Camlin Page
Date / /

Q. Can we overload static method
Yes.

Q Can we override static method
No, bcause its not bounded to object

Q. How can we prevent method overriding.
If we mark method as Static or Final or private then we can prevent overriding.

Q. Can we override a method which throws runtime exception without throws clause
yes. there is no restriction no unchecked exception while overriding.
but in case of Checked exception it is not possible

Q. Can we override a non-static method as static in java
Yes no problem (but should not be private or final)

Q Can we have non-abstract method inside interface?
from java 8 onwards you can have a non-abstract method inside interface.

Q. Can we overload main()
Yes

Q. Can we override main()
No, since main() is a static method, we can't override it. because static method is resolved at compile time without needing object info

Q. when to use singleton pattern
when we need just one instance of class & wants that to be globally available then we can use singleton

Camlin | Page
Date / /

(SVN, git, mercurial)

* How you manage your Source Code (version control
* Hybrid, Keyword, datadriven ?
* What is your framework Structure ? explain
* object repository ?
* How is your team Structure
* clone & ~~git~~ Commit ?
* maven ? Jenkins (CIT) (.xml file)
* Explain diff. ~~exp~~ exception (exception Handling)
* Logical, palindrome, prime No, rev-string, armstrong.
* Can we execute Scripts using Command (Cmd), or maven
* wrapper, utility class ? property file.
* excel, Screenshot, select dropdown, , Popup
* List of all challenges faced in your project.
* Roles & responsibilities of automation test engg.
* How to run failed test-cases
* How to take Screenshot of failed test cases.
* Diff types of waits
* return type of findElements & getWindowHandles.
* ~~How~~

* Steps involved in automation
    1. selecting the test Tool
    2. Define scope of automation
    3. planning design & development
    4. Test Execution
    5. maintenance.

* Steps involved in planning phase of automation
    1. selecting "right" automation tool
    2. Selecting automation framework if any
    3. List of scope for automation test envin. Setup.
    4. preparing grant chart for
        A. development & Execution
    5. Identify test deliverable

* The Condition where we can't use automation in agile
    1. when agile testing always ask for changes in requirments
    2. When exhaustive level of documentation is required in Agile
    3.

* primary feature of good automation tool
    1. Test environment support & easy to use
    2. Good debugging facility
    3. Robust object identification
    4. object & image testing ability
    5. object identification
    6. Testing of database
    7. support multiple framework

* Scripting std.
    1. Uniform naming Convention
    2. ~~[illegible]~~ proper Comments for every method

Camlin Page
Date / /

* Roles & Responsibilities of automation Test engg.

(1) selenium envirnment setup :-
eg down & install, eclipse, Java lang. Config.
selenium jar file, TestNG, maven, ... etc

(2) Inspect elements / object
using firepath or firebug

(3) Creating test cases using element locators & selenium webdriver Commands.
- element locator for identifying element
- selenium webdriver for performing operations on element.

(4) Enhancing Test Cases ~~&~~ using prg. features
eg. flow Control stat, conditional stat.
exception handling (selenium supp 6 lang)
adding Comments, error handling, verification etc

(5) grouping Test Cases, prioritizing test cases, ~~generating~~, Executing test batches & generating test reports using testing framework eg. TestNG / Junit for $\boxed{\text{.net - nunit}}$

(6) Data driven testing & DB testing, cross browser testing
Executing same functionality with multiple set of data

~~(7)~~
(7) Analyzing test result & reporting ~~rep~~ defects
(8) Selecting TCs for Regression testing, defect tracking
(9) Regression testing on modified builds
(10) final Regression
(11) maintenance of test automation Resources.

Camlin Page
Date / /

* How to Create batch file

1. Create New project
2. Download TestNG
3. Create class.
4. add external jars files.
5. Run the class by Creating xml file.
6. Create lib folder in folder structure
(right click on project → properties → resources
Copy project path) (Run → enter path)
7. Add all the Required Jars to that lib folder
eg. Standalone, TestNG.
8. Create batch file
→ open Notepad ++;
New file

```
Set projectlocation = ______
cd %projectlocation%
Set classpath= %projectlocation%\bin; %projectlocation%.
                                   \lib\*
Java org.testing.TestNG %projectlocation%\testng.xml
pause.
```

Save file with .lib extension
then click on that file (looks like setting symbol)

Camlin Page
Date / /

* Scroll up & Down :-

```
Webdriver driver = new firefoxdriver()
driver.manage().windows().maximize();
driver.get("http://www.jqueryui.com");

JavaScriptExecutor JS = ((JavaScriptExecuter) driver);

JS.executeScript("scroll(0, 400)");
JS.executeScript("Scroll(0, -300)");
```

+ve for scroll Down

-ve for scroll up

* Scroll Into View :-

```
Webelement ele = driver.findelement(BY.xpath("__"));

JavaScriptExecutor JS = ((JavascriptExecuter) driver);

JS.executeScript("arguments[0].ScrollIntoView(true);",
                                          ele);
```

* Alternate To SendKeys :-

```
JS.executeScript("arguments[0].Value='Sanjay';", ele);
```

* How to run failed test-Cases

In a Test Suite if 10 T.Cases are there if 2 Tc are failed then we can run those 2 TC separatelly. By 2 ways.

① (manually)
After executing Test Suite. when we refresh project it will by default generate one file \boxed{testng.failed.xml}. this xml file contains only ~~on~~ failed test cases. So by using this we can run failed TCes.

```
v test-output
        > SuiteName
            |
            | ~~test~~ [testng-failed.xml]
            | emailable-report.html  } report
            | testng-report.xml      }
```

② (programatically)
1. Create a new package (ie. runner)
2. Create a class TestRunner.
3. Create Object of TestNG class.
4. create list of String
5. In this list we can add multiple xml files
6. ~~path~~ pass that list in a method Called setTestSuites which is in runner object (TestNG)
7. Call the run() method.

```
public class TestRunner
{
    psvm (String [] args)
    {
        TestNG runner = new TestNG();
        List <String> list = new ArrayList<String>();
            list.add ("path of failed .xml file ");
        runner. setTestSuites (list);
        runner. Run();
    }
}
```

Camlin Page
Date / /

* Take Screenshot of failed Test cases.

ITestResult :-

- It is an inteface which Keeps all information about the test Case which ~~are~~ we executed
- we will Capture some information from this like ① Test case execution status.
  ② Test case Name.

```
public class Demo
{
  ~~psvm (String [] args)~~
 @ Test
   public void Test()
   {
       Webdriver d = new FirefoxDriver();
       d. get ("     ");
       d. findelement ( BY. xpath ("____")). send Keys (" ");
       d. close();
   }
  @ ~~Befor~~ Aftermethod
   public void teardown (ITestResult Result)
   {
       if (ITestResult. FAILURE == Result. getStatus())
       {
file scrfile = ((Takescreenshot) d ). getScreenshot AS
                                          (OutputType. File);
 File D file = new File (" Dest. Path ");

   FileUtils. CopyFile (scrfile, Dfile);
       }
   }
```

* List of challenges faced in project :-

1) Domain knowledge of application
2) vast appln, flows, functionalities & module interconnection were complex
3) use of dynamic Xpath.
4) Handling multiple exceptions while test execution eg.
StaleElementException, NullPointerException
5) Dependancies of test Cases on one another
6) Execution of single test Case were but failes in suite
7) use of Before method implemented to minimize dependancies
8) maintenance of test Data & pre-conditions on multiple pest.
9) Multiple unexpected pop up handling.
10) Application freeze because of multiple user using single pest.
11) many setting related test Cases were tested on diffrent Pest to minimize the effect on other test Cases

Camlin Page
Date / /

★ **How to take Screenshot in Selenium webdriver?**

Screenshot are taken in 3 steps in webdriver.

Step① Convert Webdriver object to TakeScreenshot

```
TakeScreenshot scrshot = ((TakeScreenshot) webdriver);
```

step② Call getScreenshotAS method to create image file

```
File scrFile = scrShot.getScreenshotAS(outputType.
                                        File)
```

step③ Copy file To desired location.

```
File DestFile = new File(" D:\\ Screenshot");
FileUtils. CopyFile ( scrFile , DestFile)
```

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

```
            Webdriver driver = new F.F.D();
File scrFile = ((TakeScreenshot) driver) . getScreenshot AS
                                       (outputeType.file)

            FileUtils. Copyfile ( scrfile, " Dest. path");
                                              ↓
                                     new file(" Path ")
```

* ~~Fetch~~ ~~Excel~~ ~~Data~~ :-

Camlin Page
Date / /

* TestNG Listeners *

There are 2 types of listeners in ~~Selenium~~

① Webdriver Listeners
② TestNG Listeners

① Selenium Webdriver Listeners
• Listener is defined as interface that modifies the default TestNG behaviour.
• As the name suggests Listeners "listen" to the event defined in the selenium script & behave accordingly.
• It is used in the selenium by implementing Listeners interface.
• It allows customizing TestNG reports or logs.

Types of Listeners in TestNG =
1) ITest Listener
2) I Suite Listener
3) I Reporter
4) I method Intercepter
5) I Invoked method Listner
6) I Invokedmethod Listner2
7) I Hookable
8) I Execution listner
9) I Configuration listner
10) I Configurable
11) I Annotation Transformer
12) I Annotation Transformer2

Above interfaces are called as TestNG listner these are used in Selenium to generate logs or Customize the Testing reports.

ITestListener

- ITestListner has following methods.

① OnStart :- this method is called when any Test starts

② onFinish :-
onFinish method is called when all Tests are executed

③ onTestStart :-
this method is Called when any particular test starts.

④ onTestSuccess :-
onTestSuccess method is called on the Success of any Test.

⑤ onTest Failure :-
onTest Failure method is Called on the failure of any Test.

⑥ onTestSkipped :-
onTestSkipped method will be Called when any method get Skipped in a any Test.

Camlin Page
Date / /

⑦ onTest Failured But within Success percentage -
this method is called each time Test fails but is within success percentage

* Steps To Create TestNG Listners :-

① • Create a class "Listener_Demo" & implements "ITestListener"
• move mouse over readline Test, & eclipse will suggest 4 2 quick fix.
• just click on " Add unimplemented method".
multiple unimplemented methods (without body) is added to the code.

eg.

```
public class Listener_Demo implements
{                              ITestListener
@ override
  public void onFinish (ITestContext arg)
  {  =
  }

@ override
  public void onStart (ITestContext arg)
  { =
  }

          ,
          ,
          .
          ,
          ,

@ override
  public void onTestFailure (ITestResult arg)
  { =
  }
```

② • modify the Listener class, in particular method
ie. onTestFailure, onTestSkipped, onTestStart, onTest Sucess.
• modification is simple, we just print the name of the Test.
• logs are created in Console, it is easy for user to understand which Test case get pass, fail, or skipped.

* After modification *

```
public class Listener_Demo implements ITestListener
{
 @override
     public void onFinish (ITest Context Result)
       { ≡
       }
  @ override
     public void onStart (ITest Context Result)
      { =
      }
  @ override
     public void onTestFailedButWithSucessPercentage (ITestListner R)
     { ≡
     }
  @ override
     public void onTestFailure ( ITest Result Result)
      {  S.o.p ("name of Test case failed:" + Result.getName());
      }
  @ override
   public void onTestSkipped (ITestResult Result)
    { Sop. ("name of Test case skipped: " + Result.getName());
    }
  @ override
   public void onTest~~Skipped~~Start (ITestResult Result)
    { Sop (" name of Test case Started: " + Result.getName());
    }

  @ override
   public void onTestSucess (ITest Result Result)
    { Sop ("name of Test case passed: " + Result.getName());
    }
```

③ Create another class "TestCases" for implement ListnerClass in this TestCases class.

```
      @Listener ( PackageName. ClassName)
      public class TestCases
   {
      @Test (priority = 1)
       public void Test1()
      {
         Assert.
      }

      @Test (priority = 2)
       public void Test2()
      {
         Assert.fail();
      }

   }
```

↓
o/p

```
Passed : Test1
Failed : Test2
```

* Use of Listener in TestSuite :-

eg

```
      < suite name = " " >
        < Listeners >
          < Listener class-name = " pack. Name . class_Name"/>
        < / Listeners>
      < test Name = " " >
        <classes >
          < class = " pack_Name . class_Name"/)
          </class>
      </test>
      </suite>
```

• If we write the Listener in test suite then we no need to write in Test class.

• This Listener ~~to~~ is implemented throught the test suite irrespective of the number of classes we have.

• when we run this xml file, listeners will work on all classes mentioned. We Can also declare any number of listener class.

Summary :-

① Listeners are required to generate logs or Customize TestNG reports in selenium Webdriver.

② There are many types of listeners & Can be used as per requirments.

Listener are interface used in selenium webdriver Script.

* Handling of Dynamic web Tables :-

• There are 2 types of HTML Tables present on web-page

① Static Table :- Data is Static ie. Number of rows & Columns are fixed.

② Dynamic Table :- Data is Dynamic ie. No. of. rows & Column are not fixed.

* Webelement - Webelement are nothing but HTML element like textbox, dropdowns, radio buttons, Submit button etc

① fetch No. of Rows & Column from Table

• When Table is in dynamic nature we Can't predict No. of rows & Column.

```
// No. of Rows
      List<Webelement> Row = d.findelements(BY.xpath(
                      ".//@[id=" "]/Table/tbody/tr/td[1]));
// No. of Column
   List<webelement> Col = d.findelements(BY.xpath(".//[@id=" "]
                              /Table/thead/tr/th));

   ∴ int rowSize = Row.size();
     int ColSize = Col.size();
```

Camlin Page
Date / /

* Robot Class in Selenium Webdriver :-

- In Selenium automation testing, Sometimes there is need to Control ~~Keyword~~ Keyboard or mouse to interact with OS window like download pop-up, Alert, print popup, etc or, native operation system applications like Notepad, Skype, Calculator.

- Selenium webdriver Can't handle these os popups / applications.
- In Java 1.3 Robot Class was introduced Robot Class can handle OS pop-ups / appln.

* Advantage of Robot class :-
① It Can Simulate keyboard & mouse event.
② It helps while upload/Download files using Selenium webdriver
③ Robot class can easily be integreted with Current automation framework (keyword, data-driven, hybrid).
④ alternate for robot class is AutoIt, but its drawback is that it generates an executable file (exe) which will only work on windows. So its not good option to use.

* Commonly used method :-
① Key press() :- robot.keypress (KeyEvent.VK_Down) it press key Down. of keyboard.

② mousepress() :- robot.keypress (inputEvent.BUTTON3-Down-mask); It press right click of mouse.

Camlin Page
Date / /

(3) <u>mouseMove()</u> :- robot.mouseMove(PointgetX(), point.getY())
this will move mouse pointer to Specified X & Y Coordinates.

(4) <u>KeyRelease()</u> :- robot.KeyRelease(KeyEvent.VK_Down)
this ~~[illegible]~~ method with ~~release~~ down arrow key of keyboard

(5) <u>mouseRelease()</u> :- robot.mouseRelease(inputEvent.Button3_Down_mask)
this method will release Right click of our mouse.

<u>close browser</u>

<u>eg</u> :-

```
Robot robot = new Robot();  -> throws AWT excep
robot.Keypress(KeyEvent.VK_Control);
robot.Keypress(KeyEvent.VK_W);
robot.KeyRelease(KeyEvent.VK_Control);
robot.KeyRelease(KeyEvent.VK_W);
```

* <u>Disadvantage</u> :-
(1) <u>Keyword / mouse</u> event will only work on current instance of window. eg suppose a code is performing any robot class event, & during the code execution user has moved to some other screen then keyboard/mouse event will occur on that screen

Camlin Page
Date / /

① Ignore Test

→ JUnit @Ignore
@ ~~Ignore~~ @Test ("Not ready to Run")

and.

→ TestNG
@ ~~Enable~~ Test (enable = false)

* New in TestNG
1) Before Suite
2) After Suite } suite
3) Before Test
4) after Test } Test
5) Before group
6) After group } group

Maven - 1) main ~~all~~ right version of jars
2) Execute the framework from Command prompt
ANT
3) Build automation Tool, , it generates build for u.
4) we can execute test file in exe also

* Jenkins
it will help u to schedule test to run test cases.
Constantly keeps on polling in git.

Camlin Page
Date / /

* Object Reporisitory :-

Q. what is object Repositery?
- An object Repository is a common storage location for objects.
- In selenium webdriver object would typically be the locater used to uniquely identify web element.

* Advantage of using object Repository :-
- segregation of objects from test cases, if locater value of one webelement changes, only the object respository needs to be changed rather than making changes in all test cases in which the locater has been used.
- maintaining an object Repository incrises the modularity of framework implementation.

* Types of object Repositories in Selenium webdriver?
- Selenium webdriver does not offer an in-built object repository by default.
- However object Repository can be buit using key-value pair approach where in key refers to the name given to the object & value refers to the properties used to uniquely identify an object within the webpage.

① object Repository using property file.
② object Repository using XML file.

* Create object Repository using property file.

Steps

① Create package objectRepositoryDemo. In that Create a class called "demo".

② Right click on ~~pr~~ main project → New → other → general → File → Next

③ enter filename eg. application.properties. click on finish. created file shown under Referenced libraries.

④ Store Data into library

eg 1. UN = //input[@id="abc"]
   2. pwd = pwd
   ⋮
   N.

* * Read data from property file.

① object of properties ~~file~~ class should need to be Create

properties obj = new properties();

② create object of FileInputStream class with path to properties file.

```
FileInputStream objFile = new FileInputStream(System.getProperty("user.dir")
                                          + "\\application.properties");
```

③ Reading data from properties file can be done using load method present in properties class

```
obj.load(objFile);
String username = obj.getProperty("UN");
```

String UN will Contain XPath for username.

eg. d.findelements(BY.xpath(username));

★ Mavan :-

- Mavan - a build Automation tool which is mainly used for java projects. It makes build consistent with another project.

- mavan is used to manage dependencies. (maintain the right versions of Jars).

- Build Tool :-
  - It is used to setup everything which is required to run our java code independently.
  - It does
    1. Generates Source Code
    2. Compile Source Code
    3. Generates Documentation from Source Code
    4. packages Compiled Code to Jar of zip file.
    5. Install packaged code in local repository, server repository or Central repository.

- mavan provides pom.xml which is Core of any project. this is configuration file where all required info are kept.

- mavan Stores all project Jars. library Jar is in a place called repository which could be a central, local or remote repository

- Mavan downloads the dependancies ~~from~~ Jar from Central repository. most Commonly used libraries available in http://repo1.maven.org/maven2/

Type in cmd
mvn --version

Camlin Page
Date /

• Downloaded libraries are Stored in local repository called $\boxed{.m2}$ file

• If libraries are not available in central repository then maven looks for remote repository. The user has to configure the remote reposi. in pom.xml to download from remote ~~de~~ repository.
Below is the example to configure remote repository to pom.xml.

```
<repositories>
    <repository>
    <id> LibraryId </id>
    <url> http://ComanyrepositoryId </url>
    </repository>
</repositories>
```

* <u>Build Life cycle</u> :-

(name of Jar without version)

1) <u>clean</u> :- Delete all artifacts & targets which are created already.

2) <u>Compile</u> :- use to Compile Source code of project

3) <u>test</u> :- test the Compiled code & these test do not require to be packaged ordeployed

4) <u>package</u> :- package is used to convert your project into Jar or war etc.

5) <u>Install</u> :- Install the package into local repository for use of another project

Camlin Page
Date / /

General phrases used in mavan :-

1) groupId : Generally groupId refers to domain Id for best practice Company name is used as groupId. It Identifies project uniquely.

2) ArtifactId : It is basically name of the jar without version.

3) version :- used to create version of project

4) Local repository :- mavan downloads all required jars dependencies & store in the local repository called m2.

* Advantages of mavan :-
1) Better dependancy management
2) More powerful build
3) Better debugging
4) Better Collaboration
5) More Consistent project Structure
6) More Componentized build, (code can test very quickly)

1. mavan forces you to have standard directory Structure
2. import all Library.
3. it is a project management tool, It help us to generate reports, it help in dependancy management

Camlin Page
Date /

★ <u>Jenkins :-</u>

• Jenkins is an open Source "Continuous integration server" build with Java.

• Jenkins chief usage is to moniter any Job which Can SVn checkout, or any application Status.

• It fires pre-configured actions when particular Steps occurs in Jobs.

cff

• It provides ~~455~~ 900+ plugins to support building & testing virtually any project.

★ <u>imp features :-</u>

65A

• Generates list of all changes done in repositories like SVn.

• Jenkin Can Schedule your test to run at Specific time.

• we Can Save execution history & test Report.

• Jenkins allow you to run your ~~Script~~ test every time your S/W changes & deploy S/W to a new env. when tests pass.

Camlin Page
Date / /

* Fetch Excel Data :

FileInputStream file = new fileInputStream("path of excel");

WorkBook Book = WorkBookFactory . create (file);

Sheet sheet = Book . getsheet ("Sheet name")

Row row = Sheet . get Row (Row No.)

cell cell = Row . get cell (cell No.)

Sop (cell) ;

Camlin Page
Date / /

★ **Waits in Selenium Webdriver :-**

① Implicit wait
② Explicit wait
③ Fluent wait.

① **Implicit Wait :-**

- Implicit wait is used when we know exactly what time will ~~take~~ to load the webpage eg 10 sec, 20 sec

eg. driver.manage().timeouts().implicitlyWait (10, timeunit.second)

- Even if webpage is not loaded within 10 sec, then it will through "no such element" Exception.

② **Explicit Wait :-**

- Explicitly wait (webelement) is used when we we don't know exactly How much time will take, it might load in 2 sec, or it might take in 10 sec.

•

- In this case we don't want to wait for 10 sec. if the Page is already loaded in 2 sec, in this case we can given condition

Camlin Page
Date / /

• So in Explicit wait we give a Condition plus we Can give specific waiting time. till that Condition occurs.

eg.

```
                Webdriver driver = new firefoxdriver();
                driver.navigate().to("http://www.google.com");
waiting time ← Webdriverwait wait = new webdriverWait(driver, 20);
      webelement login;
login = wait.until (Expected Conditions.visibilityofElementlocated(
                                                    By.id("——")));
      login.click();
```

expected Condition.

• Here we have taken waiting to 20 sec, but if page is loaded in 3-4 sec then it will not wait for 20 sec. moves to next Step.

③ **Fluent Wait :-**

(polling time) - 3rd parameter.

In fluent wait we can give Condition, & we ~~Cond~~ Can give specific ~~time~~ waiting time to occur that Condition Apart from this 3rd parameter is we Can give is frequency

```
Wait<Webdriver> wait = new FluentWait<webdriver>
                                               driver
          . ~~wait~~ withTimeunit(20; Timeunit.second)
          . polling Every(5, timeunit.Second)
          . Ignore(NoSuchelement Exception.class);
```

* Add Dependencies in mevan
  - → click on pom.xml
  - → In that window click on pom.xml tab,
  - → ente Contral repository in google
    click on (https:// search. mavan org) link
  - → Here search for jars which we require
    (eg. Junit, & Selenium-Java etc)
  - → click on Latest version link
    Copy the code form (Dependency information)
  - → paste the code in pom.xml Save the
    Code that particular jar will be
    added to the mavan Dependencies.

* Cucumber :-

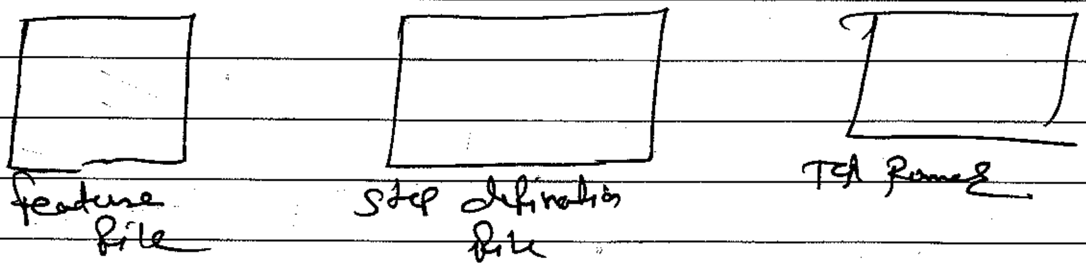
A Cucumber is a tool based on Behavior Driven Development (BDD) framework which is used to write acceptance tests for the web application.

It allows automation of functional validation in easily readable & understandable format to BA, Developers, Testers.

- It is use for performing Acceptance testing.
- Cucumber in initially implemented in Ruby & then converted to java lang.

* It is Behaviour driven development (BDD) approach to write automation test script to test an appln (Desk, web, android)

- It enables you to write & execute automated acceptance, unit ~~test~~ tests codes.
- It is cross-platform, open source & free.



★ Keyword driven :-

- Keyword driven framework is an extension to Data driven testing framework ~~to~~.

- It ~~not~~ only Segregates the test Data from the ~~te~~ script, & also keep the certain Set of code belonging to the Script into external data file

- So these set of code are known as Keyword & hence named, keyword are self guided as to what actions need to be performed on the application.

eg	Keyword	Locator/Data
	login	
	clicklink	//-[@id = '1235']
	verifylink	// *[@id='link']

* Generating Extent Report in Selenium :.

- As TestNG will generate very basic Report which is not that attractive
- So to create Attractive Report we need to use $\boxed{\text{Extent Report 3}}$ (latest version) which is 3rd party API

extend report jar is provided by small company called relevant codes which is handled by some of the bloggers

- why - It gives you amazing dashboard very good look & feel for your execution report

steps

① ~~[illegible]~~ Download the dependency & add into pom.xml

- Extent reports depends TestNG (Listners)
- Result will be displayed pie chart & also column wise