

Selenium I.W que

- ① what is selenium & its component
- ② what is locators & its type?
- ③ what is webdriver?
- ④ what are the adv & disadv of selenium webdriver?
- ⑤ what is architecture of selenium webdriver?
- ⑥

Ques what are diff forms of selenium?

Ans It has 4 four forms

① Selenium webdriver :-

It is a interface used to automate web application using browser's native methods.

② Selenium IDE :-

A firefox plugin that works on record & play back principle.

③ Selenium RC :-

Selenium Remote Control is officially deprecated by selenium & it is used to work on JS to automate the web applications.

④ Selenium Grid :-

Allows selenium tests to run in parallel across multiple mc.

Ques) What are the abstract methods available in Selenium webdrivers?

Selenium webdriver interface has many abstract methods like

get(String url), quit(), close(),
getWindowHandle(), getWindowHandles(),

getTitle() etc.

WebDriver has nested interfaces like
Window, Navigation etc.

These nested interfaces are used to
perform operations like back(),
forward() etc.

① What is selenium?

→ It automate the browser, which contain multiple component.

② Components → ① selenium IDE

- ② selenium webdriver
- ③ selenium grid

③ webdriver =

→ It is a Java Interface, which contains n no. of abstract methods & variable of that interfaces implemented by multiple classes like chromedriver class, firefoxdriver class, etc.

→ It is an API (contains n classes & methods by which we can communicate with client and server) webdriver is mediator b/w browser & client libraries.

→ It is one of component in selenium by which we can automate a browser.

→ It is browser automation framework that accepts commands & sends them to a browser.

→ It is implemented through a browser specific driver.

→ It controls the browser by using directly communicating with it.

④ adv and disadv ⇒

adv ⇒ ① It supports multiple languages & multiple browsers.

② It supports multiple OS.

③ It is open source & freely available.

disadv ⇒ ① automates only browser.

It can't generate reports.

② doesn't support excel sheet data

③ can't automate windows based application.

e.g. upload files, capture, graphs

can't automate

but can overcome by integrating 3rd party tools.

i) automate windows Based Application

AUTOT, Sikuli

ii) data driven testing using excel,

we can use POI library, we can integrate it with selenium, webdriver

iii) if you want to generate a report

we can integrate 3rd party reports like extend allure report etc.

imp ⑤

selenium web driver architecture



→ upto selenium 3.8 there were JSON wire protocol, but in selenium 4.0 W3C (world wide web consortium) introduce W3C (world wide web consortium)

→ W3C → It is a standards
Browsers, drivers are followed by W3C Standards
But JSON wire protocol doesn't follow
W3C Standard. JSON wire protocol encode/decode code then
client libraries pass to driver
Because of this there is no stability & consistency betn different 2 diff. type of
protocoll. → but client libraries are following
JSON wire protocol

→ In JSON wire protocol, encoding & decoding is required, because of this mechanism there is inconsistency b/w two diff type of protocol.

→ So, in selenium 4, they have introduced W3C protocol instead of JWP working of W3C →

→ Drivers are following W3C, Browsers are also following W3C so we are using W3C protocol for more sync. So, W3C is also part of client libraries.

→ Internally W3C client libraries will talk to directly with drivers & browser so, no need to do encoding / decoding.

Adv → All Browsers & All Browsers specific drivers follows W3C & even client libraries also following same kind of protocol. There will be stability. There will not be inconsistency. We can do actions like minimize or maximize windows or do some actions on keyboard.

Q.W \Rightarrow difference is previously use
Ans JSON wire protocol in 3.8 now
JSON wire protocol & they have
introduced W3C protocol.
here using JSW encoding & decoding
required when you sending request
to the driver.
but when you use W3C protocol,
because all protocol are in same
standard, so encoding & decoding
is not required & communication
will be faster & stable, we can avoid
inconsistency.

⑥ How to launch Browsers in webdriver?

2 diff. way

1st way → ~~locate the~~ specify driver locatⁿ

for every browser, driver is available

Chrome Browser → chromedriver.exe available

Firefox → geckodriver.exe

Edge → edgedriver.exe

Step I search for chromedriver → + download
↓ download by using chrome version of your PC

After download, you will get zip file
that will contain chromedriver.

We need to use →

Step II System.setProperty(key, value)

key → ① chromedriver → WebDriver.ChromeDriver

value → location of driver

In selenium, every class is implemented
like WebDriver is an interface,
implemented by multiple classes.
Chromedriver class, FirefoxDriver by other class
this classes implemented WebDriver
methods.

Step III → Create object of chromedriver

1 way → `Chromedriver driver = new Chromedriver();`

In chromedriver browser, there is default constructor, & that constructor will automatically invoke, before that we need to import → org.openqa.selenium.chromedriver

Chromedriver which is implemented WebDriver
so webdriver variable can hold the chromedriver object

2nd way ⇒ `WebDriver driver = new Chromedriver();`

// Launch chromedriver Browser

→ driver.get("url"); // url passing

By using WebDriverManager, we can launch browser, No need to drivers manually.

Add dependency

Search for it copy dependency

[automatically download jar files] → by using pom

```
WebDriverManager.chromedriver().setup();  
class //Automatically setup the chromedriver  
path → avoid setting path  
WebDriver driver = new ChromeDriver();  
// launch chrome
```

⑦ How to open url ?

By using get method → driver.get() # launch browser

⑧ How to capture title of the webpage?

By using driver.getTitle() # give title of webpage

⑨ How to capture current url of webpage?

driver.getCurrentUrl() # return current url of webpage.

⑩ How to capture page source?

driver.getPageSource() # return HTML data

⑪ How to check WebElement is displayed, Enabled & Selected ?

3 methods are available:-

- ① `isDisplayed()` → check webelement display or not
both use for verifying webelements only
- ② `isEnabled()` → check value is accepting or not
- ③ `isSelected()` → check value if element is selected or not generally used for radio button, checkbox, dropdown
`getDriver().methods()` → return true or false

⑫ How to Navigate Back & Forward ? How to refresh page ?

-
- 1) `driver.navigate().back()`
 - 2) `navigate().forward()`
 - 3) `navigate().refresh()` # refresh page
 - 4) `navigate().to()` # open url

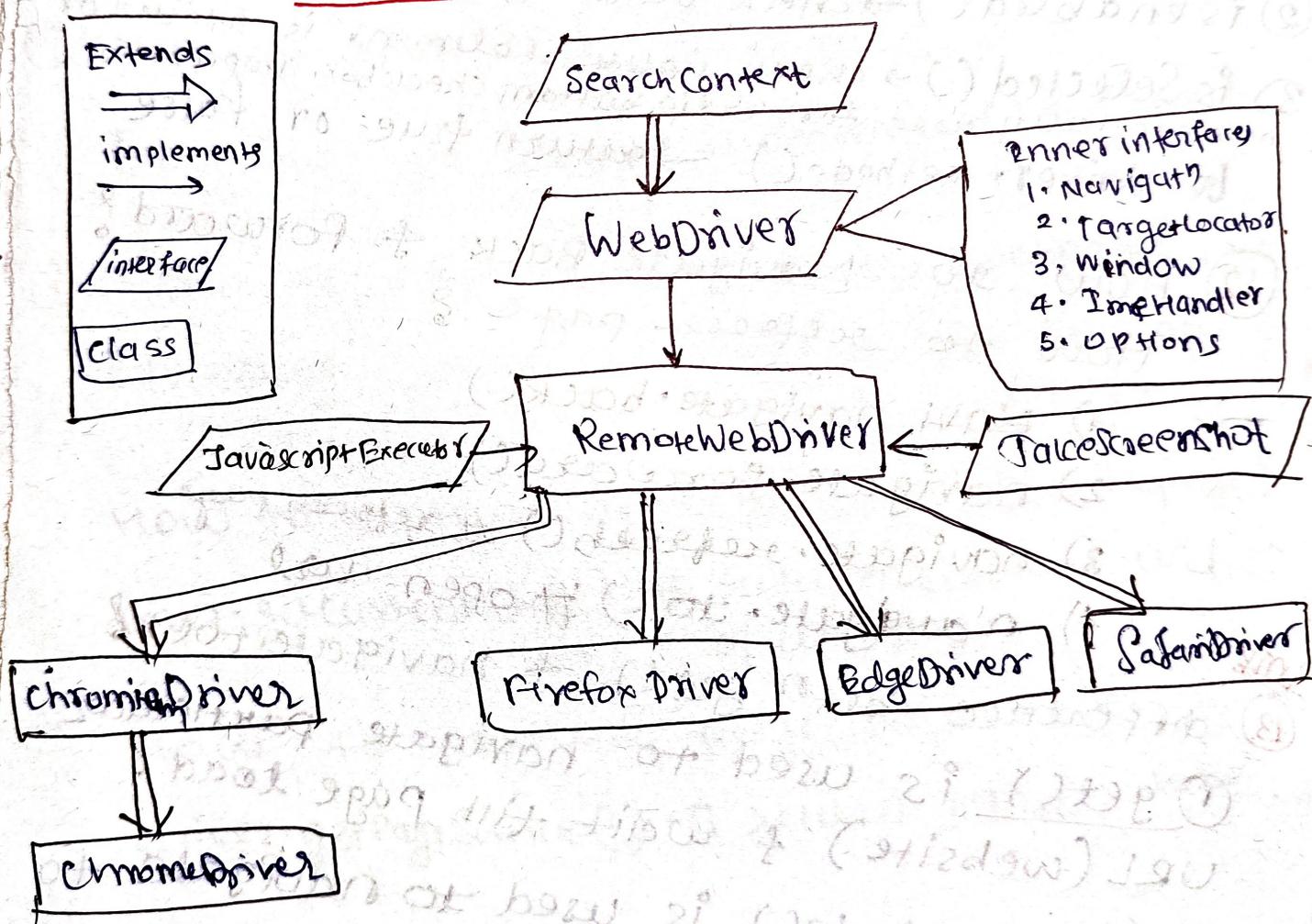
⑬ difference b/w `get()` & `navigate.to()` ?

① `get()` is used to navigate particular URL (website) & wait till page load.

② `navigate().to()` is used to navigate to particular URL & does not wait to page load. It maintains browser history or cookies to navigate back or forward.

→ The only difference is that can be found in the parameters.

- i) get() accepts only one string parameter.
- ii) navigate().to() accepts string parameter and URL instance as parameter.



relative X-path → identify element by using well known attribute or value & then directly jump to value

/ → represent html root node

Xpath

→ can traverse from top to bottom or bottom to top in a page

(can find any element)

css-selector

→ can traverse from top to bottom only. can't go to parent level.

wait → there are so many ~~at~~ wait statement To solve synchronization problem,

Sometimes application takes time to load a web element or sometimes pages will take some time to load. Sometimes our code will not wait for this. So we can use wait commands

So, we can use wait commands

In implicit wait, depends on time

In explicit wait, depends on condn

explicit wait → An explicit wait is a type of wait which is applied to a particular web element until the expected condition specified is met.

implicit wait applicable for only some all web elements.

fluent wait → here we can also specify polling interval (intervals after which driver will try to find element) along with the maximum timeout value.

context click → right click

→ tool tip defined by 'title' tag.
Tooltip → it is text that appears when mouse hovers over an object on a webpage.
→ Desired capabilities → DC are set of key-value pairs that are used for storing or configuring browser specific properties like its version, platform etc. in the browser instances.

Implicit

- ① It is time based
- ② It is applicable for all the elem. in web page

Explicit

- ① It is condⁿ based
- ② It is only for single element.

adv \Rightarrow POM \Rightarrow Reusability

test hybrid framework \Rightarrow Combination of Data driven & keyword driven framework

Data driven \Rightarrow consist of test data keyword given \Rightarrow in properties file, there prepare all kinds of variable & values of variable predefined frameworks like testNG & prepare

\hookrightarrow config files

everything in reusable folders

Selenium grid →

→ If you want to execute your test case from your local machine to remote machine on different browsers then use Selenium grid.

main system act as a hub ~~and~~ server. If the other remote system considered as nodes. Then we have to add node to the hub server then hub server will be able to recognize all the node system. So, nodes comes in a remote system. Then you can execute test case of local system on multiple browser. This is called as distributed computing or distributed test cases across multiple system.

Testing

→ NG → Next Generation
→ Built in framework, integrated
with selenium or other automated
tool to provide multiple capabilities
like assertions, reporting, parallel
test execution etc.

Adv → ① Assertion
→ By using assertions, we can validate
easily.

adv → ② Parallel execution
③ dependency → Based on priority, we can
④ priority → control the execution of test cases
④ grouping → Suppose there are two of
test cases, some of them are from sanity
test cases, some of them all belongs
to regression etc then if I want
to execute only sanity or only
regression then that type managed in
grouping.

- parametrized →
- ① Data provider parse
→ generate data & pass
for same test determined
in same class
 - ② Pass pass parameters through xml
file to test case.

Because of this we can achieve
parallel ~~test~~-runner feature.

Data Driven testing ⇒

we have to read same data
of xml & pass the data to application

Same data on script.
Script will perform testing on
application by using same data.

→ parametrized →

① Data provider which
→ generate data & passed
to same test data provider
in same class

② Pass pass parameters through xml
file to test case.

Because of this we can achieve
parallel ~~test~~ testNG feature.

Data Driven testing ⇒

We have to feed same data
of xml & pass the data to application
script.

Same data on script.
Script will perform testing on
application by using same data.

Scenario & Assertions

Soft Assertion (softAssert)

- It allows us to have multiple test method assertions within test method
- even when an assertion fails the test method continues with the remaining test execution

Hard Assertion

- This are used assert provided by testNG for e.g. assert_true, assert.assertEqual By default provide by testNG.
- If any assertion fails then we can't immediately the test execution will stop. can't continue further

- Excel Sheet etc used for maintaining data
- How to scroll down page using JS in selenium driver class into JSExecutor convert driver script by pixel and use method executeScript by scrollBy(0, 500) passing parameters like windows. scrollBy as a parameter
- Popups can be handled, windows related, authentication related

wait → used to solve synchronization problems

CI tools → Jenkins, Bamboo

framework → ① it will be designed to organize the project related file in a proper structured manner.

Hybrid framework \Rightarrow explain (8th video)
→ A framework defines a set of rules or best practices which we can follow in a systematic way to achieve the desired results.

* CI tools → Jenkins, Bamboo

There are continuous integration tools helping in quick deployment of application, testing them & reporting the issues in the code before it is too late.

It helps in getting the application into production quickly & with more quality confidence.

* Build Tools \Rightarrow Ant, maven

We used these tools to manage build activities for the java project.

* version control tools like SVN, GIT

We use version control tools like github/SVN to track the changes to

the files in a project & work in

collaboration.

* desired capabilities \Rightarrow

capabilities we need to set the

values of the browsers attributes

before we launch any browser using

selenium web driver.

CI is a software development method where members of the team can integrate their work at least once a day.

CD/CD meaning combination of CI & Continuous delivery or continuous deployment.

* How to send an email stating the execution status to all stakeholders in selenium?

We can send email in java using javax.mail library.

* How to schedule the test suit execution?

We can schedule the test suit execution using CI tools like Jenkins (hudson), Bamboo. Alternatively, we can use windows scheduler to launch the test execution.

* Database testing in selenium?

We can use JDBC driver to connect to any db in java.

* How can you use the Recovery scenario in selenium webdriver?

By using try-catch block in our selenium webdriver Java tests.

```
try {  
    driver.get(".com");  
}  
} catch (Exception e) {  
    System.out.println(e.getMessage());  
}
```

- * List some scenarios which we cannot automate using selenium webdriver ? (SW)
- ① Bitmap comparison is not possible using SW
- ② Automating Captcha is not possible
- ③ we can not read bar code using SW
- ④ windows OS based pop ups
- ⑤ third party Calender/element
- ⑥ images
- ⑦ word | PDF
- * How to get an attribute value using SW?
By usinggetAttribute("value");
- * How to get a text of a web element?
By using getText() method.

* what are the types of waits available in SW?

Implicit, explicit, fluent, pageLoadTimeout
Thread.sleep() → static wait.

* what is super interface of webdrivers?

SearchContext

* what are the open source frameworks supported by SW?

Junit, TestNG, CUCUMBER, TBEFAVR

OS supported by SW → windows, linux, mac

→ Java, Python, C#, Ruby

* lang → Perl, PHP

* HTMLUnitDriver is a fastest implemented of webdrivers because it does not use browser.

execute tests in a browser.

* types of webdriver API's available in selenium

Firefox Driver, Internet Explorer Driver, Chrome Driver, HTMLUNIT Driver, Opera Driver, Safari, Android, iPhone

* node ⇒ it is the machine which is attached to hub. there can be multiple node in selenium grid.

* hub ⇒ it is server or a central point that controls the test executions on diff. mc.

* Adv of Selenium Grid =

① It allows running test cases in parallel thereby saving test execution time.

② It allows multi browser testing

③ It allows us to execute test cases on multi-platform.

* When do you use selenium Grid ?
It can be used to execute same or diff. test scripts on multiple platforms of browsers concurrently so as to achieve distributed test execution.

* What is selenium ?
It is the language which is used to write test scripts in Selenium IDE.

* How many test cases you have automated per day ?

→ It depends on Test case scenario complexity & length.

→ I did automate 2-5 test scenarios per day when the complexity is limited.

→ sometimes just 1 or 4-5 test scenarios per day when the complexity is high.

* what type of tests have you automated?
our main focus is to automate test cases to do regression testing, smoke testing & sanity testing.
sometimes based on the project & the test time estimate, we do cases on end to end testing.

* limitation of selenium

- ① we cannot test desktop application using selenium
- ② we cannot test web services using selenium
- ③ for creating robust scripts in selenium webdriver, progr. knowledge is required.
- ④ we have to rely on external libraries & tools for performing tasks like logging(log4j), testing framework - (testNG, Junit), reading from external files (POI, for excel) etc.

* BROWSERS Supported by Selenium
Chrome, fire fox, IE, safari, HTMLUnit, Appium, iOSDriver, Android

* we can't test API's or web services using selenium webdriver
→ selenium webdriver uses browser's native method to automate the web applicatn. since web services are headless, so we can't automate web services using selenium webdriver

* X-path → Xpath or XML, path is a query language for selecting nodes from XML documents.

X-path is one of the locators supported by Selenium webdriver.

Absolute X-path → It is a way of locating an element using XML expression beginning from root node.

The main disadvantage of absolute X-path is that even with slightest change in the UI or any element the whole absolute X-path fails.

Relative X-path → It is a way of locating an element using an XML expression beginning from anywhere in the HTML document.

We prefer to use relative X-path because it can identify element even though some UI changes happened, but can't identify by absolute X-path.
Ques → How can we inspect the web element attributes in order to use them in diff. locators?
Ans → Using Chropath or developer tools like chrome developer tools to inspect the specific web elements.
Chropath is a plugin that provides Xpaths & CSS Selectors.

Ques ⇒ How can we move to nth child element using xpath?

There are 2 ways of navigating to the nth element using xpath.

① using square brackets with index position

e.g. `div[2]` will find the second div element.

② using position() →

`div[position()=3]` will find the third div element.

Ques ⇒ diff b/w XPath & CSS selector

Using XPath we can traverse up in the document i.e. we can move to parent elements. Whereas using CSS Selector we can only move downwards in the document.

Ques ⇒ How can we launch a browser

By creating an instance of driver of a particular browser.

Ques ⇒ diff b/w close & quit

`driver.close()` → used to close the current browser having session.

`driver.quit()` → used to close all the browser instances.

- All → diff b/w Absolute xpath & Relative xpath
- Absolute xpath will traverse entire HTML from the root node / html.
 - Relative xpath directly jump to node based on attribute specified.

How to switch b/w multiple windows in selenium?

- selenium has driver.getWindowHandles() & driver.switchTo().window("{windowHandle}") commands to work with multiple windows.
- The getWindowHandles() command returns a list of ids corresponding to each window & on passing a particular window handle to driver.switchTo().window("{windowHandle}") command we can switch control / focus to that particular window

```
for (String windowHandle: driver.getWindowHandles())  
{  
    driver.switchTo().window(handle);  
}
```

- driver.getWindowHandle() returns a handle of the current page (a unique identifier) whereas driver.getWindowHandles() returns a set of handles of all the pages available.

Ques ⇒ can we move back & forward in browser using selenium?

Yes, using driver.navigate().back() &

driver.navigate().forward() commands
we can move backward & forward in a browser.

GIBT commands →

get() → It automatically opens a new browser window & fetches the page that you specify inside its parentheses.

Parameter must be string object.

getTitle() → fetches the title of current page.
Returns null string if the page has no title.

getPageSource() → Returns the source code of the page as a string value.

getCurrentUrl() → fetches the string representing the current URL that the browser is looking at.

getText() → fetches the inner text of the element that you specify.

Navigate Commands

① navigate().to() →

→ automatically opens a new browser window & fetches the page that you specify inside its parentheses.

② navigate().refresh() →

It refreshes the current page.

③ navigate().back() →

It takes you back by one page on the browser's history.

④ navigate().forward() →

It takes you forward by one page on the browser's history.

close() → It closes current window of browser.

quit() → It close all windows that webdriver has opened.

- ① Can we test APIs or web services using selenium webdriver?
- No, SW uses browser's native method to automate web applications. Since web services are headless, so we cannot automate web services using Selenium web driver.
- ② What are the testing types supported by ICSE?
- SW can be used for performing automated functional & regression testing.
- ③ Which Xpath you will prefer to use? Why?
- Normally we prefer to use Relative Xpath. Because Relative Xpath can identify element even though some UI changes happened, but can't identify by Absolute Xpath.

Absolute x-path

it will traverse entire HTML from the root node /html.

Relative x-path

it directly jump to node, based on attribute specified

How to inspect the web element attributes in order to use them in diff. locators?

Using chrompath or developer tools we can inspect the specific web elements.

Xpath

using xpaths we can traverse up in the document i.e. we can move to parent elements.

css selector

whereas using css selector we can only move downwards in the document.

(HOW launch browsers in SW)

By creating an instance of driver
of a particular browser.

→ by using sendKeys() we can type
text in a textbox.

→ driver.getWindowHandle() → returns a handle
of the current page (a unique identifier)

whereas driver.getWindows()

→ driver.getWindowHandles() → returns a set of
handles of all the pages available.

Handles of the all the pages available.

Ques How can we move to particular
frame in selenium

The driver.switchTo() commands can

be used for switching to frames.

driver.switchTo().frame("frameIndex/frameId/
frameName");

For locating a frame we can
either use the index (starting from 0),
its name or Id.

Ques How to refresh browser?

- ① using driver.navigate().refresh() cmd
- ② Using sendKeys(Keys.F5) on any textbox on the webpage.

Ques How to maximize?

driver.manage().window().maximize();

getText() ⇒ we can fetch the text over an element.

getAttribute("{"attributeName}") ⇒
we can find the value of diff. attributes of an element.

deleteAllCookies() → delete cooking in Selenium

What are some expected condn
that can be used in Explicit
waits?

Some of the commonly used
expected condn of an element that
can be used with explicit waits
are

- i) elementToBeClickable(WebElement element)
or By locator
- ii) visibilityOfElementLocated (By locator)
- iii) attributeContains(WebElement element, String
Attribute, String value)
- iv) alertPresent() alertIsPresent()
- v) titleContains (String title)
- vi) titleIs (String title)
- vii) textToBePresentInElementLocated (By,
String)

Fluent Wait \Rightarrow

It is a type of wait in which we can also specify polling interval (intervals after which driver will try to find the element) along with the maximum timeout value.

```
wait = new FluentWait(driver).withTimeout  
Duration.ofSeconds(5).  
(20, SECONDS).pollingEvery(5, SECONDS).  
ignoring(NoSuchElementException.class);
```

Mouse Actions \Rightarrow

- ① click (WebElement element)
- ② doubleClick (WebElement element)
- ③ contextClick (WebElement element)
- ④ moveToElement (WebElement element)
- ⑤ dragAndDrop (source WebElement, target WebElement)

Ques ⇒ What are some commonly encountered exceptions in selenium?

- ① NoSuchElementException →
When no element could be located from the locator provided
- ② ElementNotVisibleException →
When element is present in the DOM but is not visible.
- ③ NoAlertPresentException →
When we try to switch to an alert but the targeted frame is not present.
- ④ NoSuchFrameException →
When we try to switch to a frame but the targeted frame is not present.
- ⑤ NoSuchWindowException →
When we try to switch to a window but the targeted window is not present.

⑥ TimeoutException →

When a command execution gets timeout.

⑦ InvalidElementStateException →

When the state of an element is not appropriate for the desired action.

⑧ NoSuchElementException →

When we are trying to fetch an attribute value but the attribute is not correct.

⑨ WebDriverException →

When there is some issue with driver instance preventing it from getting launched.

Ques → How can we capture screenshots in selenium?

→ Using getScreenshotAs method of TakeScreenshot interface we can take the screenshots in selenium.

```
File srcfile = ((TakeScreenshot)driver).  
getScreenshotAs(OutputType.FILE);  
FileUtils.copyFile(srcfile, new File("D:\\test.jpg"));
```

Ques → How can we handle window UI elements & window POP UPS using selenium?

Selenium is used for automating Web based application only (or browsers only). For handling window or UI elements we can use AutoIT or Fileuli.

Ques What is Robot API?

Robot API is used for handling keyboard or mouse events. It is generally used to upload files to the server in selenium automation.

```
Robot robot = new Robot(); // simulate enter key action
```

```
robot.keyPress(KeyEvent.VK_ENTER);
```

Ques → How to do file upload in selenium?

- ① using element.sendKeys("path of file") on the webElement of input tag & type file.
i.e. element should be like `<input type="file" name="fileUpload">`
- ② using Robot API
- ③ using AutoIT
- ④ using Sikuli

Ques ⇒ default port of Selenium grid?
4444

Ques ⇒ How do you handle Ajax dropdowns with the help of Selenium Sync Commands like Implicit Wait, WebDriverWait or fluent wait.

Ques ⇒ what are diff. pop-ups that you have handle in your projects?

i) Javascript Pop

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

ii) Browser Pop Ups

- Browser Profiles, Robot class, AutoIt, Sikuli

iii) Native Os Pop Ups

- Browser Profiles, Robot class, AutoIt, Sikuli

we use @findBy to identify element in the Page factory approach.

① BeforeTest → it is test level annotated.

① BeforeTest method will execute before starting the test.

② suit contain multiple tests

② AfterTest → it is test level annotated after completion of test.

① After suit method will execute after completion of entire test.

③ Before test

- it is test level annotated
- it will execute before actual test started

Before class

it is class level annotated.

- execute before the class one time

before method

it will execute multiple times before after every test method

After test

① it will execute after completion of entire test.

After class

• it will also execute one time after completion of class

after method

it will execute multiple times after every test method.

Automation life cycle →

- ① determining the scope of test automation
- ② selection of the appropriate Automation tool for test automation
- ③ developing the test plan, test design & test strategy.
- ④ The test environment should be set up
- ⑤ developing the automation test script & its execution
- ⑥ test analysis & generation of test results & reports

DevOps ⇒ it is a set of practices that combines slow development of IT operations.

⇒ it aims to shorten the system's development life cycle & provide continuous delivery with high SW quality.