How many ways to create a Thread

In two ways we can create a Thread.

① By extending Thread class

② By implementing Runnable interface

Creating a thread by extending Thread class:-

pseudo code of Thread class:-

```
public class Thread implements Runnable
{
    public static native Thread currentThread();
    public static native void yield();
    public static native void sleep(long params);
    public final void join();
    public void run();
}
```

Note:① Internally Thread class implements Runnable interface. to provide thread functionality.

Note:② Runnable (I) is functional interface.

pseudo code of Runnable interface :-

```
@FunctionalInterface
public abstract interface Runnable
{
    public abstract void run();
}
```

Note: ③: Multithreading is from "java.lang" package and which is the default package in Java.

Example: creating a thread using Thread class.

packar

Steps to follow

① Our class should extend Thread class

Runnable (I)
↓
Thread (c)
↓
ThreddDemo (c)

② Override run() to provide thread functionality.

By default Thread class implemented Runnable (I) with empty implementations.

③ call start() of Thread (c) to execute the thread.

Thread Schedular is responsible to execute the threads.

ex

```
package com.rameshsoft;

class ThreadDemo extends Thread
{
    @override
    public void run()
    {
      System.out.println("Java thread is started by child thrd");
      for(int i=0; i<10; i++)
        {
        System.out.println("Java thread");
        }//end of for loop
    }//run()
}//class

public class MultiThreadTest {
    public static void main(String[] args)
    {
        Thread thread = new Thread();
        thread.start();
        System.out.println("selenium thread is started");
        for(int j=0; j<10; j++)
          {
            system.out.println("Selenium Thread");
          }
    }// end of main()
}// end of class MultiThread Test
```

Defining a Thread

In the above program we have two threads.

① → main thread (JVM is the responsible to create main thread)

② → child thread (whenever we call thread.start() then a thread is going to be created and which is the responsible to execute child thread)

Note :- Whenever we call thread.start(), internally which calls run().

Thread Schedular :-

Thread Schedular is the component of JVM.

Thread schedular is the responsible to execute the threads based on priority of threads.

Thread priorities :-

In Java every thread has some priority. It may be default priority generated by JVM or provided by the programmer.

we can specify the Thread priorities by using the following constant provided by Thread (c)

* public static final int MAX_PRIORITY ---> (=10)
* public static final int MIN_PRIORITY ---> (=1)
* public static final int NORM_PRIORITY ---> (=5)

The valid range of thread priorities is 1 to 10

The default priority for main() thread is '5' and for remaing all threads the default priority will be inheriting from parent to child. And if you want we can set the priority by using setPriority(int a).

Syntax :-

```
public final void setPriority(int num)
{
  =
}

public final int getPriority()
{
 return this.priority;
}
```

* Even we can set and get the name of a threads as well.

Syntax :-

```
public final synchronized void setName(String threadName)
{
  =
}

public final String getName()
{
  =
}
```

Example:

```
package com.rameshsoft.multithreading;

class ThreadOne extends Thread {
  @Override
  public void run()
  {
    System.out.println("Java thread is started");
    for(int i=0; i<10; i++)
      {
        system.out.println("Java thread");
      }
    }//run()
}
public class MultiThreadTest {
public static void main(String[] args) {
    Thread t = new Thread();
    t.start();
    System.out.println("main thread default name-" + Thread.currentThread()
                                                    .getName());
    System.out.println("child thread default name " + t.getName());
    Syso("main thread default priority "+Thread.currentThread().getPrio-
                                                    -rity());
    Syso("child thread default priority "+ t.getPriority());
    t.setName("Ramesh");
    Syso("child thread name :" + t.getName());
    Thread.currentThread().setName("RameshSoft");
```

```
Syso("main thread name" + Thread.currentThread().getName());
  t.setPriority(8);
  Syso("customized priority " + t.getPriority());
  Syso("selenium thread is started");
  for(int i=0; i<10; i++)
  {
    Syso("selenium thread");
  }
 }// end of main()
}// end of MultiThreadTest
```

Creating thread by implementing Runnable (I)

We can create a thread by implementing Runnable (I) as well. and we need to override run() in the class.

* To start a thread we need to call start() of Thread (c) (as start() is not available in Runnable (I), we need to perform inter conversion).

```
ex:- MultiThreadDemo m = new MultiThreadDemo();
     Thread t = new Thread(m);
     t.start();
```

Example :-

```
package com.rameshsoft.multithreading;

public class MultiThreadOne implements Runnable {

    public void run()
    {
    Syso(" child thread is started");
     for(int i=0; i<10; i++)
        {
        Syso(" child thread");
        }
    }// end of run()
}//end of class

class MultiThreadTestOne
{
  public static void main(String[] args)
  {
    MultiThreadOne m=new MultiThreadOne();
    Thread t=new Thread(m);
     t.start(); ---------- ①
     Syso("main thread is started");

     for(int j=0; j<10; j++)
        {
        Syso(" main thread");
        }
     }
}// end of class
```

Conclusions :

(A) If we replace line —① with t.run() then no new thread will be created and that will be executed like a normal method.

(B) If we call with t.start() then only new thread will be created and which is responsible to execute the thread.

(C) If we don't override run() then thread class run() will be executed which is having empty implementations.

Thread class Constructors

① Thread t = new Thread();

② Thread t = new Thread(Runnable target);

③ Thread t = new Thread(String name)

How to prevent thread execution

we can prevent thread execution by using following methods

① yield()

syntax: public static native void yield();

② join();

syntax: public final synchronized void join();

③ sleep()

syntax: public static native void sleep(long msec);

yield():-

yield() causes to pause current executing thread for giving the chance of remaining waiting threads of same priority.

example:-

```
package com.rameshsoft.multithreading;
class MultiThreadTwo implements Runnable {
    public void run()
    {
     Syso("child thread is started");
     for(int i=0; i<10; i++)
        { Thread.yield();
        Syso("child thread");
        }
    }
}
```

Execution order: BeforeSuite

bmain()

amain()

gmain()

AfterSuite

In the above example, based on alphabetical order, execution starts at 'amain()', but this method depends on bmain(), so bmain() will be executed first and then amain() and gmain() are going to be executed.

Q: If we have 'priority' and 'dependsOn' attributes at testcase level, which will get more priority?

e.g.. @Test(priority = 0; dependsOnMethods = "bmain()")

```
public void amain()
{
≡
}
```

Ans: Even though priority is high, it depends on bmain(), so bmain() will be executed first and then amain().

INDIA'S NO1 REALTIME TRAINING INSTITUTE

RAMESHSOFT

100% REALTIME TRAINING

100% PLACEMENTS

TILL NOW

300+

STUDENTS GOT PLACED

TRAINER NAME : RAMESH ANAPATI

CERTIFIED AND REAL TIME EXPERT

Opposite Of Vindu Tiffin Center Between Canara Bank And Axis Bank 3rd Floor
Near Umesh Chandra Statue SR Nagar Sarala Apartments,HYDERABAD

PH : 9177791456, 9502695908, 040-48572456

Frameworks

- Frameworks in Real Time we will discuss in detailed in classroom.

- More than 30hrs will discuss.

  Note: This notes is not complete notes

'timeOut' attribute :-

By using 'timeOut', we can check performance of the test case.

* If the test case is not executed within the specified amount of time, then "TestNG" simply marks or makes that test case as failed.
* We need to specify 'timeOut' attribute as test case level i.e, at @Test annotation level.
* 'timeOut' attribute always takes integer as an argument
* we can specify 'timeOut' at 2 levels.
  1. Test case level (@Test)
  2. Suite level (testng.xml)

Test case level (Test level) :-

```
ex: @Test(timeOut=15)
    public void amain()
    {
    System.out.println("amain");
    }
```

Suite level :-

If we mention timeOut at suite level, then that timeOut is applicable for all the test cases available in that suite.

```
ex:  <suite name="Suite" timeOut="68">
       <test ---->
         <classes>
           <class> ----- />
           <class ----- />
         </classes>
       </test>
     </suite>
```

Grouping Mechanism in TestNG

* By using grouping, we can run the test cases at group level.
* In real time, we can use these concepts at suite level. in order to differentiate the test cases (Smoke Suite, Regression Suite..etc)
* We can specify "groups" in test case level by using "groups" attribute.
* We need to run these groups at testng.xml level.

```
ex: <groups>
      <run>
        <include name="rt" />
        <exclude name="st" />
      </run>
    </groups>
```

* If we want to run a group, then we need to write <include> tag in testng.xml

* This <include> contains name i.e here we need to specify name of the group, which we want to run.

ex: <include name="rt" /> or <include name="rt"></include>

* If you don't want to execute a particular group, then we need to write <exclude> tag in testng.xml

ex: <exclude name="st" />

example:- // GroupsDemo1.java

```
package com.rameshsoft.testng;

public class GroupsDemo1
{
@Test(groups="rt")
 public void m1()
 {
 System.out.println("m1() rt");
 }
@Test(groups="rt")
 public void m2()
 {
 System.out.println("m2() rt");
 }
```

```
@Test (groups = "st")
 public void m3()
 {
 System.out.println(" m3() st");
 }
}

 // GroupsDemo2.java

package com.rameshSoft.testng

public class GroupsDemo2
 {
@Test(groups = {" st " , " rt"})
 public void m4()
 {
System.out.println(" m4() st and rt");
 }
@Test (groups = "rt")
 public void m5()
 {
System.out.println(" m5() rt ");
 }
@Test(groups = " st")
 public void m6()
 {
 System.out.println(" m6 st"); } }
```

testng.xml:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE suite SYSTEM .... >
<suite name="suite">
  <test name="test">
    <groups>
      <run>
        <include name="rt" />
      <!-- <exclude name="st" /> -->
      </run>
    </groups>

    <classes>
      <class name="com.rameshsoft.testng.GroupsDemo1" />
      <class name="com.rameshsoft.testng.GroupsDemo2" />
    </classes>

  </test>
</suite>
```

Output: m3() rt
m2() rt
m4() st and rt
m5() rt

If we write <exclude name="st"/> in the above .xml code then output will be: m3() rt
m2() rt
m5() rt

TestNG with @parameters :-

parameterization means passing parameter or passing test data to the method.

In testng, we can achieve this by using @parameters annotations.

By using @parameters annotation, we can do the following things.

1. We can pass the test- data to the test case
2. we can perform cross-browser testing.

1. @parameters for test data :-

If we want to pass data to the test case, then we need to use @parameters annotation.

* This @parameters annotation always expects the parameters and for this parameters, we need to specify the values from testng.xml using <parameter> tag.

Example:

```
public class ParametersDemo
{
@parameters ({"username", "password"})
@Test
public void gmailLogin(String username, String password){
    {
```

```
WebDriver driver = new FirefoxDriver();
driver.manage().window().maximize();
driver.manage().deleteAllCookies();
driver.manage().timeouts().implicitlyWait(50, TimeUnit.SECONDS);
driver.get("https://www.gmail.com");
driver.findElement(By.id("Email")).sendkeys(username);
driver.findElement(By.id("next")).click();
driver.findElement(By.id("passwd")).sendkeys(password);
driver.findElement(By.id("signIn")).click();
driver.quit();
System.out.println("username is:" + username + "and password is:" +
password); }}
```

testing.xml :-

```
<? xml ------ >
<!DOCTYPE ------ >
<suite name = "suite" >
  <test name = "test" >
    <parameter name = "username" value = "rameshsoft.selenium@
                                          gmail.com" />
    <parameter name = "password" value = "123@rameshsoft" />
    ≡
  </test >
</suite >
```

Q:- Execution flow of testng annotations?

@BeforeSuite

@BeforeTest

@BeforeClass

@BeforeMethod

@parameters

@optional

@dataProvider

@Factory

@Test

- priority
- enabled
- groups
- timeOut
- dependsOnMethods
- dependsOnGroups

@AfterMethod

@AfterClass

@AfterTest

@AftorSuite

2.@parameters for Cross-browser testing:

By using @parameters annotation we can perform cross-browser testing.

* Cross-browser testing is a technique, which is used to test web application with different browsers.

ex:

```
package com.rameshsoft.testng;

public class CrossBrowserDemo
{
WebDriver driver;
@parameters ("browser")
@BeforeSuite
public void openBrowser(String browser)
{
if(browser.equalsIgnoreCase("firefox"))
  {
  driver = new FirefoxDriver();
  }elseif(browser.equalsIgnoreCase("chrome"))
  {
  System.setProperty("webdriver.chrome.driver", "c:\\users\\
                                          ....exe");
  driver = new chromeDriver();

  } elseif (browser.equalsIgnoreCase("ie")
  {
```

```
System.setProperty("webDriver.ie.driver", "c:\\users\\ - - - .exe");
                                            ↓
                                        path of IE

    driver=new InternetExplorerDriver();

    }
}
@AfterSuite
public void closeBrowser()
{
driver.quit();
}

@Test
public void script()
{
driver.manage().timeOuts().implicitlyWait(40, TimeUnit.SECONDS);

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

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

    }
}
```

testng.xml:

```
<? version - - - - >
<!DOCTYPE - - - - - >
<suite name = "suite" >
    <test name = "chromeTest">
      <parameter name="browser" value="chrome" />
        <classes>
```

```
<class name="com.rameshsoft.testng.CrossBrowserDemo" />
</classes>
</test>
<test name="ie">
<parameter name="browser" value="ie"/>
<classes>
<class name="com.rameshsoft.testng.CrossBrowserDemo" />
</classes>
</test>
<test name="firefox">
<parameter name="browser" value="firefox" />
<classes>
<class name="com.rameshsoft.testng.CrossBrowserDemo"/>
</classes>
</test>
</suite>
```


@Optional annotation in Testng:-

Basically, we can pass parameter values to the test case during runtime from testng.xml file by specifying parameters annotation to test case i.e @parameters.

To do this, we need to declare <parameter> tag in testng.xml file using "name" and "value" attributes, where the "name" attribute defines "name" of the parameter and "value" attribute refers value of the parameter.

If we don't define <parameter> in testng.xml file, then test method will receive the default value, which is specified inside the "@optional" annotation.

@optional annotation is going to be executed whenever the test method is not receiving parameter value from testng.xml file.

ex:

```
@parameters ("browser")
@Test
public void openBrowser (@optional ("chrome") String browser)
{
    ==== } statements;
}
```

@DataProvider annotation in TestNG

By using @DataProvider, we can pass the multiple sets of input data to the test case based on our requirement.

* @DataProvider is always going to return two-dimensional array.

Syntax:

```
public Object[][] dataProvidername
{
Object[][] object = new Object[ ][ ];
                          ↓    ↓
                        rows  columns
   =
return object;
}
```

* In dataProvider two-dimensional array. 'rows' indicates how many times you want to repeat the test case and 'columns' indicates number of inputs. (For how many web elements you want to pass the data).

* DataProvider name can be anything, but it should be meaningful.

* By using @DataProvider, we can pass multiple sets of input data.

* Here rows and columns are integer type values.

Example:

```
package com.rameshsoft.testng.DataProviderDemo

public class DataProviderDemo
 {
@Test (dataProvider="setdata")
public void gmailLogin (String username, String password)
{
System.out.println("username is:" + username);
System.out.println("password is:" + password);

 }
@DataProvider
public Object[][] setdata()
{
Object[][] object = new Object[2][2];
object[0][0] = "rameshsoft.selenium@gmail.com";
object[0][1] = "abc123";
object[1][0] = "rameshat914@gmail.com";
object[1][1] = "selenium";
return object;

 }
}
```

@Factory in TestNG:

@Factory allows tests to be created at runtime depending on certain data-sets or conditions.

* One of main advantage of using the Factory methods is that you can pass parameters to test Classes. while initializing them. These parameters can then be used across all the test methods present in the said classes.

Example: // FactoryAnnotationDemo.java

```
public class FactoryAnnotationDemo
 {
 @Factory
  public Object[] factoryMethod()
    {
      return new Object[]{ new Factory1("1"), new Factory1("2"),
                           new Factory1(), new Factory1("Hai", 10)
                         };
    }
public class Factory1
 {
   private String s=" ";
    private int a;
```

```
Factory1 ()
{
System.out.println ("default constructor");
}
Factory1 (String s)
{
this.s = s;
System.out.println ("parameterized (single) constructor "+this.s);
}
Factory1 (String s, int a)
{
this.s=s;
this.a=10;
System.out.println("parameterized (double) constructor "+this.s+
                   " and a is :"+ this.a);
}
@BeforeClass
void m1()
{
Systemout.println("Before class executing");
}
@Test
public void method()
{
System.out.println("Running test..."+this.s +" and "+this.a);
} }
```

RAMESHSOFT

SOFTWARE SOLUTIONS

OUR STUDENTS RECENTLY PLACED ON SELENIUM

AHMED

(SD SOFTECH PVT LTD)

RAFEEQ

(EDREEZ SOFTWARE PVT LTD)

MADHUPRIYA

(CIGNITI TECHNOLOGIES)

SWATHI

(COGNIZANT)

MRUDALA

(SOFTSOL TECHNOLOGIES)

GANESH

(PURPLETALK)

MANDEEP

(SOCTRONICS)

VISHWAJIT

(TCS)

SIBASYS

(COGNIZANT)

SRIKANTH.R

(INFOSYS)

SHRAVAN KUMAR

(IBM)

SANGAMESH

(TECH MAHINDRA)

PRAHALAD

(INFOBRAIN)

HARISH

(PRDC)

KIRAN

(CYBAZE)

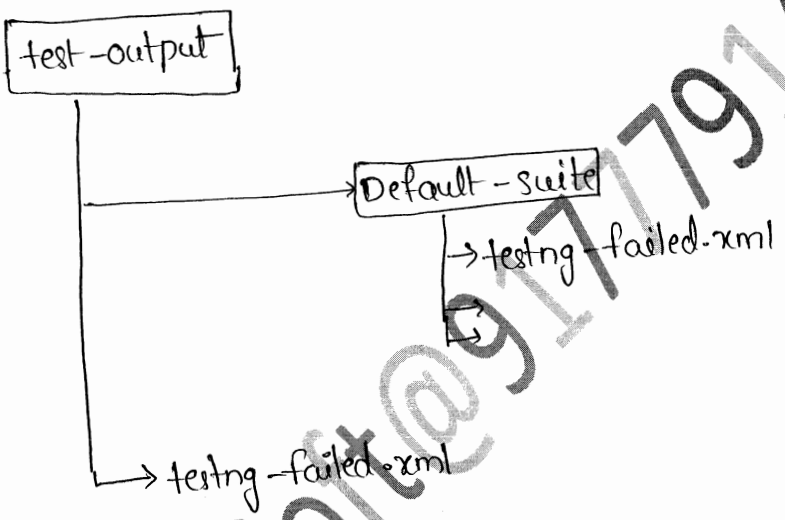
THAKIL

(UHG)

testng-failed.xml:

Whenever we run bulk of test cases(suite) using testng.xml, if any test case got failed, we can see that test cases in one ".xml" file called " testng-failed.xml", which is generated by TestNG.

path for testng-failed.xml:



alwaysRun attribute in TestNG:

* If alwaysRun set to true, this method always run even if it depends on another method that faild.

ex:
```
public class Demo
{
 @Test
  Public void method1(){
    System.out.println("method()");
  }
```

```
@Test(dependsOnMethods = {"method1"}, alwaysRun = true)
public void alwaysruntrue()
{
System.out.println("compulsory executes");
}
}
```

In this case alwaysRuntrue() will always run even if method() is pass or fail.

invocationCount attribute :-

* The number of times this method should be invoked
* We can execute the test method multiple times by specifying 'invocationCount'.

ex

```
@Test(invocationCount = 10)
public void method()
{
System.out.println("Welcome to RameshSoft");
}
```

This @Test will execute 10 time, as it is specified count as 10.

* If you don't specify invocationCount @Test will executes only once.

invocationTimeOut :-

The maximum number of milliseconds this test should take for the cumulated time of all the invocationCounts. This attribute will be ignored if invocationCount is not specified.

ex: @Test (invocationCount = 5, invocationTimeOut = 30)
public void m1()
{
System.out.println("Welcome to Rameshsoft");
}

threadPoolSize attribute :-

* The size of the thread pool for this method, i.e number of threads for the @Test method. The method will be invoked from multiple threads as specified by invocationCount.

* This attribute is ignored if invocationCount is not specified.

ex:- @Test (invocationCount = 5, threadPoolSize = "2", invocationTimeOut = 100)
public void m1()
{
System.out.println("Welcome to Rameshsoft");
System.out.println("Masters in Java with Selenium");
}

@Test will execute 5 times with 2 threads.

parallel attribute

* TestNG allows the tests to run in parallel or multi-threaded mode. This means based on the test suite configuration, different threads are started simultaneously and the test methods are executed in them.

* Parallelism can provide a lot of advantages to the users. They are

1. Reduces execution time: As tests are executed in parallel, multiple test get executed simultaneously, hence reducing the overall time taken to execute the test.

2. Allows multi-threaded tests: Using this feature, we can write tests to verify certain multi-threaded code in the applications.

There are different ways in which parallelism feature can be configured in TestNG.

① Running test methods in parallel:

* we can achieve this by writing parallel = "methods" in <suite> in testng.xml.

* This reduces the execution time.

Syntax/ex: <suite name = "Suite1" parallel = "methods" thread-count = "5" >

* TestNG will run all your test methods in separate threads. Dependent methods will also run in separate threads but they will respect the order that you specified.

② Running test classes in parallel:

* TestNG will run all the methods in the same class in the same thread, but each class will be run in a separate thread.

Ex: <suite name="Suite1" parallel="classes" thread-count="2"/>

③ Running tests inside a suite in parallel:

* Executing each test inside a suite in parallel, that is each test that is part of the test suite execution will be executed in its own separate respective thread.

* TestNG will run all the methods in the same <test> tag in the same thread, but each <test> tag will be in a separate thread. This allows you to group all your classes that are not thread safe in the same <test> and guarantee they will run in the same thread while taking advantage of TestNG using as many threads as possible. to run your test.

ex: <suite name="Suite1" parallel="tests" thread-count="3">

thread-count attribute:

* The default number of threads to use when running tests in parallel.
* This sets the default maximum number of threads to use for running tests in parallel.

* If will only take effect if the parallel mode has been selected.
* This can be overridden in the suite definition.

ex: <suite name="Suite1" parallel="classes" thread-count="3">

Example for parallel tests:

```
public class ParallelSuiteDemo
{
  String testName = " ";
   @BeforeSuite
   public void beforeSuite()
   {
   System.out.println("initializing suite");
   }
   @BeforeTest
   @parameters({"testName"})
    public void beforeTest(String testName)
    {
    this.testName = testName;
    long id = Thread.currentThread().getId();
    System.out.println("Before Test" + testName + ".Thread id is:" +
                                                        id);
    }
   @BeforeClass
    public void beforeClass(){
     long id = Thread.currentThread().getId();
     System.out.println("Before test--class"+testName+" Thread id:"+id);
                                                        }
```

```
@Test
public void testMethodOne()
{
long id = Thread.currentThread().getId();
System.out.println("test method1 " + testName + " Thread id is: " + id);
}

@AfterClass
public void afterClass()
{
long id = Thread.cwoorentThread().getId();
Syso("After class " + testName + " Thread id is: " + id);
}

@AfterTest
public void afterTest()
{
long id = Thread.currentThread().getId();
Syso("Aftertest " + testName + " Thread id is: " + id);
}
}
```

testng.xml :

```
<? xml ---------- >

<!DOCTYPE ---------- ---->

<suite name = "Suite1" Parallel = "tests" thread-count = "2" >

    <test name = "TestDemo1" >
```

```
<parameter  name = "testName"  value = "ChromeTest" />
    <classes>
        <class  name = "com.testng.ParallelInSuiteDemo" />
    </classes>
</test>
    <test name = "TestDemo2" >
    <parameter  name = "testName"  value = "FirefoxTest" />
        <classes>
            <class  name = "com.testng.ParallelInSuiteDemo" />
        </classes>
    </test> <!--- test -->
</suite> <!-- Suite -->
```

* "allow-return-values" attribute :-

Test methods are annotated with @Test. Methods annotated with @Test that happen to return a value will be ignored, unless you set "allow-return-values" to true in your testng.xml.

testng.xml :-

```
<suite allow-return-values = "true" >
    <test  allow-return-values = "true">
        <classes>
            <class  --- />
        </classes>  </test>  </suite>
```

Listeners in Selenium

* Listeners are interfaces used in Selenium Webdriver Script that modifies the default behaviour of the system.
* Listeners have ability to listen to a particular event.
* The main purpose of this Listeners is to creat logs and reports.
* There are basically two types of Listeners:
  1. WebDriver Listeners
  2. TestNG Listeners

TestNG Listeners

→ Listener is defined as interface that modifies the default TestNG's behaviour.

→ As the name suggests Listeners "listen" to the event defined in the selenium script and behave accordingly.

→ It is used in selenium by implementing Listeners Interface

→ It allows customizing TestNG report or logs.

→ TestNG Listeners help to change the default behaviour of the methods and write our own implementations when a Test fails or skips etc...

Types of Listeners in TestNG:-

1) IAnnotationTransformer,
2) IAnnotationTransformer2
3) IConfigurable,
4) IConfiguration Listener
5) IExecutionListener
6) IHookable
7) IInvokedMethod Listener
8) IInvokedMethod Listener2
9) IMethodInterceptor
10) IReporter
11) ISuiteListener
12) ITest Listener

ITestListener (I) :-

* It is an interface
* This can hold the states of our Test method like name of the Test method, whether Test method is passed or failed or skipped or not. All the information about Test method will be there as the part of this ITestListener.

pseudocode:-

```
public interface ITestListener
{
    void onTestStart (ITestResult result);
    void onTestSuccess (ITestResult result);
    void onTestFailure (ITestResult result);
    void onTestSkipped (ITestResult result);
    void onTestFailedButWithinSuccessPercentage(ITestResult res)
    void onStart (ITestContext context);
    void onFinish (ITestContext context);

}
```

→ onStart(-) : method is called when any Test starts

→ onTestSuccess(-) : method is called on the success of any Test

→ onTestFailure(-) : method is called on the failure of any Test

→ onSkipped(-) : method is called on skipped of any Test

→ onTestFailedButWithinSuccessPercentage(-) : method is called each time Test fails but is within success percentage

→ onFinish(-) : method is called after all Tests are executed.

example :-

First Let us create a class which implements ITestListener interface and add unimplemented methods into our class. And override the method according to our script.

```
import org.testng.ITestContext
import org.testng.ITestResult
import org.testng.ITestListener

public class ListenerTest implements ITestListener
{
    @override
      public void onTestStart(ITestResult result)
          {
          }
```

```
@override
public void onTestSuccess(ITestResult result)
{
  =
}
@override
public void onTestFailure(ITestResult result)
{
   try{
    ScreenShotUtility.screenshot(result.getName(),
                        DriverUtility.getDriver())
        }catch(IOException e)
            e.printStackTrace();
        }
   }
   @override
   public void onTestSkipped(ITestResult result)
   {
   }
   @override
    public void onTestFailedButWithinSuccessPercentage(
                                    ITestResult result)
     {
     =
     }
```

```
@override
  public void onStart(ITestContext context)
  {
  =
  }

  @override
   public void onFinish(ITestContext context)
   {
   =
   }

}//end of class
```

Now we will create WebDriver Utility class.

```
public class DriverUtility
{
 static WebDriver driver;
  public void openBrowser(String browser){
      if(browser.equalsIgnoreCase("firefox")){
       System.setproperty("webdriver.gecko.driver", "e://geckodriver.
                                                      exe");
        driver = new FirefoxDriver();
       driver.manage().window().maximize();
       driver.manage().timeouts().implicitlywait(20, TimeUnits.SECONDS);
       }
      elseif(browser.equalsIgnoreCase("chrome"))
        {
```

```
        System.setProperty ("webdriver.chrome.driver", "D:\\chromedriv-
                                                        -er.exe");
        driver = new ChromeDriver();
        driver.manage().timeouts().implicitlywait(19, TimeUnit.SECONDS);
    }
    else if (browser.equalsIgnoreCase("ie"))
    {
        System.setProperty("webdriver.ie.driver","E:\\IEDriverServer.
                                                        exe");
        driver = new InternetExplorerDriver();
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(19,TimeUnit.SECONDS);
    }
} //end of openBrowser()

    public static WebDriver getDriver()
    {
        return driver;
    }
} //end of class.
```

Now, we will write a ScreenShotUtility class for taking screenshot

we also can write methods in interface from Java 1.8

So, let us creat an interface.

```
import org.apache.commons.io.FileUtils;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;

import java.io.File;
import java.io.IOException;

public interface ScreenShotUtility
 {
   public static void screenshot(String imageName, WebDriver d)
                                          throws IOException
   {
  TakesScreenshot t = (TakesScreenshot)d;
    File file = t.getScreenshotAs(outputType.FILE);
    FileUtils.copyFile(file, new File("E:\\" +imageName+ ".jpeg"));
   }
}
```

Now let us creat our Test class.

```
public class TestDemo {
{
@Test
    public void login() {
      DriverUtility.openBrowser("chrome");
    DriverUtility.getDriver().get("https://www.gmail.com");
  WebElement username = DriverUtility.getDriver().findElement(
                        By.id("identifierId"));
                                  └→ id is wrong, here
                                     it will fail
      username.clear();
      username.sendkeys("rameshsoft.selenium");
    }
}//end of class.
```

Now Listener is ready, Test case is ready.

Now we need to configure Listener in testng.xml.

(we can also configure Listener at class level)

testng.xml:-

```
<?xml version = "1.0" ... ?>
<!DOCTYPE Suite SYSTEM " http://testng.org/testng-1.0.dtd ">
<suite name = "Suite">
<listeners>
  <listener class-name = "com.rameshsoft.Listeners.ListenerTest">
  </listener>
</listeners>
<test name = "Test">
  <classes>
    <class name = "com.rameshsoft.Listeners.TestDemo" />
  </classes>
</test>
</suite>
```

TestListenerAdapter class :-

TestListenerAdapter is a class, which is implemented ITestListener interface.

Ex:-

```
public class TestListenerAdapter implements ITestListener
{
=
}
```

Let us write a program to take a screenshot on failure.

For this we need write the below,

1. Our class should extend TestListenerAdaptor class.
2. Utility function for screenshot.
3. Utility functions related to webDriver like openBrowser()
4. Write the Test class for our business scenario.
5. configure listeners in testng.xml.
6. Run the testng.xml.

Pseudocode of TestListenerAdapter class :-

```
public class TestListenerAdapter
{
  public TestListenerAdapter()
  {
   =
  }
  public void onTestFailure(ITestResult res)
  {
   =
  }
  public void onTestSuccess(ITestResult res)
  {
   =
  }
  public void onTestSkipped(ITestResult res)
  {
   =
  }
  public List getFailedTest()
  {
   =
  }
=
}
```

Utility Interface for screenshot method :-

```
import java.io.File;
import java.io.IOException;
import org.apache.commons.io.FileUtils;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
    public interface ScreenShotUtility
    {
      public static void screenshot(WebDriver d, String name)
                                    throws IOException
      {
      TakesScreenshot ts = (TakesScreenshot)d;
   File file=ts.getScreenshotAs(OutputType.FILE);
  FileUtils.copyFile(file,new File("D:\\"+name+".jpeg"));
      }
} // end of interface
```

```
Public  class  DriverEngine
{
  public static  WebDriver driver;
  public void openBrowser()
   {
  System.setProperty("webdriver.chrome.driver", "D:\\chromedriver.
                                                    exe");
        Driver   = new ChromeDriver();
    }

    public void closeBrowser()
     {
      driver.quit();
     }
}
```

Now let us write a Test class.

```
public class FacebookTest {

    @Test
      public void main()
        {
          DriverEngine.openBrowser();
```

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

WebElement element = DriverEngine.driver.findElement(By.id("
                                                    emai"));
                                                    ↳ it will
    element.clear();                                fail here as id
                                                    is wrong
    element.sendkeys("rameshsoft.selenium");
  }
}//end of class.
```

Now write a class which extends TestListenerAdapter class.

```
public class ScreenShotTest extends TestListenerAdapter
  {
    @override
    public void onTestFailure(ITestResult iTestResult)
    {
      System.out.println("screenshot on failure");
      ScreenshotUtility.screenshot(DriverEngine.driver,
                                   iTestResult.getName())
      }
      catch(IOException e)
      {
        e.printStackTrace();
      }
    }
  }
```

testng.xml

```
<?xml version="1.0", encoding="UTF-8"?>
<!DOCTYPE Suite SYSTEM "http://testng.org/testng-1.0.dtd">
 <suite name="Suite">
 <listeners>
  <listener class class-name="com.rameshsoft.automation.
                                ScreenshotTest"/>
 </listeners>
 <test name="Test">
 <classes>
  <class name="com.rameshsoft.FacebookTest"></class>
 </classes>
 </test>
</suite>
```

ITestResult: is a Listener which is nothing but an interface.

→ It holds the information about Test methods.

pseudocode :—

```
public interface ITestResult
{
   // Test states
  int CREATED = -1;
  int Success = 1;
  int FAILURE = 2;
  int SkIP = 3;
  int SUCCESS_PERCENTAGE_FAILURE = 4;
  int STARTED = 16;

  int getStatus();
  void setStatus(int status);
  ITestNGMethod getMethod();
  Throwable getThrowable();
  void setThrowable(Throwable throwable);
  long getStartMillis();
}
```

example :-

```
public class ScreenShotDemo
{ @Test
 public void login ()
  {
  System.setProperty("webdriver.chrome.driver", "D:\\chrome-
                                                  driver.exe");

  WebDriver d = new Chromedriver();

  d.manage().timeouts().implicitlyWait(19, TimeUnit.SECONDS);

  d.get("https://www.gmail.com");

  d.findElement(By.id("identifierI")).sendKeys("ramesh-
                                                 -soft.selenium");
  }                                  ↳ id is wrong, it will fail
                                                        here
   @BeforeMethod
public void beforeMethod(Method m)
 {
  System.out.println("executing the Test");

 }
   @AfterMethod
   public void afterMethod(ITestResult result)
                                    throws IOException
```

```
{
if(ITestResult.FAILURE == result.getStatus())
  {
TakesScreenshot ts = (TakesScreenshot)d;
File file = ts.getScreenshotAs(outputType.FILE);

FileUtils.copyFile(file, new File("E:\\gmail"));
  }
}
}//end of class.
```


*Note:- We will discuss more elaboratively In class room

Files

1) Properties file/ Object Repository

* What is Object Repository?
* What is the use of Object Repository?
* How to create Object Repository?
* How to write data to the Object Repository?
* How to read data from the Object Repository?
* Real time usage of Object Repository?
* Real time reusable functions for Object Repository?
* Real time example using Object Repository?

2) Excel files

* What is Excel file?
* What is the use of Excel file?
* How to write data from the Excel file?
* How to Read data from the Excel file?
* Real time usage of excel file?
* Real time reusable functions for excel file?
* Real time example using Excel files?

3) Text files

* What is text file?
* What is the use of text file?
* How to write data to text file?
* How to read data from the text file?
* Real time usage of text files.
* Real time reusable functions for text files?
* Real time example using text files?

4) CSV Files:

* What is csv file?
* What is the use of csv file?
* How to write data to the csv file?
* How to read data from csv file?
* Real time usage of csv file

Properties File

* If we want to store the data in the form of key-value pairs. then we can go for properties file
* In this Properties file, key and values are by default String.
* If we want to represent a file as properties file, then compulsory we need to save that file with ".properties" extension.

Object Repository:

Object Repository is a place where we can maintain the webelement identification like xpaths, id, name, classname etc.

* If we want to convey a file as object repository, then compulsory we need to save that file with ".properties" extension.
* Object repository name can be anything, but recommended to give "OR.Properties" or "OR-Gmail.properties".

How to create ".properties file":

In two ways we can create properties file in Eclipse

1st Way:

Select the Project or package → new → other → general → file → next → provide file name as OR.properties → finish

2nd Way:

```
public class PropertyFileCreateDemo{
  public static void main(String[] args) throws IOException
  {
    Properties property = new Properties();
     property.setProperty("username", "rameshsoft.selenium@gmail.com");
     FileWriter fwriter =new FileWriter("or.properties");
       property.store(fwriter, "file created");
        fwriter.closec);
  } }
```

ex: //or.properties

username = rameshsoft.selenium@gmail.com

ele-uname = //*[@id='Email']

ele-next = //*[@id='next']

* How to write data to a Properties file

In order to write data to a properties file, first we need to have properties file.

Step1: Specify the location of a file

In order to specify the location of a file, java provided a predefined class called "File" and this "File" class is from "java.io".

* Again we need to pass the reference to a predefined class called FileInputStream class in order to specify the file location.

ex: File file = new File("specify the path of a properties file");

FileInputStream fip = new FileInputStream(file);

(or)

FileInputStream fip = new FileInputStream("specify the path of properties file");

step2: Create properties class object.

In order to read data from properties file, Java provided a predefined class called "Properties".

Pseudo code of Properties class:

```
public class Properties {
public void load(FileInputStream fip); → to load file in our java program
public void store(FileOutputStream fop); → to store data into properties file
public String getProperty(String key); → to get a properties from properties file.
public void setProperty(String key, String value); → to write data to a properties file.
}
```

Syntax: Properties properties = new Properties();

Step3: Load Properties file into java program

In order to load properties file into java program, properties class is having a method called 'load()'. This load() always asks you which file we want to load.

Syntax: public void load (FileInputStream fip);

ex: properties.load(fip);

whenever we call load(), this method loads your file into java programs.

step4: write or set the data into properties file.

In order to set the data (the data) to a properties file, Properties class has a method called "setProperty()".

Syntax: public void setProperty(String key, String value);

* This setProperty() always asks you "key" and "value" as arguments. we pass the "key" and "value" in the form of String.

ex: properties.setProperty("name", "RameshSoft");

Step5: Specify the location of a output file

In order to specify the location of the output file, java provides a predefined class called 'FileOutputStream'.

Ex: FileOutputStream fileOutputStream = new FileOutputStream(file);

Step6: Save the properties file.

After writing the data into properties file, we need to save that properties file. In order to save that file, properties class has a method called 'store()'.

This store() always ask you which file we want to store or save.

Ex: properties.store(fileOutputStream);

whenever we call store(), this method save your file.

How to read data from properties file:

In order to read data from object repository, we need to follow the following steps:

Please repeat the step1 to step3 from 'How to write data to a properties file' topic.

Step4: Read or get the data from properties.file.

In order to get the data from properties file, properties class has a method call 'getProperty()'.

Syntax: public String getProperty(String key)

This getProperty() always asks you 'key' as an argument. If you pass the key then it gives the corresponding value in the form of String.

ex: String gmail_username = properties.getProperty("username");

* Write a Program for writing data to a properties file:

```
public class ObjectRepoWriteDemo {
public static void main(String[] args) throws IOException, InterruptedException
{
Properties properties = new Properties();
FileInputStream fip = new FileInputStream("F:\\Selenium\\OR.properties");
properties.load(fip);
properties.setProperty("name", "RameshSoft");
properties.setProperty("course", "Selenium");
FileOutputStream fop = new FileOutputStream("F:\\Selenium\\OR.properties");
properties.store(fop, "Successfully added data into properties file");
```

(or)

```
properties.store(new FileOutputStream("F:\\Selenium\\OR.properties"));
} }
```

output: OR.properties

Successfully added data into properties file

name = RameshSoft

course = Selenium

* Write a Program for Reading data from properties file

```
public class ObjectRepoReadDemo {
  public static void main(String[] args) throws Exception {
    FileInputStream fip = new FileInputStream("F:\\Selenium\\OR.prope-
                                                -rties");
    Properties property = new Properties();
    property.load(fip);
  String gmail_Username = property.getProperty("username");
  String gmail_Next = property.getProperty("next");
  String gmail_Password = property.getProperty("password");
  String gmail_SignIn = property.getProperty("signIn");
  System.out.println("Gmail username xpath is: "+gmail_Username);
  System.out.println("Gmail next xpath is: "+gmail_Next);
  System.out.println("Gmail password xpath is: "+gmail_Password);
  System.out.println("Gmail paseword signIn xpath is: "+gmail_SignIn);

    WebDriver driver = new FirefoxDriver();
    driver.manage().window().maximize();
  driver.get("https://www.gmail.com");
  driver.findElement(By.xpath(gmail_Username)).sendKeys("rameshs-
                                                -oft.selenium@gmail.com");

  driver.findElement(By.xpath(gmail_Next)).click();
```

```
    Thread.sleep(2000);
' driver.findElement(By.xpath(gmail_Password)).sendkeys("123456");
  driver.findElement(By.xpath(gmail_signIn)).click();
     Thread.sleep(2000);
  driver.quit();
  } }
```

* Real time Reusable Funtions for Object Repository

```
// PropReusable.java
public class PropReusable {
 static Properties props = new Properties();
  String fileName;
  String value;
  public PropReusable(String fileName)
  {
    this.fileName = fileName;
  }
 public String getProperty(String key) throws IOException
  {
   File file = new File(fileName);
   FileInputStream fip = new FileInputStream(file);
     props.load(fip);
     value = props.getProperty(key);
     return value;
   }
```

```
public void setProperty(String key, String value) throws IOException
{
  File file = new File(fileName);
  FileInputStream fip = new FileInputStream(file);
  props.load(fip);
  props.setProperty(key, value);
  props.store(new FileOutputStream(fileName), null);
}
public void removeProperty(String key) throws IOException
{
File file =new File(fileName);
FileInputStream fip = new FileInputStream(file);
  props.load(file);
  props.remove(key);
  props.store(new FileOutputStream(fileName), null);
  }
}
```

Note: we can maintain object repository in .java files and .xml files also.

Excel Files:

Excel sheet is a place where we can maintain the test data in the form of rows and columns.

* This Excel file contains multiple sheets in order to differentiate the test data. We use Excel sheets in real time to maintain the Test data.

* In order to perform read and write operations on excel, we have one API called 'Apache POI API'. This API is from org.apache org-anization.

How to download Apache POI API:

Open Google → enter 'Download poi api' → click on the first link "Apache poi - download release Artifacts" → Binary Distribution → click on second link 'poi-bin-3.15-beta1-20160409.zip' → click on link "http://redro-ck.digimark.com/apachemirror/poi/dev/bin/---zip'.

How to integrate poi Api with eclipse:

Select the project → Right click → build path → configure build path → java build path → libraries → add external jars → add all the corre-sponding apache poi Api files → Ok.

Note: In order to read or write data from excel sheets, we have the following API's.

1) JXL 2) Apache POI API 3) FILLO

* How to write data to Excel sheet:

In order to write data to an Excel sheet we need to follow the follow-ing steps.

Step1: Add the apache POI jar files to eclipse

Step2: Specify the location of an excel file

In this, we need to specify the location of an excel file where you want to write the data and by using "FileInputStream" class we can specify the location of a file.

```
FileInputStream fip = new FileInputStream("c:\\abc.xls");
```

Step3: Create a Workbook

In order to create a Workbook apache poi api provided one predefined class called "WorkbookFactory" and this class has one non-static method called 'create()' method to create Workbook.

```
Workbook workbook = WorkbookFactory.create(fip);
```

Step4: Get the Sheet

In order to get the sheet we have a predefined method called getSheet(). in Workbook interface.

```
Sheet sheet = Workbook.getSheet("sheet1");
```

Step5: Create a Row

In order to create a row, we have predefined method called 'createR-

-ow()' in Sheet interface.

Syntax: public Row createRow(int rowNumber)

ex: Row row = sheet1.createRow(0);

step6: Create a cell

In order to create a cell, we have a predefined method called 'createCell()' in 'Row' interface.

Syntax: public Cell createCell(int cellNumber)

ex: Cell cell = row.createCell(0);

step 7: Set the values into a cell

In order to set the values into a cell, we have a method called 'setCellValue()' overloaded method.

ex: cell.setCellValue("RameshSoft");

row.createCell(1).setCellValue("Welcome");

step 8: Save the excel file

In order to save the data into excel file, 'Workbook' interface has a method called write() and this method always asks "FileOutputStream" as an argument.

ex: FileOutputStream fop = new FileOutputStream("c:\\selenium\\abc.xls");

workbook.write(fop);

* Write a Program for writing data to the Excel file.

```
public class WriteDataExcelDemo {
  public static void main(String[] args) throws Exception {
    FileInputStream fip = new FileInputStream("F:\\abc.xls");
    Workbook workbook = WorkbookFactory.create(fip);
    Sheet sheet1 = workbook.getSheet("sheet1");
    Row row0 = sheet1.createRow(0);
    Cell cell00 = row0.createCell(0);
    cell00.setCellValue("Welcome to RameshSoft");
    row0.createCell(1).setCellValue("Master in Java with selenium");
    Row row1 = sheet1.createRow(1);
    Cell cell10 = row1.createCell(0);
    cell10.setCellValue("Real Time Training Institute");
    FileOutputStream fileOutputStream = new FileOutputStream("F:\\abc.xls");
    workbook.write(fileOutputStream);
  }}
```

output:

Welcome to RameshSoft	Masters in Java with Selenium
Real Time Training Institute	

* How to read data from Excel sheets :

step1: Download apache POI API and add the Apache POI Jar files to eclipse, so that we can get the capability to access the API

step 2: Specify the location of the excel file

In order to specify the location of an excel file, java provided one predefined class called "FileInputStream' class.

- This FileInputStream class always asks you the location of excel file where it is.

ex: FileInputStream fip = new FileInputStream("c:\\Data.xls");

step 3: Create a workbook

In order to create a workbook, apache poi API provided a predefined class called 'WorkbookFactory' class.

psuedo code: public class WorkbookFactory

```
{
    public static Workbook create(FileInputStream fip)
    {
        ==
    }
    ==
}
```

→ This 'WorkbookFactory' class has a static method called create() to create the Workbook

→ This create() always asks you 'FileInputStream' as an argument i.e for which excel file you want to create a workbook.

→ Whenever we call create(), this method always returns 'Workbook' and this 'Workbook' is an interface.

Syntax: public static Workbook create(FileInputStream fip);

ex: Workbook workbook = WorkbookFactory.create(fip);

Step 4: Get Sheet

In order to get sheet, we have getSheet() in 'Workbook' interface. Whenever we call getSheet(), it always returns the corresponding Sheet object.

Methods available in Workbook interface:

→ public Sheet getSheet(String sheetname)

→ public Sheet getSheetAt(int arg);

→ public int getSheetIndex(String arg)

→ public String getSheetName(int arg)

ex: Sheet sheet1 = workbook.getSheet("sheet1");

Step 5: Get Row

In order to get a row, we have one method in "sheet" interface called "getRow();"

Whenever we call getRow() method, it always returns 'Row' object based on you parameter.

ex: Row row0 = sheet1.getRow(0); ∴ Row is an interface

step 6: Get cell

In order to get cell, Row interface has one method called 'getCell()'.

Syntax: public Cell getCell(int cellNumber)

whenever we call getCell() method it always returns corresponding cell object

ex: Cell cell00 = row0.getCell(0); ∵ Cell is an abstract class

Cell cell01 = row0.getCell(1);

≡

step 7: Get the data from cells

In order to get the data from cells, "cell" abstract class has the following methods.

→ public String getStringCellValue()

→ public String getNumericCellValue()

→ public boolean getBooleanCellValue()

* How to Read Excel Sheet or write a program for Reading Exceldata

```
public class ReadExcelData{
 public static void main(String[] args) throws Exception {
 FileInputStream fip = new FileInputStream("F:\\abc.xls");
  Workbook workbook = WorkbookFactory.create(fip);
  Sheet sheet1 = workbook.getSheet("sheet1");
    Row row0 = sheet1.getRow(0);
```

```
Cell cell00 = row0.getCell(0);
String cellValue00 = cell00.getStringCellValue();
System.out.println("Cell 00 value is: "+ cellValue00);
Cell cell01 = row0.getCell(1);
String CellValue01 = cell01.getString CellValue();
System.out.println("cellvalue 01 is :"+ cellValue01);
Row row1 = sheet1.getRow(1);
String cellValue10 = row1.getCell(0).getStringCellValue();
System.out.println("cell value10 is :"+ cellValue10);
}
}
```

getLastRowNum(): This method is available in 'sheet' interface.

By using this method, we can get the total number of rows present in a sheet.

Syntax: public int getLastRowNum()

* This method always returns row count in the form of integer.

getLastCellNum(): This method is available in 'Row' interface.

* By using getLastCellNum(), we can get the total number of cells or columns present in a row in the form of integer.

Syntax: public int getLastCellNum()

Program for how to read excel data by using cell validations

```
public class ReadExcelDataDemo {
public static void main (String[] args) throws Exception {
FileInputStream fip = new FileInputStream ("F:/ABC.xls");
Workbook workbook = WorkbookFactory.create(fip);
Sheet sheet1 = workbook.getSheet("sheet1");
    //this loop for rows iteration.
for(int i=0; i<=sheet1.getLastRowNum(); i++)
  {
   Row row = sheet1.getRow(i);
    //this loop for columns/cells iteration
     for(int j=0; j<row.getLastCellNum(); j++)
      {
       Cell cell = row.getCell(j);
       if(cell.getCellType()==cell.CELL_TYPE_NUMBER)
        {
        double d = cell.getNumericCellValue();
        System.out.println(d);
        } else if (cell.getCellType()==cell.CELL_TYPE_STRING)
         {
         String s = cell.getStringCellValue();
          System.out.println(s);
         } else {
          boolean b = cell.getBooleanCellValue();
          System.out.println(b);
         }
  } } } }
```

*Reusable functions for excel files

```
public class ExcelReader {
 String excelFilePath;
 Workbook workbook;
 String SheetName;
 public ExcelReader ( ) {  }
 public ExcelReader (String excelFilePath) throws InvalidFormatException
 {
  this.excelFilePath = excelFilePath;
  workbook = WorkbookFactory.create (new FileInputStream (excelFilePath));
 }
 public int getRowCount(String sheetName) {
  int rowCount=0;
          sheetName = sheetName;
  rowCount=Workbook.getSheet(sheetName).getLastRowNum()+1;
  return rowCount;
 }
 public String getCellData ( String sheetName, int rowNumber, int columnNum)
 {
 String cellValue="";
 if (workbook.getSheet(sheetName).getRow(rowNumber).getCell(columnNumber)
    .getCellType()= =workbook.getSheet(sheetName).getRow(rowNumber).
       getCell(columnNumber).CELL_TYPE_STRING)
   { cellValue = workbook.getSheet(sheetName).getRow(rowNumber).getCell(
          columnNumber).get StringCellValue ().trim(); }
```

```
else {
cellValue = String.valueOf ((int) (workbook.getSheet(sheetName).getRow(rowNu-
        -mber).getCell(columnNumber).getNumericCellValue().trim(); }
  return cellValue;
} }
```

XSSF and HSSF (From apache POI API)

HSSF :-

→ It stands for "Horrible spread sheet Format".

→ This class is from apache POI API.

→ By using this class, we can read and write microsoft excel format files i.e we can read and write ".xls" format files.

XSSF :

→ It stands for "xml spread sheet format".

→ This class is from "apache POI API".

→ By using this "XSSF" class, we can read and write microsoft excel files. i.e we can read and write ".xlsx" format files.

How to read data from HSSF :-

If you want to get total number of rows, we have one method called 'rowIterator()', which is available in "HSSFsheet".

Syntax: public Iterator rowIterator()

If you want to iterate the cells, we have one method called 'cellIterator()', which is available in "HSSFSheet".

Syntax: public Iterator cellIterator()

Program for how to write data to an excel sheet using HSSF:

```
public class XlsWriteDemo {
 public static void main(String[] args) {
  String excelFileName = "c:\RameshSoft\test.xls";
  String sheetName = "sheet1";
 HSSFWorkbook workbook = new HSSFWorkbook();
 HSSFSheet sheet = workbook.createSheet(sheetName);
  for(int r=0; r<5; r++)
   {
    HSSFRow row = sheet.createRow(r);
        for(int c=0; c<5; c++)
         {
          HSSFCell cell = row.createCell(c);
          cell.setCellValue("cell "+r+" "+c);
         }
 FileOutputStream fileOut = new FileOutputStream(excelFileName);
        workbook.write(fileOut);
        fileOut.flush();
        fileOut.close();
}
```

* Program for how to read data from .xls file using HSSF

```
public class XlsRead {
public class static void main(String[] args){

InputStream excelFileToRead = new FileInputStream("C:/Test.xls");
HSSFWorkbook workbook = new HSSFWorkbook(excelFileToRead);

HSSFSheet sheet = workbook.getSheetAt(0);

HSSFRow row;

HSSFCell cell;

Iterator rows = sheet.rowIterator();
while (rows.hasNext())                    row = (HSSFRow) rows.next();
{                                         Iterator cells = row.cellIterator();
                                          while(cells.hasNext()){
cell = (HSSFCell) cells.next();

if(cell.getCellType() == HSSFCell.CELL_TYPE_STRING){

System.out.println(cell.getStringCellValue()+" ");
} else if ( cell.getCellType() == HSSFCell.CELL_TYPE_NUMERIC)
    {
    System.out.println(cell.getNumericCellValue()+" ");
    }
} } } }
```

* Program for how to write data to '.xlsx' file using XSSF

```
public class XlsxWrite {
public static void main(String[] args) {
String excelFileName = "c:/Test.xlsx";
String sheetName = "sheet1";
XSSFWorkbook workbook = new XSSFWorkbook();
XSSFSheet sheet = workbook.createSheet(sheetName);
for(int r=0; r<5; r++)
{
XSSFRow row = sheet.createRow(r);
for(int c=0; c<5; c++)
{
XSSFCell cell = row.createCell(c);
cell.setCellValue("cell"+r+" "+c);
}
}
FileOutputStream fileOut = new FileOutputStream(excelFileName);
workbook.write(fileOut);
fileOut.flush();
fileOut.close();
}}
```

* Program for how to read data from '.xlsx' file using XSSF

```
public class XlsxRead {
public static void main(String[] args) {
InputStream excelFileToRead = new FileInputStream("c:/Test.xlsx");
XSSFWorkbook workbook=new XSSFWorkbook(excelFileToRead);
XSSFSheet sheet = workbook.getSheetAs(0);
```

```
    XSSFRow  row;
    XSSFCell  cell;
Iterator  rows = sheet.rowIterator();
while(rows.hasNext())
 {
row=(XSSFRow)rows.next();
 Iterator cells=row.cellIterator();
     while (cells.hasNext())
     {
     if(cell.getCellType()==XSSFCell.CELL_TYPE_STRING)
     {
     System.out.println(cell.getStringCellValue()+" ");
      } else if (cell.getCellType()==XSSFCell.CELL_TYPE_NUMERIC)
       {
      System.out.println(cell.getNumericCellValue()+" ");
       } }
   }
}
```

* Configuration file:

Configuration file is a properties file and if we want to convey a file as configuration file we need to save that file with '.properties' extension.

Name can be anything but in real time standards we will give it as config.properties.

In Configuration files(config.properties) we maintain environmental variables

like urls usernames password etc....

In order to perform any kind of operations like read and write data operations on this configuration file, we can perform like normal properties file way only.

ex: config.properties

```
#QA1
URL = https://www.rameshsoft.com
username = rameshsoft
password = selenium
ScreenShotOnFailure = true
reportsOnFailure = true

#QA2
URL = https://rameshselenium.blogspot.in/
username = rameshsoft
password = selenium
ScreenShotOnFailure = true
reportsOnFailure = true
```

* Automate a Gmail login Test case.

Test case steps:

1. Open a Firefox Browser
2. enter url as www.gmail.com
3. Enter username and password
4. Get the page title.
5. close browser.

Rules

→ take url from config.properties

→ take username and password from Excel sheet

→ take web elements from (GmailLogin.properties) properties file

Program:

```
public class GmailLoginDemo{
public static void main(String[] args) throws Exception {
 // reading url from config.properties
   File file1 = new File("F:\\Desktop\\config.properties");
   FileInputStream fip = new FileInputStream(file1);
    Properties p1= new Properties();
     p1.load(fip);
   String url = p1.getProperty("gmail_url");
    System.out.println("Gmail url is:"+ url);
```

```
// taking web elements from

File file2 = new File("c:\\Desktop\\Gmail_OR.properties");

FileInputStream fip2 = new FileInputStream(file2);

Properties p2 = new Properties();

p2.load(fip2);

String username = p2.getProperty("gmail_username");

String next button = p2.getProperty("gmail_next");

String password = p2.getProperty("gmail_password");

String signIn = p2.getProperty("gmail_signIn");

// taking or reading test data from excel sheet

File file3 = new File("c:\\Desktop\\testdata.xls");

FileInputStream fip3 = new FileInputStream(file3);

Workbook workbook = WorkbookFactory.create(fip3);

String username data = workbook.getSheet("sheet1").getRow(0).getCell(0).
                       .getStringCellValue();

String password data = workbook.getSheet("sheet1").getRow(0).getCell(1).
                       getStringCellValue();

WebDriver driver = new FirefoxDriver();

driver.manage().timeouts().implicitlyWait(50, TimeUnit.SECONDS);
```

```
driver.get(url);
driver.findElement(By.xpath(gmail_username).sendkeys(usernamedata);
driver.findElement(By.cssSelector(gmail_next)).click();
Thread.sleep(2000);
driver.findElement(By.id(gmail_password)).sendkeys(passworddata);
driver.findElement(By.xpath(gmail_signIn)).click();
String pageTitle = driver.getTitle();
System.out.println("Page Title is:" + pageTitle);
}
}
```

Text files:

Any file with ".txt" extension is called as text file.

* Text files are used to store the data or some information
* In olden days, these files are used for database management.
* If we want to create a text file, then we need to save that file with '.txt' extension.

How to write data to a text file:

Step1: Create a file

* In order to write data to a file, first we need to create a file.
* In order to create a file, java provided one predefined class called 'File' and this class is from 'java.io' package.

'File' class has one non-static method called 'createNewFile()' in order to create the file.

Syntax: public boolean createNewFile()

* Whenever we call createNewFile(), it wont create a file directly First it is going to check whether there is any with the same name or not and if the file exists, it wont create new file, but if the file doesn't exist, it creates new file and returns true.

ex: File file=new File("c:\\users\\abc.txt");

file.createNewFile();

Step2: Create FileWriter and and BufferedWriter objects

In order to write data to a file, java provided two predefined

called 'FileWriter' and 'BufferedWriter'.

```
FileWriter fileWriter = new FileWriter(file);

BufferedWriter bwriter = new BufferedWriter(fileWriter);
```

step 3: Write data to a file

"BufferedWriter" class is having the overloaded write() methods in order to write data to file.

ex:
```
write(String str)

write(int a);

write(char c);

=
```

step 4: Save the data into file

In order to save the data permanently or store the data permanently we need to call 'flush()' method, which is available in BufferedWriter class.

Syntax: public void flush()

whenever we call flush(), then the total data will be moved from buffer to text file permanently.

* Program for how to write data to a text file.

```
public class WriteTextData {
  public static void main(String[] args) throws IOException {

    File file = new File("F:/selenium/hello.txt");

      boolean b = file.createNewFile();
```

```
        System.out.println(b);
    FileWriter fwriter = new FileWriter(file, true);
    BufferedWriter bwriter = new BufferedWriter(fwriter);
     bwriter.write("Welcome to Rameshsoft");
      bwriter.newLine();
      bwriter.write("Masters in selenium");
       bwriter.flush();
      }
}
```

Note: If you want to override the new data with existing data in a file, then we need to write the following.

```
FileWriter fwriter = new FileWriter(File file);
```

If you wan to append the data, i.e if you want to add new data to the existing data, then we need to write the following:

```
FileWriter fwriter = new FileWriter(File file, boolean append);
```

*How to read data from text files:

Step1: Specify the location of a file

we can specify the location of a file by using 'File' class provided by java, which is from 'java.io' package.

ex: File file = new File("F:/selenium/hello.txt");

Step2: Create FileReader and BufferedReader objects.

In order to read data from text files, java provided two

predefined class called FileReader and BufferedReader.

ex: FileReader fileReader = new FileReader(file);

BufferedReader bureader = new BufferedReader(fileReader);

step 3: Call the corresponding methods to get the data

BufferedReader class has the following methods to read the data.

1) public boolean ready()
2) public String readLine()
3) public int read()
4) public void reset()

≡

1) ready() :- Whenever we call ready(), it checks whether there is any data in that file or not. If data is there, it returns 'true' else it returns false.

2) readLine() : By using readLine(), we can read one line data and it returns that line the form of String.

Programs for how to read data from text files.

```
public class FileReadDemo {
Public static void main(String[] args) throws Exception {
        File file = new File("D:\\selenium.txt");
    FileReader fileReader = new FileReader(file);
```

```
BufferedReader bufferedReader = new BufferedReader(fileReader);

String line1 = bufferedReader.readLine();

System.out.println(line1);

String line2 = bufferedReader.readLine();

System.out.println(line2);

String line3 = bufferedReader.readLine();

System.out.println(line3);

while(bufferedReader.ready())
{
String text = bufferedReader.readLine();

System.out.println(text);

}
}
}
```

Real time usage of text files:

Program:

```
public class RealTimeDemo {
public static void main(String[] args) throws Exception {

WebDriver driver = new FirefoxDriver();

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

driver.manage().timeouts().implicitlyWait(58, TimeUnit.SECONDS);

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

```
WebElement electronics = driver.findElement(By.xpath("//*[text()='Electronics']");

Actions actions = new Actions(driver);

actions.moveToElement(electronics).build().perform();

String electronicsText = electronics.getText();

File file = new File("F:/selenium/Flipkart.txt");

boolean b = file.createNewFile();

if(b){
 System.out.println("File is created");
 }else{
  System.out.println("File is already existed");}

FileWriter fileWriter = new FileWriter(file);

BufferedWriter bwriter = new BufferedWriter(fileWriter);

bwriter.write(electronicsText);

bwriter.flush();

}
}
```

output: flipkart.txt

ELECTRONICS

Reusable functions for Text files:

```
public class TextFileReusable {
 static String filePath;
 static File file;
 static FileWriter fwriter;
 static BufferedWriter bwriter;
 static FileReader freader;
 static BufferedReader bufferedReader;
 public TextFileReusable (String filepath)
 { this.filePath = filepath;
 }
public static void fileWriting() throws Exception
{
 file =new File(filepath);
  file.createNewFile();
  fwriter=new FileWriter(file);
  bwriter =new BufferedWriter(fwriter);
   }
public static void fileReading() throws FileNotFoundException
 {
  file =new File(filepath);
```

```
        freader = new FileReader(file);
    bufferedReader = new BufferedReader(freader);
                }
public static void writeDataStr(String data) throws IOException
{
                bwriter.write(data);
                bwriter.flush();
    }
    public static void writeDataInt(int data) throws IOException
    {
    bwriter.write(data);
    bwriter.flush();
    }
    public static String readData() throws IOException
    {
    String data = bufferedReader.readLine();
    return data;
    }
}
```

CSV Files:-

If you want to work with comma separated files (csv) in java.

As java doesnot support parsing of csv files natively, we have 3rd party api "open.csv". is one of the best library available for this purpose. It is open source.

Required tools and technologies:

1) Java JDK 1.5 or above

2) OpenCsv library v1.8 or above

3) Eclipse Luna.

* Reading CSV file in Java :-

we will use the following csv sample file for this example.

File : sample.csv

RameshSoft, Masters in Selenium' Realtime, Training, SR Nagar, Hyd

Program:

```
public class CsvReadDemo{
  public static void main (String[] args) {
    String csvFileName = "c:\\Rameshsoft\\abc.csv";
    CsvReader csvReader=new CsvReader(new FileReader(csvFileName));
     String[] row = null;
    //read csv file line by line
    while((row = csvReader.readNext()) != null)
    {
    System.out.println(row[0] + " # "+row[1] + " # " + row[2]);
     }
    //reading multiple lines
    List content = csvReader.readAll();
```

```
        for (Object object : content)
          { row = (String[]) object;
        System.out.println(row[0] + " #" + row[1] + "# " + row[2]);
          }
         csvReader.close();
        }
}
```

In above code snippet, we use readNext() of csvReader class to read csv File line by line. It returns a String array for each value in row. It is also possible to read full csv file once. The 'readAll()' returns a List of String[] for given csv file.

Writing csv file in Java:

Creating a csv file is as simple as reading one. All you have to do is create the data list and write using csvWriter class.

Program:

```
public class CsvWriteDemo {
  public static void main(String[] args) {
    String csv = "c:\\Abc.csv";
    CsvWriter writer = new CsvWriter(new FileWriter(csv));
    String[] country = "RameshSoft # Masters in Selenium # Real Time
                        Training".split("#");
    writer.writeNext(country);
    List<String[]> data = new ArrayList<String[]>();
```

```
        data.add(new String[] {"Java with Selenium", "SR Nagar"});
        data.add(new String[]{"Hyderabad", "91777 91456"});
        writer.writeAll(data);
        writer.close();
    }
}
```

We created Object of class CsvWriter and called its 'writeNext()' The 'writeNext()' method take String[] as an argument. We used 'writeAll()' of class CsvWriter to write a List of String[] as csv file.

How to handle pop-ups :-

we have two kinds of pop-ups available.

1) web-based pop-up

2) window-based pop-up

web-based pop-ups :-

Generally, Java script pop-ups are generated by web application. and hence they can be easily controlled by the brawser.

* webDriver API allows us to handle web-based pop-ups.

* There are three kinds of web-based pop-ups available.

a) Prompt pop-up

b) Configuration pop-up

c) Java Script alert pop-up.

In order to handle web-based pop-ups, Selenium has provided API called 'Alert API'.

Alert is an abstract class to handle web-based pop-ups in Selenium.

* whenever you want to perform alert operations on pop-ups first we need to switch to that pop-up.

* we can switch to the pop-up in the folbwing way.

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

* In order to switch to the pop-up, we have one method called alert().

Syntax: public Alert alert();

Methods present in Alert abstract class:-

1) accept():- In order to click 'ok' button on pop-up, we need to call accept(). function

Syntax: public void accept()

2) dismiss():- In order to click 'cancel' button on pop-up, we need to call dismiss() function.

Syntax: public void dismiss()

3) sendkeys(): In order to send some text to the pop-up, we need to call sendkeys() function.

Syntax: public void sendkeys(String data)

4) getText():- In order to get text of the pop-up, we need to call getText(). whenever we call getText(), it always returns that pop-up text in the form of String.

Syntax: public String getText()

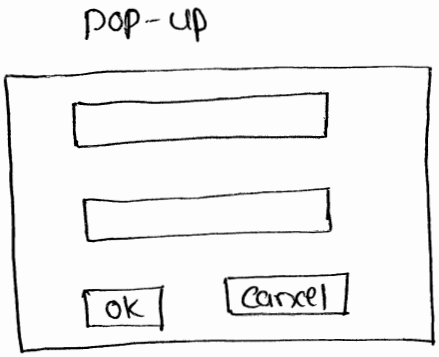
prompt.html :-

```
<html>
  <head>
   <title> Selenium by Ramesh Anapati </title>
  </head>
  <body>
    <h2> Selenium by Ramesh Anapati </h2>

   <fieldSet>
   <legend> Prompt Box </legend>
   <p > click</p>
   <button on click = "promptFunction()" > click </button>
    <                                    >
     <script >
     function  promptFunction()
      {
       var  x;
       var  person = prompt ("please enter your name for new batch",
                                                    "your name") ;
       if(person!=null)
        {
       x= "Hello" + person + "/welcome to selenium" ;
      document.getElementById("prompt demo").innerHTML = x ;
        }
      }
     </script>
   </fieldSet>
   </body>
</html>
```

Program:

```
public class WebBasedPopupDemo {
    public static void main(String[] args) throws Exception {

     WebDriver driver = new FirefoxDriver();
     driver.manage().window().maximize();
     driver.manage().timeouts().implicitlyWait(49, TimeUnit.SECONDS);
     driver.get("file://c:/users/----");
                          ↳ html file url ex: prompt.html
     driver.findElement(By.xpath("html/body/fieldset/button")).click();
     Alert alert = driver.switchTo().alert();
     String popupText = alert.getText();
     System.out.println("popup text is :"+ popupText);
     alert.sendkeys("Welcome to RameshSoft");
     Thread.sleep(3000);
     alert.accept();
     Thread.sleep(3000);
     driver.quit();
    }
}
```

Q:-If the pop-up is having two text boxes, how can you enter data into second text box.



we can enter text into second text box by using "Actions" class TAB key.

```
Alert alert = driver.switchTo().alert();
alert.sendkeys("RameshSoft");
Actions actions = new Actions(driver);
actions.sendkeys(keys.TAB).build().perform();
actions.sendkeys("selenium");
alert.accept();
```

Window-based pop-up:-

As selenium cannot handle window-based pop-ups, we need to take support from third party tools called 'Autoit'.

Autoit:- it is an open source tool.

* Being a selenium developer, we take support of "Autoit" to handle window-based pop-ups.

* Other than this window handling, we can also perform keystrokes and mousehover actions.

How to download Autoit :-

open google → enter "download Autoit" → click on "Autoit" — Downloads → Autoit (http://www.autoitScript.com/site/download"/) link → click on "Download Autoit" → save file.

How to install Autoit :-

After downloading Autoit setup file → double click on the setup file → next → click on "next" until 'finish'.

* How to check whether 'Autoit' is installed or not ?-

Go to my computer → Program files → Check for Autoit folder.

Autoit editor :-

If you want to write the Autoit program, Autoit has editor called "SciTE" editor (a scintilla based Text Editor).

Path :- My computer → 'c' drive → Program files → Autoit 3 → SciTE → double click on "SciTE" icon.

Finder Tool :-

By using Finder tool in Autoit, we can find elements.

path : → My Computer → 'c' drive → Program files → Autoit3 → Au3Info_64 → double click.

* If you want to find the element with finder tool. Hold and drop on corresponding element → now it will give you the corresponding attributes for that element like "title", "control id", "class name", "text" etc.

1) ControlFocus() :-

By using controlFocus() method, we can focus on a particular element i.e, input Focus to a given control on a windows.

syntax: controlFocus ("title", "text", "control id");

ex: ControlFocus("File Upload", "", "Edit1");

2) ControlSetText() :-

By using ControlSetText(), we can set or we can pass some input to the element.

syntax: ControlSetText("title", "text", "controlID", "InputPath");

3) ControlClick() :-

By using controlClick() method, we can perform click operation on element.

syntax: ControlClick("title", "text", "controlID");

ex: ControlClick("Fileupload", "", "Button1");

Program: // windowbased.au3

```
ControlFocus("File upload", "", "Edit1");

ControlSetText("File upload", "", "Edit1", "c:\Users\Desktop\
                                    seleniumfile.txt");
```

ControlClick ("File upload" "" , "Button1");

After writing this code in "Autoit" editor, we need to save that file with ".au3" extension and save as type "Autoit(au3)". After saving this file, we can see "au3" file and that we need to compile.

How to compile "au3" file:-

In order to compile "au3" file, there are certain rules.

1) If the operating system is windows x64, use compile script with 64'.

2) If the operating system is windows x32, use compile script with 86.

Select the "au3" file → compile → now it generates ".exe" file.

How to execute '.exe' file:-

In order to execute ".exe" file, Java provided a predefined class called "Runtime" class, and it has static method called 'getRuntime()' to execute the ".exe" file.

* Runtime class is from java.lang package and getRuntime() is a static method, which is available in "Runtime" class.

* exec() method always asks path of executable file.

window.html :-

```
<input type="file" , id="1", name="File upload" >
```

Program :-

```
public class AutoitFileUploadDemo {
    public static void main (String[] args) throws InterruptedException {
        WebDriver driver = new FirefoxDriver();
        driver.manage().window().maximize();
        driver.manage().deleteAllCookies();
        driver.manage().timeouts().implicitlyWait(59, TimeUnit.SECONDS);
        driver.get("file:///c:/users/Desktop/window.html");
                                    (window html path)
        driver.findElement(By.xpath("html/body/input")).click();
        Thread.sleep(2000);
        Runtime.getRuntime().exec("c://users//Desktop//windowbased.exe");
                                                    ↓
        Thread.sleep(2000);                    path of .exe file
        driver.quit();
    }
}
```

INDIA'S NO1 REALTIME TRAINING INSTITUTE

RAMESHSOFT

100% REALTIME TRAINING

100% PLACEMENTS

TILL NOW

300+

STUDENTS GOT PLACED

TRAINER NAME : RAMESH ANAPATI

CERTIFIED AND REAL TIME EXPERT

Opposite Of Vindu Tiffin Center Between Canara Bank And Axis Bank 3rd Floor
Near Umesh Chandra Statue SR Nagar Sarala Apartments,HYDERABAD

PH : 9177791456, 9502695908, 040-48572456

Frameworks

- Frameworks in Real Time we will discuss in detailed in classroom.

- More than 30hrs will discuss.

  Note: This notes is not complete notes

Reports

There are two types of reports available

1) Static Reports

2) Dynamic Reports.

Static Reports :- Static Reports are not user-friendly reports i.e, by seeing that reports most of the people may not understand.

* Basically, we can see static reports in testing by default generated by TestNG after execution of our scripts.

* Even we can customize the TestNG reports, still it will not completely user friendly.

* "TestNG" has 'Reporter' class which is used to customize the reports.

Sample Program:-

```
public class TestDemo1 {
@Test
public void testScript() throws InterruptedException {
  WebDriver driver = new FirefoxDriver();
  Reporter.log("Firefox browser is launched");
   driver.manage().window().maximize();
   Reporter.log("Window is maximized");
   driver.manage().deleteAllCookies();
   Reporter.log("Deleted all cookies");
```

```
driver.manage().timeouts().implicitlyWait(45, TimeUnit.SECONDS);

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

Reporter.log("Gmail application is opened");

driver.findElement(By.id("Email")).sendkeys("rameshsoft.selenium
                                              @gmail.com");

driver.findElement(By.id("next")).click();

Reporter.log("clicked on next button");

Thread.sleep(2000);

driver.findElement(By.id("passwd")).sendkeys("******");

driver.findElement(By.id("signIn")).click();

Reporter.log("successfully opened Gmail inbox");

Thread.sleep(2000);

driver.quit();

}

}
```

2) Dynamic Reports :-

These are user-friendly reports, by seeing this report everyone can understand.

* we can generate dynamic reports by using "Extent Reports API".

Extent Reports API :-

By using this API, we can generate user-friendly html codes.

There are two types of versions available.

1. 1.0v extent reports.
2. 2.0v extent reports.

* By using extent reports, we can generate reports in the form of graphical or pictorial representation.

How to download Extent Reports API :-

Open google → type "Download extent reports" → Extent reports for awesome selenium WebDriver Reporting (relevantcodes.com/extentreports for selenium) → download version 1 → save.

Once it is downloaded, extract the zip folder and then you will get normal folder with a jar file.

Add this jar file(s) to eclipse to use extent API.

Steps to develop API:-

Step1:- Download and add jar file(s) to eclipse

Step2: Get extent reports object

```
ExtentReports ext = ExtentReports.get(Test.class);
```

↓ class name for which we want generate reports.

Step3:- Initialize where you want to place the reports in local machine

```
ext.init("c://rameshsoft//report.html", true);
```

If we write true, the new reports will be generated with same name of old report and the data will be overridden and it is always recommended to write "true".

Step4:- Start the Test case and then log the report.

```
ext.startTest("GmailLogin");
```

Step5:- Log the information

In order to log the information extent reports is having log().

```
ext.log(LogStatus.INFO, "clicked on next button");
```

Step6:- While we are logging, there are different types of log status are there.

1) INFO :- for logging additional info relevant to the test (or for debugging purpose).

2) ERROR :- for logging any error (non-fatal) that occurs during test execution.

3) WARNING :- for logging warnings.

4) FAIL :- Fail for checks that are executed with a negative result.

5) FATAL :- for logging fatal errors occuring during test execution.

6) PASS :- for checks that are executed with a positive result.

7) SKIP.

Sample Program :-

```
public class ReportDemo {
 static ExtentReports extent;
 public static void screenshot (String fileName, WebDriver driver) throws
                                                        IOException {
    File srcFile = ((TakeScreenShot)driver).getScreenShotAs (OutputType.FILE);
    FileUtils.copyFile(srcFile, new File(fileName));
    }
  public static void main (String [] args) throws InterruptedException {
     extent = ExtentReports.get (ReportDemo.class);
     extent.init ("D://extent report// reports.html", true);
```

```
extent1.startTest("Gmail Login Testcase");

WebDriver driver = new FirefoxDriver();

extent1.log(LogStatus.INFO, "Firefox browser is launched");

Thread.sleep(2000);

screenshot("D://extentreport//image1.png", driver);

Thread.sleep(1000);

extent1.attachScreenShot("D://extent-report//image1.png");

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

extent1.log(LogStatus.INFO, "Gmail is opened");

 Thread.sleep(2000);

 screenShot("D://extentReport//imag2.png", driver);

   Thread.sleep(2000);

   extent1.attachScreenShot("D://extent-report//image2.png");

   Thread.sleep(2000);

driver.findElement(By.id("Email")).sendkeys("rameshsoft.selenium@
                                              gmail.com");

   extent1.log(LogStatus.INFO, "username is entered with test data
                                  @rameshsoft.selenium@gmail.com");

    Thread.sleep(2000);
```

```
Screenshot("D://extentreport//image3.png", driver);
extent1.attachScreenShot("D://extentreport//image3.png");
driver.findElement(By.id("next")).click();
extent1.log(LogStatus.INFO, "clicked on next button");
    Thread.sleep(1000);
Screenshot("D://extentreport//image4.png", driver);
extent1.attachScreenShot("D://extentReport//image4.png");
 extent1.endTest();
 driver.quit();
 }
}
```

* After execution of the test cases, if we want to see the reports, go to the driver specified in init() and open the reports.

2.0v Extent Reports:-

step1: Download the extent reports jar file(s).

Open google → type 'download extent reports' → click on ExtentReports for awesome Selenium WebDriver reporting → click on Extent Reports - java - v2.40.2 → ok.

Once it is downloaded, extract zip file and we get folder format, that

contains jar files. and add the jar file into your project.

Step2: Get the object of ExtentReport class.

ex:- ExtentReports extentReports = new ExtentReports ("c:\\users\\report.html");

Step3: Get the object of ExtentTest class.

ExtentTest logger = extentReports.startTest("DemoExtentReport");

step4: Log the reports.

step5: End the Test.

Sample Program:-

```
public class DemoReport{
   public static ExtentReports extentReport;
    public static ExtentTest logger;
    public static WebDriver driver;
 @BeforeMethod
  public void method()
   {
    System.out.println("Before method");
    }
 @Test
  public void verifyTitle()
  {
  extentReport = new ExtentReports("c:\\users\\report.html");
```

```
        logger = extentReport.startTest("DemoExtentReport");
        driver=new FirefoxDriver();
        logger.log(LogStatus.INFO, "Browser is launched");
    driver.manage().timeouts().implicitlyWait(35, TimeUnit.SECONDS);
    driver.manage().window().maximize();
    logger.log(LogStatus.INFO, "Browser is maximized");
    driver.get("https://www.google.com");
   String title=driver.getTitle();
   Assert.assertTrue(title.contains("Gmail"));
   logger.log(LogStatus.PASS, "Title is verified");
    }
@AfterMethod
public void method(ITestResult result) throws Exception
{
  if(result.getStatus() == result.FAILURE)
      {
      File file = ((TakeScreenShot)driver).getScreenShotAs(outputType.FILE);
      FileUtils.copyFile(file, new File("C:\\Users\\failure.png"));
       }
   }
}
```

JavascriptExecutor :-

JavascriptExecutor is an interface, which is available in org.openqa.selenium.JavascriptExecutor.

→ It Provides mechanism to execute Javascript through selenium WebDriver. Javascript in the context of the currently selected Frame(or) window.

→ we can use Javascript to perform actions using selenium WebDriver.

→ By using "JavascriptExecutor", we can perform these actions.

Why we use it?

To Enhance the capabilities of the existing scripts by performing Javascript injection into our application under Test.

In simple words "Java script can be executed within the browser with the help of "JavascriptExecutor".

Syntax: JavaScriptExecutor js = (JavascriptExecutor) driver;

(Typecasting as JavascriptExecutor is an interface of org.openqa.selen-ium.JavascriptExecutor).

js.executeScript(script, Arguments);

The arguments to the script is optional.

script → The Javascript to execute.

1) Generating Alert pop window

```
JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("alert('Hello World!');");
```

2) Mouse hover :-

```
WebElement element = driver.findElement(By.xpath("/a"));

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("arguments[0].onmouseover()", element);
```

3) Click() :

```
JavaScriptExecutor js =(JavascriptExecutor) driver;

js.executeScript ("arguments[0].click();", element);

                (or)

js.executeScript("window.document.getElementById("Email").click()");

js.execute Script ("document.getElementById("Email").click());
```

4) refresh window

```
js.executeScript("history.go(0)");
```

5) Get title :-

```
String title = js.execute Script("return document.title;").toString();
```

6) How to get innertext of entire web page

```
String text = js.executeScript("return document.documentElement.innerText;")
                                                              .toString();
```

7) Scroll Bar operations

```
js.executeScript("window.scrollBy(0, 50)");
```

8) Highlight element

```
WebElement username = driver.findElement(By.id("email"));

    JavaScriptExecuter js = (JavaScriptExecutor) driver;

js.executeScript("arguments[0].style.border='4px groove green'", username);
```

9) sendkeys():

```
js.executeScript("document.getElementById('Email').value='ramesh'");
```

10) How to uncheck checkbox in selenium if checkbox is disable

```
js.executeScript("document.getElementById('abc').checked=false");
```

11) Get url:-

```
String url = (String) js.executeScript("return document.URL");

    System.out.println("Page url is:" + url);
```

12) Get Attribute Text

```
Object obj = ((JavaScriptExecutor) driver).executeScript("return document.
                getElementById('main')");

String name = ((WebElement) obj).getAttribute("class");
```

13) Get Domain name

```
String domain = (String) js.executeScript("return document.domain");
```

o/p: domain : http://www.google.co.in

14) Last modified

```
String lastmodified = (String) js.executeScript("return document.lastmodified
                                                 ");
```

o/p: Lastmodified : 01/01/2016 14:00:46

15) Navigate to different page

```
js.executeScript("window.location = 'https://www.facebook.com/ ' ");
```

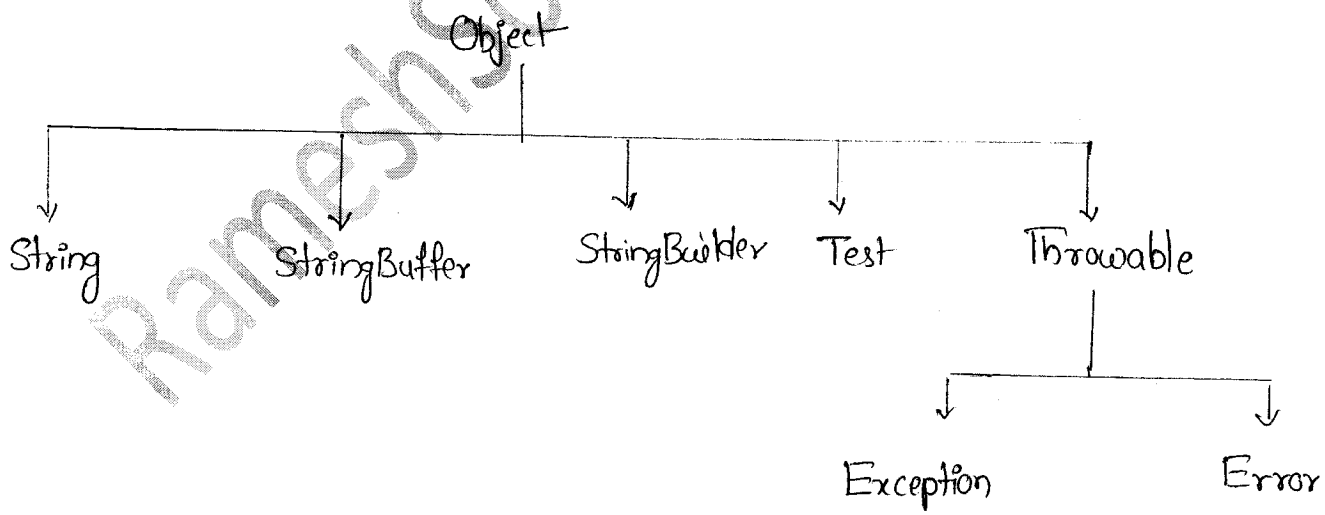
Exception Handling

Exception:- An unwanted or an unexpected event that disturbs normal flow of the program is called exception.

* Whenever we get exception, it is always an abnormal termination.
* In general, the program should terminate normally. So, it is highly recommended to handle exceptions using 'Exception Handling'.

"Exception Handling means providing some alternative to continue rest of the program successfully."

In Java for all classes, either predefined or customized classes, 'Object' class is the Parent Class or Super class, either you 'extends' or not, by default 'Object' class will become Parent class.



Example:

```
class Test
{
==
}
```

⇒

```
class Test extends Object
{
==
}
```

* Throwable class acts as a root for Exception Hierarchy.

* Throwable class contains two child classes

1) Exception 2) Error.

Exception:- Most of the cases Exceptions are caused by our program & these are recoverable.

For example, if our programming requirement is to read data from Remote file.

At Runtime if Remote file is not available then we will get RuntimeException saying FileNotFoundException.

If FileNotFoundException occurs we can provide a local file to continue rest of the program normally.

Error :- Most of the cases errors are not caused by our program & these are due to lack of System resources.

* Errors are non-recoverable.

ex: If 'outofmemoryError' occurs being a programmer we can't do anything then the program will be terminated abnormally.

System or server Admin is responsible to increase Heap memory.

Difference b/w Checked Exception and Unchecked Exception

Checked Exception :-

* The Exceptions which are checked by compiler for smoth execution of the program at runtime are called Checked Exceptions

Ex: FileNotFoundException, HallTicktMissingException --etc.

* Compiler will check whether we are handling checked Exception or not.
* If we are not handling then we will get compile time error.

Unchecked Exception:-

* The Exceptions which are not checked by compiler whether the programmer handling or not are called Unchecked Exceptions.

ex:- NullPointerException, ArithemeticException, ShortCircuitException

Note:-

1) Whether Exception is checked or unchecked compulsory every exception should occurs at runtime only & there is no chance of occuring any Exception at compile time.

2) RuntimeException and its child classes, Error and its child classes are unchecked. Except these the remaining are checked Exceptions.

Difference blw Fully checked and Partially checked Exceptions

Fully checked Exceptions:-

* A checked Exception is said to fully checked if and only if all its child classes also checked.

ex: IOException, InterruptedException

Partially checked Exceptions:-

* A checked Exception is said to be partially checked if and only if some of its child classes are unchecked.

Ex: Exception, Throwable. (these are only available partially checked exceptions in Java)

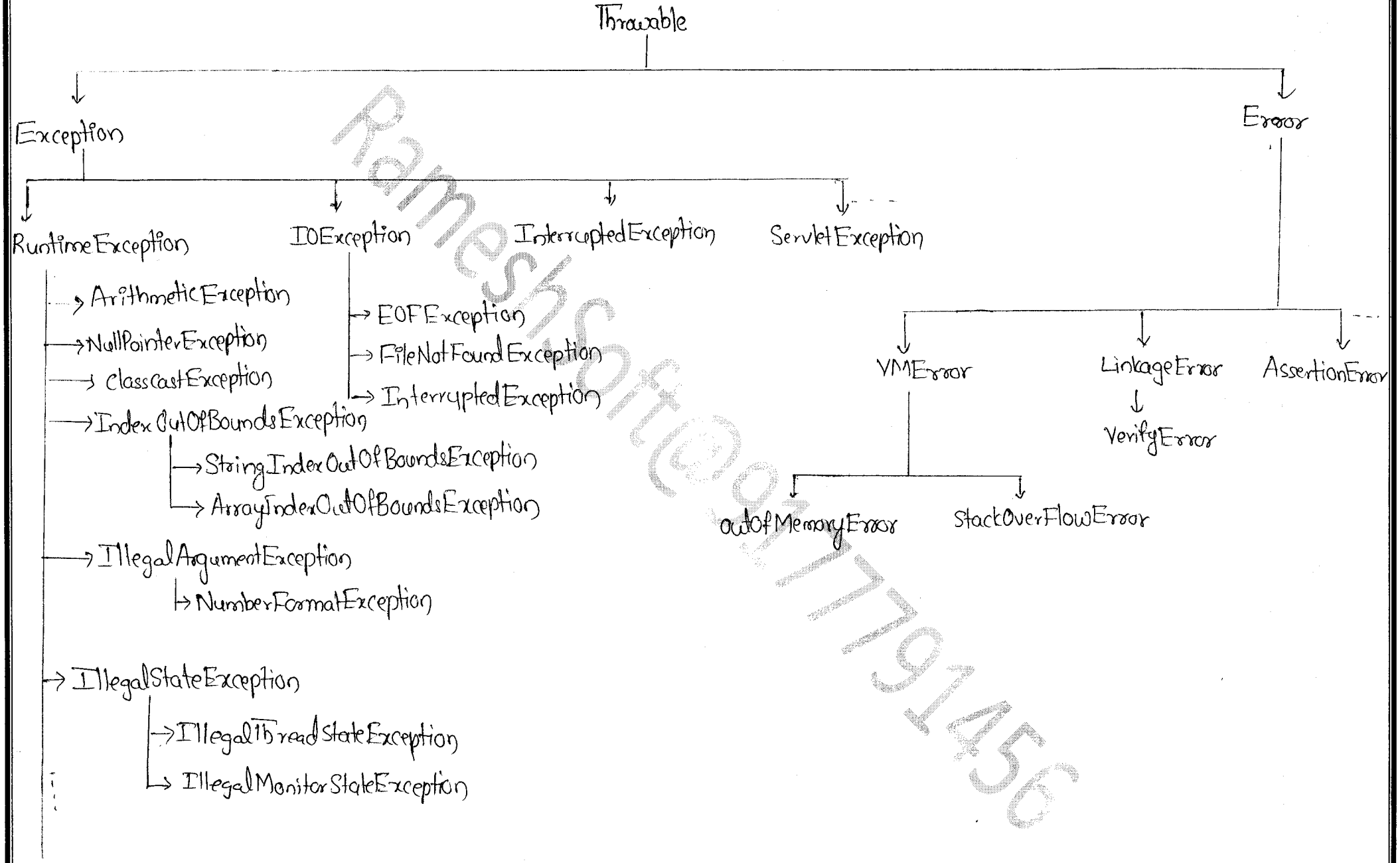


Figure: Exception Hierarchy chart

Handling the Exceptions:

For handling the exception sun developers has given 5 keywords.

They are

① try
② catch
③ finally
④ throw
⑤ throws

Handling exceptions using try-catch

try :- 'try' is a block, inside 'try' block always place risky code only.

* Risky code means because of that code you may get the problems.
* Whenever we get exception from that line onwards the rest of the code is not going to be executed.

```
syntax:-   try
           {
            = risky code;
           }catch(Exception e)
            {
            =
           }
```

catch:- catch is a block and always catches the exception raised in try-block

* Catch always associates with try-block.

```
syntax:    try
           {
            =
           }catch (Exception e)
            {
            =
           }
```

* try-block we cannot define without 'catch' block. that is compulsory we need to write or associate catch block.

Note: Between try block and catch block if we write any executable statements it is going to given compile-time error.

Control Flow in try-catch:

```
try
{
 Statement1;
  statement2;
  statement3;
}
catch( X  e)
 {
  statement4;
 }
statement5;
```

case i) if there is no Exception

1, 2, 3, 5, Normal termination

case ii) If an exception raised at statement2 and the corresponding catch block matched.

1, 4, 5, Normal termination

case iii) If an Exception raised at statement2 and the corresponding catch block not matched.

1, Abnormal termination

RAMESHSOFT

SOFTWARE SOLUTIONS

OUR STUDENTS RECENTLY PLACED ON SELENIUM

AHMED

(SD SOFTECH PVT LTD)

RAFEEQ

(EDREEZ SOFTWARE PVT LTD)

MADHUPRIYA

(CIGNITI TECHNOLOGIES)

SWATHI

(COGNIZANT)

MRUDALA

(SOFTSOL TECHNOLOGIES)

GANESH

(PURPLETALK)

MANDEEP

(SOCTRONICS)

VISHWAJIT

(TCS)

SIBASYS

(COGNIZANT)

SRIKANTH.R

(INFOSYS)

SHRAVAN KUMAR

(IBM)

SANGAMESH

(TECH MAHINDRA)

PRAHALAD

(INFOBRAIN)

HARISH

(PRDC)

KIRAN

(CYBAZE)

THAKIL

(UHG)

Case iv) : If an exception raised at statement4 (or) statement5 it is always abnormal termination.

Note:-

① within the try block if any where an Exception raised rest of the try block won't be executed eventhough we handled that exception.

Hence length of the try block should be as less as possible and we have to take only risky code within the try block, but not normal java code.

② In addition to try block there may be a chance of raising exception inside catch & finally block also.

③ If any statement raises exception and if it is not part of try block then it is always abnormal termination of the program.

④ Sometimes even we get exceptions in catch-blocks as well

Example :-

```
public class ExceptionTryCatchDemo
{
    public static void main(String[] args)
    {
        System.out.println("Login");
        try
        {
            FileInputStream fip = new FileInputStream("abc.xls");
        }catch(Exception e)
        {
            System.out.println("provide alternative or handling the exception");
        }
        System.out.println("Log out");
    }
}
```

Examples for flow within try & catch blocks:—

Example/case i):— If exception raised in try block then output is: Hi

```
try
{
System.out.println("Hello");
}catch(Exception e)
 {
 System.out.println("Hi");
 }
```

if exception not raised output: Hello.

case ii):—

```
try
{
System.out.println("outer-try");
    try
    {
    System.out.println("Inner try");
    }catch(Exception e)
      {
      System.out.println("Inner-catch");
      }
}catch(Exception e)
{
System.out.println("outer-catch");
}
```

i) if no exception raised

output is: Outer try
Inner try

ii) If exception raised in outer try

output: outer Catch.

iii) if exception raised in inner try

output: Outer try ; Inner catch

try with multiple catch blocks :-

* The way of handling an exception is varied from Exception to Exception.
* Hence for every exception type is recommended to take separate catch block.
* Hence try with multiple catch blocks is possible and recommended to use.

```
try
{
 ≡
 }
catch(Exception e)
  {
  =
  }
```

Not Recommended.

```
try
{
≡
}
catch(ArithmeticException e)
{
perform the alternative arithmetic operations;
 }
catch(FileNotFoundException e)
{
use local file instead of remote file
}
catch (SQLException e)
 {
  use mysql db instead of Oracle db
  }
catch(Exception e)
{
 Default hierarchy
 }
```

Highly recommended

* If try with multiple catch blocks present then the order of catch blocks is very important. and it should be from child to Parent by mistake if we are taking from Parent to child then we will get Compile time error saying,

Exception xxx has already been caught.

ex:
```
try
{
=
}
catch(Exception e)
{
=
}
catch(ArithmeticException e)
{
=
}
```
(x)

```
try
{
=
}
catch(ArithmeticException e)
{
=
}
catch(Exception e)
{
=
}
```
(✓)

CE: exception java.lang.ArithmeticException has already been caught

*If we are trying to take multiple catch blocks for same Exception then we will get Compile time error.

ex:
```
try
{
=
}catch(ArithmeticException e)
{
=
}
catch(ArithmeticException e)
{
=
}
```
(x)

COmpile time error: exception java.lang.ArithmeticException has already been caught.

finally block :-

→ It is not recommended to maintain clean-up code inside try block because there is no guarantee for the execution of every statement inside try block always.

→ If is never recommended to maintain clean up code inside catch block because if there is no exception then catch block won't be executed.

→ we required some place to maintain clean up code which should be executed always irrespective of whether exception raised or not, whether handled or not handled such type of best place is nothing but finally block.

→ Hence the main purpose of finally block is to maintain clean up code.

ex:
```
try
{
 risky code
}catch(X e)
{
 Handling code
}
finally
{
 clean up code
}
```

→ The speciality of finally block is it will be executed always irrespective of whether exception raised or not and handled or not handled.

* Finally is mainly used to perform cleanup activities like db connection closing, file o/p, o/p stream closing etc.

Control flow in try-catch-finally :-

```
try
{
 statement 1;
 statement 2;
 statement 3;
}catch(X e)
 {
 statement 4;
 }
finally
 {
 statement 5;
 }
 statement 6;
```

case i) : If there is no Exception

flow : 1, 2, 3, 5, 6, Normal Termination

case ii) : If an Exception raised at statement 2 and corresponding catch block matched.

flow : 1, 4, 5, 6, Normal Termination

case iii) : If an Exception raised at statement 2 and corresponding catch block not matched

flow : 1, 5, Abnormal Termination

case iv) : If an exception raised at statement 4 then it is always Abnormal Termination, but before that finally block will be executed.

case v) :- If an exception raised at statement 5 or statement 6 then it always Abnormal Termination.

Control flow in nested try-catch-finally :-

```
try
{
 statement1;
 statement2;
 statement3;
    try
    {
      statement4;
      statement5;
      statement6;
    }
    catch(X e)
     {
      statement7;
     }
    finally
    {
      statement8;
     }
   statement9;
}
catch(X e)
  {
  statement10;
  }
finally
  {
  statement11;
  }
 statement12;
```

case i): If there is no exception. Then flow is 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, Normal Termination

case ii): If an Exception raised at statement 2 and corresponding catch block matched.

flow : 1, 10, 11, 12, Normal termination.

case iii) If an Exception raised at statement 5 and corresponding inner catch block matched.

flow : 1, 2, 3, 4, 7, 8, 9, 11, 12, Normal Termination.

case v) : If an exception raised at statement 5 and corresponding inner catch block not matched. Then flow : 1, 2, 3, 4, 8, 10, 11, 12, Normal Termination.

case vi) : If an Exception raised at statement 5 and both inner and outer catch blocks are not matched.

Flow : 1, 2, 3, 4, 8, 11, Abnormal Termination.

case vii) : If an Exception raised at statement 7 and corresponding catch block matched.

flow: 1, 2, 3, 4, 5, 6, 8, 10, 11, 12, Normal Termination

case viii) If an exception raised at statement 7 and corresponding catch block not matched.

Flow: 1, 2, 3, 4, 5, 6, 8, 11, Abnormal Termination

case ix) : If an exception raised at statement 8 and corresponding catch block matched.

flow: 1, 2, 3, 4, 5, 6, 10, 11, 12, Normal Termination

case x) : If an exception raised at statement 8 and corresponding catch block not matched.

flow: 1, 2, 3, 4, 5, 6, 11, Abnormal Termination

case xi) : If an exception raised at statement 9 and corresponding catch block matched.

Flow : 1, 2, 3, 4, 5, 6, 8, 10, 11, 12, Normal Termination

case xii) : If an exception raised at statement 9 and corresponding catch block not matched. Then flow: 1, 2, 3, 4, 5, 6 8, 11, Abnormal Termination

case xiii) : If an exception raised at statement 10 and then it is always Abnormal Termination but before that finally block will be executed.

case xiv) : If an exception raised at statement 11 then it is always Abnormal Termination.

** Note:—

Default Exception Handler can handle only one exception at a time which is the most recently raised exception.

```
public class Demo
 { public static void main(String[] args)
    {
       try
        {
         System.out.println(10/0);
        }
       catch(ArithmeticException e)
        {
         System.out.println(10/0);
        }
       finally
        {
         String s=null;
        System.out.println(s.length());
        }
    }
 }
```

Output: Runtime Exception : NullPointerException

throw :—

throw is a keyword, by using throw, we can throw the exception or we can create customized exceptions.

* Sometimes we can create Exception object explicitly and we can handover that object to the JVM manually, for this we have to use throw keyword.

ex: throw new ArithmeticException("/ by zero");

Ex: class Demo
{
public static void main(String[] args)
{
System.out.println(10/0);
}}

Exception in thread "main": java.lang.ArithmeticException
at Demo.main()

In this case, main() is responsible to create Exception object and handover to the JVM implicitly.

class Demo
{
public static void main(String[] args){
throw new ArithmeticException("/by zero");
}
}

Exception in thread "main":
java.lang.ArithmeticException:
/ by zero at Demo.main()

In this case, programmer is responsible to create exception object explicitly and handover to the JVM.

* Most of the cases, we can use throw keyword for customized exceptions (our own exceptions) but not for predefined Exceptions.

throws keyword :-

*In our program, if there is any chance of raising checked Exception compulsory we should handle that checked exception otherwise we will get compiletime error saying, Unreported exception xxx; must be caught or declared to be thrown.

*We can handle this checked Exception by using the following 2 ways.

1) by using try-catch
2) by using throws. keyword.

* we can use throws keyword to delegate responsibility of Exception Handling to the caller (it may be method or jvm) then caller is responsible to handle that checked Exception.

ex:
```
public class Demo
{
  public static void main(String[] args) throws InterruptedException
  {
    Thread.sleep(2000);
  }
}
```

① we can use throws keyword to delegate responsibility of Exception handling to the caller.

② It is required only for checked Exceptions and usage of throws keyword for unchecked Exceptions there is no use.

③ throws keyword required only to convince compile and usage of throws keyword doesn't prevent Abnormal Termination of the program.

Q Difference b/w final, finally and finalize()?

final:-

→ final is a keyword applicable for classes, methods and variables

* when a class declared as final, you cannot inherit the class. So inheritance is not possible.

* when a method is declared as final, then we cannot override that method.

* when a variable declared as final, then we cannot perform reassignment. because it will become constant.

finally :-

* finally is a block, which always associates with try-catch block.
* This block will be executed always whether exception raised or not and whether we handled or not handled.
* we can use finally block to perform clean-up activities like database closing, connections etc.

finalize() :-

* finalize() is a method used to perform clean up activities like object destruction.
* Before destroying any useless objects, garbage collector calls the finalize() to perform clean-up activities.
* Garbage Collector is the assistance of JVM.

Note:-

when compared with finalize() method finally block is recommended to maint-ain clean up code because we can't expect exact behaviour of Garbage Collector.

Methods to print Exception information

* Throwable class defines the following methods to print Exception information.

1) printStackTrace():-

Printable format:- Name of Exception : Description StackTrace

2) toString():-

Printable format: Name of Exception : Description

3) getMessage():

Printable format: Description

Example:-

```
public class ExceptionPrintDemo
{  public static void main(String[] args)
    {
      System.out.println("Login");
    }
   try
    {
    FileInputStream fip = new FileInputStream ("D://ABC.xlsx");
    }catch(Exception e)
    {
      e.printStackTrace();
   // System.out.println(e.getMessage());
    // System.out.println(e.toString());
    // System.out.println(e);  } } }
```

Jenkins

→ Jenkins is an Open-Source continuous integration tool, which we can download, we can see the code, and we can modify the code.

→ Jenkins is written in java programming language designed to test and report an isolated changes in a larger code base in real time.

→ Jenkins enables developers to find and solve defects in a code base rapidly and to automate testing of their build.

→ Continuous integration is a process in which all development work is integrated at a pre-defined time or event and the resulting work is automatically tested and built. The idea is that development.

→ The basic functionality of Jenkins is to execute a predefined list of steps based on a certain trigger or condition.

ex: build for every 20 minutes.

Jenkins also monitors the execution of the steps and allows to stop the process if one of the step fails. Jenkins allows to notify user about the build success or failure.

→ Jenkins can be started via the command line or can run on a web application server.

→ Jenkins can be extended by additional plug-ins.

Advantages of Jenkins:

1) When the test cases are ready and we want to handover to client or manual testing team, so that they can trigger these test cases using click, then jenkins is the best choice.

2) Using Jenkins, we can create build and we can run easily using batch file or build .xml etc.

3) In Jenkins, we can schedule the build periodically.

4) Email-notification: Jenkins provides notification mails too, once build passed or failed to respective receipients (depends on configuration)

How to download Jenkins:

Open browser → navigate to http://jenkins-ci.org → download jenkins.war from jenkins home page.

How to configure jenkins:

→ Go to the location where jenkins.war is available.

→ Open command prompt known as cmd and navigate till the project home directly and start jenkins server.

start cmd prompt → project-home-directory>

>java -jar jenkins.war

info: jenkins is fully up and running.

→ Once jenkins server is up and running, you will get above success message.

by default, jenkins run on 8080 port, so you can navigate to below url for jenkins UI.

open the browser and type url as http://localhost:8080

Now jenkins is up and running, so now you have to configure jenkins so that we can execute our test case via jenkins.

→ Once jenkins is running, so we are almost done. build before moving to create build, we need to configure jenkins, so that jenkins can identify other tools as well like java, maven etc.

* click on > Manage Jenkins
* Click on > Configure System
* How we are telling jenkins that our java is located at this location so we don't have to worry about explicit path.
* Navigate to JDK section and click on add JDK button.
* uncheck install automatically checkbox, so jenkins will only take java, which we have mentioned above.
* Give the name as Java-Home and specify the jdk path
* In jenkins, we have very good feature that you can configure email notification for user.
* This is optional, but if you want to configure email notification then you have to do little setting while configuring jenkins.

Go to Email Notification:

SMTP Server xxx.gmail.com This is Gmail smtp server

Default - user email - suffix @gmail.com This suffix will be same for all

☑ use SMTP authentication

username abc@gmail.com

password * * * * *

use SSL ☑

SMTP port 465 This SMTP port will be same for all.

reply to address def@gmail.com

Charset UTF-8

☐ Test configuration by sending test email

save apply ← (once done click on save and apply)

Now you jenkins is configured now.

→ Before creating job, jenkins execute job in 4 ways.

1) Execute windows batch command i.e bat file

2) Execute shell

3) Invoke Ant

4) Invoke top-level maven target

In this, we will execute using window batch command.

1) Create a batch file first then, we will add the same file to Jenkins.

* To create batch file, we need to set class path of TestNG, so that we can execute testng.xml files.
* Open command prompt and set the classpath while setting classpath, we will set the path of bin folder and this libs folder.
* open notepad and type the below command and save as xxx.bat file

ex: run.bat

```
java -cp bin;libs/* org.testng.TestNG testng.xml
```

How to create a job:

create a job in jenkins, which will execute the build open jenkins on browser (type: http://localhost:8080)

→ click on new item

→ Give the job name, select build a free style software project and click on 'ok' button.

→ To advanced project options → check the use custom workspace in director we will specify the batch file which we created and click on apply and save.

* We finally run the build :

click on build now option

* Check build history and console output and verify the output.

Schedule the build in jenkins for periodic execution:—

Jenkins comes with very good functionality in which we can schedule jobs which we created.

we can schedule build for existing jobs which already created and while creating new project also, we can specify the same.

Let us schedule the job:

1) open job which we created now and click on configure > select the checkbox build periodically.

2) specify the time here, we need to be careful about the syntax.

Jenkins works on cron-pattern.:—

Jenkins will accept 5 parameters, let us discuss one by one.

* * * * *

* Here 1st parameter specify 'minute', and range will vary from 0-59.

* 2nd parameter specify 'Hours' and range will vary from 0-23.

* 3rd parameter specify 'Day' and range will vary from 0-30|31

* 4th parameter specify 'Month' and range will vary from 1-12

* 5th parameter specify 'Weekday' and range will vary from 0-6 or 1-7. (Weekdays Sunday, monday ----) (1- Monday , 7- Sunday).

Ex1: if you specify 00 22 * * * it means the build runs daily at 11 pm.

Ex2: if you specify 50 * * * * it means the build run daily 50 minutes after every Hour.

Maven

* Maven is from "Apache" Organization
* Maven is a software project management tool, formally known as build tool.
* Maven provides developers or testers complete build life cycle process.
* Maven makes life of testers easy while creating test automation scripts.
* Based on the concept of a project object model (pom), maven can manage project build, reporting and documentation that is called central Repository.
* Maven has it's own repository where it keeps all plug-ins, jars etc.
* We can create Maven Project for writing script and create dependencies using pom.xml. Once dependency is set, maven will download all the dependent jar files automatically and in future if any update comes from selenium or TestNG side, it will simply update all the required changes.

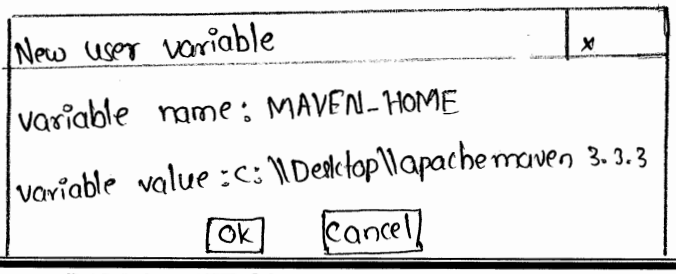
How to download and configure maven:

Open google → navigate to maven official site and download stable version → https://maven.apache.org/download.cgi → apache maven-3.3.3-bin.zip → save → ok

Maven Configuration:

Extract the downloaded folder → click on my computer → right click → properties → advanced system setting → advanced → Environment variable → user → new → MAVEN_HOME.

Ex: New user variable



under system variable → go to path → edit → write maven bin path

Note: Please don't touch other variables, go to last point and then enter path till bin folder.

Now click on save and apply, you can verify now that maven is installed or not.

Open command prompt and type the below command

```
> mvn -version
```

if maven is installed properly then you will get maven version and java version on console.

How to create maven Project :-

File → new → other → maven → maven project → next → check the simple project → next → Group id: (give some name) → Artifact id: (give some name) → finish.

Src/main/resources :-

under this keep .xml files

Src/test/java :-

Under this we need to maintain all the source files (.java)

How to download dependencies :-

We can get dependencies/repository from maven-repository.com

Maven Plugins :-

we have two types of plugins in maven

1) maven-compiler-plugin

2) maven-surfire-plugin

i) Maven-compiler-plugin

By using this plug-in, we can compile all source files like .java/.jar files we need to configure these plugins in pom.xml by using <build> and <plugins> after defining configuration.

ii) Maven-Surfire-plugin :-

At the time of build process, it is going to execute the .class files. we need to configure these plugins in pom.xml by using <build> and <plugins> after defining configurations.

It generates reports in two different file formates.

1) plain text file 2) xml file.

Structure of pom.xml :-

```
<project xmlns="http://maven.apache.org/pom/4.0.0" xmlns:xsi:Schemab-
                                                    -cation="http" />

    <model versions> 4.0.0 </model versions>
     <groupId> Hello </groupId>
     <artifactId> Hello </artifactId>
     <version> 0.1.1-SNAPSHOT </version>
     <name> mavendemo </name>
     <description> mavendemo example project </description>

     <properties>
        <suiteXmlfile> c:/selenium/Hello/src/main/testng.xml</suiteXml file>
        <skipTests> false </skipTeste>
     </properties>
```

```
<dependencies>
    <dependency>
        <groupId> org.testng </groupId>
        <artifactId> testng </artifactId>
        <version> 6.1.1 </version>
        <scope> test </scope>
    </dependency>
    <dependency>
        <groupId> org.seleniumhq.selenium </groupId>
        <artifactId> selenium-java </artifactId>
        <version> 2.46.0 </version>
    </dependency>
</dependencies>
<build>
    <plugins>
        <plugin>
            <groupId> org.apache.maven.plugins </groupId>
            <artifactId> maven-compiler-plugin </artifactId>
            <version> 3.0 </version>
            <configuration>
                <compileversion> 1.8 </compileversion>
                <source> 1.6 </source>
                <target> 1.6 </target>
            </configuration>
        </plugin>
```

```
<plugin>
  <groupId> org.apache.maven-plugins </groupId>
  <artifactId> Maven-surfire-plugin </artifactId>
  <version> 2.6 </version>
    <configuration>
      <suiteXmlFiles>
        <suiteXmlFile> {SuiteXmlFile} </suiteXmlFile>
      </suiteXmlFiles>
      <testFailureIgnore> true </testFailureIgnore>
    </configuration>
  </plugin>
 </plugins>
</build>
</project>
```

Cookies Management

What is Cookie?

* A Cookie is a small piece of text file stored on user's comp-uter in the form of name-value pair. Cookies are used by website to keep track of visitors e.g to keep user information like username etc.

* If any web application using Cookies, Server send cookies and client browser will store it. The browsers then returns the cookies to the server at the next time the page is requested. The most common example of cookie is to store user information, user preferences password remember option etc.

Cookies which are files created by websites you have visited and your browser's Cache, which helps page load faster, make it easier for you to browse the web.

Points to remember about Cookies:

1) Cookies are domain specific i.e a domain cannot read or write to a cookie created by another domain.

This is done by the browser for Security purpose.

2) Cookies are browser specific. Each browser stores the cookies in a different location. The cookies are browser specific and so a cookie created in one browser (e.g in google chrome) will not be accessed by another browser (e.g Internet Explorer).

3) Most of the browser store cookies in text files in clear text, so it is not secure at all and no sensitive information should be stored in cookie.

4) Some browsers limit the number of cookies stored by each domain (20 cookies). If the limit is exceeded, the new cookies will replace the old cookies.

5) Cookies can be disabled by the user using the browser properties. So unless you have control over the cookie settings of the users (for e.g intranet application), cookies should not be used.

6) Cookie names are case-sensitive e.g username is different than username.

Advantages :-

1) Here are some of the advantages of using cookies to store session state.

2) Cookies are simple to use and implement.

3) Occupies less memory, do not require any server resources and are stored on the users computer so no extra burden on server.

4) Cookies persists a much longer period of time than session state.

Disadvantages :-

1) As mentioned previously, cookies are not secure as they are stored in clear and text, they may pose a possible security risk as anyone can open and tamper with cookies, you can manually encypt and decrypt cookies, but it requires extra coding and can affect application performance because of the time that is required for encryption and decryption.

2) Several limitations exist on the size of the cookie text (4kB in general), number of cookies (20 per site in general). etc

3) User has the option of disabling cookies on his computer from browser's to browser.setting.

4) Users can delete a cookies.

5) Complex type of data not allowed (e.g dataset etc). It allows only plain text (i.e cookie allows only string content).

Why to Clear Cookies ?

If you are having a tough time logging in, try clearing you cookies, If your account is not loading correctly, you are not seeing what you expect, or there is an error, try clearing the cache to make sure you are seeing the latest info.

Clearing your browser's cache and cookies means that website settings (like username and password) will be deleted and some sites might appear to be a little slower because all of the images have to be loaded again.

Most browsers work in a similar way, but here are the major browsers and how to clear cache and delete cookie for them. And it's best practice to run the latest version of your favourite browser, keeping it updated helps pages load better and gives you more security.

Cookies API in Selenium WebDriver :-

Add Cookie :

Method name : addCookie (Cookie cookie)

Syntax : `public void addCookie (Cookie cookie)`

eg: driver.manage().addCookie(arg 0);

Description: To add a specific cookie into cookies. If the cookies domain name is left blank, it is assumed that the cookie is meant for the domain of the current document.

ex: Cookie name = new Cookie("firefoxCookie", "123456789123");

driver.manage().addCookie(name);

Get the Cookie with specific name:

Method name: getCookieNamed (java.lang.String name)

Syntax: `public Cookie getCookieNamed (arg0)`

Description: To get a cookie with a given name.

e.x: driver.manage().getCookieNamed ("firefoxCookie");

Get Cookies:

Method name: getCookies()

Syntax : `public Set<Cookie> getCookies()`

Description: Get all the cookies for the current domain

```
ex:  Set<Cookie> cookieList1 = driver.manage().getCookies();

     for(Cookie getCookies : cookieList1)
       {
       System.out.println(getCookies);
       }
```

Delete Cookie with name:

Method name: deleteCookieNamed(java.lang.String name)

Syntax: `public void deleteCookieNamed(argo);`

Description: Delete the name cookie from the current domain

```
ex: driver.manage().deleteCookieNamed("firefoxCookie");
```

Delete Cookie :-

Method name: deleteCookie(cookie)

Syntax: `public void deleteCookie(argo)`

Description: Delete the cookie from the current domain

```
ex: driver.manage().deleteCookie(cookie);
```

Delete All Cookies :-

Method name: deleteAllCookies()

example: driver.manage().deleteAllCookies();

Syntax: public void deleteAllCookies()

Description: It will delete all the cookies for the current domain

Example Program:-

```
public class CookiesDemo
{
  public static void main(String[] args){
   WebDriver driver = new FirefoxDriver();
   driver.manage().window().maximize();
   // add cookie
    Cookie cookie = new Cookie("abc", "123");
   driver.manage().addCookie(cookie);
   // get all cookies
   Set<Cookie> cookies = driver.manage().getCookies();
    for(Object object : cookies)
        {
        System.out.println("Cookie name is: "+object);
        }
```

```
// get Named cookie
Cookie name = driver.manage().getCookieNamed("-ga");
System.out.println("Cookie name is:" + name);
// delete all cookies
driver.manage().deleteAllCookies();

//delete cookie
driver.manage().deleteCookie(cookie);

// delete CookiedNamed
driver.manage().deleteCookieNamed("abc");

  }
}
```

Frameworks

Framework :- A framework is a semi-completed and reusable component and based on our requirements, we can customize it.

A Software framework in computer programming is an abstract in which common code providing generic functionality can be selectively. Frameworks are a special case of software libraries in that they are reusable abstractions of code wrapped in a well defined application programming interface (API), yet they contain some key distinguishing features that separate them from normal libraries.

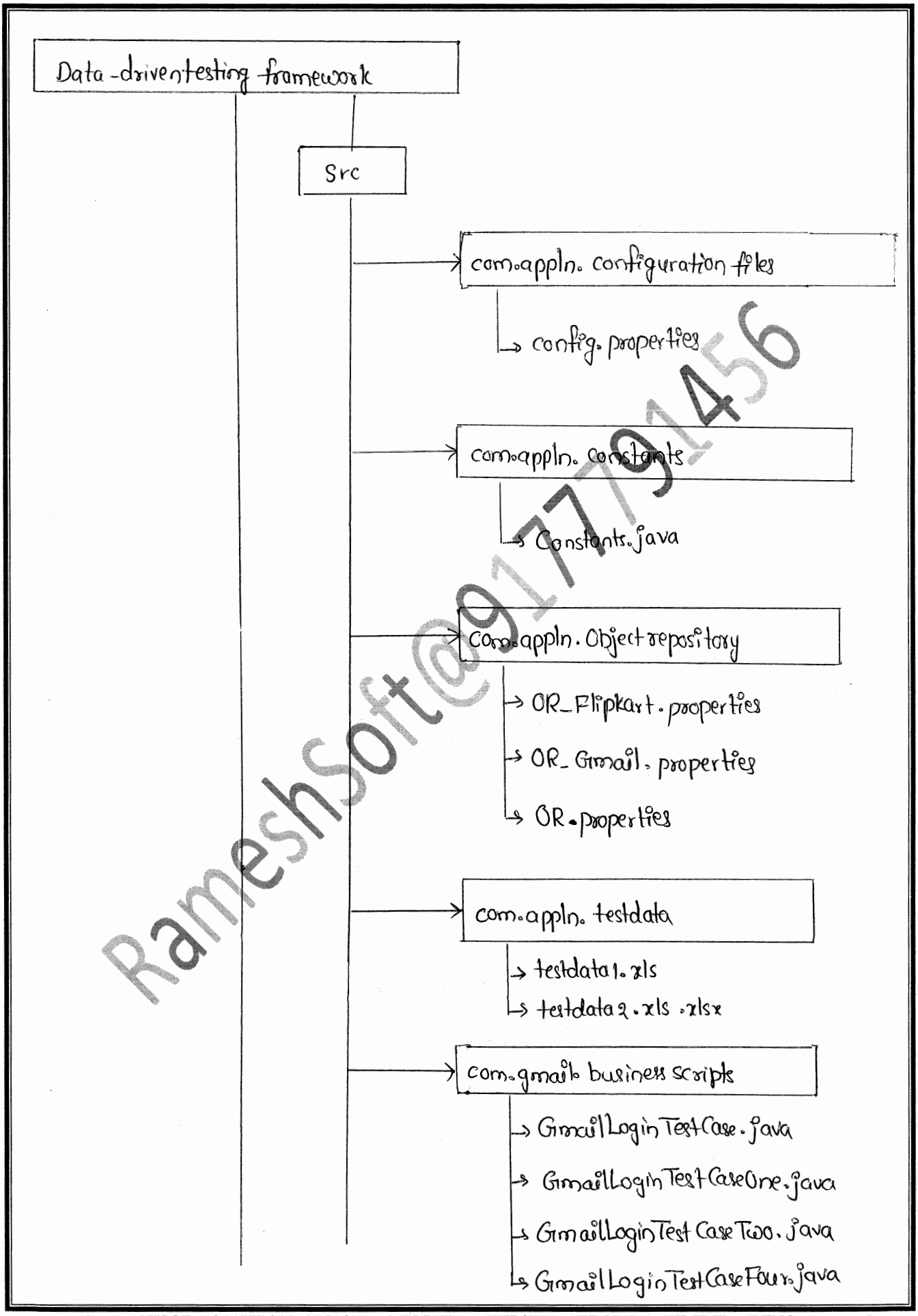
Data-driven testing framework :-

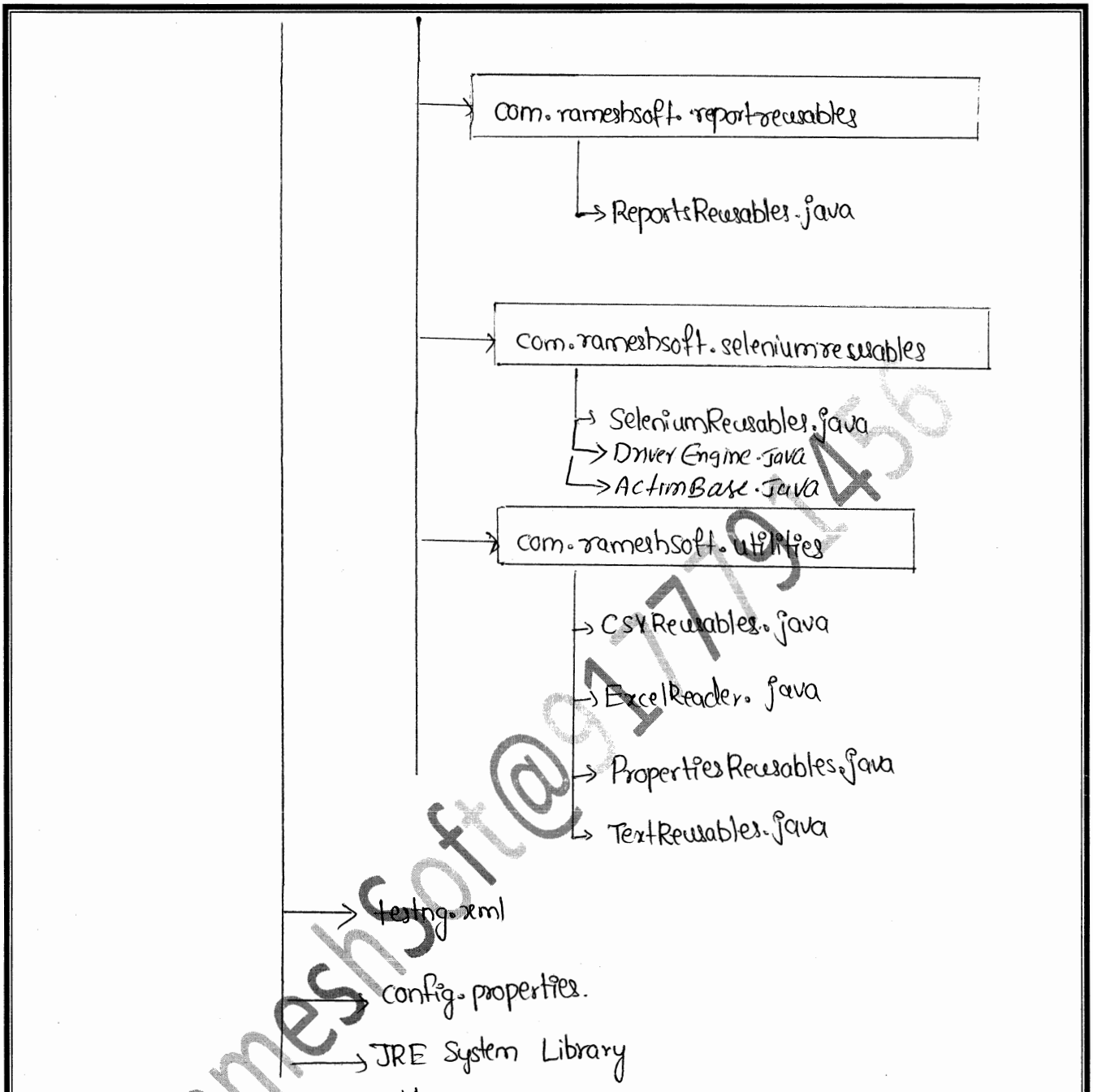
Data Driven Testing Framework is creation of test scripts where test data and/or output values are read from data files instead of using the same hard-coded values each time the test rule.

Structure of Data Driven Testing Framework :-

com.rameshsoft.selenium reusables :-

This package contains reusable functions for selenium in the form of .java files.





com.rameshsoft.reports.reusables:—

In this package, we are going to maintain all the reusable functions for reports in the form of .java files.

com.appln.ObjectRepository (or) com.project.objectRepository:—

In this package, we are going to maintain webelement identifications in the form of .properties files. we can maintain any no of properties files in this package.

com.project.businessscripts :—

In this package, we are going to maintain all our testcases in the form of .java files, one class for one test case.

config.properties:—

In this file, we are going to maintain environmental variables like urls and usernames and passwords etc.

testng.xml:—

By using this xml file, we can execute bulk of test cases.

com.project.testdata :—

In this package, we are going to maintain test data in the form of .xls and .xlsx files.

com.project.constants:—

In this packages, we maintain the all the Filepaths in the form of Strings in .java class.

com.project.utilities :—

In this package, we are going to maintain reusable functions for all properties, excel files, csv files and text files.

How to implement Data Driven Testing Framework :—

we will discuss more elaboratively in the class room in real time.

Page Object Model Framework

Page object model framework has now a days become very popular test automation framework in the industry and many companies are using it because of its easy test maintenance and reduces the duplicate of code.

What is POM :-

* Page Object Model is a design pattern to create Object Repository for web UI elements.
* Under this model, for each web page in the application there should be corresponding page class.
* This 'Page' class will find the web elements of that web page and all contai-ns 'Page' methods which perform operations on those web elements.

The main advantage of Page Object Model is that if the UI changes for any page, it doesn't require us to change any tests, we just need to change only the code within the page objects (only at one place).

Advantages of POM :

1) There is a clean separation between test code and page specific code such as locator (or their use if you are using a UI map) and layout.

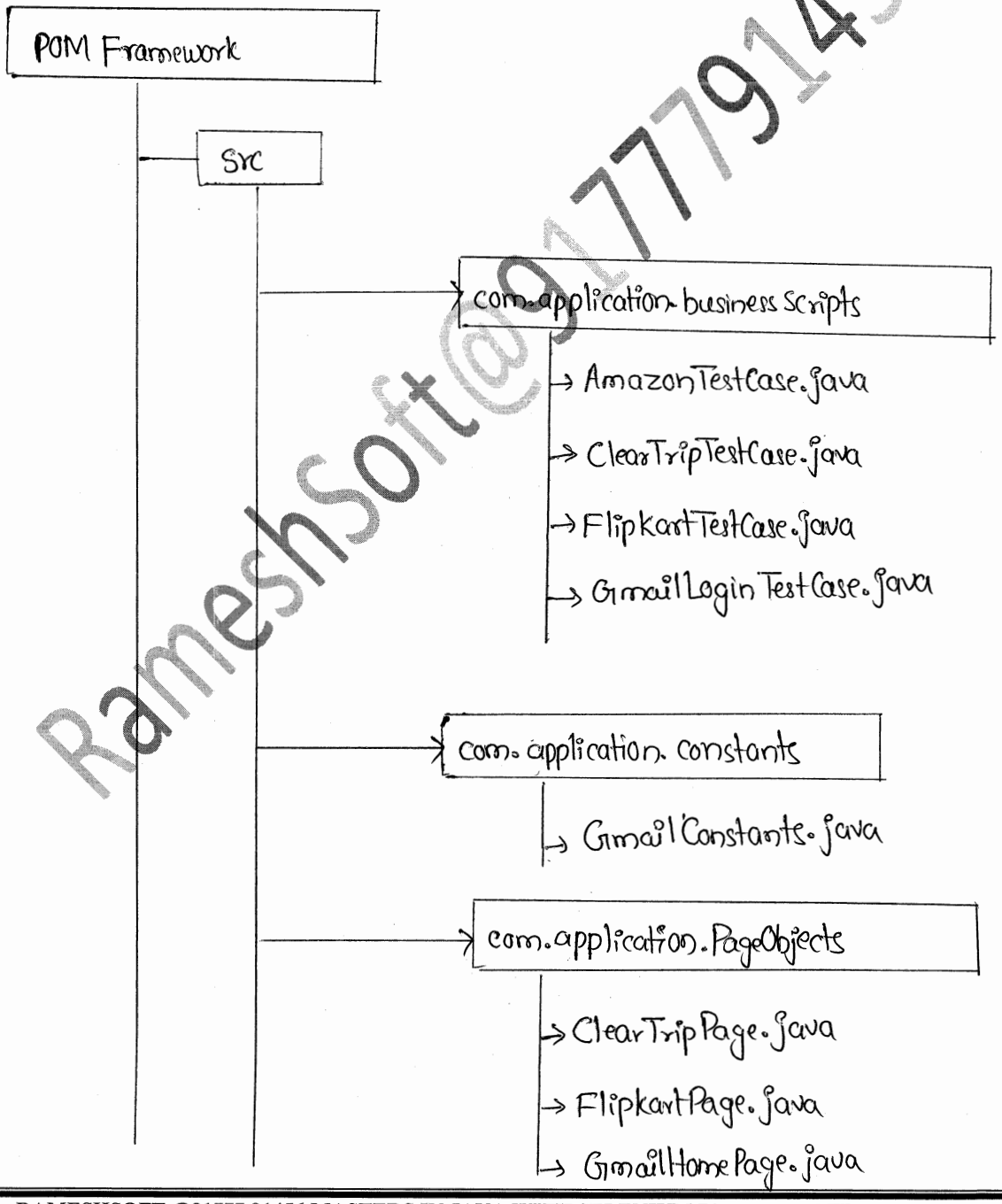
2) If the UI changes for any page, it doesn't require us to change

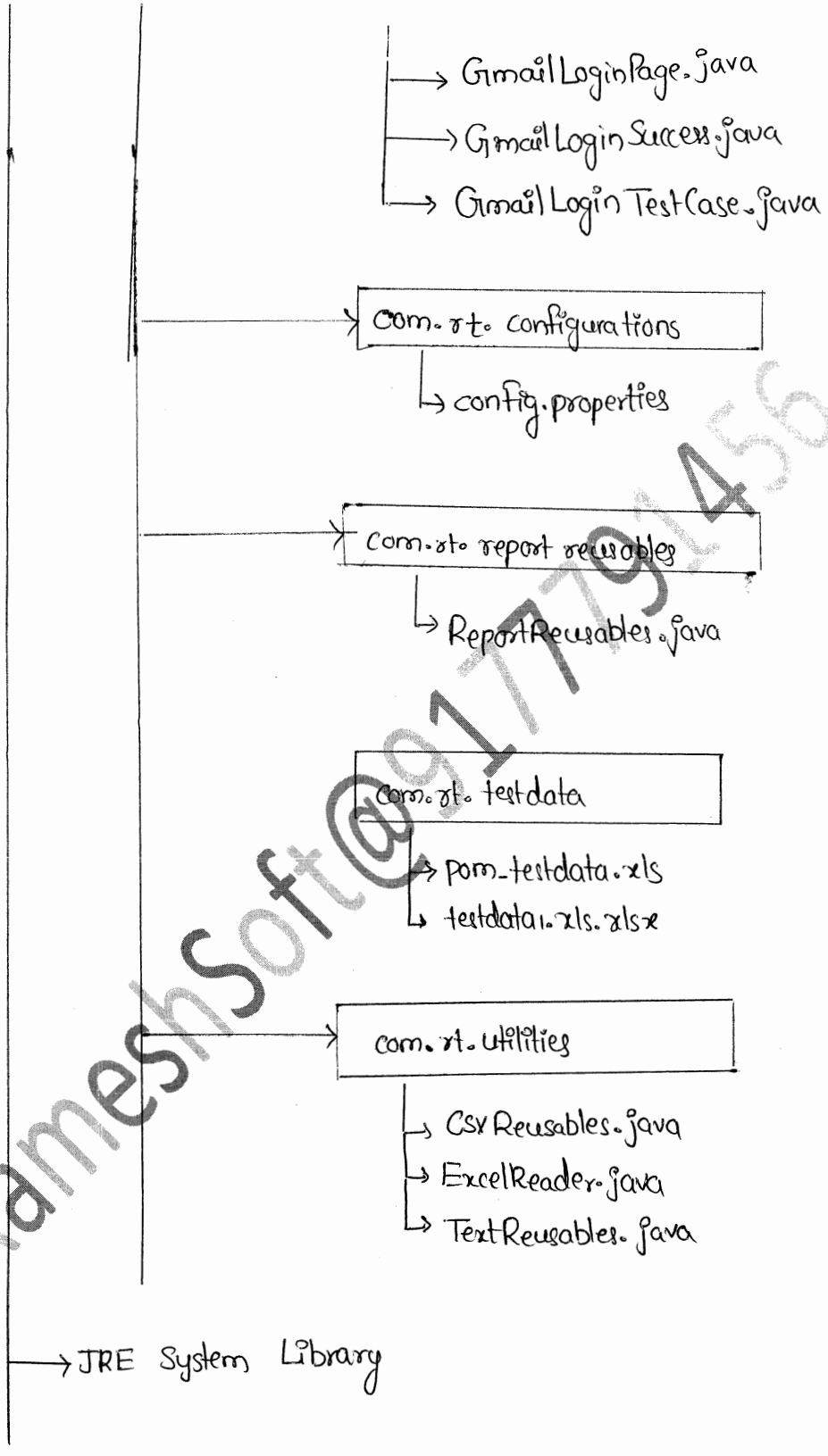
any tests, we just need to change only the code within the page.

3) Code becomes less and optimized because of the reusable page methods in the POM classes.

4) POM makes our code cleaner and easy to understand.

Structure of Page Object Model Framework :-





How to implement Page Object Model Framework:—

We will discuss Frameworks more elaboratively in the class room in Real Time.

**** WE WILL DISCUSS MORE TECHNICAL DISCUSSIONS IN THE CLASS ROOM ****

Handling Date Pickers and Webtabels:-

```
public class DatePickerDemo {
  public static void main(String[] args) throws InterruptedException {
    Date date = new Date();
    String expectedDate = date.getDate() + " ";
    WebDriver driver = new FirefoxDriver();
    driver.manage().window().maximize();
    driver.manag().deleteAllCookies();
    driver.manage().timeouts().implicitlyWait(59, TimeUnit.SECONDS);
    driver.get("https://www.jquerygui.com");
    driver.findElement(By.linkText("Datepicker")).click();
     driver.switchTo().frame(0);
    driver.findElement(By.id("datepicker")).click();
      Thread.sleep(2000);
List<WebElement> dates = driver.findElements(By.xpath(".//*[@id=
                                    "ui-datepicker-div"]/table/tbody/tr/td]"));
    /* for (WebElement webelement : dates)
       { System.out.println(WebElement.getText());
       }
    */
```

```
for(WebElement webelement : dates)
 { if(webelement.getText().equalsIgnoreCase(expectedDate))
  {webelement.click();
    break(); }// if block
  }//for loop
 Thread.sleep(3000);
 }
}
```

If we want to get disabled dates, then we have to write xpath as:

```
driver.findElement(By.xpath("//*[contains(@class,"ui-state-disabled")]"));
```

How do we perform window Authentication in selenium:—

* we can perform or handle window authentication in two ways.

1. Using selenium webDriver
2. Using Autoit tool.

Using Selenium WebDriver:—

Syntax: http://username:password@url

ex:-
```
public class BrowserAuthenticationDemo {
    public static void main(String[] args) throws Exception {
     WebDriver driver = new FirefoxDriver();
     driver.get("http://seleniumrealtime:9177791456@www.rameshsoft.com");
      Thread.sleep(2000);
     driver.quit(); }}
```

```
driver.get("https://rameshselenium.blogspot.in/");
 try {
   driver.findElement(By.id("a")).click();
    }catch(Exception e)
      {
      takeScreenShot();
      }
public void takeScreenShot() throws Exception
{
File file=((TakeScreenShot)driver).getScreenShotAs(OutputType.FILE);
FileUtils.copyFile(file, new File("c:\\test\\failed.png");
}
}
```

Internationalization in Selenium WebDriver :—

```
public class InternationalizationDemo{
  public static void main(String[] args){
    FirefoxProfile profile=new FirefoxProfile();
    profile.setPreference("intl.accept-languages","s1");
    WebDriver driver=new FirefoxDriver(profile);
    driver.get("https://www.google.com");
    driver.findElement(By.name("q")).sendkeys("Selenium");
    driver.findElement(By.name("btnG")).click();
  }}
```

How to take screenshots @remote in selenium

```
public class RemoteScreenShotDemo {

    public static void main(String[] args) throws Exception {

    WebDriver driver = new RemoteWebDriver(new URL("http://localhost:
                4444/wd/hub"), DesiredCapabilities.find());

     driver.get("https://rameshselenium.blogspot.in/");
  /* Remote WebDriver doesn't implement TakeScreenShot, then Augumentor
   will add the TakeScreenShot methods to the instance. */

   WebDriver augumentedDriver = new Augmenter().augument(driver);
  File screenShot = ((TakeScreenShot)augumentedDriver).getScreenShotAs(
                                OutputType.FILE);

    }
}
```

How to take screen shot on failure in Selenium:-

```
public class TestScript {

    public WebDriver driver;

     @Test
      public void testScript() {

        driver = new FirefoxDriver();
```

**

How to launch IE & chrome browser without System.setProperty()

Let us consider for example take the IE browser, the very first time if we will launch without System.setProperty(..) we will get Exception.

```
Ex:- public class IEBrowserTest{
        public static void main(String[] args){
        InternetExplorerDriver driver = new InternetExplorerDriver();
        driver.get("https://rameshselenium.blogspot.in");
        }}
```

when we execute the above code, selenium will throw an exception

Exception in thread "main" java.lang.IllegalStateException: The path to the driver executable must be set by the webdriver.ie.driver System property; for more information.

The Exception clearly says that the path to the driver must be set by the webdriver.ie.driver System Property. this error typically caused by either not having the required IEDriverServer.exe on your local machine or not having it setup in your PATH environment variable.

How to set path environment variable for IEBrowser:-

Open Control Panel

↓

System or security

↓

Right click on 'My Computer'

↓

properties

↓

Advanced system settings

↓

Under advanced tab choose the environment variables option

↓

now we need to specify the location in the PATH variable

↓

Select Path

↓

Edit (choose edit option)

↓

Add location of IEdriver.exe path (where IEDriver is downloaded)

↓

click ok

*Now try to run the IEBrowser without specifying System.setProperty() in selenium script and it will launch IEBrowser successfully now.

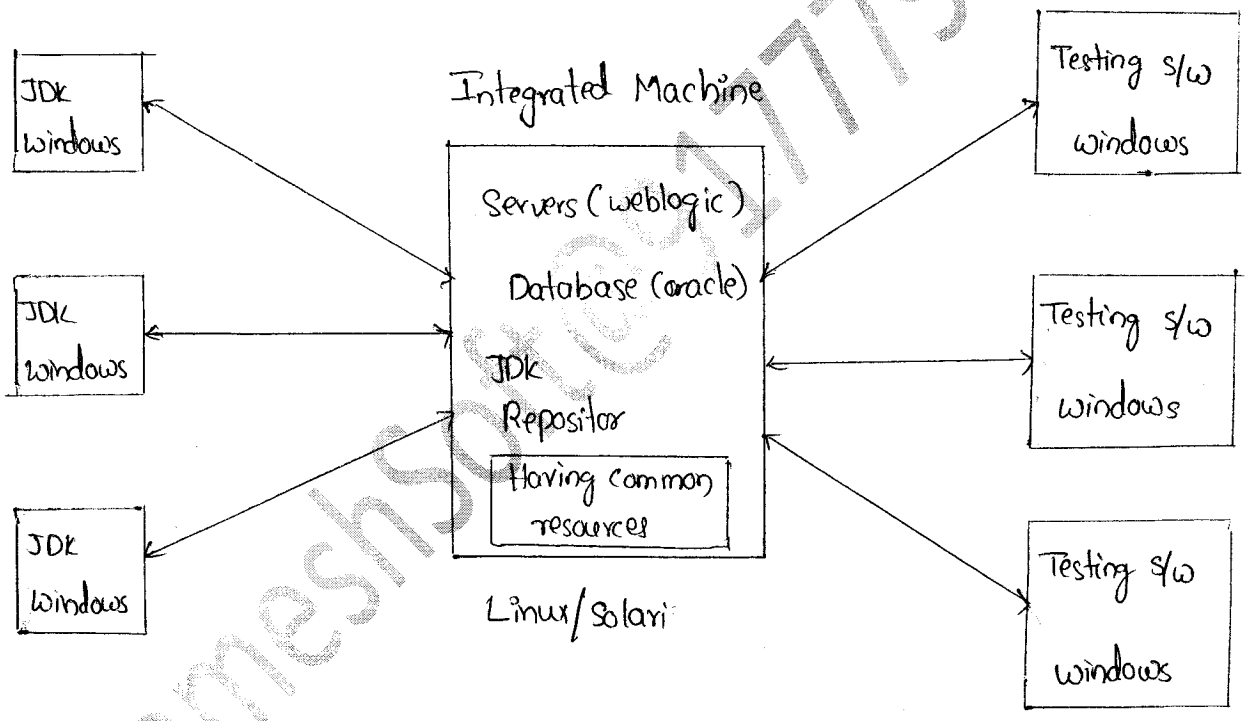
ex:—

```
public class IEBrowserTest {
    public static void main(String[] args) throws Exception {
        try
        {
            InternetExplorerDriver driver = new InternetExplorerDriver();
            Thread.sleep(2000);
            driver.get("http://rameshselenium.blogspot.in");
        } catch (Exception e)
        {
            e.printStackTrace();
        }
}}
```

SVN

Introduction: In Real time all developers and testers machine will be there running in windows environment but all these machines will be connected to a common machine of a company called integrated machine.

This integrated machine contains the common software that are required for multiple project of a company.



*During development mode of the project, the project will be maintained in the cvs repository or svn repository to make the resources of the project visible and accessible for all developers of the project.

Definition of SVN :- SVN is a version control system using it developers can record the history of their source files.

This is also called as source code management.

* The SVN repository keeps tracks of various operations that are done in the files by developers by accessing the files from SVN repository.

Benefits :-

① All the code changes are tracked.

② Supervisor can see how the code evolved overtime.

③ Make it easy to backup your source code.

④ code changes across several developers can be synchronized.

Repository :- Repository is a place where it maintains the data or information.

* Repository (or) subversion is a centralized system for sharing information and which is a central store of data.

* The repository stores information in the form of file system tree. i.e A typical hierarchy of files and directories.

* Any number of clients connect to the repository and then read or write to these files.

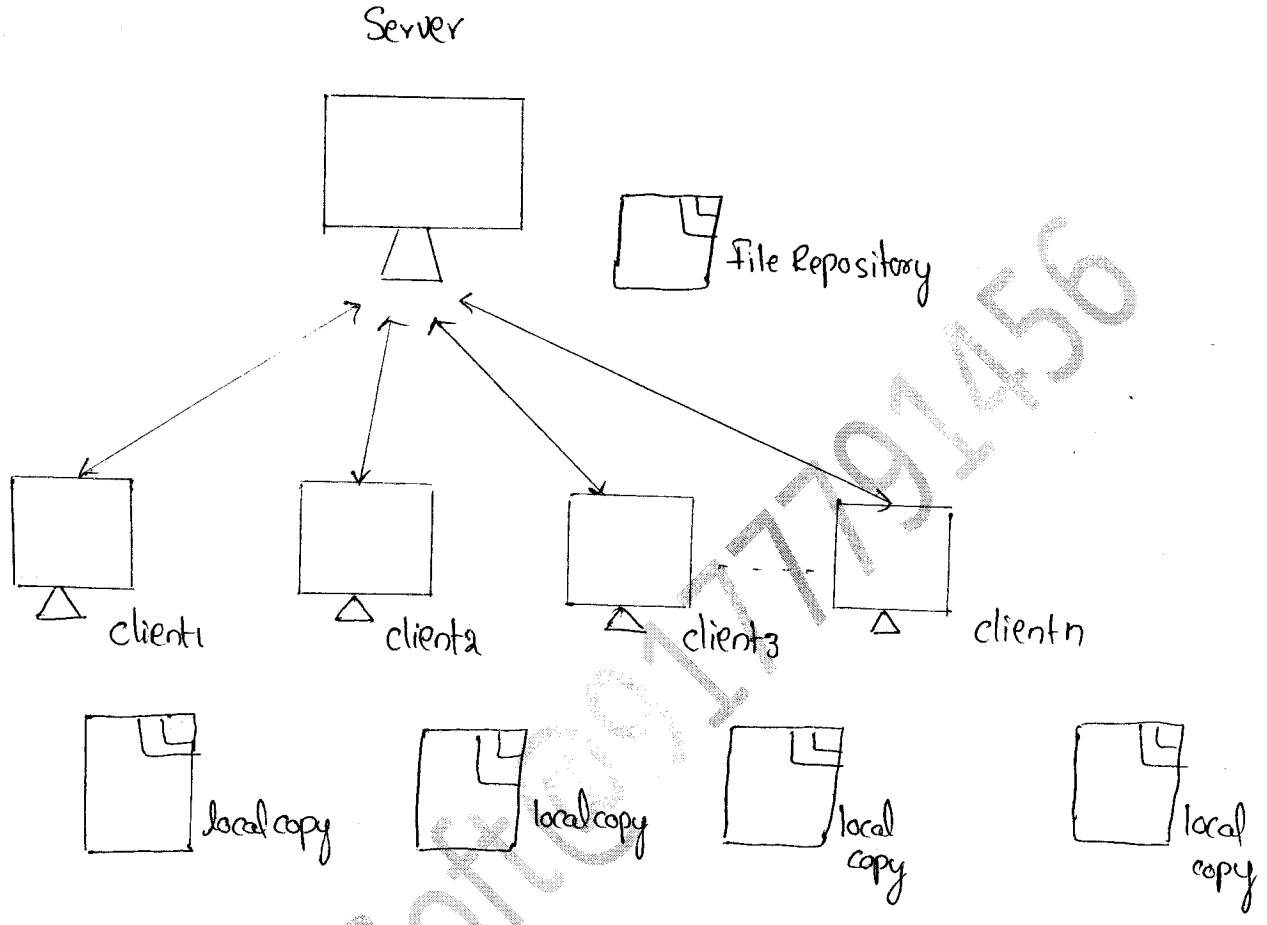
Merge :- Combining changes between two versions of the same file.

Hello.java (Origional)

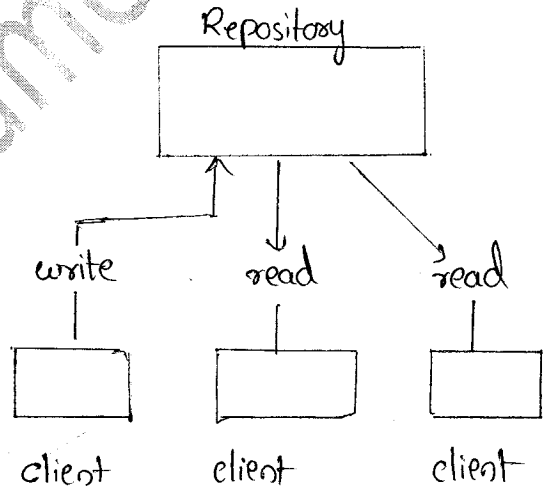
Hello.java 1.1 (after first modification)

Hello.java 1.2 (after second modification)

History :- shows a list of commit message, times and it shows who committed for a particular file.



* By writing data client makes the information available to others. by creating data, the client receives information from others



Must know :-

Repository :- Repository is a place or area on the server where the files are stored.

Sandbox :- A local copy of the code which you work on and then commit to the repository

checkout :- Process of collecting resource or project from SVN repository

update :- Getting code changes that have been committed since you checkedout the project.

check in / commit :- Process of keeping ressources back into repository after doing modifications called as commit operation or check in operation.

Regular Expressions

Regular Expressions represents a group of Strings according to a partic-ular pattern.

While functional test design, testers are using regular expressions as optional. In general regular expressions are used to match with type of values. Due to this reason, testers are using regular expressions to specify expected type of fields in Equivalence Class Partition (ECP).

* Regular Expressions are coded Strings that define an infinite number of possible matches.

Examples:

1) we can write a regular expression to represent all valid email id's.

2) we can write a regular expression to represent date formats. all valid.

3) we can write a regular expression for mobile number all valid matches

The main important application areas of Regular expressions are

i) To develop validation frameworks

ii) To develop pattern matching applications.

iii) To develop communication protocol ... etc.

Pattern class :-

* A Pattern object is compiled representation of regular expression.

i.e pattern object is Java equivalent from of regular expression.

* we can create a Pattern object by using compile() of Pattern class.

Syntax: public static Pattern compile (String re)

Ex Pattern pattern = Pattern.compile("xy");

Matcher class

* A Matcher object can be used to match the given pattern in the target string.

* we can create Matcher object by using match() of Pattern class.

Syntax: public Matcher matcher (String target)

ex: Matcher match = pattern.matcher("xyxyyyxx");

Methods of Matcher class

1) boolean find():-

It attempts to find next match and returns true if the match is available. otherwise it returns false.

2) int start() : This method returns the starting index of matched pattern.

3) int end() :

This method returns end+1 index of matched pattern

4) String group() :- This method returns matched pattern.

Some of Regular Expressions :-

character classes :-

[abc] → one character either 'a' or 'b' or 'c'

[^abc] → except 'a' or 'b' or 'c'

[ ] → one position

[a-z] → one character in lower case from a to z

[A-Z] → one character in upper case from A to Z

[a-z A-Z] → one character in lower case or upper case.

[0-9] → one digit from 0 to 9

[a-z A-Z 0-9] → Any alphanumeric character.

[a-z][A-Z] → First character as lower case and second character as upper case.

[^a-z A-Z 0-9] → Any special character.

{ } → specifies no. of positions.

[KLM][a-z] → Any character K or L or M and any lower case letter.

[0-9]{4} → Numerics with 4 digits only

[0-9]{2,4} → Two digits to Four digit number

[0-9]{,4} → No digit to four digit number

[0-9]{4,} → min 4 digit to infinit digit number

[0-9]+ → one digit to infinite number.

[0-9]* → No digit to infinite digit number

[0-9]? → No digit or one digit number.

[A-Z][a-z]* → first character upper char and second character is no or more lower character.

[A-Z][a-z]+ → first Character uppercase and second character is one or more lower case character.

pre-defined character classes

\s → space character

\S → Any character except space

\d → Any digit from 0 to 9

\D → Any character except digit

\w → Any word character [a-z, A-Z 0-9]

\W → special character

. → any character including special character also

ex

```
Pattern p = Pattern.compile("x");
Matcher m = p.matcher("aqc@z8#l4");
while(m.find())
{
System.out.println(m.start()+"---"+m.group());
}
```

x=[abc]	x=[^abc]	x=[a-z]	x=[0-9]	x=[a-z A-Z 0-9]
0---a	1----9	0---a	1---9	0---9
2---c	3---@	2---c	5---8	1---9
	4----z	4---z		2---c
	5---8	7---l		4---z
	6---#			5---8
	7---l			7---l

x=[^a-z A-Z 0-9]

3--@
6---#

Q write a regular expression to represent all valid 10 digit mobile number.

Rules/conditions:-

1. It should contain exactly 10 digits
2. should starts with 7 or 8 or 9

[7-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]

(or)

[7-9][0-9]{9}

10-digit (or) 11-digit number:

0?[7-9][0-9]{9}

Q:- write a regular expression to represent all valid mail id's.

[a-zA-Z0-9][a-zA-Z0-9._]*@[a-zA-Z0-9]*([.][a-zA-Z]+)+

Note: To alternate more than one regular expression, we use pipe (|)

ex: (regular expression1)|(regular expression2)

ex cat|dog , [0-9]|([1-7][0])

** We will discuss more in the class room ***

Agenda

1) What is database?

2) Why database?

3) Types of Database? DBMS?

4) What is SQL?

5) DML, DDL, DCL, TCL Commands?

Database :-

It is an organised collection of inter related data.

* Once data stored in a database it can be shared, integrated, simultaneously access number of users.
* It is the collection of schemas, tables, queries, reports, views and other objects.

Why Data base :-

Before Databases, A flat files are used as Backend for storing the data.

* With files as a data repository, the following problems are occured.

File:- A flat file is a file which is not related to any particular programming language. ex:- .txt file.

Problems with files :-

1) Data redundency :- Sometimes we are maintaining multiple copies of

Same data in different locations these data is also called as redundant data.

2) Data inconsistency :- In flat file mechanism where we are modifying data in one file it is not affected in another location/file, this is called inconsistency.

3) Security :- A file can be opened directly without authentication. so files are not secured. Flat files doesn't support security mechanism.

4) Complex :- To work with files we need some programming code to open a file, to close a file, to read/write the data we need to write programming statements. So files are complex to work.

To overcome above problems we go for Database.

Advantages of Database over flat file :-

1) It help's to avoid data duplication and reduce data redundancy
2) Greater data integrity and independence from applications programs.
3) Data bases provide security, you can control the security like you can set up the permission on different levels where only specified users can add, update or delete the data.
4) To retrive/fetch data from database we are using SQL (Structured query language). ... etc.

DBMS :-

* A Data Base Management System (DBMS) is a collection of programs (s/w) written to manage database.
* A DBMS is computer software application that interacts with user, other applications, and the database itself to capture and analyze data.
* A general purpose of DBMS is designed to allow definition, creation, querying, update and administration of databases.

RDBMS (Relational database Management Systems) are the most common database systems. They include database like SQLServer, Oracle, MySQL, Sybase and Microsoft SQL Server etc.

* RDBMS is the most common of all the different types of Databases.

Types of Databases

1) Relational Databases: This is most common of all the types of databases, In this, the data in a relational database is stored in various data tables. Each table has a key field which is used to connect it to other tables. Hence all the tables are related to each other through several key fields. Examples of relational databases are Oracle, Sybase and Microsoft SQL Server ... etc.

2. Operational Database:-

In its day to operation, an organization generates a huge amount of data. such as inventory management, purchases, transactions and financials. All this data is collected in a database which is often known by several names. such as operational/production database, subject-area database (SADB) or transaction databases.

An operational database is usually hugely important to organizations as they incl-ude the customer database, personal database and inventory database ie the details of how much of a product the company has as well as information on the customers who buy them. The data stored on operational databases can be chang-ed and manipulated depending on what the company requires.

3. Database Warehouses:-

Organizations are required to keep all relevant data for several year. In the US it can be as long as 6 years. This data is also as important source of inf-ormation for analysing and comparing the current year data with that of the past years which also makes it easier to determine key trends taking place. All this data from previous years are stored in a database warehouse. Since data stored has gone through all kinds of screening, editing and integr-ation it doesn't need any further editing or alteration.

With this database ensure that the software requirement specification (SRS) is formally approved as a part of the project quality plan.

4. Distributed Database:- Many organizations have several office locations. manufacturing plants, regional offices, branch offices and a head office, at different geographic locations. Each of this work groups may have their own database which together will form the main database of the company This is known as a distributed database.

5. End-user database:- There is a variety of data available at the workstation of all the end users of any organization. Each workstation is like a small database in itself which includes data in spreadsheets, presentations, word files, note pads and downloaded files. All such small databases form a different type of database called the enduser database.

RDBMS vs DBMS

Although DBMS and RDBMS both are used to store information in physical database but there are some remarkable differences between them.

DBMS	RDBMS
1. DBMS applications store data as file.	RDBMS applications store data in tabular form.
2. In DBMS, data is generally stored in either a hierarchical form or a navigational form.	In RDBMS, the tables have an identifier called primary key and the data values are stored in the form of tables.

3. DBMS doesn't apply any security with regards to data manipulation.	RDBMS defines the integrity constraint for the purpose of ACID (Atomocity, Consist-ency, Isolation and durability) property.
4. DBMS doesn't support distributed database.	RDBMS supports distributed database.
5. DBMS uses file system to store data so there will be no relation b/w tables	In RDBMS, data values are stored in the form of tables, so a relationship b/w these data values will be stored in the form a table as well.
6. Example of DBMS are file systems and xml.. etc	Examples of RDBMS are mysql, postg-re, sql server, oracle etc

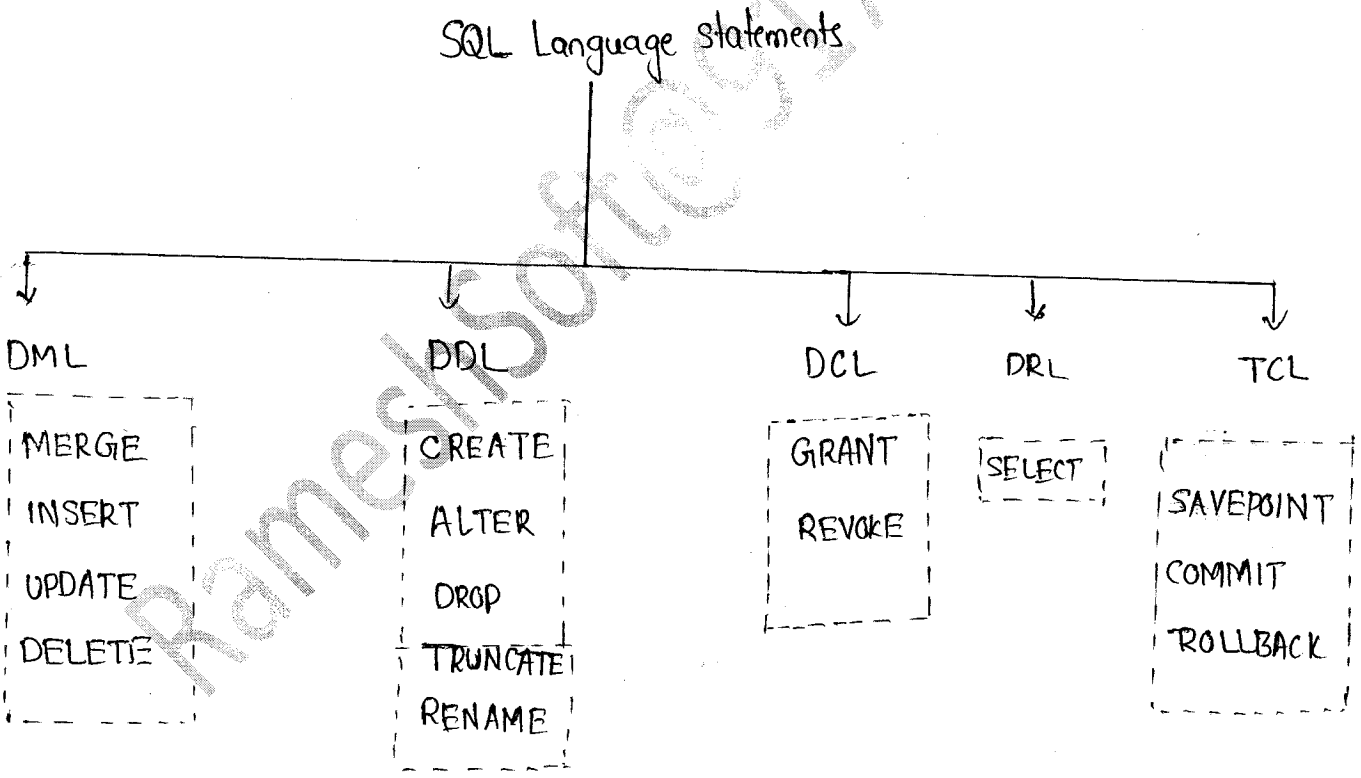
Structured Query Language

* SQL pronounced as S-Q-L or sometimes as see-quel.
* SQL is a language for dealing with Relational Databases.
* SQL programming can be effictively used to insert, search, update delete database records.
* SQL language is diveded into four types of primary language statem-ents; DML, DDL, DCL and TCL.

Using these statements, we can define the structure of a database by creating and altering database objects, and we can manipulate data in a table through updates or deletions.

The five main categories of SQL statements are:

1) DML (Data Manipulation Language)
2) DDL (Data Definition Language)
3) DCL (Data Control Language)
4) TCL (Transaction Control Language)
5) DRL (Data Retrival / query language)



DDL (Data Definition Language)

DDL statements are used to alter/modify a database or table structure and schema. These statements handle the design and storage of database objects.

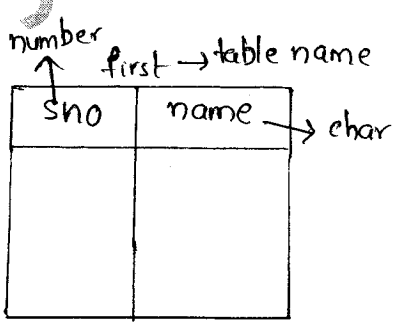
① CREATE :-

It is used to create database objects like table, view, index, sequence...

syntax: Create table tablename (column1 datatype(size), ...);

ex:- create table first (sno number(4), name varchar2(10)); ↵

table is created.

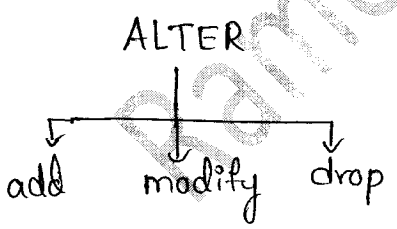


to view structure of the table :-

syntax:
sql > desc tablename;
sql> desc first; ↵

② ALTER :-

It is used to change existing table structure.



i) add :- It is used to add number of columns into the existing table.

Syntax: sql> alter table tablename add (column1 datatype(size), ...);

ex: sql> alter table first add (Age number(2));

* first table is already exist and then we added one column Age.

ii) modify :-

It is used to change column data type or column data type size only.

Syntax : sql> alter table tablename modify (column_name <new datatype>);

ex: alter table first modify (sno date);

sql> desc first

iii) drop :- It is used to remove columns from table.

method1: If you want to drop a single column at a time without using paranthesis then we are using following syntax.

syntax : alter table tablename drop column column_name;

ex: sql> alter table first drop column sno;

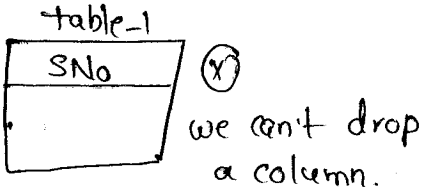
method2:- If you want to drop a single or multiple columns the we are using following syntax.

syntax : alter table table_name drop (column1, column2 ...);

ex: sql> alter table first drop (name); ✓

sql> alter table first drop name; x

Note : we cannot drop all columns in a table (if atleast one column means we can't but we can drop a table).



③ DROP :-

It is used to remove database objects from database.

Syntax: drop object-type objectname;

sql> drop table tablename;

sql> drop view viewname;

sql> drop procedure procedurename;

ex sql> drop table first

To drop temporarily :-

sql> drop table first ↵ (dropped into recyclebin)

sql> flashback table first to before drop; ↵ (get back from recyclebin)

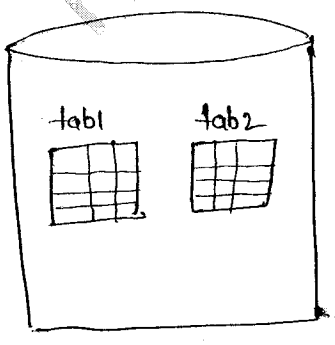
sql> desc first;

To drop permanently :-

sql> drop table first purge; ↵ permanently deleted from database.

sql> flash back table first to before drop; ↵

error: object not in recyclebin



Oracle DB

sql> drop table tab1;

tab1 is dropped into recycle bin

sql> drop table tab2 purge;

tab2 dropped / deleted permanently.

④ TRUNCATE :-

This command is used to delete all rows from table permanently and here structure (columns) will remain same.

Syntax : truncate table tablename;

ex sql> create table first as select * from emp; (Creating duplicate table)

sql> truncate table first;

sql> select * from first;

no rows selected

sql> desc first ↵

columns will come.

⑤ RENAME :- It is used to rename a table and rename a column also

renaming a table :-

Syntax : rename oldtablename to newtablename;

ex :-

sql> rename first to second; ↵

sql> desc second; ↵

renaming a column :-

Syntax : Alter table tablename rename column columnname to newcolumn-name;

ex : sql> alter table first rename column sno to xyz; ↵

sql> select * from first; ↵

*** Note :- All DDL commands are automatically committed.

Data Manipulation Language (DML) :-

1) INSERT :-

It is used to "insert" data into a table.

method1:

Syntax: insert into tablename values (val1, val2, val3 ...);

ex: sql> create table first (sno number(10), name varchar2(10));

sql> insert into first values (1, 'RameshSoft');

sql> insert into first values (2, 'Selenium Training');

sql> select * from first;

first

sno	name
1	RameshSoft
2	Selenium Training

method2:-

syntax: insert into tablename (col1, col2, ...) values (val1, val2...);

ex: sql> insert into first (sno, name) values (3, 'Real Time Training');

2) UPDATE :-

It is used to modify data in a table.

Syntax: Update tablename set columnname = newvalue where columnn-
-ame = oldvalue ;

ex: update first set sno='8' where sno='2' ; ↵

update first set name = 'Selenium Training!' where name='seleniumTr-
-aining'; ↵

first

sno	name
1	Rameshsoft
6	Selenium Training 1
3	Realtime Training

③ DELETE :- It is used to delete all rows or particular rows from a table.

syntax1 : delete from tablename ; → To delete all rows.

syntax2 : delete from tablename where condition ; → delete particular rows.

ex :- sql> delete from first ;↵

(Also)

whenever we are using delete from tablename, truncate table tablename then total data automatically deleted, but we can get it back data when we are using delete from tablename to delete data because these deleted data temporarily stored in a buffer. we can get it back this data using rollback.

ex :-

sql> delete from first ; ↵

sql> rollback ;

sql> select * from first; ↵

INDIA'S NO1 REALTIME TRAINING INSTITUTE

RAMESHSOFT

100% REALTIME TRAINING

100% PLACEMENTS

TILL NOW

300+

STUDENTS GOT PLACED

TRAINER NAME : RAMESH ANAPATI

CERTIFIED AND REAL TIME EXPERT

Opposite Of Vindu Tiffin Center Between Canara Bank And Axis Bank 3rd Floor
Near Umesh Chandra Statue SR Nagar Sarala Apartments,HYDERABAD

PH : 9177791456, 9502695908, 040-48572456

Frameworks

- Frameworks in Real Time we will discuss in detailed in classroom.

- More than 30hrs will discuss.

  Note: This notes is not complete notes

Data Retival / Data Query Language (DRL / DQL) :-

SELECT :- This command is used to retrive data from table.

syntax: Select col1, col2, ---from tablename where condition

There are 4 ways to retrieve data

1) Select all columns and all rows (*) :-

- if we are not writing where clause we can get all columns & rows

ex: select * from emp;

2) select all columns & particular rows :- (where condition)

ex: select * from emp where sal > 2000;

3) select particular columns & all rows;

ex: select ename, sal from emp;

4) Select particular columns and particular rows :-

ex: Select ename, sal from emp where sal > 2000;

Data Control Language (DCL) :-

Oracle provides extensive security features in order to safeguard information stored in its tables from unauthorized viewing and damage.

Depending on a user's status and responsibility, appropriate rights on Oracle's resources can be assigned to the user by the DBA. The rights

that allow the use of some or all of oracle's resources on the server, are called PRIVILEGES.

Objects that are created by a user are owned and controlled by that user. If a user wishes to access any of the objects belonging to another user, the owner of the object will have to give permissions for such access. This is called GRANTING of priviliges.

Privileges once give can be taken back by the owner of the object. This is called REVOKING of privileges.

* This privileges are used to perform some operations on the object.

there are :- 1) insert
2) update
3) delete
4) select
5) execute.

GRANT:-

syntax: GRANT <object_privileges> ON <object_name>
To <user_name> [WITH GRANT OPTION]

ex: GRANT ALL ON first TO Ramesh WITH GRANT OPTION

ex: GRANT SELECT, UPDATE, ON first TO Ramesh WITH GRANT OPTION

The user Ramesh has been given permission to view and modify records in table first.

REVOKE :-

This command is used to cancel system or object privileges.

Syntax: REVOKE <object_privileges> ON <object_Name> FROM <User_name>

ex: REVOKE UPDATE ON first FROM Ramesh;

Transaction Control Language (TCL) :-

A series of one or more SQL statements that are logically related, or a series of operation performed on Oracle table data is termed as a Transaction.

* Oracle treats changes to table data as a two step process, first the changes requested are done. To make these changes permanent a COMMIT statement has to be given at the SQL prompt. A ROLLBACK statement given at the SQL prompt can be used to undo a part of entire Transaction.

* A Transaction begins with the first executable SQL statement after a commit, Rollback or Connection made to the Oracle engine. All the changes made to an Oracle table data via a transaction are made or undo at one instance.

Specially, a Transaction is a group of events that occurs between any of the following events:

* Connecting to Oracle
* Disconnecting from Oracle
* Committing changes to the table
* Rollback

Closing Transaction :-

A Transaction can be closed by using either a commit or Rollback statement. By using these statement, table data can be changed or all the changes made to the table data undo.

COMMIT:

A Commit ends the current transaction and makes permanent any changes made during the transaction. All transaction locks acquired on tables are released.

Syntax: COMMIT;

ROLLBACK:-

A Rollback does exactly the opposite of COMMIT. It ends the transaction but undoes any changes made during the transaction. All transaction locks acquired on tables are released.

Syntax: ROLLBACK [WORK] [TO SAVEPOINT] <save_point_Name>

WORK :- It is optional and is provided for ANSI compatibility.

SAVEPOINT :- It is optional & it is used to rollback a partial transaction, as far as specified savepoint.

SAVEPOINTNAME :- It is savepoint created during the current transaction.

SAVEPOINT :-

SAVEPOINT makes and saves the current point in the processing of transaction. When a SAVEPOINT is used with a ROLLBACK statement, parts of a transaction can be undone. An active savepoints in one that is specified since the last COMMIT or ROLLBACK.

syntax : SAVEPOINT <SavePointName>

ex :
```
DECLARE
    Total_sal number(9);
BEGIN
    insert into emp values ('E01' 'Ramesh', 10000);
    insert into emp values ('E02', 'Rameshkumar', 8500);
    SAVEPOINT no-update;
update emp SET salary = salary + 2000 where emp_name = 'Ramesh';
update emp SET salary = salary + 2000 where emp_name = 'Rameshkumar';
SELECT sum(salary) into Total_sal from emp;
if Total_sal > 1500 then RollBack To SAVEPOINT no-update;
end if;
Commit;
end;
```

RAMESHSOFT

SOFTWARE SOLUTIONS

OUR STUDENTS RECENTLY PLACED ON SELENIUM

AHMED	**RAFEEQ**
(SD SOFTECH PVT LTD)	(EDREEZ SOFTWARE PVT LTD)
MADHUPRIYA	**SWATHI**
(CIGNITI TECHNOLOGIES)	(COGNIZANT)
MRUDALA	**GANESH**
(SOFTSOL TECHNOLOGIES)	(PURPLETALK)
MANDEEP	**VISHWAJIT**
(SOCTRONICS)	(TCS)
SIBASYS	**SRIKANTH.R**
(COGNIZANT)	(INFOSYS)
SHRAVAN KUMAR	**SANGAMESH**
(IBM)	(TECH MAHINDRA)
PRAHALAD	**HARISH**
(INFOBRAIN)	(PRDC)
KIRAN	**THAKIL**
(CYBAZE)	(UHG)

Data Types

Java datatypes	Database Types	JDBC types
byte	number/int	Types.TINYINT
short	number/int	Types.SMALLINT
int	number/int	Types.INTEGER
long	number/int	Types.BIGINT
String	varchar2/varchar/text	Types.VARCHAR
double	number/double	Types.DOUBLE/Types.DECIMAL
Date	date	Types.DATE

* Types is a class from java.sql package.
* Types class has defined mediator types b/w java datatypes and database datatypes.
* CollableStatement of JDBC can understand only JDBC 'Types'. So while registering out parameters, we should inform an out parameter type with Types class only.

CONSTRAINTS

* Constraints are used to prevent invalid data entry into our tables
* constraints are created on table columns.
* Oracle server supports following types of constraints.

1) NOT NULL
2) UNIQUE
3) PRIMARY KEY
4) FOREIGN KEY
5) CHECK

* All the above constraints defined in two levels.

i) column level constraints
ii) table level constraints.

① column level constraints:—

In this method we are defining constraints on individual columns. i.e whenever we are defining the columns, immediately we are using constraint types.

syntax: create table tablename (columnname1 datatype(size) constraint type,
columnname 2 datatype (size) constraint type...);

② table level constraints:—

In this method we are defining constraints on group of columns i.e here we are first defining all columns and then last only we

must specify group of columns along with constraint type.

Syntax: create table tablename (column1 datatype, column2 datatype---
constraint type (column1, column2 ---));

NOT NULL:- NOT NULL constraint doesn't support table level.

→this constraint doesn't accepts null values but it will accept duplicate values.

Column level:-

```
sql> create table t1(sno number(10) not null, sname varchar(10));
sql> insert into t1 values (1, 'Ramesh');
sql> insert into t1 values (null, 'Rameshsoft'); // error ORA-1400; cannot insert null
```

UNIQUE:- This constraint defined on two levels.

* this constraint doesn't accepts duplicate values but it will accept null values.
* Also oracle server internally creates Btree indexes on unique constraint columns.

column level:-

```
sql> create table t2 (sno number(10) unique , sname varchar(10));
```

table level:-

```
sql> create table t2( sno number(10), sname varchar2(10), unique(sno, sname));
```

PRIMARY KEY:— Primary key uniquely identifies a record (or) row is a table.

* There can be only one primary key is a table and also primary key doesn't accept duplicate, null values and also internally oracle server creates btree index on primary key columns.

column level:—

create table w1(sno number(10) primary key, name varchar2(10));

table level:—

create table w2(sno number(10), sname varchar2(10), primary key(sno, name));

* This is also called as composite primary key i.e it is the combination of columns as a single primary key.

FOREIGN KEY:—

If you want to establishes relationship between tables then we are using referential integrity constraints foreign key.

* Here one table foriegn key must belongs to another table primary key or unique key and also these columns must belongs to same data type.

column level: foriegn key creation (references):—

create table tablename (any column datatype(size) references master tablename (primary key columnname), column2 datatype(size));

ex: sql > create table a1(sno number(10) references w1(sno));

table level: (foreign key, references)

syntax:-

create table tablename (column1 datatype(size), column2 datatype(size)...,
foreign key (column1, column2) references master tablename(
primary key columns);

ex: sql> create table a3 (sno number(10), name varchar2(10), column3 number(
(10) foreign key (sno, sname) reference w2);

CHECK:-

This constraint is used to define logical conditions according to business rules.

syntax: create table tablename (col1 datatype(size) check (logical condition),
col2 datatype(size) check(logical condition),...);

ex:
sql> create table a4(name varchar2(10) check(name = upper(name)));

Java Database Connectivity (JDBC)

*JDBC is a Java API to connect and execute query with database.

"JDBC API is a Java API, which consists a group of classes and interfaces in the form of packages of Java and it allows a Java applications to connect with a database independent manner."

Why JDBC?

Before JDBC, ODBC API was the database API to connect and execute query with the database. But ODBC API uses ODBC driver which is written in C language (i.e. platform dependent and unsecured). That is why Java has defined its own API (JDBC API) that uses JDBC driver (which is written in Java language).

JDBC API :-

JDBC API is a part of Java Standard Edition (JAVASE) and it is divided into 2 types of packages.

1) java.sql
2) javax.sql

Driver (I) :- The interface that every driver class must implement.

Connection (I) :- A Connection (session) with a specific database.

Statement(I)- The object used for executing a static SQL Statement and returning the result it produces.

PreparedStatement (I) :- An object that represents a precompiled SQL statement.

CallableStatement(I) :- The interface used to execute SQL stored procedures.

DatabaseMetaData (I) :- Comprehensive information about the database as a whole.

ParameterMetaData (I) :- An object that can be used to get information about the types and properties for each parameter marker in a PreparedStatement object.

ResultSet (I) :- A table of data representing a database result set, which is usually generated by executing a statement that queries the database.

ResultSetMetaData (I) :- An object that can be used to get information about the types and properties of the columns in a ResultSet object.

SavePoint (I) :- The representation of savepoint, which is a point within the current transaction that can be referred from the Connection.rollback().

Classes :-

Date(c) :- A thin wrapper around a millisecond value that allows JDBC to identify this as an SQL DATE value.

DriverManager(c) :- The basic service for managing a set of JDBC Drivers.

Time(c) :- A thin wrapper around the java.util.Date class that allows the JDBC API to identify this as an SQL Time value.

Timestamp(c) :- A thin wrapper around the java.util.Date that allows the JDBC API to identify this as an SQL TIMESTAMP value.

Types(c) :- The class that defines the constants that are used to identify generic SQL types, called JDBC types.

SQLException(c) :- An exception that provides information on a database access error or other error.

JDBC Drivers

JDBC Driver is a software component that enables java application to interact with the database.

There are 4 types of JDBC Drivers

1) JDBC-ODBC Bridge Driver (Type-1) [Type-1 driver is removed in java8]

2) Native-API Partly Java driver (Type-2)

3) Network Protocol driver (Type-3)

4) Native-protocol pure Java driver (thin driver) (Type-4)

In real time applications mostly this Type-4 driver is used.

* This Type-4 driver is a pure Java driver, it means a driver software is 100% implemented in Java language.
* Type-4 driver uses a database server native protocol to establish the connectivity between a Java application and a database.
* Using Native protocol the connectivity is faster than using Native-API, so Type-4 driver is faster than other drivers.
* The Type-4 driver is a platform independ, but a database dependent.
* The type-4 driver uses the following information to open a connection with a data base.

i) ip address of server machine
ii) port number of a database server
iii) service id

Here,
ip address is used to identify a machine across a network
port number is used to identify a server/service in a machine (system)
service id is to identify a database in a DB server.

Steps to Connect to the Database

There are five steps to connect any Java application with the Data-base in using JDBC.

1) Register the driver class.

2) Creating connection

3) Creating Statement

4) Executing queries

5) closing Connection

① Register the driver class

The forName() of 'Class' class is used to register the driver class.

This method is used to dynamically load the driver class.

Syntax :—

public static void forName (String className) throws ClassNotFoundExce-ption.

ex:— Class.forName ("oracle.jdbc.driver.OracleDriver); ← register the oracleDriver class.

② Creating Connection / Open Connection :—

To open a connection, we call getConnection() of DriverManager class.

Syntax of getConnection() is as follows,

1) public static Connection getConnection(String url) throws SQLException

2) public static Connection getConnection(String url, String username, String password) throws SQLException

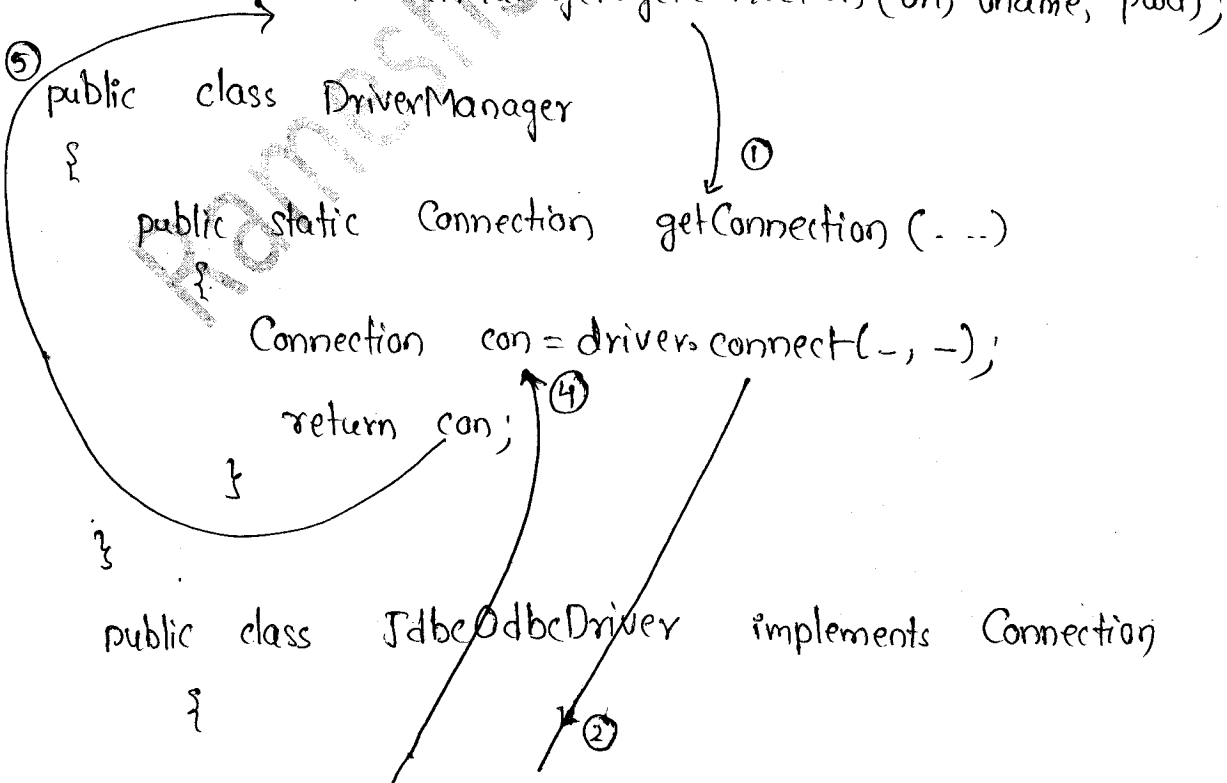
* getConnection() of DriverManager class internally calls connect() of driver class to open a connection with a database.

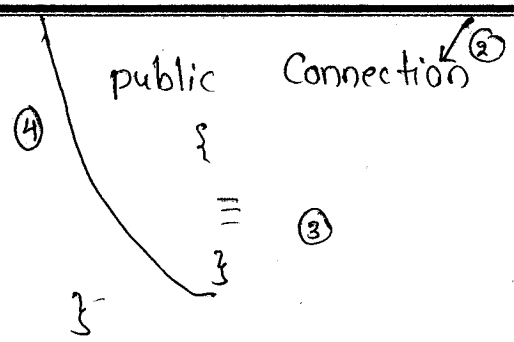
* connect() returns a connection to a getConnection() and then it will return that connection back to Java program.

* getConnection() is a static method of DriverManager class.

Syntax:-

Connection con = DriverManager.getConnection(String url, String uname, String pwd)





* connect() of driver class returns an object of implementation class of Connection interface.
* getConnection() of DriverManager class stores it in a reference variable of Connection (I).
* finally getConnection() returns reference variable back to our java program.
* getConnection() is static factory method.
* The url is used to identify one driver among multiple driver registered with a DriverManager class.

The url of a driver will be supplied by the driver vendor.

③ Creating Statement :-

1) To transfer a sql command from java application to a database or to transfer the result from the database to a java application we need Statement object.

2) we can obtain a statement object by calling createStatement() of Connection interface.

Syntax: Statement stmt = con.createStatement();

public statement createStatement() throws SQLException

Here createStatement() is a non-static factory method.

④ Execute queries / sql commands

To run sql command on a database, we call the following methods of Statement (I).

i) public int executeUpdate("command") → for non-select operation

ii) public ResultSet executeQuery("command") → for select operation

iii) public boolean execute("command") → for both select & non-select operations.

If it is true then select operation is done,

if it is false the non-select operation is done.

ex:- ResultSet rs = stmt.executeQuery("select * from table1");

```
while(rs.next())
{
    System.out.println(rs.getInt(1)+" "+rs.getString(2));
}
```

⑤ Closing Connection :-

By closing Connection object statement and ResultSet will be closed automatically.

The close() of Connection (I) is used to close the connection.

Syntax:

public void close() throws SQLException

ex: con.close();

***

Write a JDBC Program to connect to Database (oracle Database) using Type-4 driver.

```
public class OracleDBTesting{
    public static void main(String[] args){
        Connection con=null;
        Statement stmt;
        String driver = "oracle.jdbc.driver.OracleDriver";
        String url = "jdbc:oracle:thin:@localhost:1521:ORCL";
                                                    ↑service id
        String uname = "system";
        String pwd = "tiger";

        try{
            Class.forName(driver).newInstance();
            con=DriverManager.getConnection(url, uname, pwd);
            System.out.println("Connection is opened");
```

```
        stmt = con.createStatement();

    ResultSet  rs = stmt.executeQuery ("select * from employee");


     while (rs.next())
      {
     System.out.println(rs.getString(1) + "---"+rs.getString(2)+"---"+rs.getInt
                                                          (3));


      }
    } catch(Exception e)
       {
        e.printStackTrace();
       }
    finally {
         try{
            if ((con!=null) && (!con.isClosed()))

                con.close();
              } catch(Exception e)
                {
                 e.printStackTrace();
                }
          } //finally

        }

    }
```

Note:-

* By default Oracle database server is 1521 port number.
* when we are working with Type-4 with Oracle DB we need to add ojdbc14.jar to program.(ie,in our eclipse build path)

② Write a JDBC program to connect to MySQL Database using Type-4 driver.

```
public class MYSQLDBTestingWithType4
 {
  public static void main(String[] args){

    Connection con = null;
     Statement stmt = null;
     String url = "jdbc.mysql://localhost:3306/";
      String driver = "com.mysql.jdbc.Driver";
       String dBname = "testing";

      try {
        Class.forName(driver);
        con = DriverManager.getConnection(url+dBname,"root","root");
        System.out.println(" Database connection is opened");
        stmt = con.createStatement();

  int x = stmt.executeUpdate(" create table student_info(sid int,
                                sname varchar(10), marks int);");

    if (x==-1)
      { System.out.println("table student_info is created");
       }
```

```
stmt.executeUpdate("insert into student_info values(001,'Ramesh',950)");

stmt.executeUpdate("insert into student_info values(002,'kumar',850)");


ResultSet rs = stmt.executeQuery("select * from student_info");

 while (rs.next())
  {

System.out.println(rs.getInt(1)+"---"+rs.getString(2)+"---"+rs.getInt
                                                          (3));

  }
} catch (Exception e)
  {
   e.printStackTrace();
  }
 finally {
      try{
          if ((con!=null)&& (!con.isClosed()))
             con.close();
        }catch(Exception e)
            {
              e.printStackTrace();
            }
      }
   }
}
```

* By default MySQL database server port no is 3306.

Note:- * When we are working with Type-4 with mysql we need to add mysql-connector-java-version-bin.jar to our program. (i.e in our eclipse build path)

** Automate a Gmail Login Test Case, where username and password take from mysql database. with Type-4 driver.

```
public class GmailLoginTest {
    public static void main (String[] args) throws SQLException, Interrup-
                                                    -edException {
        Connection con = null;
        Statement stmt = null;

        String url = "jdbc:mysql://localhost:3306/";
        String dBName = "testing";
        String driver = "com.mygql.jdbc.Driver";
        String username = "";
        String password = "";

    try {
        Class.forName(driver);
        con = DriverManager.getConnection(url+dBName, "root", "root");
        System.out.println(" Database testing is closed :"+con.isClosed());

            stmt = con.createStatement();

            ResultSet rs = stmt.executeQuery("select * from Gmail_info");
```

```
while (rs.next())
 {
System.out.println(rs.getString(1) + "---" + rs.getString(2));

 username=rs.getString(1);

 password=rs.getString(2);

} catch(Exception e)
 {
  e.printStackTrace();

  }

finally {

    con.close();

     }

System.out.println("username  is : " + username);

System.out.println("password is : " + password);

   WebDriver webDriver = new FirefoxDriver();

   webDriver.manage().window().maximize();

  webDriver.manage().deleteAllCookies();

webDriver.manage().timeouts().implicitlyWait(70, TimeUnit.SECONDS);

webDriver.get("https://www.gmail.com");
```

```
webDriver.findElement(By.id("Email")).sendkeys(username);

webDriver.findElement(By.id("next")).click();

    Thread.sleep(2000);

webDriver.findElement(By.id("Passwd")).sendkeys(Password);

webDriver.findElement(By.id("signIn")).click();

webDriver.quit();

}
```

Important methods in Statement (I):-

The object is used for executing a static SQL statement and returning the result it produces.

1. void close():-

Releases this Statement objects database and JDBC resources immediately instead of waiting for this to happen when it is automatically closed.

Syntax: public void close() throws SQLException

2. boolean execute(String sql):-

Executes the given SQL statement, which may return multiple results.

syntax: public boolean execute(String sql) throws SQLException

3. ResultSet executeQuery(String sql):-

Executes the given SQL statement, which returns a single ResultSet object

Syntax: public ResultSet executeQuery(String sql) throws SQLException

4. int executeUpdate(String sql):-

Executes the given SQL statement, which may be an INSERT, UPDATE, or DELETE statement or an SQL statement that returns nothing, such as SQL DDL statement.

Syntax: public int executeUpdate(String sql) throws SQLException

5. boolean isClosed():-

Retrives whether this Statement object has been closed.

syntax: public boolean isClosed() throws SQLException

Important methods in PreparedStatement (I):-

The object is used for executing a static SQL statement and returning the results it produces.

1. ResultSetMetaData getMetaData():-

Retrieves a ResultSetMetaData object that contains information about the columns of the ResultSet object that will be returned when this PreparedStatement object is used.

Syntax: public ResultSetMetaData getMetaData() throws SQLException

2. boolean execute():-

Executes the SQL statement in this PreparedStatement object, which may be any kind of SQL statement.

syntax: public boolean execute() throws SQLException

3. ResultSet executeQuery():-

Executes the SQL query in this PreparedStatement object and returns ResultSet object generated by the query.

Syntax: public ResultSet executeQuery() throws SQLException)

4. int executeUpdate():-

Executes the SQL Statement in this PreparedStatement object, which must be an SQL statement, which may be INSERT, UPDATE, or DELETE or an SQL statement that returns nothing such as an SQL DDL statement.

syntax: public int executeUpdate() throws SQLException)

5. void setDate(int parameterIndex, Date x):-

Sets the designated parameter to the given java.sql.Date value using the default time zone of the virtual machine that is running the application.

Syntax: public void setDate(int parameterIndex, Date x) throws SQLException

6. void setDouble(int parameterIndex, double x):-

Sets the designated parameter to the given Java double value.

Syntax: public void setDouble(int parameterIndex, double x) throws SQLException

***

We Will discuss more in Real time Scenarios in the classroom

***

Selenium Grid

1) What is Grid?

2) Advantages of Grid?

3) Grid Architecture in detail?

4) When to use it

5) what is hub and working with hub in detail?

6) What is node and working with node in detail

7) Configuring hub and nodes.

8) Execution flow of hub, nodes and our scripts.

9) Develop and execute the scripts in different browsers

10) Executing Scripts in multiple systems.

11) Conclusion on Grid in detail.

Multi Threading

Q: What is Multithreading?

Multithreading is a process of executing multiple threads simultaneously.

Q: What is a Thread?

A Thread is a program and a Thread is a light weight sub process.

Q: What is Multitasking?

Multitasking is a process of executing multiple tasks simultaneously.

we can achieve multitasking by two ways:

* Process-based Multitasking (Multiprocessing)
* Thread-based Multitasking (Multithreading)

Process-based Multitasking :-

→ Each process have its own address in memory.

→ Process is heavy weight

→ Switching from one process to another requires time.

→ This is also called Multi processing.

Thread-based Multitasking :-

→ Threads shared the same address space.

→ Thread is the light-weight program

→ Switching between the process is very easy.

→ we can have control over the threads.

→ This thread-based multitasking also called as "Multithreading".

Life Cycle of a thread :-

A Thread can be in five states.

The Life cycle in of a thread in Java is controlled by JVM

The thread is having the following states.

1. New
2. Runnable
3. Running
4. Non-Runnable
5. Terminated. (Halted)