Class 4

**Type of method**

Default

Parametrised

**return type**

Void – Will not return

Will return – (any data type)

**with body are concrete methods**  
{

}

**without body (Abstract method)**

public void abc(); // We are not defining body there !

**OOPS concept**

**Abstraction** – hiding the internal information from the external world

When any requirements come to architect, then he/developer will not work on that directly. Java Architect will first define the functionality.

In Java,

Abstraction can be achieved by **2 ways**

Abstract class and Interface

Abstract class – there can be abstract method and concrete method

Interface – only abstract or without body

**100 % abstraction can be given by Interface.**

To make abstract class we use '**abstract**' keyword

eg:- public abstract class ClassB

In this abstract class we can define abstract methods and concrete methods

We cant create object of interface or abstract class.

To inherit:-

class – class --- extends

interface – class --- implements

interface – interface --- extends

We define with and without body method in abstract class and treat that class as parent and now we need to make another class child and need to inherit by keyword 'extends' and we need to give body to parent method which was without body.

And then we need to make object of child class and we can call child method and parent method both.

Interface:- It is like a class (It's a blueprint of the class), not fully class

public interface interface1

{

}

We define without body method in interface and treat that as parent and now we need to make another class child and need to inherit by keyword 'implements' and we need to give body to parent method which was without body.

And then we need to make object of child class and we can call child method and parent method both.

**Static keyword**

For this we need to create a variable a and we make a method and use a in that method because a is global as it is defined in class.

We will make a method to increment and then in main method we will create the object and call the increment method and it will display 1 and a is not initialized in class and by default global values have default value of 0.

Now, we will create another object for that same class and call that same method and again output will be 1.

As every object is getting separate memory, and that's why different object starts with 1 whenever any new object call this increment method. (**Memory will be allocated at object level**)

If we call this method with same object, then it will increment the value by 1 every-time we call the function.

Now we are making global variable static by **static** keyword.

static int a;

**Static means memory will be allocated at class level not object level.**

Now what it means is that if we call it n number of times let it be with any obj whether same or different, value will be incremented every-time when we call this method.

As, now memory will be allocated at class level by using static keyword not at object level.

Now, let suppose there is a static method

public static void display()

{

syso(“This is static method”)

}

When class loads, earlier we have studied that whenever a class object is created only then memory will be allocated but now in this case (when we define static methods or static variables), memory will be created as soon we define them, no need to create object in this case.

In case of static method call, we don't need to create object of class as we can call static method from main method by class\_name as main method is static too and static to static we can call by class\_name.

class\_name.method();

Like same if there is any static variable a then we can call that variable directly as well.

syso(class\_name.a);

**Benefit:- Memory utilisation**

Static method can use only static variable.

Non Static method can use both static and non static variables.

**Calling**

Static to Static (3 ways)

1. By object reference
2. By class name
3. direct call

Static to Non Static

1. By object reference

Non Static to Static

1. By object reference
2. By class name

**direct call is applicable when we are working in same class only.**

**parse int and Integer**

There is a data type int and there is a class corresponding to int which is Integer

float – Float

class name starts with Upper case (Integer and Float)

**This is wrapper class. (Corresponding class)**

Benefit of Wrapper Class

As in java, we need to create object of every class but while declaring int value we just give it directly int a;

no need to create a class.

int a = 12;

Otherwise it would have been

Integer a = new Integer(12);

Benefit

String s = “123”;

String is a class and int is data type and we want to convert String to int

For conversion, w need to use class Integer

int a = Integer.parseInt(s);

syso(“a”);

Now, a contains 123 (int)

Above, Integer is a Class and parseInt is a method and we are calling it like class\_name.method

As main is static and Integer.parseInt(s); is also static and static to static, we can call it by class\_name.

Wrapper class – class to class coversion

Implicit/Explicit – data types to data types

**Class 6**

**Double/Nested Loop**

Loop in loop.

**File Handling**

Data read from a .txt file.

**First of all, we have to make a connection with a file from which we need to read data by using:**

**File class.**

**File f = new File(“path of file”);**

In path make sure to use '**\\**'

to convert special meaning to normal meaning as there can be \n or \t (like special characters)

**Read data character by character by using FileReader class.**

**FileReader fr = new FileReader(f);**

file reader is dependent on file object.

and throws exception.

There is an ASCII value corresponding to every character.

ASCII value is a decimal value

decimal to binary conversion (Digital Electronics)

and then binary value is stored in system.

Now we use while loop as we don't know how many times this loop will execute.

**While ((int a = fr.read())!=-1)**

**-1 will indicate end of file.**

Throws exception.

Now we will convert int to char by explicit conversion.

**syso((char)a);**

And this will read character by character from .txt file.

If not conver to character then it will print decimal value or ASCII value.

**Now we will read line by line.**

**File f = new File(“path of file”);**

**FileReader fr = new FileReader(f);**

**BufferedReader br = new BufferedReader(fr);**

to read line by line we use BufferedReader Class like above and it is dependent on FileReader as file reader is giving us character by character and buffered reader is converting them in line.

String s;

**While((s = br.readLine())!=null)**

In case of string, null will indicate end of file.

**syso(s);**

This will print line by line.

**Write in .txt file**

Will start with file class to make a connection and this will create file when we want to write data in a file. If file is already present, then it will overwrite that file else will create a new file.

**File f = new File(“path of file”);**

Now in order to write in file we will use class FileWriter.

**FileWriter fw = new FileWriter(f);**

In order to print line by line and to pass new line within a file, we will use BufferedWriter class.

**BufferedWriter bw = new BufferedWriter(fw);**

Now, to write data we use method:- write

**bw.write(“My name is Jitender Ahuja”);**

**bw.newLine();**

**bw.write(“I am working for Goibibo”);**

**bw.close();**

And at last close the file by using method close.

If user, now changes the text and file is already present in system, then it will overwrite the file.

Throws exception, wherever it comes.

**Append in a file**

**File f = new File(“path of file”);**

**FileWriter fw = new FileWriter(f);**

**BufferedWriter bw = new BufferedWriter(fw);**

**bw.write(“My name is Jitender Ahuja”);**

**bw.newLine();**

**bw.write(“I am working for Goibibo”);**

**bw.close();**

In FileWriter class we need to use only **true**

And above will look like:-

**File f = new File(“path of file”);**

**FileWriter fw = new FileWriter(f, true);**

true indicates, we are going to append.

**BufferedWriter bw = new BufferedWriter(fw);**

**bw.write(“My name is Jitender Ahuja”);**

**bw.newLine();**

**bw.write(“I am working for Goibibo”);**

**bw.close();**

**Exception Handling**

Abnormal conditions which terminate the program are called Exceptions.

When that condition comes, control will be out of the program and program will be terminated and no code will execute after this type of condition.

Eg:-

**syso(“First Line”);**

**int a=5/0;**

**syso(“Second Line”);**

In the above program, output will only be First Line and after that **divide by zero** exception comes and program will get terminate and second line will not get print.

Exception is a class in java.

And to inherit class we use 'extends' keyword.

**And there are 2 types of classes**

Checked Exceptions

Unchecked Exceptions

Checked exceptions are called as Compile time exceptions.

When code is being written, it is compile time

Eg:- File not found Exceptions.

Unchecked exceptions are called as Run time exceptions.

When we run the code, it is run time.

Eg:- Arithmetic Exceptions, Null Pointer Exceptions

**How to handle exceptions**

**Using try catch**

**Using throws**

If we think at particular point in code, if an exception might comes, we can put that code in try block.

And catch block will handle the exception.

**try**

**{**

**int a = 5/0;**

**}**

**catch(Exception e)**

**{**

**syso(“Exception handled”);**

**}**

Try will throw an exception when it comes and catch block will handle that exception.

When an exception comes, try will throw an exception and catch block will handle and execute else catch block will not get executed.

**Basically,**

Try block will throw an object that will go inside e where e is reference variable and Exception is a class and try block is throwing an object.

To find the reason for exception, we can print:-

**catch(Exception e)**

**{**

**e.printStackTrace();** //Method

**}**

**Finally**

Exception is coming or not, finally will execute !

**try**

**{**

**int a = 5/0;**

**}**

**catch(Exception e)**

**{**

**syso(“Exception handled”);**

**}**

**finally**

**{**

**syso(“final block”);**

**}**

We can do that work with finally block like we want to close the connectivity with database or we want some resources to get free in last.

Basically, what we want that should happen no matter what , we can put that in finally block.

**We can not write try alone, there must be at-least catch or finally or both with try.**

**Q. What is the difference between final, finally and finalize ?**

**A. Final is use to make the variable constant, finally to free the resources and finalize is use for garbage collections.**

**Q. Difference between try catch and throws?**

**A. Try catch will handle only one type of exception at a time but throws can handle multiple exceptions at a time.**

**Q. Can we write multiple catch with try ?**

**A. Yes, but most specific to most general.**

**eg.**

**try**

**catch (ArithmeticException e1)**

**catch (ArrayIndexOutofBoundException e2)**

**catch (NullPointerException e3)**

**catch (Exception e)**

**Q. Can we put try catch inside try catch ?**

**A. Yes !**

**try**

**try**

**catch**

**catch**

What is the use of try catch inside a selenium

In case of failed scenario,we want screen-shot.

So, in that case, we will put all the code in try and screen-shot code in catch.

So, in this case what happens is if there is any failed scenario, exception comes and catch will take screen-shot and it get executed, otherwise program will execute normally without any failed scenario.

In normal terms, we can put even all the code in try block too. Wherever, we can think exception can comes just put that code in try block.

When we want to make our own exception.

Then we can use **throw** keyword.

In this, we throw the object of **myException**

try

{

**throw new myException(“message”);**

int a = 5/0;

}

catch(Exception e)

{

syso(“Exception handled”);

}

finally

{

syso(“final block”);

}

**Difference in throw and throws**

throw is basically for one type of exception at a time but with throws we can handle multiple exceptions at a time.

Class 7

**String Handling**

What is String ?

Anything in double quotes is a string.

String is a class in java.

String is a sequence of characters.

String is a immutable class. (which can't be modified)

There are 2 ways of object creation of String Class.

String s = new String (“abc”); // Memory will be allocated in heap in this case.

String s1 = “abc”; // Memory will be allocated in SCP in this case. (String Constant Pool)

**String s1 = “abc”;**

**String s2 = “abc”;**

Now in above example s1 object is created and point to abc in SCP. (abc is stored in SCP).

Now s2 object will not get created and will point to same abc in SCP. (because it is already present)

So, in SCP we can do memory utilization.

But String in case sensitive.

So When **String s3 = “Abc”;**

Now it will create another object s3 and that will point to Abc in SCP.

But in case of Heap whether string is same or not, new object will get created every-time.

**So, recommend using SCP way to create object of String Class.**

**String Comparison**

== (double equal)

It will compare the address. (Which object is pointing to that value, object is same or not)

Eg:-

String s1 = “abc”;

String s2 = “abc”;

Ans:- Address is same.

String s1 = “abc”;

String s2 = “Abc”;

Ans:- Address is not same.

String s1 = “abc”;

String s2 = new String (“abc”);

Ans:- Address is not same.

String s1 = new String (“abc”);

String s2 = new String (“abc”);

Ans:- Address is not same

**equals(), it will compare the value.**

S1.equals(s2)

just compare the values, does not matter if it is available in heap or SCP.

**compareTo(), it will compare the ASCII value.**

S1.compareTo(s2)

It will compare character by character wherever the first diff comes, it will give the difference between their ASCII values.

**concat(), it will append.**

String s1 = new String (“abc”);

s1.concat(“def”);

Answer is:- abc

Because String is a immutable class.

Now,

String s1 = new String (“abc”);

s1 = s1.concat(“def”);

Answer is:- abcdef

Now a new object s1 is created again, and now it will append.

String s2 = s1.concat(“def”);

print s2 = it will also work and answer is abcdef

**Traversing a string character by character**

for loop

String s1 = “Deepak”;

for (int i=0; i<s1.length(); i++)

{

syso(s1.charAt(i));

}

**Traversing a string word by word**

split, for loop

String s1 = “Deepak is my name”;

String[] s2 = s1.split(“ ”);

for (int i=0; i<s2.length; i++)

{

syso(s2[i]);

}

length is a variable in case of array of string and length() is a method in case of string.

**StringBuilder**

String Builder is a mutable class, which we can modify.

Not thread safe. It is fast as it is not thread safe.

StringBuilder sb = new StringBuilder(“abc”);

sb.append(“def”);

syso(sb);

Answer is:- abcdef

**StringBuffer**

String Buffer is also a mutable class, which we can modify.

Each method in StringBuffer is synchronized. It is thread safe.

**Excel Handling**

**xls format**-------JXL API

**xlsx format**-----Apache POI API

Apache POI API supports both xls and xlsx format.

**API**

built in methods and classes. Just we have to use.

Every API has its own corresponding jar files which contains built in methods and classes.

**Read from Excel(.xls)**

File f = new File(“Path”); //Connection with file

WorkBook wb = Workbook.getWorkbook(f); //Returning object of Workbook Class

Sheet ws = wb.getSheet(0); //Returning object of Sheet Class

int r = ws.getRows();

int c = ws.getColumns();

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

{

for (int j=0; j<c; j++)

{

Cell c1 = ws.getCell(j,i); //Cell Position

syso(c1.getContents()); //Read data of cell

}

}

**Class**

WorkBook

Sheet  
Cell

**Method**

getWorkbook()

getSheet()

getRows()

getColumns()

getCell()

getContents()

**Write into Excel(.xls)**

File f = new File(“Path”); //Connection with file

WritableWorkBook wb = Workbook.createWorkbook(f);

WritableSheet ws = wb.createSheet(“Jeet”, 0);

for (int i=0; i<5;i++)

{

for (int j=0; j<5; j++)

{

Label l = new Label(j, i, “Jeet”);

ws.addCell(l);

}

}

wb.write();

wb.close();

**Class**

WritableWorkBook

WritableSheet

Label (jxl.write)

**Method**

createWorkbook()

createSheet()

addCell()

write()

close()

Class 8

**Apache POI**

Apache POI is used for xlsx support.

Every API has its corresponding jar files.

Every jar files has its own dependencies

As we make Maven Project, we get pom.xml

In this pom.xml, if we do entry of any dependency, then its corresponding jar files will get downloaded automatically and other jar files as well which are interdependent to that jar.

Eg:-

Now, we will make a java project, and make a package in source and then class in package.

Now right click on project → configure → convert to Maven Project and finish

Then we will get pom.xml file.

Now go in pom.xml file,

Add dependency of jar files which we need, and maven itself maintain jar files and the benefit is if we need one jar and that jar is further using any other jars, so maven itself maintains all the jars on its own.

We just need to know the dependency of the jar files required.

In pom.xml file → we need to add tag 'dependencies' (open and close tag) both and in between we need to add dependencies and press control S.

After saving, we will get folder of Maven dependencies.

Maven maintain .m2 folder and in that all the jar files are maintained, which we can access directly by run command window and typing .m2

.m2 → repository → jar files

If any problem comes like;-

* if jar files are not downloaded properly, then we can delete .m2 folder and then go to project and right click and go to maven and update project.
* Pom.xml → right click and run as maven clean and maven test.

Why problem comes

In some companies, proxies are there and due to which jar files wont get download properly and we can check this by the size of jar files and it is 1kb and 2kb like, so these types of problems can come and above is the solution.

Or in worst case we have to maintain all the jar files in .m2 folder like written above.

Or else bypass proxy, eg:- where .m2 folder get created there we can maintain setting.xml file and bypass proxy.

**Maven maintain jar files of dependencies and build the project means compile or run the project.**

**Ques:- Maven build lifecycle ?**

**validate** - validate the project is correct and all necessary information is available

**compile** - compile the source code of the project

**test** - test the compiled source code using a suitable unit testing framework. These tests should not require the code be packaged or deployed

**package** - take the compiled code and package it in its distributable format, such as a JAR.

**verify** - run any checks on results of integration tests to ensure quality criteria are met

**install** - install the package into the local repository, for use as a dependency in other projects locally

**deploy** - done in the build environment, copies the final package to the remote repository for sharing with other developers and projects.

**Read from .xlsx file**

File f = **new** File("C:\\Users\\jitender.ahuja\\Desktop\\Latest.xlsx");

FileInputStream fi = **new** FileInputStream(f);

XSSFWorkbook xw = **new** XSSFWorkbook(fi);

XSSFSheet xs = xw.getSheetAt(0);

**int** r = xs.getPhysicalNumberOfRows();

**for**(**int** i=0; i<r; i++)

{

XSSFRow xr = xs.getRow(i);

**for**(**int** j=0; j<xr.getPhysicalNumberOfCells(); j++)

{

XSSFCell xc = xr.getCell(j);

System.***out***.println(xc.getStringCellValue());

}

}

**Class**

File //Connection

FileInputStream //dependent on File

XSSFWorkbook //dependent on FileInputStream

XSSFSheet //dependent on XSSFWorkbook

XSSFRow //dependent on XSSFSheet

XSSFCell //dependent on XSSFRow

**Methods**

getSheetAt() //Method of XSSFWorkbook

getPhysicalNumberOfRows() //Method of XSSFSheet

getRow() //Method of XSSFSheet

getPhysicalNumberOfCells() //Method of XSSFRow

getCell() //Method of XSSFRow

getStringCellValue() //Method of XSSFCell

**Benefit** :- Loop will execute as many times as many cells are there and if there is any cell as empty then it will get error.

Optimize way of reading, how many cells are there only till that time loop will execute.

**Write to .xlsx file**

File f = **new** File("C:\\Users\\jitender.ahuja\\Desktop\\Latest1.xlsx");

FileOutputStream fo = **new** FileOutputStream(f);

XSSFWorkbook xw = **new** XSSFWorkbook();

XSSFSheet xs = xw.createSheet("Jeet");

**for** (**int** i=0; i<5; i++)

{

XSSFRow xr = xs.createRow(i);

**for** (**int** j=0; j<5; j++)

{

XSSFCell xc = xr.createCell(j);

xc.setCellValue("Jitender");

}

}

xw.write(fo);

fo.flush();

fo.close();

File //Connection

FileOutputStream //dependent on File

XSSFWorkbook // **Not Dependent**

XSSFSheet //dependent on XSSFWorkbook

XSSFRow //dependent on XSSFSheet

XSSFCell //dependent on XSSFRow

**Methods**

createSheet() //Method of XSSFWorkbook

createRow() //Method of XSSFSheet

createCell() //Method of XSSFRow

setCellValue() //Method of XSSFCell

**To make connection between FileOutputStream and XSSFWorkbook**

xw.write(fo) //Method of XSSFWorkbook

fo.flush() //Method of FileOutputStream

fo.close() //Method of FileOutputStream

**Where fo:- object of file output stream**

**xw:- object of XSSFWorkbook**

Class 9

**Junit**

It's a unit testing tool which is used to run the classes and which do not have main method.

Developers also use it and the benefit of Junit is if we do all the code in main method, it becomes lengthy so in order to avoid this – we can write the code and run it with the help of Junit.

With the help of Junit – we can break the code and run in any sequence what we want.

It is totally based on annotations.

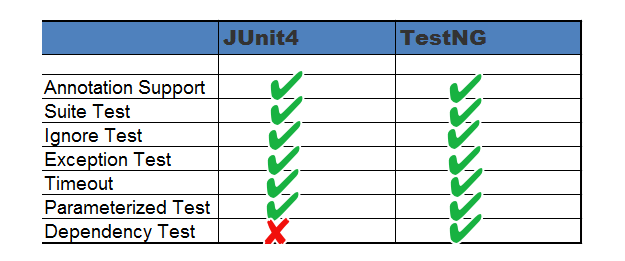
Annotations will be applied on the methods. Annotations are the metatag which defines the order of execution.

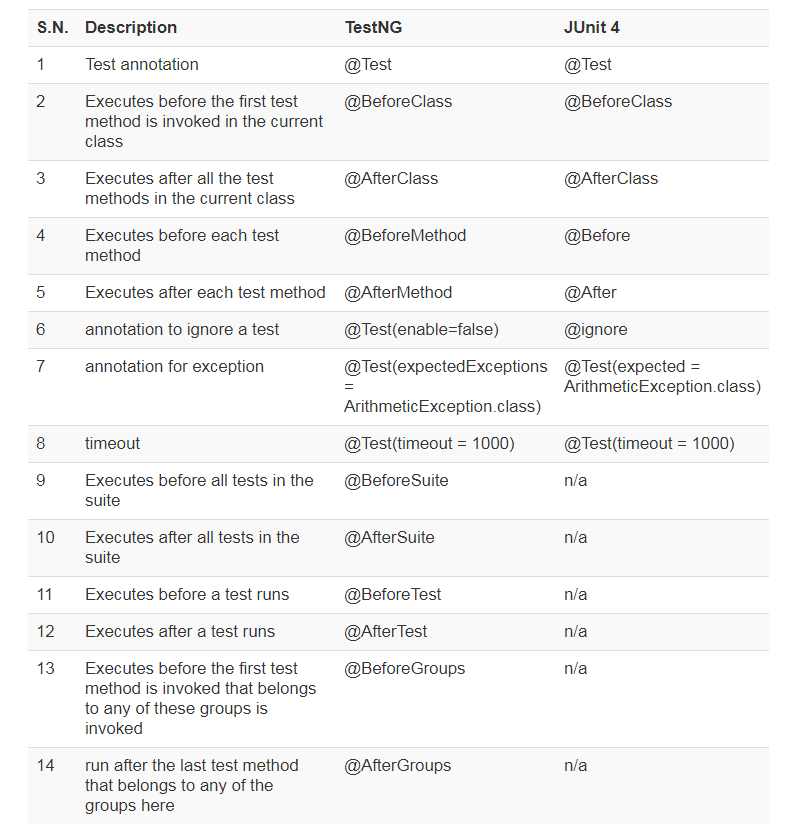
**Diff between Junit and Testng.**

Both Testng and Junit are Testing framework used for Unit Testing.

TestNG is similar to JUnit. Few more functionalities are added to it that makes TestNG more powerful than JUnit.

TestNG is a testing framework inspired by JUnit and Nunit.





Add these 2 jars to work with Junit

Selenium jars - 3.14 jar

old jars 2.48

Annotations will be applied over the method.

**@Test** is mandatory and is like main method, without it, program will not give any result.

**Annotations**

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

@Test --- mandatory annotation

@Before --- Before Test

@After --- After Test

@BeforeClass --- Before Class

@AfterClass --- After Class

@Ignore --- Ignore that case //write it over any @Test and it will then ignore that test to get execute

Before Class and After Class methods should be static methods.

Static gives memory at class level.

**Sequence**

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

BeforeClass

Before

Test //In alphabet order – execution takes place

After

Before

Test //In alphabet order – execution takes place

After

AfterClass

**Switch**

To run with multiple conditions.

**int** a = 10;

**int** b = 10;

**int** z;

**char** c = '\*';

**switch** (c) {

**case** '+':

z=a+b;

System.***out***.println("Addition is: "+z);

**break**;

**case** '-':

z=a-b;

System.***out***.println("Subtraction is: "+z);

**break**;

**case** '\*':

z=a\*b;

System.***out***.println("Multiplication is: "+z);

**break**;

**case** '/':

z=a/b;

System.***out***.println("Division is: "+z);

**break**;

**default**:

System.***out***.println("Invalid Opeartor");

**break**;

}

Class 10

**Selenium Introduction**

**Selenium** is a web based applicaion automation tool. It is a open source tool. It supports multiple languages(Java/python/c#). It supports multiple platforms like different browser (windows/linux). It supports android and ios automation.

Apium is the tool in the market for Android and ios automation, which use selenium as well in backend. (use one of the components)

**Disadvantages**

1. lots of programming required (there are some tools in which we need to learn any programming eg --> QTP, Katalon Studio etc. etc. (these work on record and play concept))

2. lots of configurations required

3. no support - when it comes initially

4. support only web based applications

**There are 4 components in Selenium**

1. Selenium IDE

2. Selenium RC

3. Selenium Webdriver

4. Selenium Grid

Selenium IDE

support only Firefox driver

there is plug in for record and play back.

Selenium RC

support multiple browsers

Need to start server everytime

code --> RC Server --> browser

statement to start server ---> java -jar RC Server jar file

browser switching was a problem there

Selenium Webdriver(3.14) Selenium 3

support android and ios

browser switching is handled in this

Apium, use web driver in backend

Selenium Grid

use for parallel executions (execution for parallel test cases)

**Interview question**

Webdriver is a interface or an API (built in methods and classes), and to inherit interface we use implements keyword.

Classes are implementing the webdriver interface.

Like - Chromedriver - selenium

Firefoxdriver - selenium

IEdriver - selenium

iosdriver - apium

androiddriver - apium

**Chromedriver driver = new Chromedriver();**

**Webdriver driver = new Chromedriver(); //upcasting**

**In General,**

Anything on the page is Webelement.

like to automate login page we will be having username , password and login button.

So, all the above mention 3 things are Webelement.

**How Selenium works**

First of all, we need to identify the web element **UNIQUELY** on the web page and then we can perform action on the same.

eg --> on username we need to pass some value (.send)

on button we need to click (.click)

**How to identify uniquely**:-

There are some element locating techniques

1. id

2. name

3. classname

4. css

4.1 tag with id --> tagname#idvalue

4.2 tag with classname --> tagname.classvalue

4.3 tag with attribute --> tagname[attributename='attributevalue']

4.4 tag with class with attribute --> tagname.classvalue[attributename='attributevalue']

4.5 tag with innertext --> tagname:contains('innertextvalue')

<label for ="email">**Email or Phone**</label>

**Email or Phone** --> innertextvalue

HTML has a top to down approach:-

Technique should be unique because it traverse and pick the first one and in case of not unique there might be issue in locating webelement.

Code --> Exe file of browser --> browser

Class 11

**Drop-Down**

If it is inside the select tag --- then use the **Select** class

If it is inside some other tag apart from select tag --- then we will use the keyboard key (arrow down) --- **Actions** class

Keyboard --- Any key --- **Actions** class

**3 ways to use drop down in Select class**

1. selectByIndex

2. select ByValue

3. selectByVisibleText

WebElement day = driver.findElement(By.*id*("day"));

Select s = **new** Select(day);

s.selectByIndex(6);

WebElement month = driver.findElement(By.*id*("month"));

Select s1 = **new** Select(month);

s1.selectByValue("8");

WebElement year = driver.findElement(By.*id*("year"));

Select s2 = **new** Select(year);

s2.selectByVisibleText("1990");

**Actions Class**

Actions a = **new** Actions(driver);

WebElement year = driver.findElement(By.*id*("year"));

year.click();

**for** (**int** i=0; i<3; i++)

{

a.sendKeys(Keys.***ARROW\_DOWN***);

}

a.sendKeys(Keys.***ENTER***).perform();

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

WebElement login = driver.findElement(By.*id*("loginbutton"));

Actions a = **new** Actions(driver);

a.keyDown(Keys.***CONTROL***).click(login).keyUp(Keys.***CONTROL***).perform();

Class 12

**Collection**

Collection is used for storing objects. It is a readymade framework given by java.

There are 3 interfaces in collection

1. Set ------ no duplicate allow

2. List ------ duplicate allowed, insertion order maintian

3. Map ------ key:value

To inherit interface, we use implements keyword.

**Set**

There are 3 classes which are implementing the set interface

1. HashSet --- not maintain the insertion order

**HashSet<Integer> hs = new HashSet<Integer>();**

2. LinkedHashSet --- insertion order maintain

**LinkedHashSet<Integer> lhs = new LinkedHashSet<Integer>();**

3. TreeSet -- Ascending order

**TreeSet<Integer> ts = new TreeSet<Integer>();**

**List**

There are 2 classes which are implemting the list interface

1. ArrayList -- searching fast on the basis of indexing, insertion deletion slow and tough

**ArrayList<Integer> al = new ArrayList<Integer>();**

2. LinkedList -- searching slow because of nodes, insertion deletion fast and easy

node - info (data) and linked part (address of next node) and last one will contain null

**LinkedList<Integer> ll = new LinkedList<Integer>();**

**Map**

There are 3 classes which are implementing the map interface

1. HashMap -- will not maintain the insertion order, and we can give duplicate keys but in this case latest value will come to that key. And also we can give duplicate values(that is ok)

**HashMap<Integer, String> hm = new HashMap<Integer, String>();**

**Set<Integer> all = hm.keySet();**

2. LinkedHashMap -- will maintain the insertion order, and we can give duplicate keys but in this case latest value will come to that key. And also we can give duplicate values(that is ok)

**LinkedHashMap<Integer, String> lhm = new LinkedHashMap<Integer, String>();**

**Set<Integer> all = lhm.keySet();**

3. TreeMap -- ascending order but as per key, and we can give duplicate keys but in this case latest value will come to that key. And also we can give duplicate values(that is ok)

**TreeMap<Integer, String> tm = new TreeMap<Integer, String>();**

**Set<Integer> all = tm.keySet();**

**Use of collection in selenium**

To Locate:-

single element -- findelement (Return type:- WebElement)

multiple elements -- findelements <List of WebElements which have same tag>

Example - Youtube

Class 13

**Modifier Key**

Shift, Control, Alt, Command, Option, and Function

Control + Click on Login Button from Action Class

modifier key -- keyup() and keydown()

non modifier -- sendkeys()

**Use of collection**

getWindowHandle -- give the address of current window

getWindowHandles -- give the address of all the tabs and windows

System.*setProperty*("webdriver.chrome.driver", "C:\\Users\\jitender.ahuja\\Downloads\\Jar Files\\chromedriver\_win32\\chromedriver.exe");

ChromeDriver driver = **new** ChromeDriver();

driver.get("https://www.facebook.com");

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

WebElement login = driver.findElement(By.*id*("loginbutton"));

Actions a = **new** Actions(driver);

a.keyDown(Keys.***CONTROL***).click(login).keyUp(Keys.***CONTROL***).perform();

System.***out***.println("Current page url: "+driver.getCurrentUrl());

String currentwindowaddress = driver.getWindowHandle();

System.***out***.println("Current Window Address: "+currentwindowaddress);

java.util.Set<String> allwindows = driver.getWindowHandles();

**int** i=0;

**for**(String s : allwindows)

{

i=i+1;

**if** (i==2)

{

driver.switchTo().window(s);

**break**;

}

}

System.***out***.println("After Switching: "+driver.getCurrentUrl());

driver.switchTo().window(currentwindowaddress);

System.***out***.println("Original Page: "+driver.getCurrentUrl());

driver.close();

**Verification Points**

System.*setProperty*("webdriver.chrome.driver", "C:\\Users\\jitender.ahuja\\Downloads\\Jar Files\\chromedriver\_win32\\chromedriver.exe");

ChromeDriver driver = **new** ChromeDriver();

driver.get("https://www.facebook.com");

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

WebElement signup = driver.findElement(By.*name*("websubmit"));

signup.click();

String a = signup.getText(); // Inner Text

System.***out***.println("Inner Text: "+a);

String b = signup.getAttribute("name"); // Attribute value

System.***out***.println("Attribute value of name: "+b);

System.***out***.println("Element is visible: "+signup.isDisplayed()); // Element is visible or not

System.***out***.println("Element is enabled: "+signup.isEnabled()); // Element is enable or not means click-able or not

System.***out***.println("Element is selected: "+signup.isSelected()); // Element is selected or not

driver.close();

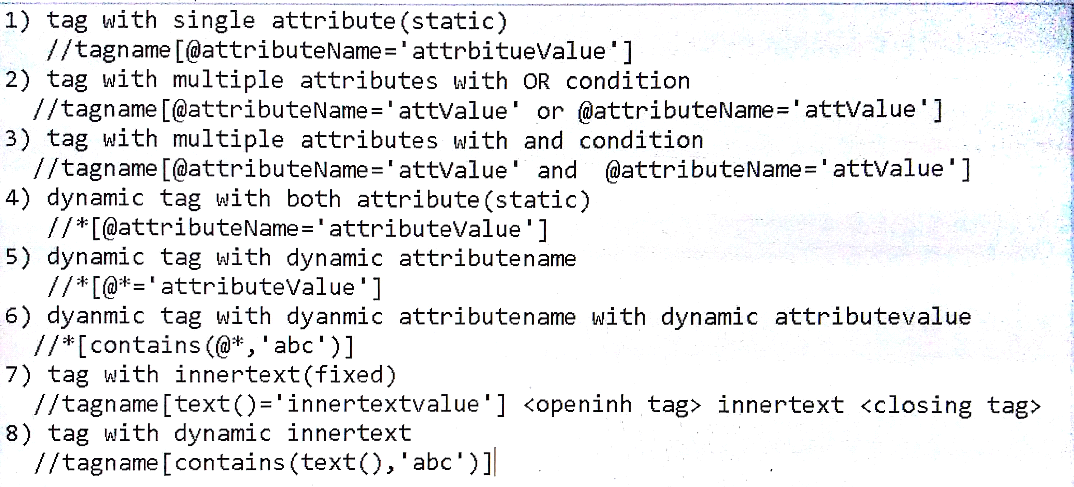
Class 14

**XPATH**

Dynamic - Change

Static - Fixed

**Techniques**



9) tag with dynamic attribute value

//tagname[contains(@attributeName,'abc')]

**Driver Navigation**

driver.get("url");

driver.navigate().to("url");

driver.navigate().back();

driver.navigate().forward();

**WAITS**

Wait is an interface.

**Normal wait**

Thread.sleep(1000); //Add throws declaration

// Will wait for next statement for the mentioned time no matter page is launched or not

**Implicit wait**

driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);

//Will wait for all the find elements below this implicit wait with a max of mentioned time and if find before time, it will go forward.

**Explicit wait**

Wait is an interface.

WebElement is an interface.

WedDriver is an interface.

We can make reference variable of interface but not object.

To inherit interface, we use "implements".

"Webdriverwait" is a class which implements interface "Wait".

Explicit wait is used for handling particular condition.

condition such as;-

text to be present

element to be clickable

//Will wait for elements under condition with a max of mentioned time and if find before time, it will go forward.

For Example:-

WebElement signin = driver.findElement(By.*xpath*("//yt-formatted-string[text()='Sign in']"));

WebDriverWait w = new webDriverWait(driver, 5);

w.until(ExpectedConditions.elementToBeClickable(signin))

**Fluent Wait**

Fluent wait is a part of the Explicit Wait type. Fluent wait instances define the maximum amount of time to wait for a condition, as well the frequency with which to check the condition.

**IFrame**

If on a page there are many windows and in this window if there is one button and if we need to click that button we can not locate that button, because at that time control is in main window.

So, now here comes the concept of IFrame and we need to switch the control of driver to the window where the button is.

eg:-

HTML be like,

<HTML>

<HEAD>

<TITLE>

<BODY>

<DIV>

<IMG>

<IFRAME ID = "XYZ">

<HTML>

<HEAD>

<TITLE>

<BODY>

<BUTTON ID = "ABC">

Sol is we need to locate the IFRAME first

Webdriver iframe = driver.findElement(By.id("iframe"));

driver.switchTo().frame(iframe)

Now locate the button and do normal work.

Now, to go the main window or frame

driver.switchTo().parentFrame();

To check how many frames are there in 1 window, we use

List <WebElemenst> all = driver.findElements(By.tagname("iframe"));

all.size();

**Interview Question**

Array of Integer -- Define Size

List of Array of webElements -- Define size is not required

Class 15

**TestNG** -- It is a unit testing tool which is used to run the classes which don't have main method.

It needs to be set up unlike junit.

It is based on annotations

Annotations will be applied on menthods

Annotations are the meta tag which defines the order of execution

@Test----it is mandatory

Order of Annotations execution

@BeforeSuite

@BeforeTest

@BeforeClass

@BeforeMethod

@Test

@AfterMethod

@AfterClass

@AfterTest

@AfterSuite

If 2 test cases are there, then order will be

@BeforeSuite

@BeforeTest

@BeforeClass

@BeforeMethod

@Test

@AfterMethod

@BeforeMethod

@Test

@AfterMethod

@AfterClass

@AfterTest

@AfterSuite

more annotations in testng

we can set the priority -- @Test(priority=0)

we can set the dependency -- @Test(dependsonMethods="")

Report feature -- refresh the project and get index file from testng folder in project

@Test(enabled=false) -- to not run the particular test case

Multiple classes run at the same time --> Project --> TestNG --> Convert to TestNG