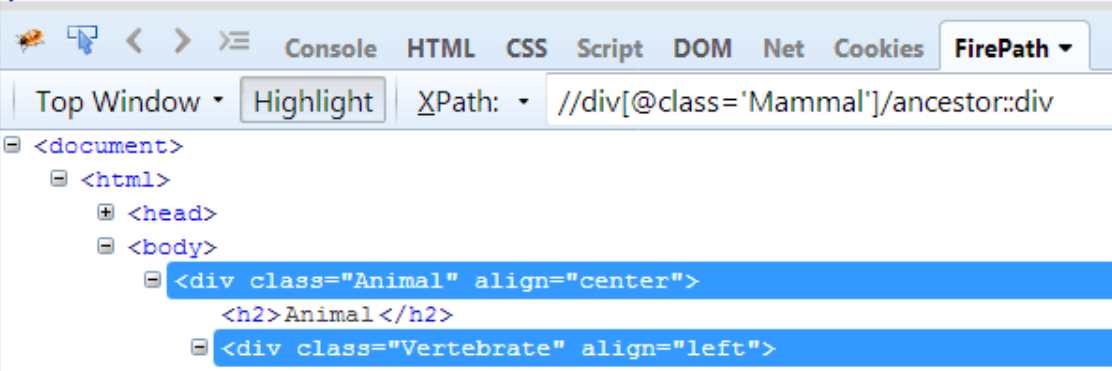
----> Basically locators are used to fined the location of webelements in dom(document obj model) structure. There are different locators like

ID, Name, ClassName, Link text, partiallinkext, TagName, CSS Selector

**2.Tell me about Xpath axes**

------> It is method to search multiple nodes in current DOM from the current node context. Mainly used for finding dynamic web elements like which changes dynamically when we refresh pages or do any other options and which are not possible to find with help of ID,name, class name, link text or css selector.





**3. Synchronization methods**

---->Thread.sleep()

Implicit wait() - driver. manage(). timeouts().implicitlyWait(Duration.ofSeconds(10));

Explicit Wait() -

WebDriverWait wait= new WebDriverWait(driver,20);

WebElemet e= wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath(" xpath")));

Other methods related to conditions :

alertIsPresent()

elementToBeClickable()

elementToBeSelected()

visibilityOfElementLocated()

frameToBeAvaliableAndSwitchToIt()

invisibilityOfTheElementLocated()

invisibilityOfElementWithText()

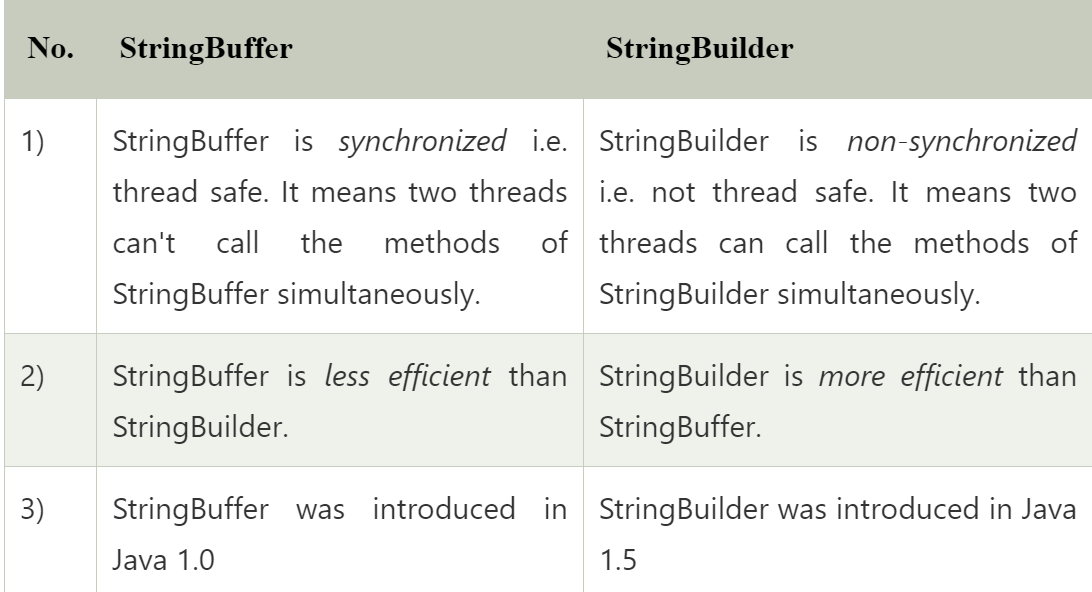
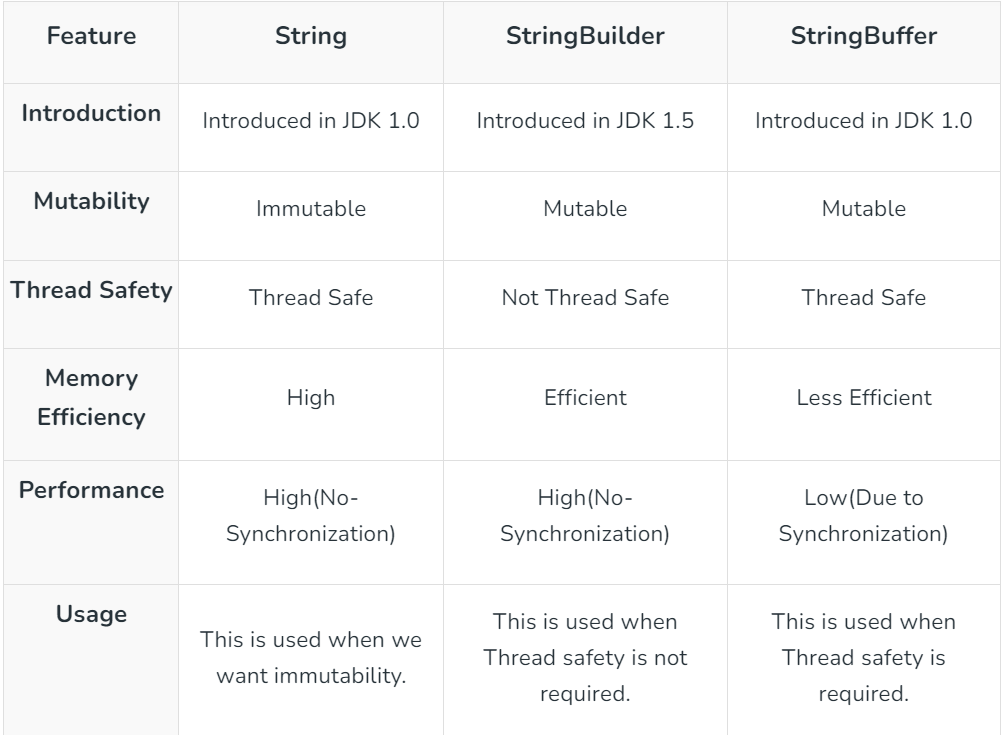
**4. Where you had used Encapsulation**

--->

***Encapsulation*** in Java is a mechanism of wrapping the data members and member functions together as a single unit. In encapsulation the variables of a class will be hidden from other classes, and can be accessed only through the methods of their current class, therefore it is also known as data hiding.

I have used encapsulation in POM pages where we declare our data members for webelement as private and all other getter and setter methods as public that can be accessed by outside test methods.

**5.WHat is difference between Stringbuffer and StringBuilder which is synchroinized**



**Why we use synchronization**

For thread safe means two threads can access method of String builder class simultaniously.

**6.Write code for Waits**

Inplicit wait: driver.manage().timeOuts().implicitlyWait(Duration.OfSeconds(5));

Explicit wait :

WebDriverWait wait=new WebDriverWait(driver,20);

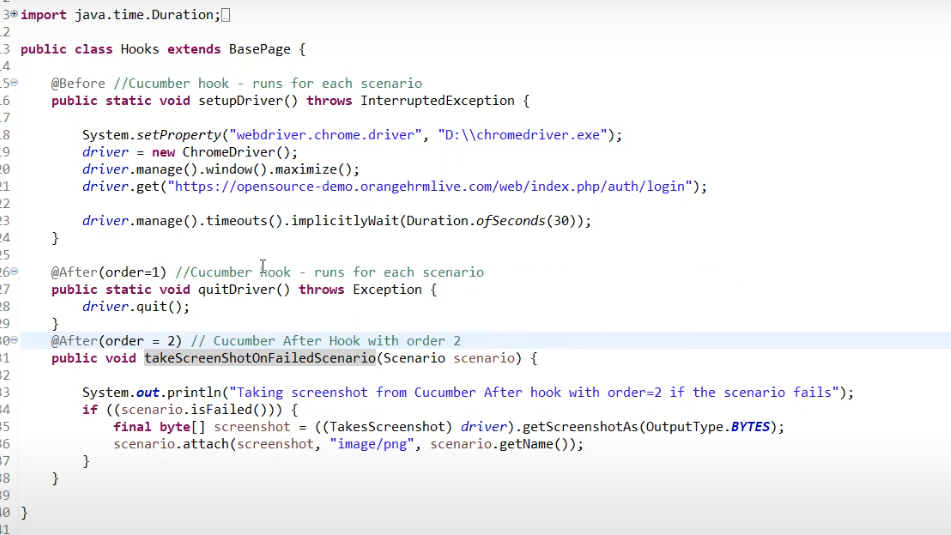
WebElement w= wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath(“xpath”)));

**7. Write code for taking screenshots**

File src= ((TakeScreenshot)driver).getScreenshotAs(OutputType.FILE);

File dest=new File(“Path”);

FileHandler.copy(src,dest);



**8.Write code for Scrollingwindows**

JavaScriptExecutor js= JavaScriptExecutor(driver);

Js.executeScript(scroll(100,200));

**9.Write code for handling alerts**

**Driver.switchTo().alert().accept();**

**Driver.switchTo().alert().dismiss();**

**Driver.switchTo().alert().sendKeys(“Text”);**

**Driver.switchTo().alert().getText();**

**10. Write code for Handling windows**

//Printing window handles

**String parentHandle=driver.getWindowHandle();**

**Set<String> handle= driver.getWindowHandles();**

**For(String handles:handle)**

**{**

**//System.out.println(handles);**

**If(!handles.equals(parentHandle))**

**{**

**Driver.switchTo().window(handle);**

**Driver.findElement(By.Id(“ text”)).sendKeys(“Value”);**

**}**

**}**

// how to find how many number of widows open

Set<String> allWindowHandles = driver.getWindowHandles();  
ArrayList<String> tabs = new ArrayList<String>(allWindowHandles);  
System.out.println("No. of tabs: " + tabs.size());

//

**11. Tell me about API methods**

### **GET**

The GET method is used to retrieve data on a server. Clients can use the GET method to access all of the resources of a given type, or they can use it to access a specific resource. For instance, a GET request to the /products endpoint of an e-commerce API would return all of the products in the database, while a GET request to the /products/123 endpoint would return the specific product with an ID of 123. GET requests typically do not include a request body, as the client is not attempting to create or update data.

### **POST**

The POST method is used to create new resources. For instance, if the manager of an e-commerce store wanted to add a new product to the database, they would send a POST request to the /products endpoint. Unlike GET requests, POST requests typically include a request body, which is where the client specifies the attributes of the resource to be created. For example, a POST request to the /products endpoint might have a request body that looks like this:

{  
 "name": "Sneakers",  
 "color": "blue",  
 "price": 59.95,  
 "currency": "USD"  
}

### **PUT**

The PUT method is used to replace an existing resource with an updated version. This method works by replacing the entire resource (i.e., the specific product located at the /products/123 endpoint) with the data that is included in the request’s body. This means that any fields or properties not included in the request body are deleted, and any new fields or properties are added.

### **PATCH**

The PATCH method is used to update an existing resource. It is similar to PUT, except that PATCH enables clients to update specific properties on a resource—without overwriting the others. For instance, if you have a product resource with fields for name, brand, and price, but you only want to update the price, you could use the PATCH method to send a request that only includes the new value for the price field. The rest of the resource would remain unchanged. This behavior makes the PATCH method more flexible and efficient than PUT.

### **DELETE**

The DELETE method is used to remove data from a database. When a client sends a DELETE request, it is requesting that the resource at the specified URL be removed. For example, a DELETE request to the /products/123 endpoint will permanently remove the product with an ID of 123 from the database. Some APIs may leverage authorization mechanisms to ensure that only clients with the appropriate permissions are able to delete resources.

**12. Tell me about API codes**

### **1xx informational responses**

These codes indicate that the server has received the request and is processing it. They are primarily used to manage communication between the client and server during the early stages of a request-response cycle. Some examples of this type of code are:

**100 Continue**: This status code indicates that the initial part of the request has been received and the server would like the client to send the rest of it.

**101 Switching Protocols**: This status code is used to inform the client that the server is changing the protocol that is being used in the connection.

**102 Processing**: This status code is an interim response that indicates the server is still processing the request.

### **2xx success responses**

These codes indicate that the client’s request was successfully received, understood, and processed by the server. Some of the most common 200 responses are:

**200 OK:** This status code indicates that the request was successful, and the server returned the requested data.

**201 Created:** This status code means that the request was successful, and the server created a new resource.

**204 No Content:** This status code indicates that the request was successful, but the server did not return any data.

### **3xx redirection responses**

These codes indicate that the client needs to take additional actions to fulfill the request. They are often used when the requested resource has moved to a different location. Some examples include:

**301 Moved Permanently:** This status code indicates that the requested resource has been permanently moved to a new URL. Clients should respond by updating their bookmarks and links to point to the new URL, and search engines should update their indexes with the new location.

**303 See Other:** This status code indicates that the response is available at a different URL, and the client should perform a GET request to that URL to retrieve the resource.

### **4xx client error responses**

These codes indicate that there was an issue with the client’s request, such as a mistyped URL or invalid credentials. The most common 4xx responses include:

**400 Bad Request:** This status code indicates that the request was malformed or invalid.

**401 Unauthorized:** This status code lets the client know that it is not authorized to access the requested resource.

**403 Forbidden:** This status code communicates that the client is authenticated but not authorized to access the requested resource.

**404 Not Found:** This status code indicates that the requested resource was not found on the server.

### **5xx server error responses**

These codes, which indicate that the server encountered an error while trying to fulfill the client’s request, include:

**500 Internal Server Error:** This generic error code indicates the server encountered an unexpected condition that prevented it from fulfilling the request.

**502 Bad Gateway:** This status code indicates that a server acting as a gateway or proxy received an invalid response from an upstream server.

**503 Service Unavailable:** This status code is returned when the server is temporarily unable to handle the request. It’s often seen during periods of increased traffic or when the server is undergoing maintenance.

**13 Tell me about Feature file**

A ***Feature File*** is an entry point to the Cucumber tests. Which is created with .feature extention. This is a file where you will describe your tests in Descriptive language (*Like English*). Where we use Gherking language.A feature file can contain a scenario or can contain many scenarios in a single feature file . Here we can write test steps in Given, When,Then, And format.and every step is this file is mapped with Steps in step definition files where we write logic for responsible step.For comman features we can use Background keyword

**14. Tell me about TestNG**

**-->**

*TestNG organizes tests into suites and classes. It utilizes annotations to define the execution order and dependencies between tests within a suite.*

*Some of TestNG’s notable features include:*

* ***Annotations:*** *@BeforeTest, @AfterTest, and others provide control over test execution flow.*
* ***Data-Driven Testing:*** *Allows running the same test with different data sets for comprehensive testing.*
* ***Grouping:*** *Organize related tests together for better management and execution.*
* ***Parallel Testing:*** *Enables running tests concurrently to reduce overall test execution time.*
* ***Reporting:*** *Generates detailed reports on test execution results.*
* [@BeforeSuite](https://www.geeksforgeeks.org/testng-annotations-beforesuite/)
* [@AfterSuite](https://www.geeksforgeeks.org/testng-annotations-aftersuite/amp/)
* [@BeforeTest](https://www.geeksforgeeks.org/testng-annotations-beforetest/)
* [@AfterTest](https://www.geeksforgeeks.org/testng-aftertest-annotation/)
* [@BeforeClass](https://www.geeksforgeeks.org/testng-annotations-beforeclass/)
* [@AfterClass](https://www.geeksforgeeks.org/testng-afterclass-annotations/)
* [@BeforeMethod](https://www.geeksforgeeks.org/testng-beforemethod-annotations/)
* [@AfterMethod](https://www.geeksforgeeks.org/testng-aftermethod-annotations/)
* [@BeforeGroups](https://www.geeksforgeeks.org/testng-beforegroups-annotations/)
* [@AfterGroups](https://www.geeksforgeeks.org/testng-annotations-aftergroups/)

**15. Tell me about your automation experience.**

**16. How you will execute smoke tests in Cucumber**

*Cucumber has already provided a way to organize your scenario execution by using* ***tags*** *in feature file. We can define each scenario with a useful tag. Later, in the runner file, we can decide which specific tag (and so as the scenario(s)) we want Cucumber to execute. Tag starts with “***@***”. After “***@***” you can have any relevant text to define your tag like* ***@SmokeTests*** *just above the scenarios you like to mark. Then to target these tagged scenarios just specify the tags names in the* ***CucumberOptions*** *as* ***tags = {"@SmokeTests"}****.*

*Other than Interview :*

### Is there a way to type in a textbox without using sendKeys()?

Yes! Text can be entered into a textbox using JavaScriptExecutor

JavascriptExecutor jse = (JavascriptExecutor) driver;

[jse.executeScript("document.getElementById(‘email').value=“abc.efg@xyz.com](mailto:jse.executeScript("document.getElementById(‘email').value=“abc.efg@xyz.com)”);

### How to upload a file in Selenium WebDriver?

You can achieve this by using sendkeys() or Robot class method. Locate the text box and set the file path using sendkeys() and click on submit button

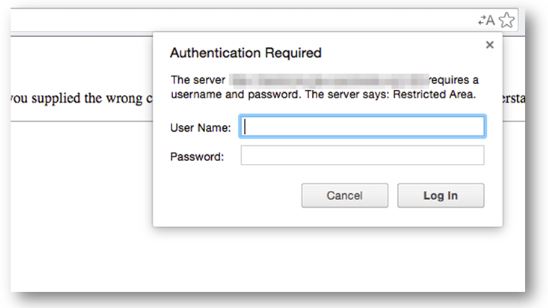
Locate the browse button

WebElement browse =driver.findElement(By.id("uploadfile"));

Pass the path of the file to be uploaded using sendKeys method

browse.sendKeys("D:\\SeleniumInterview\\UploadFile.txt");

### How to login to any site if it is showing an Authentication Pop-Up for Username and Password?



To handle authentication pop-ups, verify its appearance and then handle them using an explicit wait command.

Use the explicit wait command

WebDriverWait wait = new WebDriverWait(driver, 10);

Alert class is used to verify the alert

Alert alert = wait.until(ExpectedConditions.alertIsPresent());

Once verified, provide the credentials

alert.authenticateUsing(new UserAndPassword(<username>, <password>));

**Siemens Interview questions:**

***1. How to find Broken links :***

*Create a list of type WebElement to store all the Link elements in it.*

*for(int i=0; i<links.size(); i++)*

*{ WebElement element = links.get(i);*

*String url=element.getAttribute("href");*

*verifyLink(url);*

*}*

*Now Create a Connection using URL object( i.e ., link)*

*URL link = new URL(urlLink);*

*Connect using Connect Method*

*HttpURLConnection httpConn =(HttpURLConnection)link.openConnection();*

*Use getResponseCode () to get response code*

*if(httpConn.getResponseCode()!== 200)*

*Through exception, if any error occurred*

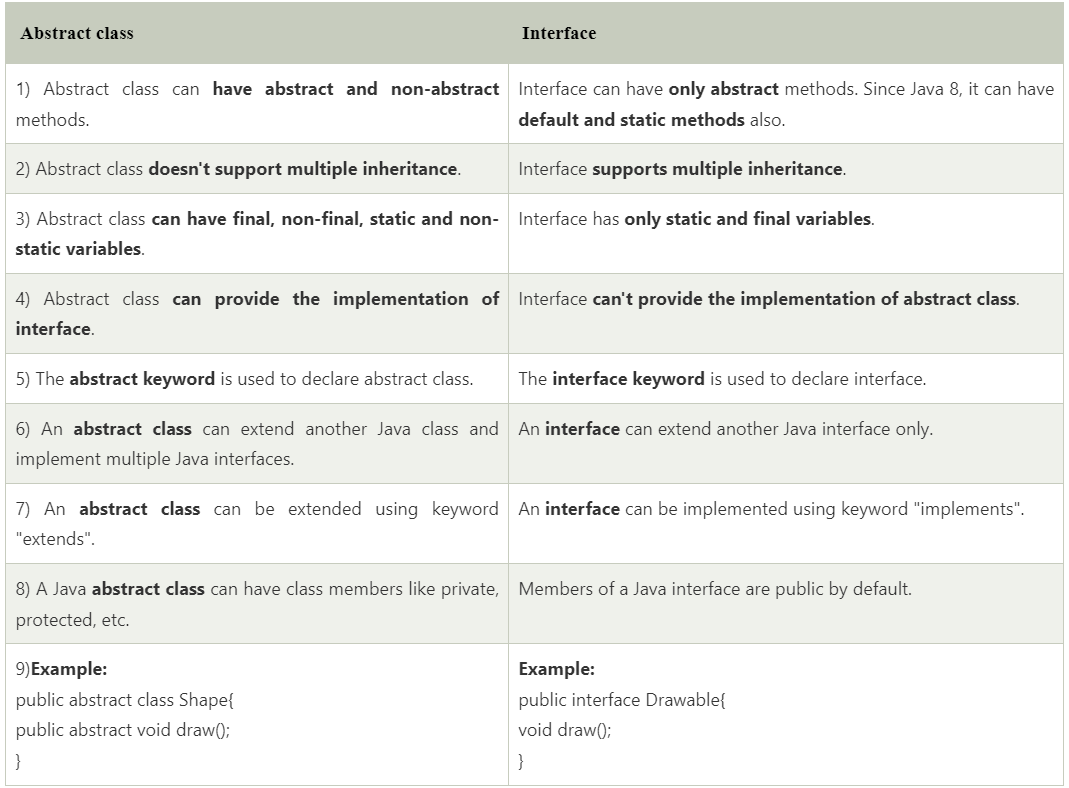
*System.out.println(“Broken Link”);*

**Git Pull Method?**

* The [git pull command](https://www.simplilearn.com/tutorials/git-tutorial/git-pull-request) is used to fetch and merge changes from the remote repository to the local repository.
* The command "git pull origin master" copies all the files from the master branch of the remote repository to the local repository.
* git pull <branch\_name> <remote URL>
* Git pull origin master

**One feature file to run 100 times how to do it ?**

**Abstact and Interface difference?**

  
 **Count characters from word**

String str= ”Count Characters”;

Char ch[] =str.toCharArray();

HashMap<Character,Integer> map= new HashMap<Character,Integer>();

For(int i=0;i<=ch.length();i++)

{

If(map.containsKey(ch[i]))

{

Int count=map.get(ch[i]);

Map.put(ch[i],count+1);

}

Else{

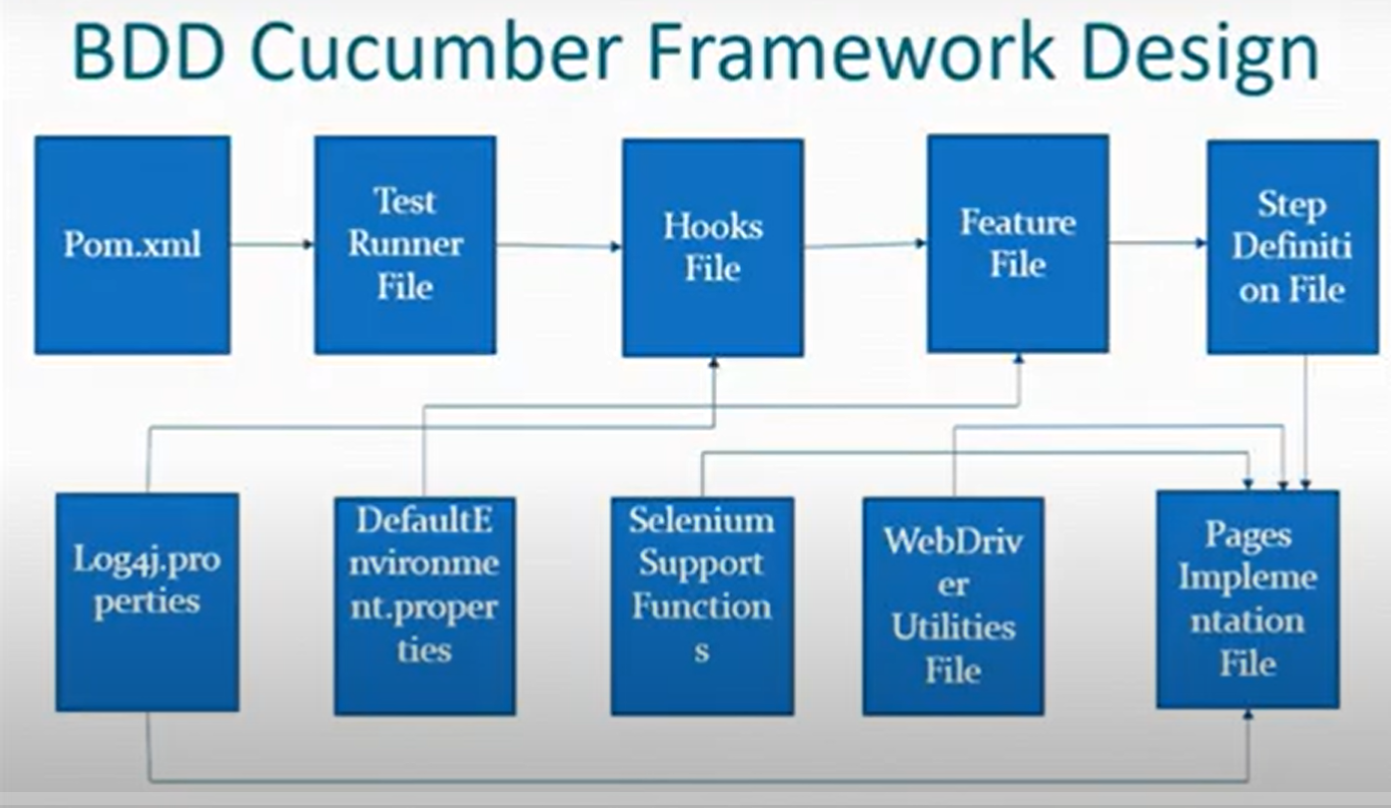
map.put(ch[i],1);

}

System.out.println(map);

}

**Framework folder structure**

  
 **How to automate Rest API?**   
 **Cucumber Disadvantages?**

### **Maintainability**

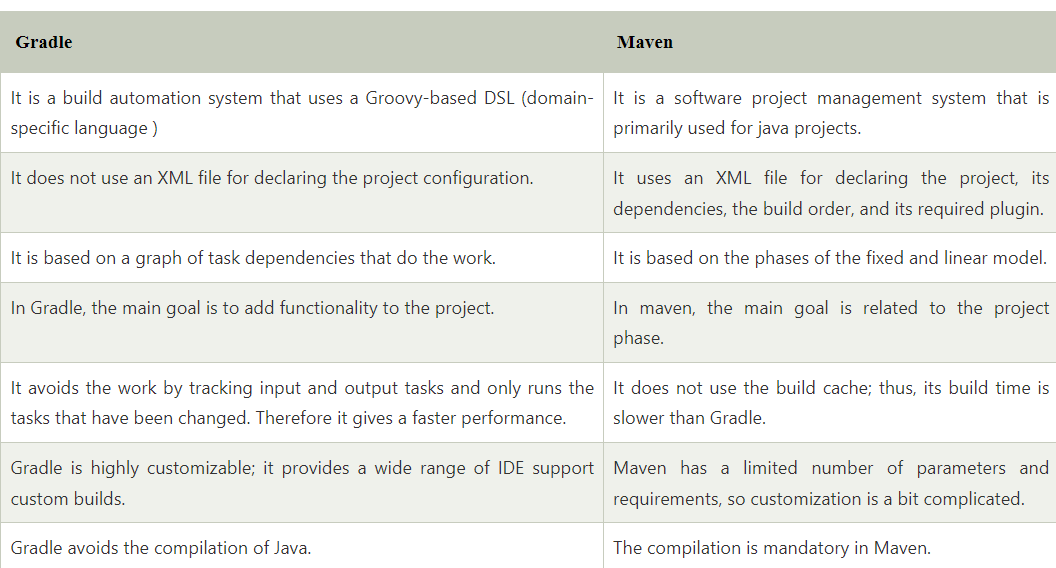
First, if you don't actually need to have others involved in the automation development, just having well-written methods & classes is more than enough. Cucumber can add an additional layer that just more tedious to maintain. As you can see, we have three different step definition classes just in this "simple" skeleton project.

### **Technical limitations**

Cucumber also has some technical limitations by design, chief among which being that you can't easily share information (aka state) between steps. You need to rely either on class-level fields or use additional libraries like Cucumber Picocontainer, Spring, or Guice.

A trivial example can be seen in the **frontend.feature** file. My expected page title contains a pipe character. This character is used by Cucumber to delimitate cells within a data table. Unless I escape the character, it would fail compilation due to a mismatch in table cell count.

**Mavan and Gradle difference?Maven disadvantage**



**Maven is limited to Java only**, whereas Gradle supports multiple languages and can be used to build projects in Java, C++, Python, and more.

  
 **When JAVA file created and when class file created**

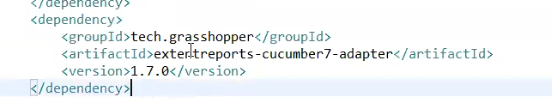
When we compile the Java source code (. java file), it generates a . class file. If a Java program has more than one class, in such cases after compiling the source file, we get the same number of .  
 **Where you give path for running feature file and class file ?**  
 **Scenario : If you are going on leave for one week for emergency reason (last week of sprint) , You have 20 scenarios to complete within sprint how you will do it ?**

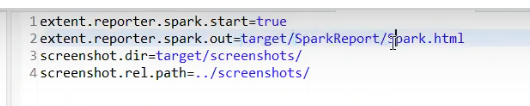
In this situation, Instead of spilling over all the sprint work, I would try and minimize the impact on the sprint. Run the priority test cases which are required to finish 1 or 2 critical stories.

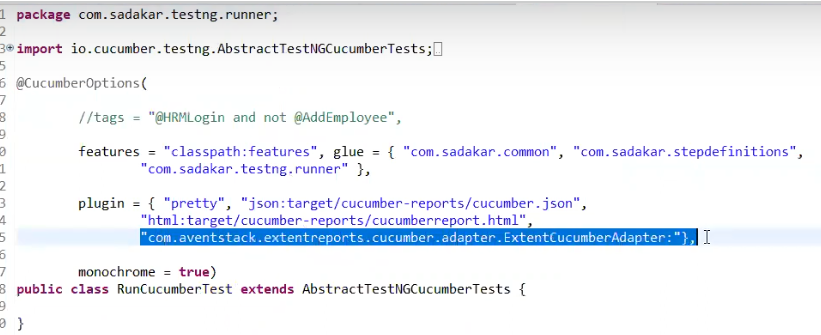
In case there are automation scripts available then impact on sprint can further be minimized.

Having said this, team will have to identify concrete action items so that such situation can be avoided in future. Inspect and Adapt!

**Which type of extent reports you are using in current framework ?**





  
 **JAVA code : count words from input string ? If you are written code then they will ask alternate method to write code(Tell them with collections such as set and HashMap)**  
 **From command line how you execute java code?**

Javac finename.java java filename

**Can Static method be overridden ?**

The short answer is No. Static methods in Java cannot be overridden. This is because static methods are not associated with the instance of a class, but with the class itself. Therefore, when a subclass inherits a static method from its parent class, it cannot modify the behavior of the static method in any way.

**Webtable having data to fill**

Feature :Login to Luma website

Scenario Outline: As a valid user user can login to website

Given User is on homepage of Luma website

When User gives valid <userId>

When User gives valid <password>

Then User successfully login with his account

Examples:

|userId | password |

| admin1| 12345 |

| user1| 123456 |

**How to upload files through selenium?**

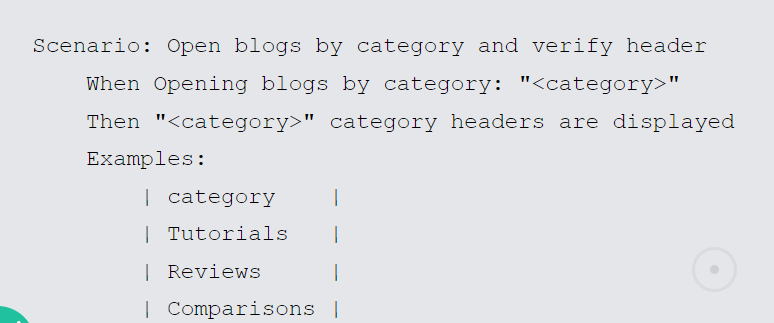
WebElement chooseFile = driver.findElement(By.id("custom-input"));

chooseFile.sendKeys("/Users/ankita/Downloads/edureka.png");

**When to use Scenario outline?**

The Scenario Outline keyword can be used to run the same Scenario multiple times, with different combinations of values. The keyword Scenario Template is a synonym of the keyword Scenario Outline . We can collapse these two similar scenarios into a Scenario Outline .  
 **How to execute One scenario with different data?**

That can be achieved using the Cucumber feature called [Examples](https://cucumber.io/docs/gherkin/reference/#examples). A special keyword that includes header names as a first row and their values starting from the second row. If for example, we have 3 rows in the examples that means that the scenario will run two times for different parameters.

  
 **How to swap two string without using third Variable?**

public class SwapString {

static String string1;

static String string2;

public static void main(String[] args) {

s1=s1+s2;

s2=s1.substring(0,(s1.length()-s2.length()));

s1=s1.substring(s2.length());

System.out.println("s1="+s1+" s2="+s2);

}

}

**How to find second max number and max number from array?**

public class MaxNumber {

public static void main(String[] args) {

int arr[]= {23,12,45,1,90};

int len=arr.length;

int max1=0;

int max2=0;

for(int i=0;i<len;i++)

{

if(arr[i]>max1)

{

max2=max1;

max1=arr[i];

}

}

System.out.println("max1="+max1+"max2="+max2);

}

**How to create Abstract class?**

**Abstraction** is a process of hiding the implementation details and showing only functionality to the user.

Another way, it shows only essential things to the user and hides the internal details, for example, sending SMS where you type the text and send the message. You don't know the internal processing about the message delivery.

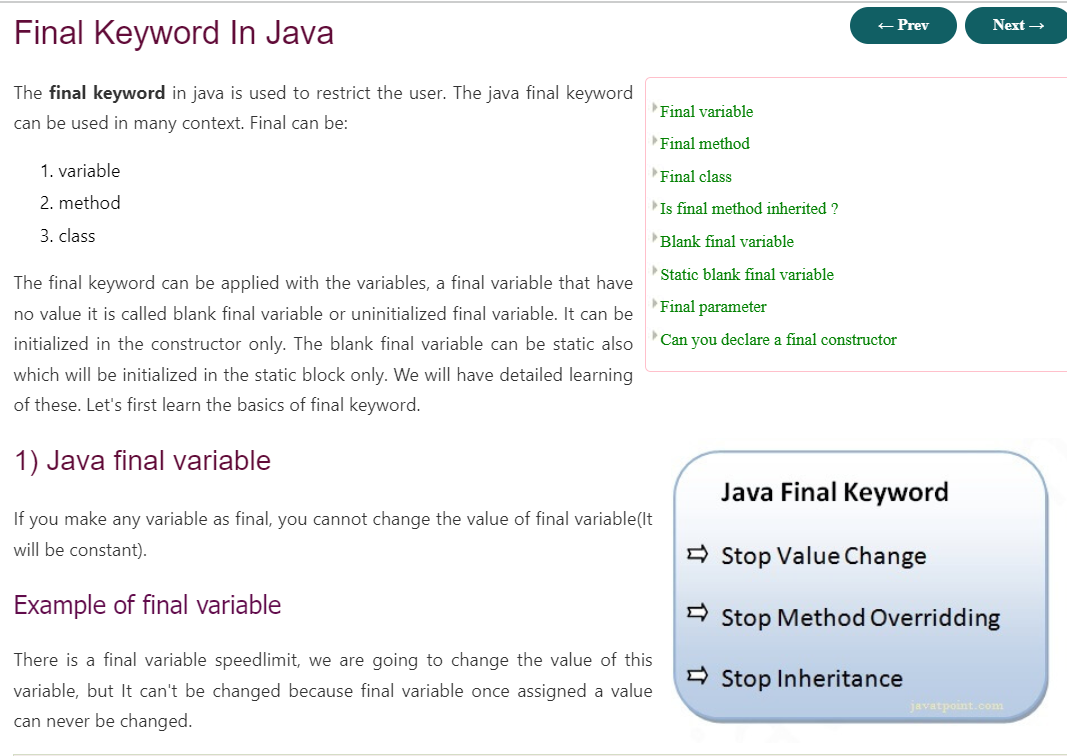
A class which is declared as abstract is known as an **abstract class**. It can have abstract and non-abstract methods. It needs to be extended and its method implemented. It cannot be instantiated.

#### Points to Remember

* An abstract class must be declared with an abstract keyword.
* It can have abstract and non-abstract methods.
* It cannot be instantiated.
* It can have [constructors](https://www.javatpoint.com/java-constructor) and static methods also.
* It can have final methods which will force the subclass not to change the body of the method.

1. **abstract** **class** Bike{
2. **abstract** **void** run();
3. }
4. **class** Honda4 **extends** Bike{
5. **void** run(){System.out.println("running safely");}
6. **public** **static** **void** main(String args[]){
7. Bike obj = **new** Honda4();
8. obj.run();
9. }
10. }

**Final keyword**

  
 **Git Push process**

git init  
git add README.md  
git commit -m "first commit"  
git branch -M main  
git remote add origin <https://github.com/Shubhangi-Sirsat/Testing-notes.git>git push -u origin main

### …or push an existing repository from the command line

git remote add origin <https://github.com/Shubhangi-Sirsat/Testing-notes.git>git branch -M main  
git push -u origin main

**Status code -2XX, 4XX, 5XX**  
 **How to validate response code ?**