

Self Intro:

=====

Hi this is kavin, currently working in Tpf software solutions for 4.5 years in both automation and Manual.

I worked on different domain such as ecommerce & Airline.

My last project of Ecommerce which is Newzealand based client.

In that we follow agile Methodology with 2 week of sprint.

We implemented BDD framework with cucumber tool integration of Junit.

In the current project we use Maven as Build Management tool,

we use design pattern such as Page object model which is used to maintain the locator and improve code reusability

And singleton design pattern model which used to avoid multiple object creation.

For maintain test data we use Datadriven framework with use of apache poi dependency.

We use Jira as Defect tracking tool, Git->Source code management tool, CI/CD- we use Jenkins.

Apart from this i have experience on other framework such as TDD testNG which we used to execute parallel and cross browser experience.

Roles and responsibility

=====

Once the requirement comes to me i will understand the requirements and understand the requirements .

if i have any doubts in the requirements ill ask for the walk through session .

BA will give me the proper walk through session on the requirement that i hold.

and after the walk through session ill be very clear with what is the testing that i have to perform.

So i can easily scope what to test and what not to test.

After that i ll be asking for the application access.

In mean time my team lead will prepare a test strategy and test plan document and once the planning is done we ll go the test case design.

Since i know the requirements, ill writing the testcases.

For manual stories we ll go for the manual test cases and for automated stories i need to create automated test cases.

So we will be waiting for the build to come.

after creating the testcases and for all the testcases ill be in need of test data.

And after creating test data ill go for the reviews Of my testcases.

There are three level of reviews one is peer review, next is lead review and BA review.

So after completing these review i might have some review comments which i havent covered before.

Ill update the review comments and resubmit my test caes for reviewing it and ill het it approved.

Once i got approved ill be waiting for the build from the developer to come to me.

and once the developer gives me the build ill start executing manually for manual test cases and my scripts will execute the automated test cases like automation related functionalities.

So we ll following all the retesting, regression, sanity and smoke based on the time

which the project might ask and we ll be executing every single stories in case of any bugs we ll raise bug

and entire defect management will happen over there

after the developer has fixed the bug ill perform retesting and in that way ill rerunning all the failed testcases and complete the execution on time.

Once the execution is completed ill be closing my stories and thereby we ll be closing one by one stories and we ll be giving signoff to our client.

GIT

===

GIT we can say source code management tool or version control tool, used to maintain our code in a repository.

Using GIT only we can push our code from local to repository and pull the code from repository. Initially i will create my own branch and i will take a new clone from using git clone URL command.

And i will checkout to my branch by git checkout command.

Then once the requirement came, i will start scripting and push my code to my branch on daily basis by using git init command which i will do at the starting of the project And git status to check out what are the files that i had changed.

Git add . to add the files Git commit -m message to commit

git push -u origin branch name, i will push my code to branch.

Once i have dome with the requirement, i need to take a latest pull from master to avoid conflicts.

Because if one of the colleague has merged his/her code to the master, then if i'm not aware of it and tries to push my code means, conflict will occur.

To avoid that, i will get updated code from the master by git pull origin master and i will remove the header and tail and update my code below the updated codes from master.

And give pull request to my manager. My manager will review my codes and he/she will merge our code to master. This happens on daily basis

Apl postman:

===== -

Get -we can validate and we can get details

Post- its used for creation

Put-it's used update and modification

Patch - its used to key and value pair

Delete - its used to delete

Agile methodology

=====

Intially, product owner will prepare a user stories and maintain all the user stories in a product backlog.

Then sprint grooming meeting will be conducted, this meeting will be attended by product owner, scrum master and scrum team.

Here product owner will take a set of user stories, and explain it in the meeting. We should try to understand the user stories.

If we have any queries means, we have to mark it in the query log and get clarified in the approve session which will be conducted later on.

Then sprint planning meeting will be conducted. Here scrum master and scrum team will give user story points to the user story which we are going to work on next sprint.

In my company, user story points are calculated based on poker card technique which is based on fibonacci series

The user story pionts will be given based upon the complexity of the user stories. Once it is done, the particular user stories will be moved from product backlog to sprint backlog.

These two meeting will be conducted before the sprint starts. Once sprint starts, we will be having a daily stand up calls in all days of this prints. It is a short meeting.

Here scrum team will share the what we have done yesterday and what we are going to do today to the scrum master. If we have any blockers also, we will get it clarified here.

So, once the sprint starts, we will do our regular testing works like scripting and execution. Daily stand up calls will be based upon the burn down charts. If we have any defects, we have to raise

bug and have to follow it.

Once sprint ends, sprint review meeting will occur.

This meeting is to showcase the work to the product owner. Here, scrum team will do demo the product owner. If the product owner satisfies the work, then he will accept. If product owner is not satisfied, he will reject it.

Once sprint review is completed, we will be having sprint retrospective meeting. It is type of appreciation and feedback session.

Here scrum master will discuss with the scrum team about what are all the positive and negative of this sprint and scrum team will rate the sprint out of 10.

TestNG:

=====

We can generate the report as HTML. We can pass the input using parameter, in testNG.xml file we have to pass the parameters with the name and value. We have to call the name in the execution class using DataProvider annotation. We can change the order of execution by giving priority to the each test cases. We can pass the bulk of data using Data provider where we have to give name to the each testcases using dataprovider and call that name using @DataProvider annotations.

We can achieve parallel execution by three ways, class method and test

We can achieve cross browser testing using testNG, for that we have to pass the parameter tag in testNg.xml giving a diff browser values we have to call the in my execution class using @parameters annotations.

We can run particular testcases, we can go with grouping concepts

We have to give the name to the each testcases, we have to call the name in testNg.xml using include or exclude tag

To verify and validation, we go with assert concept

We can achieve hard(if particular line gets failed, it wont execute the program) and soft assert
Rerun - listener tags in testNG.xml.

Annotations:

@beforesuite

@beforetest

@beforeclass

@beforegroup

@beforemethod

@test

@Aftermethod

@Aftergroup

@Aftertest

@Afterclass

@Aftersuite

Priority:

When we want to change the order of the test method we can go for the concept called priority.

Default priority for any test is 0.

priority we can give +ve and -ve values.

Enable:

if we are passing enable=false it will ignore the particular test from execution.

Dataprovider:

Using dataprovider we can pass bulk of data to a testcase.

We can test all the +ve and -ve test case very easily.

@optional:

if the class name and xml value not same we are using @optional(value)

parameter:

Hiding the datas.if we can bulk of data input from testng.xml we can pass using parameters

Cucumber:

=====

we are using maven as build management tool

We are using cucumber with JUnit

In pom.xml we add all the dependences required for the frame works.

TestRunner class - @RunWith(cucumber.class) and @cucumberoptions(feature-path of feature file, glue - name of stepdefiniton package, dryrun - used to generate the missing snippets, plugin - where i will pass the path of the folder where reports gets generated and stored, In plugin i can give pretty(what are executed and skipped),)

Featute - where we write user stories what im going to test

Scenario/scenaroi outline - will give test cases

Given - will pass the condition

When - action and

Then -validation purpose

And - connection the action of two validations

Scenario Outline - If im going to pass the inputs from feature file itself with example as a mandaory keyword

next i will create Hooks class- in this hooks we are have @before and @after annotations

In this @Before contain starting a web driver,setup the maximize browser,starting the wb page by using get method.

and @After annotation having takes screenshots method.

In this method used to take pass and failure test cases screen shots.

Next i will create Baseclass in this base class having predefined methods.

Once i run the testrunner class, it will generate the snippets

I will paste that in step definition class

STepdef - i will call the methods from base class and locators from pojo class

JVM report(will create a class, here i will pass the path of the folder where i want the reports to be stored)

Featue;

To mention the path of feature file.

Feature define the logical test functionally test.

Feature keyword is present at the starting of the feature file.

scenerio;

In the scenerio we are passing one set of datas.

scenerio outline-

To execute same scenerio which more time with differend set of datas.

It is common all the steps in the scenerio outline

we need to use keyword called Example to pass our test datas.

Backgrounds;

To perticular feature file in background.

In background we can maintain all the common functionalities.

Hooks;

To all the feature file runn in hooks class.

Common feature file in the feature to be given at @Before and @After

But;

To Pass the represent negative statements

And;

connection the action of two validations and Pass the positive commends

Given;

It will pass the condition of test case

Maven:

=====

Software project tool of build management tool for framework.

Its used to get central reciprocity of dependencies.

And maintain common structure across the organization.

flexibility in integration with continuous integration tool.

plug in for test framework execution.

Maven Execution:

=====

Starts from pom.xml file, it reads the pom.xml file and start the execution

It is used to build the software app, manage our dependencies, run our test and create reports

Structure:

src/main/java

src/main/resources

src/test/java

src/test/resources

JRE system library

Maven Dependency

src

target

pom.xml

defect Life cycle:

Clean - It will clean the existing jar file

Install - It will install the newer version of jar file

Test - Execution

Repo:

Local - .m2 folder(all the jar files will be there)

Remote - Eclipse

Central - Apache maven file 3.8.2

Data Driven:

Data Driven is nothing but when we are using bulk of data to use from pages in packages for that scenario i use to go for Data Driven.

First i have to create a class called "File" then i have to create a obj name inside the class we have to pass the file path and then i will create FileInputStream (class) then i use to give the XSSFWorkbook because i use the excel extension of .xlsx then i create Sheet and then i will go for row using row reference name we create column name

```
File(Class) f(object name) = new File(file path);
```

```
FileInputStream stream(object name) = new FileInputStream (f);
```

```
//create a workbook
```

```
Workbook(Interface) w = new XSSFWorkbook(stream);
```

```
Sheet(Interface) sh = w.getSheet("Sheet Name");
```

```
Row r = s.getRow(row number);
```

```
Cell c = r.getCell(cell number);
```

POM:

===

It is used to maintain all our locators in pagewise.

In my project, i have morethan 20 pages. For each page, we will have pojo(plain old java object) class.

In pojo, we will hav a construct which contains pagefactory.intelelements. Pagefactory is a class used to store all the webelements.

intelelements is a method used to initialise the webelement.

Then we will have some annotations

@findBy(we can call only one locator)

@findBys(we can call morethan one locators, it should satisfy Andconditions i:e all the locators should be satisfied)

@findAll(we can call morethan one locators, it should satisfy Orconditions)

@cachelookup(for memory management)

In Pojo, we will declare all locators in private, which is a coding standard.And we will generate getters and setters,

By creating a object in another class, we can access the locator indirectly.

We can overcome staleElementReferenceException

Exceptions:

=====

Timeout - When the command takes longer than the given wait time

NosuchFrame - when webdriver tries to switch to invalid frame

NoALretPresent - when webdriver tries to switch to invalid alert

NosuchWindow - when webdriver tries to switch to invalid window

Nullpointer - when the sheet name is wrong

Seviority;

its impact of a particular defect on the software.

severity means how severe defect is affecting the functionality.

perioridy:

its used to decide the order the perticular defect fixed based on high,medium,low.

smoke Test;

smoke testing is done to make sure if the build we received from the development team is testable or not.

sanity Test;

sanity testing is done during the the release phase to check for the main functionality of the app without going deeper.

ReTesting:

to ensure that the defect which were found and posted in the earlier build were fixed or not in the current build.

regration testing;

Repeated testing of an already tested program, after modification,to discover any defect intriduce.

REAL TIME :

Class:in my project we are using lots of predefined class like chromedriver

*firefox*ie*actions*robot*select*remotewebdriver

Method:

In our project we are using lotz of predefined methods get,gettext,getcurrenturl,gettext,

Encapsulation:

maintain all the locators in pojo class as object repository .. using object indirectly access the locators in differnet class.. using the concept of pom with page factory

INHERITANCE:

We used to maintain all

The reusable methods in sendkeys,click and using extends keyword

METHOD OVERLOADING :

In our project using method over loading concept like println,sendkeys,frame,waits

METHOD OVER RIDING:

To take screenshot we need to over the get screenshot as method which is available in get screenshot interface

* list(get,index of)

*set(add,remove)

Map(put,contains key)

*we create the user defined exception we need to over ride GetMessage method in exception class..

ABSTRACT:

In our project we are using abstract class to declare the locaters

To get the response code we are having HTTPURL CONNECTION

INTERFACE:REAL time

=====

We are using lotz of interface

Search context

Wendriver

Javascript

Takascreenshot

Retryanalyzer

Alerts

Webelement

Waits

List

Set

Map

1.What is Selenium?

=====

Selenium is automation testing tool used to automate web based applications

Open source tool

2.Components of Selenium?

=====

Selenium IDE(Integration Development Environment)

Selenium RC(Remote Control) [server]

Selenium WebDriver(3.141.59)

Grid

3.WebDriver:

=====

WebDriver is predefined interface

methods:

=====

get()
findElement()
findElements()
getCurrentUrl()
getTitle()
quit()
close()
getWindowHandle()
getWindowHandles()
switchTo()
maximize()
navigate().to()
forward()
back()
refresh()

4.difference b/w get() & navigate():

=====

get()

=====

used to load the url of the page and it will not store browser cookies(history)
here we cannot perform navigation(forward,back,refresh)

navigate()

=====

used to load the url of the page and it will store browser cookies
here we can perform navigations(forward,back,refresh)

5.WebElement:

=====

WebElement is predefined interface

findElement()
findElements()
sendKeys()
click()
getText()
getAttribute()
clear()
isDisplayed()
isSelected()
isEnabled()

6.What are all the locators present in selenium?

=====

id
name
ClassName

xpath
tagName
Css Selector
linkText
partialLinkText

7.what Xpath:

=====

xpath is one of the locator it is used to locate element from web page

Types:

=====

Absolute Xpath ---->element is taken from root tag to desired tag(/)

Relative Xpath ---->element is taken from desired tag(//)

8.Why we go for xpath?

=====

if id,name,className not present
to validate the locators

9.xpath syntax:

=====

attribute xpath -->//tagName[@attributeName='attValue']

xpath with index -->(//tagName[@attributeName='attValue'])[index]

text xpath -->//tagName[text()='textValue']

contains text xpath->//tagName[contains(text(),'partialText')]

contains attribute ->//tagName[contains(@attributeName,'partialAttValue')]

10.Actions:Class

=====

by using actions class we can handle mouse related & keyboard actions like mouse over,drag & drop,double click,right click

Actions objectRef = new Actions(WebDriver ref);

moveToElement() -->mouse over

dragAndDrop() -->drag & drop

contextClick() -->right click

doubleClick() -->double click

keyUp()

keyDown()

perform()

11.Robot:Class

=====

To perform keyboard related actions

it's a predefined class which is present in java.awt package
it will throw AWTException

keyPress()
keyRelease()

KeyEvent:

=====

VK_Keys -->virtual keys

12.Selenium Interfaces:

=====

WebDriver
WebElement
TakesScreenShot
JavaScriptExecutor
OutputType
TargetLocator
Alert
Navigation
Cell
Row
Sheet
Workbook

13.Alert:

=====

it's like a popup,it will show some message to user
we cannot inspect by switching the alert we can handle it

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

Types:

=====

- 1.Simple Alert -->it contains only ok button we can accept this alert
- 2.Confirm Alert -->it contains ok & cancel but either we can accept or dismiss this alert
- 3.Prompt Alert -->it contains textbox with ok & cancel button user can insert the value and accept or dismiss

Alert:I

=====

accept()
dismiss()
sendKeys()
getText()

14.Why we go for JavaScriptExecutor?

=====

JavaScriptExecutor is a predefined interface

executeScript()-m

-if sendKeys() and Click() methods are not working
-we can scrollDown and scrollUp operations

- it is fastest and easy to interact with hidden elements
- it is used to highlight the WebElements

15.what is frame:

=====

html document is embedded with another html document
security purpose

switching frame:

=====

```
driver.switchTo().frame(String id);
driver.switchTo().frame(String name);
driver.switchTo().frame(WebElement ref);
driver.switchTo().frame(int index);
```

16.diffrence b/w parentframe() & defaultContent()

=====

parentframe()

=====

return back to previous frame[child to parent frame]

defaultContent()

=====

return back to web page

17.How to handle drop downs(Select)?

=====

by using Select class we can handle dropdown

```
Select ref = new Select(WebElement ref);
```

Select:C

=====

```
selectByIndex()
selectByValue()
selectByVisibletext()
deselectByIndex()
deselectByValue()
deselectByVisibletext()
deselectAll()
getOptions()
getFirstSelectedOption()
getAllSelectedOptions()
isMutiple()
```

18.Difference b/w findElement() & findElements()?

=====

findElement()

=====

- it is used to find only one element from web page
- return type is WebElement
- if the specified element is not found it will throw NoSuchElementException Exception

findElements():

=====

- it is used to find morethan one elements from web page
- return type List<WebElement>
- if the specified element is not found it will return empty list

19.Difference b/w getWindowHandle() & getWindowHandles()

=====

getWindowHandle()

=====

- it is used to get parent window id
- return type is String

getWindowHandles()

=====

- it is used to get all windows id's including parent window id
- return type is Set<String>

20.Difference b/w quit() & close()

=====

quit()

=====

it will quit entire browser

close()

=====

it will close current focused window

1.What is mean by OOPs:

=====

Object Oriented Programming Structure

it is a method of implementation in which program is organised as collection class,methods & Objects

principles:

=====

class

method

object

inheritance

polymorphism

abstraction

encapsulation

Class:

=====

collection of objects and methods

In my project I have used lot of predefined classes like

- ChromeDriver
- FireFoxDriver
- InternetExplorerDriver
- Actions
- Robot
- Select

method:

=====

set of actions to be performed, here only we used to write business logics

In my project I have used lot of predefined methods like

- get()
- getTitle()
- getCurrentUrl()
- getText()
- getAttribute()

object:

=====

it is a runtime memory allocation

it is used to call the methods

ClassName objectName = new ClassName();

inheritance:

=====

Process of accessing one class property to another class with the help of extends keyword

why we go for inheritance (or) advantages:

=====

1. to reduce the memory wastage (to reduce no of object creations)

2. code re usable purpose

Types:

=====

single inheritance ----> combination of one parent class and one child class

multiple inheritance ---> combination of more than one parent class and one child class in parallel at time

multi level inheritance ---> combination of more than one parent class and one child class in tree level

hierarchical inheritance ----> combination of more than one child and one parent class

hybrid inheritance ----> it's combination multiple inheritance & hierarchical inheritance

multiple inheritance:

=====

multiple inheritance won't work in java by using extends keyword due to syntax error, priority problem, compile time error

by using interface concept we can achieve multiple inheritance.

polymorphism:

=====

executing the methods in more than one form (or) completing the same task in many ways

Types:

=====

method overloading

method overriding

Method Overloading (or) Static Binding (or) Compile time polymorphism

=====

in a same class and same method overload with different arguments

arguments depends on data type

arguments depends on data type count

arguments depends on data type order

In our project we are using println, waits

Method Overriding (or) Dynamic Binding (or) Run time polymorphism.

=====

if we are not satisfied with parent class business we can override a same method in child class and write

our own logics

in our project we are using user defined exception, get message(), getScreenshotAs() method.

Abstraction:

=====

Hiding the implementation details or method business logics

Types:

=====

partial abstraction (abstract class)

fully abstraction (Interface)

abstract class:

=====

it support both abstract method & non abstract method

we can't create object

by using extends keyword we can access methods from abstract class

In our project we are using BY. is an abstract class.

interface:

=====

it support only abstract method

we can't create object

by using implements keyword we access methods from interface

by default access specifier of methods in interface is public abstract

multiple inheritance won't work in java by using extends keyword due to syntax error, priority problem, compile time error

by using interface concept we can achieve multiple inheritance.

In our project we are using WebDriver

WebElement

TakesScreenshot

JavaScriptExecutor

OutputType

TargetLocator

Alert

List

set

map

String:

=====

-String is a predefined class which is present in java.lang package

-It is a set of character or collections of words enclosed within the double quotes

-String is index based one. It starts from 0 and end with n-1

Methods:

☒ length () – to find length of the string.

☒ charAt() – to pick a particular character in the string.

☒ toUpperCase() – convert string into uppercase.

☒ toLowerCase() – convert string into lowercase.

☒ replace() – to replace a particular character in the string.

☒ indexOf() – to find the index of a particular character in the string.

☒ lastIndexOf() – to find the last index of a particular character in the string.

☒ startsWith() – to check string starts with a particular character.

☒ endsWith() - to check string ends with a particular character.

☒ isEmpty() – to check whether the string is empty or not.

☒ isConcat() – to join two strings.

☒ equals() – to check two strings are equal or not.

☒ equalsIgnoreCase() - to check two strings are equal or not without case.

☒ trim() – to remove the unwanted spaces in the string.

☒ substring() – to print from any index to another index in the string.

String refName = " ";

Types

=====

Literal string

Non-literal string

Immutable string

mutable string

Literal string

=====

-It is stored in string pool/constant memory

-It shares the memory when strings are duplicate

String refName = " ";

Non-literal string

=====

-It is stored in heap memory

-It will create new memory even though strings are duplicate

String refName = new String(" ");

To take memory address

system.identityHashCode()

Immutable string

=====

-It is stored in string pool/constant memory

-It reduce memory wastages

-We can't modify the values in memory

Mutable string

=====

- It is stored in heap memory
- We can able to modify the values
- Joining operation is performed in same memory
- We are using string buffer and string builder

String buffer

=====

- Synchronize(allow user one by one)
- Thread safe
- less efficient

String builder

=====

- Asynchronize(allow user parallely)
- Not a thread safe
- More efficient

Array

=====

- Array is used to store a collection of similar datatypes in a single variable name
 - Array is index based one
 - Index starts from 0 and end with n-1
 - Array is fixed in size
 - Memory waste will be high since memory will be allocated at compile time
- Datatype variable[] = new Datatype[size];

Disadvantages

=====

- Memory wastage is very high(memory is allocated at compile time)
- It only supports similar data type

Collection:

=====

- Storing multiple values of dissimilar datatypes in a single variable name
- Memory wastage is low due to memory is allocated at runtime

List:

=====

- List is a predefined interface which is present in java.util package
- It is index based one
- List allow duplicates
- List prints in insertion order

Types

=====

Arraylist
Linked list
vector

Set:
==

- Set is a predefined interface which is present in java.util package
- Set is value based one
- Set doesn't allow duplicates
- Order of set is based on the class which we implemented

Types
====

HashSet - Random
LinkedHashSet - Insertion
TreeSet - Ascending order

Generics
=====

- Generic supports similar datatypes, inside the generics we need to pass wrapper class
- Classes of all datatypes is wrapper class

Map:
====

- Map is a key, value pair combination\
- Key doesn't allow duplicates
- Value allow duplicates

Types
====

HashTable - Random
HashMap - Random
LinkedHashMap - Insertion
TreeMap - Ascending order

HashTable
=====

- Synchronize(allow user one by one)
- Thread safe
- less efficient

HashMap
=====

- Asynchronize(allow user parallely)
- Not a thread safe
- More efficient
- ☒ HashMap- Random order, Key allows one null and values allow n null.
- ☒ LinkedHashMap – Insertion Order, Key allows one null and values allow n null.
- ☒ TreeMap – Ascending order, Key ignores null and values allow n null.
- ☒ HashTable – Random order, Key ignores null and values also ignore null.

Constructor:

=====

- it is a special method,when object is created to the class it get invoke automatically
- ClassName and Constructor Name should be Same
- Constructor doesn't have any return type
- purpose is to intialize the variable

Types:

=====

- 1.Default Constructor/Non Parameterized constructor
- 2.Parameterized Constructor

Constructor Chaining

=====

Process of Callign one constructor from another constructor with respectively current object creation

To acheive constructor chaining --->this & super

this() --->it used to call the current class constructor

super() --->it used to call the super class construtor

Types Of Variables:

=====

- 1.Local Variables
- 2.Instance Variables
- 3.Static Variables

Local Variable(method level variable)

=====

- 1.it is declared inside of method or block or constructor.
- 2.it's get activated when the control is enter into the method and it's get de-activated when the control leaving from method
- 3.it is stroed in stack memory
- 4.we cannot give any access specifier for this variable
- 5.we have to initialize the variable before it use,doesn't have any default value

Instance Variable(object level variable):

=====

- 1.it is declared inside of class and out of method or block or constructor.
- 2.it's get activated when the object is created it's get de-activated when the object is destroyed.
- 3.it is stored in heap memory
- 4.we can declare any access specifiers
- 5.we no need to intialize the value.it take default value automatically.

Static Variables(Class level variables):

=====

- 1.it is declared inside of class and out of method or block or constructor with static.
- 2.it's get activated when the control enter into class and it's get de-activated when the control leaving from class
- 3.it is stored in static memory.
- 4.we can declare any access specifiers

5. we no need to initialize the value. it take default value automatically.

Access Specifier:

=====

1. Private ---> it is class level access
2. default ---> it is package level access, with in a package by using extends keyword and object creation
3. protected -> samepackage(extends + object), different package only extends
4. public ---> global access samepackage(object+extends), differentpackage(object+extends)

Access Modifiers:

=====

1. abstract
2. static
3. final

abstract:

=====

1. if we are declaring abstract in class level we can't create object
2. if we are declaring abstract in method level we can't write business logics
3. we can't declare abstract keyword in variable level

static:

=====

1. we can't declare static keyword in class level, but nested class is possible
2. if we are declaring static keyword in method level we no need create object we can call static method directly, if we want access static method out side of class `ClassName.methodName()`, if we are using extends keyword then we can call the method directly
3. if we are declaring static keyword in variable level we no need create object we can call static variable directly, if we want access static variable out side of class `ClassName.variableName`, if we are using extends keyword then we can call the variable directly

final:

=====

1. if we are declaring class as final we can't inherit that particular class
2. if we are declaring method as final we can't override that particular method
3. if we are declaring variable as final we can't modify the value

Exception:

=====

it is like error, whenever it occurs in the program will be terminated that line itself

Checked Exception(Compiletime Exception) ----> before execution of program it will occur

=====

FileNotFoundException
IOException
AWTException
InterruptedException
ClassNotFoundException

Unchecked Exception(Runtime Exception) ---->after execution of program it will occur

=====

ArithmeticException
NullPointerException
IndexOutOfBoundsException
StringIndexOutOfBoundsException
ArrayIndexOutOfBoundsException
NumberFormatException
InputMismatchException

Exception Handling:

=====

by using try,catch & finally block we can handle the Exception

try ---->it's a block it contains only error code

catch ->it's a solution block,it contains solution for Exception,when the Exception is occurred it will get Execute

finally ->it's a success block,always run this block of code one time

Super Class of all the Exception is Throwable/Exception

throw

=====

1.it is a keyword which used to throw the Exception Explicitly

2.it is declared inside of method

3.at time we can throw only one Exception

throws

=====

1.it is a keyword which is used to handle the Exception

2.it is declared in the method level

3.at time we can able to handle morethan one Exceptions

User defined Exception:

=====

once i create the exception manually.in this exception i will pass the some kind of msg in this.

Then i will call the method from sysout.i will be over rided.

Encapsulation:

=====

1.wrapping up of data and code acting on a data is binding together in a single unit

2.structure of creating folders

3.example for Encapsulation is POJO

POJO:

=====

Plain Old Java Object

The class which contains only private variables, getters & setters and constructor

Exceptions in Selenium?

=====

ElementNotVisibleException --> The element being present in the DOM but it is not visible(cannot be interactive)

ElementNotSelectableException --> An element is disabled(can not be clicked/selected) inspite

of being present in DOM

NoSuchElementException --> WebDriver cannot able to find the particular attribute name from the DOM

NoSuchFrameException --> WebDriver attempts to switch to an invalid frame, which is unavailable

NoAlertPresentException --> Webdriver is trying to switch to an invalid alert, which is unavailable

NoSuchWindowException --> Webdriver is trying to switch to an invalid window, which is unavailable

StaleElementReferenceException --> The referenced element is no longer present on the DOM page

SessionNotFoundException --> Webdriver is acting immediately after 'quitting' the browser

TimeoutException --> The command didn't complete in the specified time

WebDriverException --> Webdriver is acting immediately after 'closing' the browser

Xpath:

-->Xpath is a XML path which is used to find locators on the webpage using DOM Structure.

-->There are two types of Xpath: Absolute which is denoted by single slash and Relative xpath which is denoted by double slash.

-->In that in our project we are using Relative xpath because we can directly find the element anywhere at the webpage but in Absolute xpath we have to find from the head of the DOM structure.

Actions:

-->Actions is a predefined class which is used for mouse over actions in a webpage. There are few methods as

-->moveToElement() --> used to do mouse over actions

-->doubleClick() --> used for double click

-->contextClick() --> used for right click

-->dragAndDrop() --> to move a webelement from one place to another

-->keyUp() --> for key release

-->keyDown() --> for key press

Robot:

-->Robot is a predefined class present in java.awt package which is used for performing keyboard actions in a webpage. There are few methods as

-->keyPress() --> used to press a key

-->keyRelease() --> used to release a key

Select:

-->Select is a predefined class which is used to perform dropdown in a webpage. There are few methods as

-->isMultiple() --> to verify whether dropdown is multiselect will returns true in boolean value

-->selectByIndex() --> to select an option in dropdown by using index

-->selectByValue() --> to select an option in dropdown by using value

-->selectByVisibleText() --> to select an option in dropdown by using visible text

-->getAllSelectedOptions() --> to get the selected options in dropdown

-->getFirstSelectedOptions() --> to get the first selected option in dropdown

-->deselectByValue() --> to deselect an option in dropdown by using index

-->deselectByIndex() --> to deselect an option in dropdown by using index

-->deselectByVisibleText() --> to deselect an option in dropdown by using index

-->deselectAll() --> to deselect all the selected option in dropdown

-->getOptions() --> to get all options present in dropdown using list

JavaScript Executor:

-->If locator is found in DOM Structure but still it shows NoSuchElementException due to hidden elements in the webpage, if sendkeys not working, if click not working and for Scrollup and Scroll down, we go for JavaScript Executor. It is an Interface.

There are few methods as

-->executeScript()

-->("arguments[0].setAttribute('value')",WebElementRef) --> sendkeys

-->("arguments[0].getAttribute()") --> get the particular value

-->("arguments[0].click()",WebElementRef) --> click

-->("arguments[0].scrollIntoView(true)",WebElementRef) --> scroll down

-->("arguments[0].scrollIntoView(false)",WebElementRef) --> scroll up

TakeScreenShot:

-->It is an Interface. It is used for capturing the webpage. This captured images will defaultly stored in temporary location.

We can store the captured image in our preferred location using CopyFile() Method present in FileUtils Class.

-->Here we have method called getScreenShotAs().

WebTable:

-->All the tables present in webpage is called webtable. First we need to fetch the rows using tagname "tr" and iterate the rows and then fetch the data using tagname "td" and then iterate the datas and pick a particular data from the webtable using if conditions and then fetch the data using tagname "th" and then iterate the headers.

-->There are two types -->Static - Data's are fixed.

-->Dynamic - Data's keeps changing

Windows Handling:

-->Consider a webpage has multiple windows example 10 windows, if any actions performed in 2nd window it will throw NoSuchElementException because the control will be still in the parent window (i.e)1st window. Now if you want to switch to the second window, windows Handling is used.

-->Method Used: driver.switchToWindow();

-->We can switch window by using --> String Id, String URL, String Title and There are few methods as

-->getWindowHandle() --> to get parent window ID

-->getWindowHandles() --> to get all windows ID which returns Set<WebElement> because Set does not allows duplicate.

Alert:

>Alert is a kind of popup window or popup message. We can't find locators for Alert, so to handle an alert we use a method called

driver.switchToAlert(); If we are not handling alert, we can't do any operations.

>Alert has 3 types-->

>Simple ALert --> accept();

>Confirm Alert --> accept(); dismiss();
>Prompt Alert --> sendKeys(); accept(); dismiss();

Frames:

>It is webpage embedded inside a webpage.
>If any locators are placed inside a particular frame then we need to switch in to that particular frame and need to access the locators.
>It is mainly for security purpose. We can switch to particular frame using ID, Name, Index and WebElement Reference.
> Methods Used: driver.switchToFrame();
>parentFrame(); --> switch to previous frame
>defaultContent(); --> switch to parent window

Waits:

>If locator is found but still NoSuchElementException is thrown due to webpage loading and wait problems, we go for Waits concept.

Two types

> Static Wait

Will wait for the maximum time given though the locator is found. Example: Thread.Sleep

> Dynamic Wait

Will not wait for the maximum time, if locator is found it will navigate to next step. There are two types here

>Implicit wait

Given common for all the locators

>Explicit wait

Given only for a particular locators. There are two types here

>WebDriver wait

Given only in terms of seconds.it can't handle Timeout Exception.

>Fluent wait

We can give time travel interms of Millisecond,macroseconds.it can handle Timeout Exception.

Implicit syntax:

```
driver.manage().timeouts().implicitlywait(5,Timeunit.SECONDS);
```

Explicit Syntax:

```
WebDriverWait w = new WebDriverWait(driver,5);
```

```
w.until(ExpectedConditions.visibility of ElementLocated(By.name("locator")));
```

fluentwait syntax;

```
Wait w = new
```

```
FluentWait(driver).withTimeout(Duration.ofseconds(10).pollingEvery(Duration.ofseconds(2)).ignoring(webDriverException.class));
```

```
w.until(ExpectedConditions.visibility of ElementLocated(By.name("locator")));
```