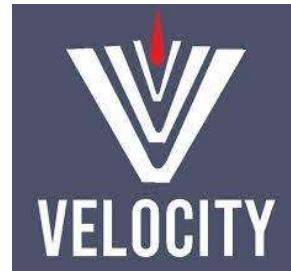


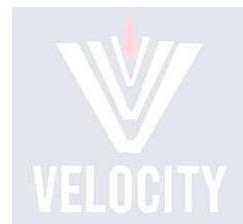
Selenium

Learn Selenium by Sanjay Sir



Contents

	<u>Page No.</u>
1. Popup (Hidden Division, Alert, Child Browser, Authentication, File Up/Down)	02
2. Handling of Multiple Element	09
3. Handling of Autosuggestion	11
4. Action Class	13
5. Handling of Customized ListBox	17
6. Different Types of Exceptions	22
7. WebTable	23
8. FrameWork (Without Data Driven)	27
9. Encapsulation(OOPS)	31
10. POM	32
11. POM (Page Object Module)	34
12. POM with Data Driven Framework	38
13. TestNG	40
14. TestNG Annotations (BeforeTest, BeforeClass, BeforeMethod, Test, After)	41
15. TestNG Keywords (invocationCount, priority, enabled, TimeOut, dependsOnMethods)	44
16. TestNG Suit	50
17. TestNG Failed Test Cases	52
18. Verification using TestNG (equals, notEquals, true, false, null, notNull)	54
19. Assert Class (Hard Assert: Static Method)	55
20. Soft Assert (Soft-Assert Class : Non Static Method)	57
21. Disable Test Case Execution (enable = false, exclude/include)	58
22. Grouping (test_groups_run_include_exclude_)	61
23. Parallel	63
24. POM DDF with TestNG	70
25. Capture the Screenshot of Failed Test Cases	77
26. Maven Project Setup	82
27. Interview Questions	83
28. Github	88



Popups

Popups:

- Popups are small or separate window which will be displayed when we perform action on any components present in a webpage.
- These popups can be handled by selenium directly, but sometimes we may need to use 3rd party tools to handle these popups.

Types of popups:

1. Hidden-Division Popup

***2. Alert popup**

***3. Child browser Popup**

4. Authentication Popup

5. File Upload Popup

6. File Download Popup

- If we are able to inspect element present in a popup then we can use selenium directly to handle that popup.
- If we are unable to inspect element present in a popup then we need to use 3rd party tool to handle that popup.

1. Hidden-Division Popup:

1. These popups are colorful.
2. We can inspect the elements present in pop up.
3. As we can inspect element present in popup then using selenium we can handle it & no need to switch.
4. URL: <https://www.flipkart.com/>

e.g. popup on flipkart(When we open flipkart then click on the login page then a colorful new window/box will open.

```
package Popup;
```

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class A_Hidden_Division_popup {
    public static void main(String[] args) throws
InterruptedException {
    System.setProperty("Somepage3.chrome.driver",
    "H:\\Selenium\\chromedriver.exe");
    WebDriver driver = new ChromeDriver();
```

```

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

//Click on Close button
driver.findElement(By.xpath("//button[@class='_2KpZ6l_2doB4z']")).click();

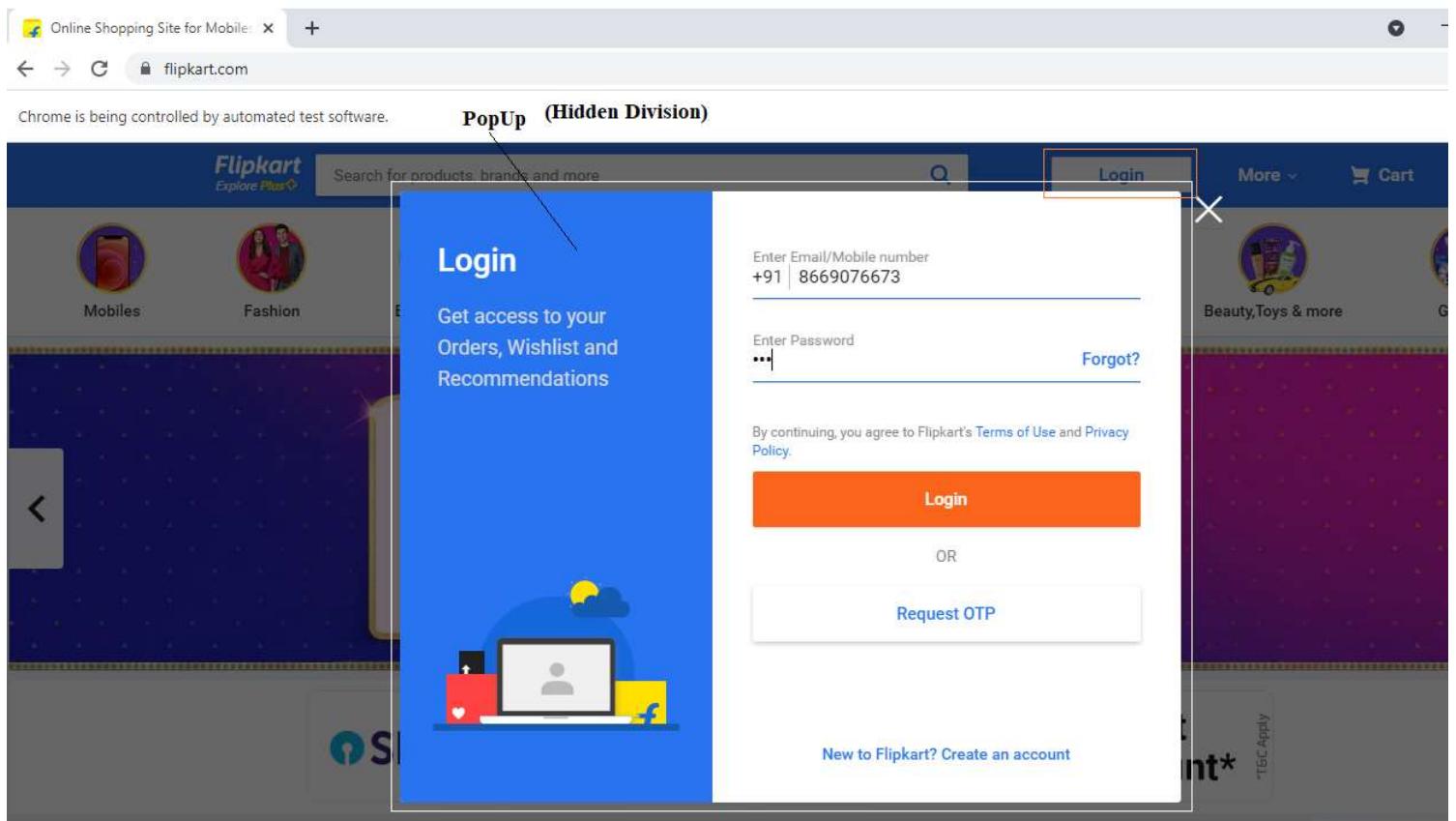
Thread.sleep(3000);

//click on the login button
driver.findElement(By.xpath("//a[text()='Login']")).click();

Thread.sleep(3000);

//Enter UN and Password
driver.findElement(By.xpath("//input[@class=' _2IX_2VJZDxU ']")).sendKeys("8669076673");
driver.findElement(By.xpath("//input[@class=' _2IX_2-
_3mctLhVJZDxU ']")).sendKeys("XYZ");
}
}

```



2. Alert Popup:

1. We cannot inspect the elements present in pop up.
2. These popup will contain **ok** button or **cancel** button & **Text**.
3. Sometimes these type of popup also contains? Or! Symbol.
4. Url: http://demo.guru99.com/test/delete_customer.php

-----Procedure to handle Alert popup-----

2. To handle alert popup we need to switch selenium focus from main page to alert popup by using syntax

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

2. Alert is an interface which contains abstract methods like:

1. **accept()**: use to click on ok button.
2. **dismiss()**: use to click on cancel button.
3. **getText()**: use to get text present in an **alert popup**.

- To click on OK button → alt.accept();
- To click on Cancel button → alt.dismiss();
- To get the text present in a webpage: → String text = alt.getText();

```
package Popup;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class B_Alert_popup {
    public static void main(String[] args) {
        System.setProperty("Somepage5.chrome.driver",
        "H:\\Selenium\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();

        driver.get("https://demo.guru99.com/test/delete_customer.php");
        driver.manage().window().maximize();

        //Enter customer Id
        driver.findElement(By.xpath("//input[@name='cusid']")).sendKeys("XYZZ123");

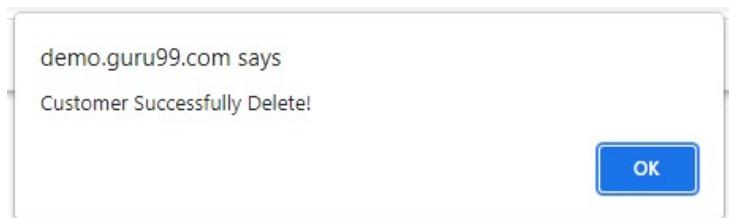
        //Click on the submit button
        driver.findElement(By.xpath("//input[@name='submit']")).click();
        //Output : New popup has opened asking for Do you really want to delete
        //this Customer?
        Unable to find the WebElement using locator i.e selenium Somepage5
        methods cant able to find element

        //Get Text from Alert Popup
        String text = driver.switchTo().alert().getText();
```

```
System.out.println(text); //Do you really want to delete
this Customer?
```

```
//Click on the OK button → Use .accept()
driver.switchTo().alert().accept();

//Click on the cancel button → use .dismiss
//driver.switchTo().alert().dismiss();
}
}
```



```
package Popup;
import org.openqa.selenium.Alert;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class C_Alert_Popup {
    public static void main(String[] args) throws
InterruptedException {
        System.setProperty("webdriver.chrome.driver",
"H:\\Selenium\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();

        driver.get("https://demo.guru99.com/test/delete_customer.php");
        driver.manage().window().maximize();

        driver.findElement(By.xpath("//input[@name='cusid']")).sendKeys("XYZZ123");

        driver.findElement(By.xpath("//input[@name='submit']")).click();

        Thread.sleep(3000);

        //Switch to alert() popup
        Alert alt = driver.switchTo().alert();

        String text = alt.getText();
        System.out.println(text);

        Thread.sleep(3000);
        //Click on Cancel → use .dismiss()
        alt.dismiss();

        //OR
        //Click on Submit → use .accept()
        //alt.accept();
    }}}
```

3. Child browser Popup/window popup

1. We can inspect elements present in popup.
2. This popup will contain address field (**URL**, **maximize**, **minimize** and **close** options).
3. URL: <https://skpatro.github.io/demo/links/>

-----Procedure to handle Child browser popup-----

1. To handle Child browser popup we need to switch selenium focus from main page to Child browser popup by using syntax

```
driver.switchTo().window("String ChildwindowId")
```

2. To get address of main page as well child browser we need to use Homepage

```
Set<String> IdOfMainPage=driver.getWindowHandles();
```

return type of getWindowHandles method is set<String>

3. To get address of only main page we need to use syntax

```
String IdOfMainPage=driver.getWindowHandle();
```

return type of getWindowHandle method is String

4. To get address of child browser we need to use

```
Set<String> ids = driver.getWindowHandles();
```

```
ArrayList al=new ArrayList(ids);
```

```
System.out.println(al.get(0)); // address of main page
```

```
System.out.println(al.get(1)); // address of child browse
```

4. Authentication Popup—autoIT tool / Robot class

1. Pop up displayed on Page load
2. We cannot inspect the elements present in pop up.
3. URL: <https://chercher.tech/auth/>

5. File upload—autoIT tool/ Robot class

6. File download—autoIT tool/ Robot classs

For handling the file upload and file download we need to use third party tool autoIT/Robot class.

Selenium can't handle the file upload or file download popup

```

package Popup;

import java.util.ArrayList;
import java.util.Set;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class D_ChildBrowser_WindowPopup {
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver", "H:\\Selenium\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(60, TimeUnit.SECONDS);
        driver.get("https://skpatro.github.io/demo/links/");
    }
}

//Click on the New Tab button

    driver.findElement(By.xpath("//input[@name='NewTab']")).click();
//It will open a new Tab

//Get child window ID→.getWindowHandles()→ get Id of both Main and
Child Page
    Set<String> AllIDs = driver.getWindowHandles();
    System.out.println(AllIDs); // get main page Id as well as
Child Page ID (Id1, ID2)

//Get only Child Window Id
    String ChildID = driver.getWindowHandle();
    System.out.println(ChildID); //will get Id of only Child browser

    ArrayList<String> al = new ArrayList<String>(AllIDs); // [mainPage ID(0), ChildWindowID(1)]
    String ChildWindowID = al.get(1);
    System.out.println(ChildWindowID);

    //Switch to Child Window
    driver.switchTo().window(ChildWindowID);

    //Click on Element of the Child Browser
    driver.findElement(By.xpath("//span[text()='Training'][1]")).click();
}
}-----

```

```

package Popup;

import java.util.ArrayList;
import java.util.Set;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class E_ChildBrowser_Popup_SwitchToMainWindow {
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver",
        "H:\\Selenium\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://skpatro.github.io/demo/links/");
        driver.manage().timeouts().implicitlyWait(60,
        TimeUnit.SECONDS);
        //Click on the NewTab Button

        driver.findElement(By.xpath("//input[@name='NewTab']")).click();

        //Get Child window Id
        Set<String>AllIDs = driver.getWindowHandles(); //Main Page
        ID(0), ChildId(1)

        ArrayList<String> al = new ArrayList<String>(AllIDs);

        //Switch to child Window
        driver.switchTo().window(al.get(1));

        //Click on Element Present on Child Popup

        driver.findElement(By.xpath("//span[text()='Training'][1]")).click();

        // Switch to Main Window
        driver.switchTo().window(al.get(0));

        driver.findElement(By.xpath("//input[@name='NewWindow']"));
    }
}

```

● Multiple Link/Elements Handling

```

package Handling_of_Multiple_Element;
import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

public class A_HandlingMultipleElements1 {
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver", "H:\\Selenium\\chromedriver");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://en-gb.facebook.com/");

        List<WebElement> AllLinks = driver.findElements(By.xpath("//a"));
        //Use "findElements"
        //for finding all links always use //a
        System.out.println(AllLinks.size());

        for(WebElement Links:AllLinks)// Use for Each loop to print Multiple Elements
        {
            System.out.println(Links.getText());
            // Returns all the Text of the different Elements Link
        }
    }
}

```

The screenshot shows an IDE interface with two tabs: *A_HandlingMultipleElements1.java* and *A_SwitchTo_iframe_From_MainPage.java*. The code in the first tab is identical to the one above. The right side of the interface displays the execution console output:

```

Nov 01, 2021 10:56:36
INFO: Detected dialect
48
Forgotten password?
Create New Account
Create a Page
?????
?????
?????
?????
?????
?????
?????
?????
?????
?????
Sign Up
Log In
Messenger
Facebook Lite
Watch
Places
Games

```

```

package Handling_of_Multiple_Element;
import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

public class B_HandlingMultipleElementsOn_CHECKBOX {
    public static void main(String[] args) throws
InterruptedException {
        System.setProperty("webdriver.chrome.driver",
"\\Selenium\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();

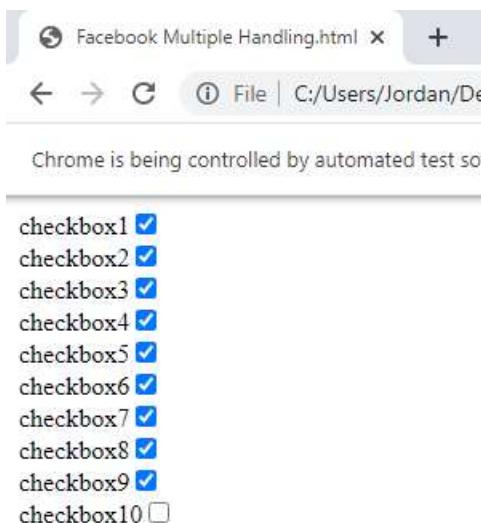
        driver.get("file:///C:/Users/Jordan/Desktop/Velocity%20Testing1/H
TML/Facebook%20Multiple%20Handling.html");

        List<WebElement> MultipleCheckboxes =
driver.findElements(By.xpath("//input[@type='checkbox']"));

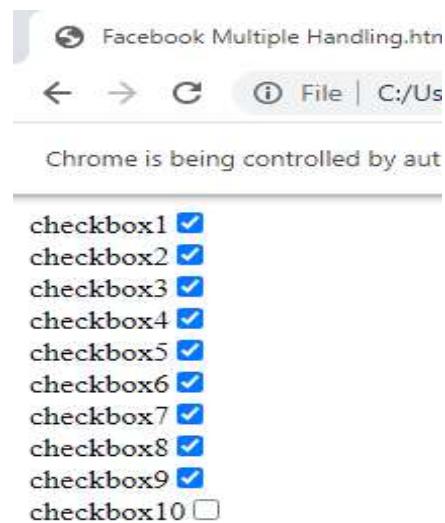
        //For Printing in Forward and Backward use For loop
otherwise ForEach loop
        for(int i=0; i<=MultipleCheckboxes.size()-1;i++)
{
            MultipleCheckboxes.get(i).click();
            Thread.sleep(500);
}

        //In Reverse Order Uncheck the Boxes
        for(int i=MultipleCheckboxes.size()-1; i>=0; i--)
{
            MultipleCheckboxes.get(i).click();
            Thread.sleep(500);
}
    }
}

```



Selecting the Checkbox



Deselecting the Checkbox

- Autosuggestion

```

package Autosuggestion;
import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
public class B_Autosuggestion_RedMi {
public static void main(String[] args) throws InterruptedException {

System.setProperty("webdriver.chrome.driver","H:\\Selenium\\chromedriver.exe");

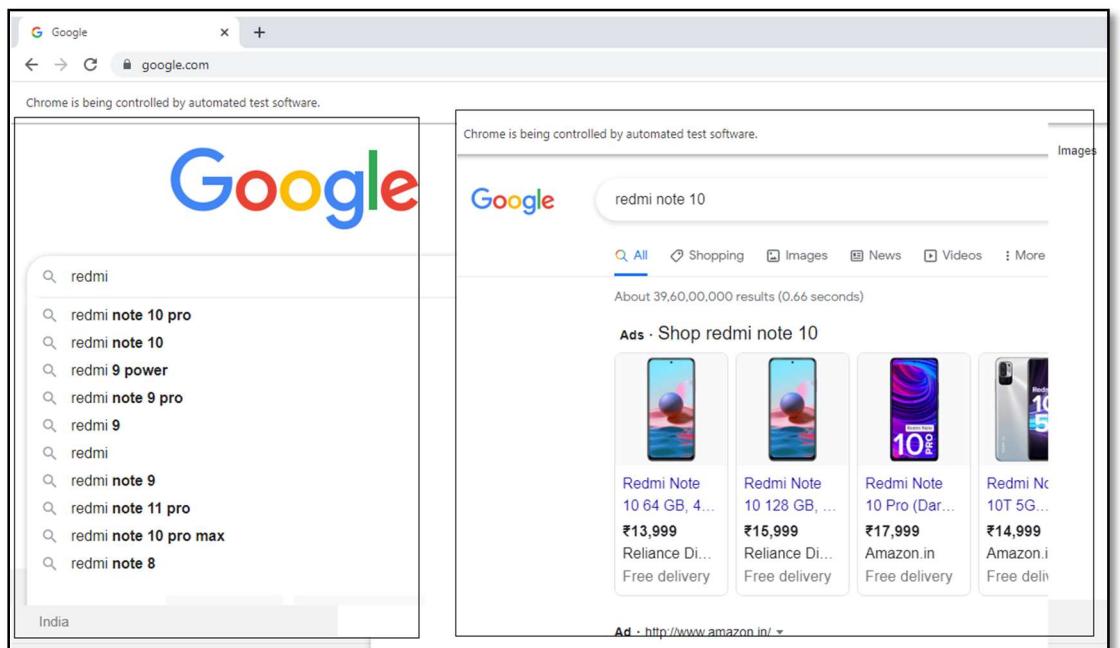
WebDriver driver=new ChromeDriver();
driver.manage().window().maximize();
driver.get("https://www.google.com/");

String expText="redmi note 10";
driver.findElement(By.xpath("//input[@class='gLFyfgsfi']")).sendKeys("redmi");

Thread.sleep(2000);

List<WebElement> MultipleElements
=driver.findElements(By.xpath("(//ul[@class='G43f7e '])[1]/li"));

for( WebElement singLeElement: MultipleElements)
{
    String actText = singLeElement.getText();
    if(actText.equalsIgnoreCase(expText))
    {
        singLeElement.click();
        break;
    }
}
}
}
-----
```



Action Class

Actions class is **an ability provided by Selenium for handling keyboard and mouse events.**

Q. How to Handling of DropDown?

Step1: Identify dropdown element

```
WebElement login= driver.findElement(By.xpath("//a[text()='Login']"));
```

step2: create an object of Actions class with Webdriver object as a input

```
Actions act = new Actions(driver);
```

step3: call Actions class methods → moveToElement

```
act.moveToElement(login).perform();
```

Action class methods

1. moveToElement() : Move one element.
2. contextClick() : Click on Right Button
3. perform : To perform the Action.
4. build() : When multiple action perform/call in single statement then use build.
5. click() : click on the element
6. doubleclick() : double click on element
7. dragAndDrop() : drag one element and drop the same element at other place
8. clickAndHold() : click and hold
9. release() : release the click and hold element.

URLs:

DragAndDrop: http://demo.guru99.com/test/drag_drop.html

DoubleClick: http://demo.guru99.com/test/simple_context_menu.html

Other: <https://www.flipkart.com/>

```

package Autosuggestion;                                //perform();

import org.apache.xml.security.stax.ext.XMLSecurityConstants.Action;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.interactions.Actions;
public class A_HandlingDropDown {
    public static void main(String[] args) throws
InterruptedException {

    System.setProperty("webdriver.chrome.driver","H:\\Selenium\\chromedriver.exe");
    WebDriver driver=new ChromeDriver();
    driver.manage().window().maximize();
    driver.get("https://www.flipkart.com/");
driver.findElement(By.xpath("//button[@class='_2KpZ6l_2doB4z']")).click();           //Click on close button
    Thread.sleep(3000);

//Step.1 Identify the dropdown Element which need to be handle
    WebElement login =
driver.findElement(By.xpath("//a[text()='Login']"));

//Step.2 Create an object of the Action class with web driver object
as input
    Actions act = new Actions(driver);

//Step.3 Call Action Class Method
    act.moveToElement(login).perform();

Thread.sleep(3000);

//Click on object present in Dropdown //Gift Card
driver.findElement(By.xpath("//div[text()='Gift Cards']")).click();
}

}-----



package Autosuggestion;                                //moveToElement(login)
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.interactions.Actions;

public class B_MoveToElement {
    public static void main(String[] args) throws
InterruptedException {
    {
    System.setProperty("14omepage14.chrome.driver","H:\\Selenium\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();
}
}

```

```

        driver.get("https://www.flipkart.com/");
driver.findElement(By.xpath("//button[@class='_2KpZ6l
_2doB4z']")).click(); //Click on close button
Thread.sleep(3000);

//Step1 Identify the Dropdown Element which need to be handle
WebElement moreElement =
driver.findElement(By.xpath("//div[@class='exehdJ']"));

// Step2 : Create an object of actions class and pass Web driver as
input
Actions act = new Actions(driver);

//Step3:Use Action class methods
act.moveToElement(moreElement).perform();
}

}}-----
```

// ContextClick()

```

package ActionClass;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.interactions.Actions;

public class C_ContextClick {
    public static void main(String[] args) throws
InterruptedException {
System.setProperty("webdriver.chrome.driver","H:\\Selenium\\chromedriver.exe");

    WebDriver driver=new ChromeDriver();
    driver.manage().window().maximize();
    driver.get("https://www.flipkart.com/");
driver.findElement(By.xpath("//button[@class='_2KpZ6l
_2doB4z']")).click(); //Click on close button
    Thread.sleep(3000);

//Step.1 Identify the Element which need to be handle
    WebElement cart =
driver.findElement(By.xpath("//span[text()='Cart']"));

//Step.2 Create object of Action class
    Actions act = new Actions(driver);

//Step.3 Use Action class method to perform action
    act.moveToElement(cart).perform();
    act.contextClick().perform(); //Use to click on right button
                                //OR
    act.moveToElement(cart).contextClick().build().perform();
}
}}-----
```

```

package ActionClass;                                     // Click()
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.interactions.Actions;
public class C_Click {
    public static void main(String[] args) throws
InterruptedException {
    System.setProperty("16omepage16.chrome.driver","H:\\Selenium\\chr
omedriver.exe");
    WebDriver driver=new ChromeDriver();
    driver.manage().window().maximize();
    driver.get("https://www.flipkart.com/");
    driver.findElement(By.xpath("//button[@class=' _2KpZ6l
_2doB4z ']")).click(); //Click on close button
    Thread.sleep(3000);

//Step.1 Identify the element which need to be handle
    WebElement cart =
driver.findElement(By.xpath("//span[text()='Cart']"));

//Step.2 Create the object of the action class and pass 16omepage16
input into its constructor
    Actions act = new Actions(driver);

//Step.3 Use Action class Method
    act.click(cart).perform(); // OR
    act.moveToElement(cart).click().build().perform(); //OR

    act.moveToElement(cart).perform();
    act.click().perform();
}
}-----
```

// doubleClick()

```

package ActionClass;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.interactions.Actions;
public class E_DoubleClick {
    public static void main(String[] args) {
    System.setProperty("16omepage16.chrome.driver","H:\\Selenium\\chr
omedriver.exe");
    WebDriver driver=new ChromeDriver();
    driver.manage().window().maximize();

    driver.get("http://demo.guru99.com/test/simple_context_menu.html");
}

```

```

//Step.1 Find the Element which need to be Handle
    WebElement ClickDouble =
driver.findElement(By.xpath("//button[contains(text(),'Double-
Click')]]"));
//Step.2 Create an object of Action class and pass input Web driver
into Constructor
    Actions act = new Actions(driver);

//Step.3 Use Action class method to perform action
    //Method.1 OR
    act.doubleClick(ClickDouble).perform();

    //Method.2 OR
act.moveToElement(ClickDouble).doubleClick().build().perform();

    //Method.3
act.moveToElement(ClickDouble).perform();
act.doubleClick().perform();
}
}-----
```

// dragAndDrop()

```

package ActionClass;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.interactions.Actions;
public class F_DragAndDrop {
    public static void main(String[] args) {
        System.setProperty("17omepage17.chrome.driver","H:\\Selenium\\chr
omedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("http://demo.guru99.com/test/drag_drop.html");
        //Step.1 Identify the Souce and Destination Element
        WebElement src =
driver.findElement(By.xpath("//a[@class='button button-
orange'])[4]"));

        WebElement dest =
driver.findElement(By.xpath("//div[@class='ui-widget-
content'])[3]"));

        //Step.2 Create an Object of Select Class and Pass Web
driver as input in its constructor
        Actions act = new Actions(driver);

        // Step.3 Use Action class methods to perform actions
        act.dragAndDrop(src, dest).perform(); //OR

act.moveToElement(src).clickAndHold().moveToElement(dest).release().bu
ild().perform();
}}-----
```

Handling of Customized ListBox

Q. What is Customized ListBox?

The ListBox which are created without using Tagname “**Select**” is referred as Customized ListBox

Q. How to Handle Customized ListBox

Step.1 Identify the ListBox which need to be handle and store it in a reference variable of Type **WebElement**

```
WebElement month = driver.findElement(By.xpath("//select[@id='month']"));
```

Step.2 Create an Object of the **Action class** which accept the **Webdriver object** as an input.

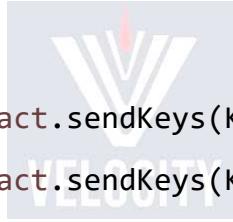
```
Actions act = new Actions(driver);
```

Step.3 **Click** on the **identified element** using Action class method

```
act.click(month).perform();
```

Step.4 Use Select Options

To move one option upward	: act.sendKeys(Keys.ARROW_UP).perform();
To Move one option downward.	: act.sendKeys(Keys.ARROW_DOWN).perform();
To Select Option	: act.sendKeys(Keys.ENTER).perform();
To Move Arrow at the Top.	: act.sendKeys(Keys.HOME).perform();
To move arrow at the end.	: act.sendKeys(Keys.END).perform();



```

package Customized_ListBox;

import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import
org.openqa.selenium.interactions.Actions;

public class A_SelectUpDown_Option {
public static void main(String[] args) throws
InterruptedException {

```

The screenshot shows a 'Sign Up' form with a 'Date of birth' field. The dropdown menu is open, showing the month 'Nov' highlighted with a red oval. Other options like 'Oct' and 'Dec' are visible.

```

System.setProperty("19omepage19.chrome.driver","H:\\Selenium\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://en-gb.facebook.com/");
driver.findElement(By.xpath("//a[contains(text(),'NewAccount')]")).click();
Thread.sleep(3000);

//Step.1 Identify the Customized ListBox Which need to handle of
WebElement Type
        WebElement month =
driver.findElement(By.xpath("//select[@id='month']"));

//Step.2 Create an object of the Action Class and Pass Input Web
Driver
        Actions act = new Actions(driver);

//Step.3 Use Action class method .click to click on element
        act.click(month).perform();
        Thread.sleep(3000);

//Step.4 To Move one Option upward
        act.sendKeys(Keys.ARROW_UP).perform();
        Thread.sleep(3000);

//Step.5 To Move One Option Downward
        act.sendKeys(Keys.ARROW_DOWN).perform();
    }
}

```

```

package Customized_ListBox;

import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.interactions.Actions;

```

The screenshot shows a 'Sign Up' form with a 'Date of birth' field. The dropdown menu is open, showing the month 'Jun' highlighted with a red oval. Other options like 'May' and 'Jul' are visible.

```

public class B_SelectUpward_forLoop {
    public static void main(String[] args) throws
InterruptedException {
        System.setProperty("20omepage20.chrome.driver",
" H:\\Selenium\\chromedriver.exe");

        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://en-gb.facebook.com/");
        driver.findElement(By.xpath("//a[contains(text(),'NewAccount')]")).click();
        Thread.sleep(3000);

//Step.1 Identify the WebElement that need to be handle of the
WebElement Type
WebElement month =
driver.findElement(By.xpath("//select[@id='month']"));

//Step.2 Create the object of the Action class and pass the input
driver
        Actions act = new Actions(driver);

//Step.3 se Action class Method to click
        act.click(month).perform();
        Thread.sleep(3000);

        for(int i=0; i<=4;i++)
        {
            act.sendKeys(Keys.ARROW_UP).perform();
        }
        Thread.sleep(3000);

        act.sendKeys(Keys.ENTER).perform();
    }
}-----
```

VELOCITY

```

package Customized_ListBox;

import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.interactions.Actions;

public class C_Select_Option_Using_HomeKey {
    public static void main(String[] args) throws
InterruptedException {

        System.setProperty("20omepage20.chrome.driver"," H:\\Selenium\\chr
omedriver.exe");

        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();
```

```

driver.get("https://en-gb.facebook.com/");
driver.findElement(By.xpath("//a[contains(text(),'New
Account')]")).click();
Thread.sleep(3000);

//Step.1 Identify the WebElement that need to be handle of
the WebElement Type
WebElement month =
driver.findElement(By.xpath("//select[@id='month']"));

//Step.2 Create the object of the Action class and pass the
input driver
Actions act = new Actions(driver);

//Step.3 se Action class Method to click
act.click(month).perform();
Thread.sleep(3000);

//Step.4 Move the option to Top
act.sendKeys(Keys.HOME).perform();

for(int i =0; i<=7; i++)
{
    act.sendKeys(Keys.ARROW_DOWN).perform();
}

act.sendKeys(Keys.ENTER).perform();
}

}-----
```



```

package Customized_ListBox;

import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.interactions.Actions;

public class D_Select_Option_Using_EndKeys {
    public static void main(String[] args) throws
InterruptedException {

    System.setProperty("webdriver.chrome.driver","H:\\Selenium\\chromedriver.exe");

    WebDriver driver=new ChromeDriver();
    driver.manage().window().maximize();
    driver.get("https://en-gb.facebook.com/");
    driver.findElement(By.xpath("//a[contains(text(),'New
Account')]")).click();
    Thread.sleep(3000);
}
}
```

```

//Step.1 Identify the WebElement that need to be handle of
the WebElement Type
WebElement month =
driver.findElement(By.xpath("//select[@id='month']"));

//Step.2 Create the object of the Action class and pass the
input driver
Actions act = new Actions(driver);

//Step.3 Use Action class method to Select Option
act.click(month).perform();
Thread.sleep(3000);

//Step.4 Move to End
act.sendKeys(Keys.END).perform();

for(int i=1; i<=4; i++)
{
    act.sendKeys(Keys.ARROW_UP).perform();
}

act.sendKeys(Keys.ENTER).perform();
}
}-----

```



Different Types of Exceptions in Selenium.

35	<p>Different type of exceptions in selenium :-</p> <ul style="list-style-type: none"> ① WebDriver Exception :- Url in not proper way. ② Unreachable browser Exception :- when we interrupt the browser while running the test script we will get unreachable exception. ③ UnHandled alert exception → alert popup display and perform the action on browser ④ no alert present Exception :- without alert pop up we perform on other pop up ⑤ Unexpected tagName exception :- ⑥ Not connected exception :- when selenium sat file unable to connect with browser then we will get not connected exception ⑦ Unsupported operation exception :- ⑧ no such a element exception :- when HTML code wrong to identify element / because of synchronization
----	---

VLLUGUTI

WebTable

- Webtable is a WebElement used for the tabular representation of data.

```
<html>
<body>
    <table border=3 id='1234'>
        <tr>
            <th>Sr.No</th>
            <th>BookType</th>
            <th>Cost</th>
        </tr>
        <tr>
            <td>1</td>
            <td>Selenium</td>
            <td>100</td>
        </tr>
        <tr>
            <td>2</td>
            <td>Manual</td>
            <td>200</td>
        </tr>
        <tr>
            <td>3</td>
            <td>Java</td>
            <td>300</td>
        </tr>
    </table>
</body>
</html>
```

Sr.No	BookType	Cost
1	Selenium	100
2	Manual	200
3	Java	300



```
package WebTable;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

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

        System.setProperty("webdriver.chrome.driver", "H:\\Selenium\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();

        driver.get("file:///C:/Users/Jordan/Desktop/Velocity%20Testing1/HTML/Table.html");

        int RowRize =
        driver.findElements(By.xpath("//table[@id='1234']//tr")).size();
        System.out.println(RowRize); //4
    }
}
```

```

package WebTable;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

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

        System.setProperty("25omepage25.chrome.driver","H:\\Selenium\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();

        driver.get("file:///C:/Users/Jordan/Desktop/Velocity%20Testing1/HTML/Table.html");

        int colSize =
driver.findElements(By.xpath("//table[@id='1234']//tr[2]/td")).size();
        System.out.println(colSize);      //3

        //for Header
        int HeadColSize =
driver.findElements(By.xpath("//table[@id='1234']//tr[1]/th")).size();
        System.out.println(HeadColSize); //3
    }
}-----
package WebTable;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

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

        System.setProperty("25omepage25.chrome.driver","H:\\Selenium\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();

        driver.get("file:///C:/Users/Jordan/Desktop/Velocity%20Testing1/HTML/Table.html");

        String text =
driver.findElement(By.xpath("//table[@id='1234']//tr[3]/td[2]")).getText();
        System.out.println(text);    //Manual

        //Header
        String HeaderText =
driver.findElement(By.xpath("//table[@id='1234']//tr[1]/th[2]")).getText();
        System.out.println(HeaderText);      //BookType
    }
}-----

```

```

package WebTable;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class D_getAll_Data_In_Row {
    public static void main(String[] args) {
System.setProperty("webdriver.chrome.driver","H:\\Selenium\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
driver.get("file:///C:/Users/Jordan/Desktop/Velocity%20Testing1/HTML/Table.html");

        for(int i=1; i<=3; i++)
        {
            String CompleteRow =
driver.findElement(By.xpath("//table[@id='1234']//tr[4]/td["+i+"]")).getText();
//RuntimeXpath
            System.out.print(CompleteRow+" ");
//3 Java 300
        }
    }
}-----
```



```

package WebTable;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class E_get_all_Data_In_Col {
    public static void main(String[] args) {
System.setProperty("webdriver.chrome.driver","H:\\Selenium\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
driver.get("file:///C:/Users/Jordan/Desktop/Velocity%20Testing1/HTML/Table.html");

        for (int i = 1; i <=4; i++) // No of Rows
        {
            if(i==1) //for Header
            {
                String HeaderText =
driver.findElement(By.xpath("//table[@id='1234']//tr["+i+"]/th[1]")).getText();
                System.out.println(HeaderText);
            }
            else //for Body
            {
                String bodyText =
driver.findElement(By.xpath("//table[@id='1234']//tr["+i+"]/td[1]")).getText();
                System.out.println(bodyText);
            }
        }
    }
}-----
```

```

package WebTable;

import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

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

        System.setProperty("webdriver.chrome.driver", "H:\\Selenium\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();

        driver.get("file:///C:/Users/Jordan/Desktop/Velocity%20Testing1/HTML/Table.html");

        //int RN =
        driver.findElement(By.xpath("//table[@id='1234']//tr")).size();
        //int RC =
        driver.findElement(By.xpath("//table[@id='1234']//tr[2]/td")).size();

        // You can use RN and RC at i<= RN; j<=RC
        for(int i=1; i<=4; i++)
        {
            for(int j=1; j<=3;j++)
            {
                if(i==1)
                {

String text1 =
driver.findElement(By.xpath("//table[@id='1234']//tr["+i+"]/th["+j+"]"))
.getText();
                System.out.print(text1+" ");
            }
            else
            {
String text2 =
driver.findElement(By.xpath("//table[@id='1234']//tr["+i+"]/td["+j+"]"))
.getText();
                System.out.print(text2+" ");
            }
            System.out.println();
        }
    }
}
}

```

How to Disable Notification

```

//Disable Notification
ChromeOptions options = new ChromeOptions();
options.addArguments("--disable-notifications");

```

Selenium Framework

Q. What is Selenium Framework?

- The Selenium Framework is a code structure that makes code maintenance easy and efficient.
- Without frameworks, users may place the “code” and “data” at the same location which is neither reusable nor readable.
- Frameworks produce beneficial outcomes like increased code reusability, higher portability, reduced cost of script maintenance, better code readability, etc.

Q. Types of Framework?

- There are mainly three type of frameworks created by Selenium WebDriver to automate manual test cases
 - 1. Data Driven
 - 2. Keyword Driven
 - 3. Hybrid

1. Data Driven

- Data Driven Framework in Selenium is a method of separating Test data from the test case.
- Once the Test data are separated from the test case, it can be easily modified for a specific functionality without changing the code.
- It is used to fetch Test Data from external files like Excel, .csv, .xml or some database tables.



FrameWork (Without Data Driven)

```

Package Framework;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.Alert;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class A_Without_DDF {
    public static void main(String[] args) throws
InterruptedException {
    System.setProperty("29omepage29.chrome.driver",
"H:\\Selenium\\chromedriver.exe");
    WebDriver driver = new ChromeDriver();
    driver.manage().window().maximize();
    driver.get("https://kite.zerodha.com/?next=%2Fmargins");

    driver.manage().timeouts().implicitlyWait(25, TimeUnit.SECONDS);

        //Enter UN
    driver.findElement(By.xpath("//input[@id='userid']")).sendKeys("DV1510");

        //Enter Pass
    driver.findElement(By.xpath("//input[@id='password']")).sendKeys("Pass@123");

        //Click on Login
    driver.findElement(By.xpath("//button[text()='Login ']")).click();

        //Enter Pin
    driver.findElement(By.xpath("//input[@id='pin']")).sendKeys("959594");

        //Click on Continue
    driver.findElement(By.xpath("//button[text()='Continue ']")).click();

        //Get Actual User ID
String actUserID = driver.findElement(By.xpath("//span[@class='user-id']")).getText();
System.out.println(actUserID); //DV1510
String ExpUserId = "DV1510"; //Already declared by User

        boolean result = actUserID.equals(ExpUserId);
        System.out.println(result); //true
        if(result=true) //Test Case Pass
        {
            System.out.println("Test Case Pass");
        }
        else
        {
            System.out.println("Test Case Fail");
        }
    }
}
-----
```

```

package FrameWork;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.concurrent.TimeUnit;
import org.apache.poi.EncryptedDocumentException;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.WorkbookFactory;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class B_WithDataDefinF1 {
    public static void main(String[] args) throws
EncryptedDocumentException, IOException {
        System.setProperty("30omepage30.chrome.driver",
"H:\\Selenium\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://kite.zerodha.com/?next=%2Fmargins");
        driver.manage().timeouts().implicitlyWait(25, TimeUnit.SECONDS);
        FileInputStream file = new
FileInputStream("C:\\Users\\Jordan\\Desktop\\Velocity
Testing1\\XcelSheet\\Kite Login.xlsx");

        Sheet sh = WorkbookFactory.create(file).getSheet("Sheet1");
        //Enter UN
        String UN = sh.getRow(0).getCell(0).getStringCellValue();
        driver.findElement(By.xpath("//input[@id='userid']")).sendKeys(UN);

        //Enter Pass
        String Pass = sh.getRow(0).getCell(1).getStringCellValue();
        driver.findElement(By.xpath("//input[@id='password']")).sendKeys(Pass);

        //Click on Login
        driver.findElement(By.xpath("//button[text()='Login ']")).click();

        //Enter Pin
        String Pin = sh.getRow(0).getCell(2).getStringCellValue();
        driver.findElement(By.xpath("//input[@id='pin']")).sendKeys(Pin);

        //Click on Continue
        driver.findElement(By.xpath("//button[text()='Continue ']")).click();

        //Get Actual User ID
        String actUserID = driver.findElement(By.xpath("//span[@class='user-
id']")).getText();
        System.out.println(actUserID); //DV1510
        String ExpUserId = "DV1510";
        boolean result = actUserID.equals(ExpUserId);
        if(result=true) //Test Case Pass
        {
            System.out.println("Test Case Pass");
        }
        else{
            System.out.println("Test Case Fail");
        }
    }-----
}

```

```

package Framework;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.concurrent.TimeUnit;
import org.apache.poi.EncryptedDocumentException;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.WorkbookFactory;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.remote.server.handler.SendKeys;
public class B_With_Data_Driven_F1 {
    public static void main(String[] args) throws EncryptedDocumentException, IOException {
System.setProperty("31omepage31.chrome.driver","H:\\Selenium\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://kite.zerodha.com/?next=%2Fmargins");
driver.manage().timeouts().implicitlyWait(25, TimeUnit.SECONDS);

FileInputStream file = new
FileInputStream("C:\\Users\\Jordan\\Desktop\\Velocity
Testing1\\XcelSheet\\Kite Login.xlsx");
        Sheet sh = WorkbookFactory.create(file).getSheet("Sheet1");

        //UN
        driver.findElement(By.xpath("//input[@id='userid']")).sendKeys(sh.getRow(0).getCell(0).getStringCellValue());

        //Pass
        driver.findElement(By.xpath("//input[@id='password']")).sendKeys(sh.get
        tRow(0).getCell(1).getStringCellValue());
        //Click on Login
        driver.findElement(By.xpath("//button[text()='Login ']")).click();

        //Enter Pin
        driver.findElement(By.xpath("//input[@id='pin']")).sendKeys(sh.getRow(0).getCell(2).getStringCellValue());

        //Click on Continue
        driver.findElement(By.xpath("//button[text()='Continue ']")).click();
        //Get Actual User ID
String actUserID = driver.findElement(By.xpath("//span[@class='user-
id']")).getText();
        System.out.println(actUserID); //DV1510
        String ExpUserId = "DV1510";
        boolean result = actUserID.equals(ExpUserId);
        System.out.println(result); //true
        if(result=true) //Test Case Pass
        {
            System.out.println("Test Case Pass");
        }
        else
        {
            System.out.println("Test Case Fail");
        } }-----

```

● Encapsulation

Encapsulation is an OOPS concept where Hiding data from other classes using private access specifier is called encapsulation

1. Data member should be declared **globally** with access level **private**
2. Initialize within a **constructor** with access level **public**
3. Utilize within a method with access level **public**

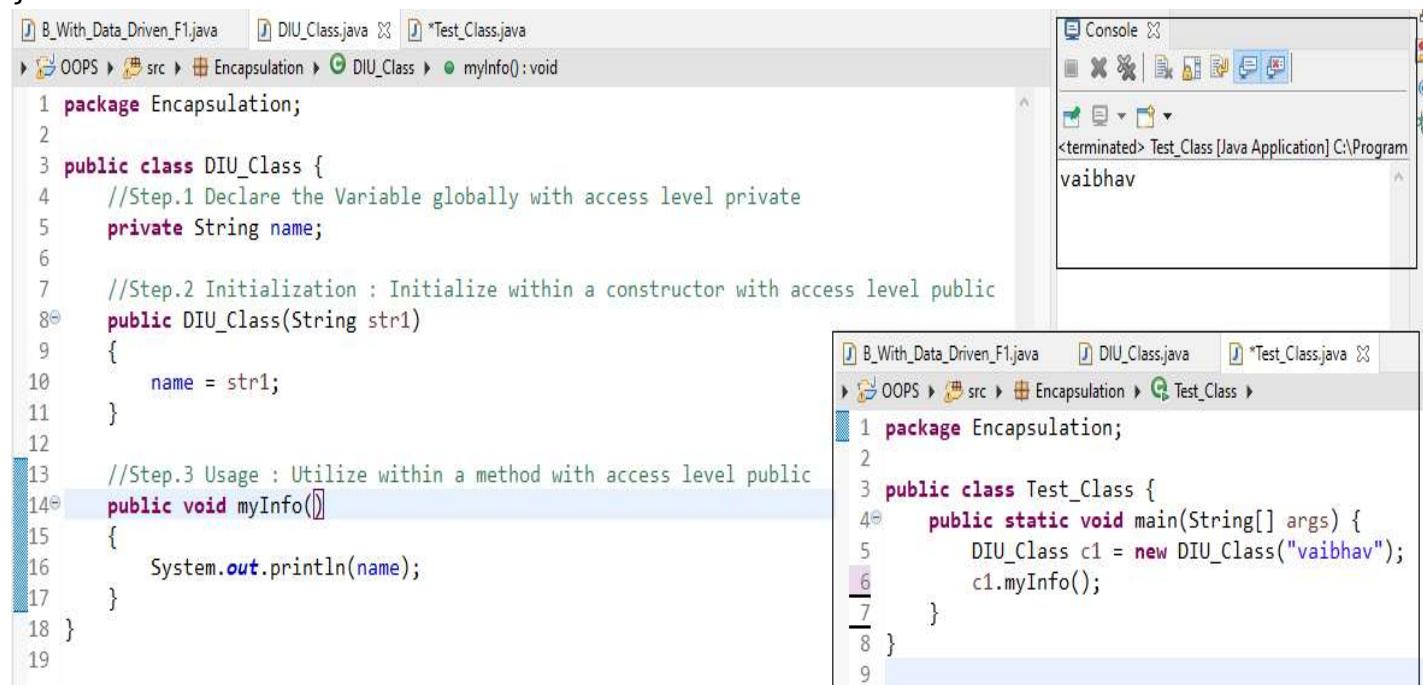
```
package Encapsulation;
public class DIU_Class {
    //Step.1 Declare the Variable globally with access level private
    private String name;

    //Step.2 Initialization : Initialize within a constructor with access
    //level public
    public DIU_Class(String str1)
    {
        name = str1;
    }

    //Step.3 Usage : Utilize within a method with access level public
    public void myInfo()
    {
        System.out.println(name);
    }
}-----
```



```
package Encapsulation;
public class Test_Class {
    public static void main(String[] args) {
        DIU_Class c1 = new DIU_Class("vaibhav");
        c1.myInfo();
    }
}
```



The screenshot shows an IDE interface with two tabs open: DIU_Class.java and Test_Class.java. The code for DIU_Class.java is as follows:

```
1 package Encapsulation;
2
3 public class DIU_Class {
4     //Step.1 Declare the Variable globally with access level private
5     private String name;
6
7     //Step.2 Initialization : Initialize within a constructor with access level public
8     public DIU_Class(String str1)
9     {
10         name = str1;
11     }
12
13     //Step.3 Usage : Utilize within a method with access level public
14     public void myInfo()
15     {
16         System.out.println(name);
17     }
18 }
```

The code for Test_Class.java is as follows:

```
1 package Encapsulation;
2
3 public class Test_Class {
4     public static void main(String[] args) {
5         DIU_Class c1 = new DIU_Class("vaibhav");
6         c1.myInfo();
7     }
8 }
```

To the right of the code, there are two windows: 'Console' and 'Output'. The 'Console' window shows the output of the program: 'vaibhav'. The 'Output' window shows the command: '<terminated> Test_Class [Java Application] C:\Program'.

Pagefactory:

- It is a class which contains static method like `initElements`.
- To initialize global variable in `PageFactory` we need to use `initElements` method within the constructor.

Syntax:	<code>PageFactory.initElements(driver, this);</code>
---------	--

- `initElements` will initialize D.M by identifying each element present in a webpage by using `@findBy` annotation, Which takes locator type as an input.

Syntax:	<code>@FindBy(locator Type =”locator value/exression”)</code>
	<code>private WebElement D.M. ;</code>

Working of PageFactory:

1. While executing Test Script `initElement` method will convert all the data members `@findBy` annotation to `findElement()`, This process is known as basic/early initialization.-->after creating object of pom class

`@FindBy(xpath=”//span[text()=’KV’]”)` private WebElement PN;

--→ `private WebElement PN = driver.findElement(By.xpath(“//span[text()=’KV’]”));`

2. To perform action on component we need to call a methods.
3. Before performing each action `initElement` method will identifies component present or not, then it will do complete initialization. This process is known as late/lazy initialization. --→after method call

Disadvantage of simple POM:

- simple POM will initialize all the component before performing actions(after object creation), but sometimes application may contains few components which will be hidden &
- Displayed once we perform action on other components, that hidden component will not be displayed while pom initializing, so it throws “**No such element**” exception.

To overcome drawback of simple pom, we need to use “`PageFactory`” which is an extension of pom.

Difference between POM & PageFactory

simple POM	Pom with PageFactory
1. It will initialize/identify all the D.M. present in class completely before performing action on components	1. It will initialize/identify the D.M. present in a class before performing each action.
2. It will use if webpage is not containing hidden elements.	2. It will be used if webpage is containing hidden elements.

Advantages of POM:

- 1) Makes the code reusable
- 2) It is useful in reducing code duplication
- 3) It makes ease in maintaining the code (flow in the UI is separated from verification)—TestNG
- 4) Makes code readable (Methods get more realistic names)
- 5) The Code becomes less and optimized.

Pom class:

1. Pom class depends on webpage present in an application.
2. For each webpage pom class will be created, number of POM class depends on number of webpages present in an application.
3. In each POM class D.M./variable are created in encapsulation concept by using pagefactory.
4. No of D.M. created in POM class will depend on no of elements present in a webpage.
5. Each declared D.M. should initialized & utilized in POM class.

**Test class:**

1. Test class depends on number of Test cases written by manual Test engineer.
2. Test class will contains navigation steps & inputs that need to be given to the components/elements.
3. In test class data/inputs that can be given directly or through external source like Excel sheet.

POM with pagefactory

POM (page object module):

- It is a **java design pattern** use for **design of classes** in **Test script**.
- Page Object module is an object design pattern in Selenium, where
 1. **Web pages** are represented as **POM** classes &
 2. The various elements on the webpage are defined as **variables** on the **POM** class.

In this case we will use **Page Factory** to initialize **web elements** that are defined in web page classes or Page Objects.

POM strictly follows encapsulation concept where

1. Data member should be declared globally with access level private using **@findBy** Annotation
2. Initialize within a constructor with access level public using pagefactory class
3. Utilize within a method with access level public



Note:

1. No of Data Member that need to be created under a **POM** class will depends on no of element that need to be handle in a webpage.
2. **POM** class will not contain a main method, to run a **POM** class we require another class with main() i.e. **Test class**
3. Test class will contain all the navigation steps to test an application

```

package POM_With_PageFactory;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;

//Regular class i.e. main method not present      //POM Class
public class A_kiteLogin1Page {
//Step.1 Declaration of variable Globally with access level private
//Use @findBy Annotation
@FindBy(xpath="//input[@id='userid']") private WebElement UN;

@FindBy(xpath="//input[@id='password']") private WebElement PASS;

@FindBy(xpath="//button[text()='Login ']") private WebElement SbtBtn;

```

```

//Step.2 Initialization : Initialize within a constructor with access
level public
    public A_kiteLoginPage(WebDriver driver)
    {
        PageFactory.initElements(driver, this);
    }
//Step.3 Usage : Utilize within a method with Access level public
    public void enterUN()
    {
        UN.sendKeys("DV1510");
    }
    public void enterPass()
    {
        PASS.sendKeys("Pass@123");
    }
    public void clickButton()
    {
        SbtBtn.click();
    }
}-----
package POM_With_PageFactory;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
//Regular Class : No Main Method //POM Class
public class A_kiteLoginPage {
    //Step.1 Declare the variable globally with access level private
    @FindBy(xpath="//input[@id='pin']") private WebElement pin;
    @FindBy(xpath="//button[text()='Continue ']")private WebElement
cntBtn;

    //Step.2 Initialize within constructor with access level public
    public A_kiteLoginPage(WebDriver driver)
    {
        PageFactory.initElements(driver, this);
    }

    //Step.3 Usage: Utilize within a method access level public
    public void enterPin()
    {
        pin.sendKeys("959594");
    }

    public void clickButton2()
    {
        cntBtn.click();
    }
}-----
package POM_With_PageFactory;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;

```

```

import org.openqa.selenium.support.PageFactory;
//POM Class
public class A_kiteLogin3HomePage {
//Step.1 Declare the variable globally with access level private
@FindBy(xpath="//span[@class='user-id']") private WebElement UserID;

//Step.2 Initialize within the constructor with access level public
public A_kiteLogin3HomePage(WebDriver driver)
{
    PageFactory.initElements(driver, this);
}
//Step.3 Utilize within the method with access level public
public void verifyID()
{
    String actId = UserID.getText();
    String expId = "DV1510";

    if(actId.equalsIgnoreCase(expId))
    {
        System.out.println("Pass");
    }
    else
    {
        System.out.println("Fail");
    }
}
}-----  

package POM_With_PageFactory;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class A_TestClassKite {
    public static void main(String[] args) {
System.setProperty("webdriver.chrome.driver","H:\\Selenium\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://kite.zerodha.com/");
        driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);

        A_kiteLogin1Page page1 = new A_kiteLogin1Page(driver);
        page1.enterUN();
        page1.enterPass();
        page1.clickButton();

        A_kiteLogin2Page page2 = new A_kiteLogin2Page(driver);
        page2.enterPin();
        page2.clickButton2();

        A_kiteLogin3HomePage 37omepage = new
A_kiteLogin3HomePage(driver);
        37omepage.verifyID();
    }
}
-----
```

```

1 package POM_With_PageFactory;
2 import org.openqa.selenium.WebDriver;
3 import org.openqa.selenium.WebElement;
4 import org.openqa.selenium.support.FindBy;
5 import org.openqa.selenium.support.PageFactory;
6 //Regular class i.e. main method not present //POM Class
7 public class A_kiteLoginPage {
8 //Step.1 Declaration of variable Globally with access level private
9 //Use @FindBy Annotation
10 @FindBy(xpath="//input[@id='userid']") private WebElement UN;
11 @FindBy(xpath="//input[@id='password']") private WebElement PASS;
12 @FindBy(xpath="//button[text()='Login ']") private WebElement SbtBtn;
13
14 //Step.2 Initialization : Initialize within a constructor with access level public
15
16 //Step.3 Usage : Utilize within a method with Access level public
17 public A_kiteLoginPage(WebDriver driver)
18 {
19     PageFactory.initElements(driver, this);
20 }
21
22 //Step.3 Usage : Utilize within a method with Access level public
23 public void enterUN()
24 {
25     UN.sendKeys("DV1510");
26 }
27 public void enterPass()
28 {
29     PASS.sendKeys("Pass@123");
30 }
31 public void clickButton()
32 {
33     SbtBtn.click();
34 }
35 }
36

```

1

```

1 package POM_With_PageFactory;
2
3 import org.openqa.selenium.WebDriver;
4 import org.openqa.selenium.WebElement;
5 import org.openqa.selenium.support.FindBy;
6 import org.openqa.selenium.support.PageFactory;
7
8 //Regular Class : No Main Method //POM Class
9 public class A_kiteLogin2Page {
10 //Step.1 Declare the variable globally with access level private
11 @FindBy(xpath="//input[@id='pin']") private WebElement pin;
12 @FindBy(xpath="//button[text()='Continue ']") private WebElement cntBtn;
13
14 //Step.2 Initialize within constructor with access level public
15 public A_kiteLogin2Page(WebDriver driver)
16 {
17     PageFactory.initElements(driver, this);
18 }
19
20 //Step.3 Usage: Utilize within a method access level public
21
22 public void enterPin()
23 {
24     pin.sendKeys("959594");
25 }
26
27 public void clickButton2()
28 {
29     cntBtn.click();
30 }
31
32
33
34
35
36
37

```

2

```

1 package POM_With_PageFactory;
2 import org.openqa.selenium.WebDriver;
3 import org.openqa.selenium.WebElement;
4 import org.openqa.selenium.support.FindBy;
5 import org.openqa.selenium.support.PageFactory;
6
7 //POM Class
8 public class A_kiteLogin3HomePage {
9 //Step.1 Declare the variable globally with access level private
10 @FindBy(xpath="//span[@class='user-id']") private WebElement UserID;
11
12 //Step.2 Initialize within the constructor with access level public
13 public A_kiteLogin3HomePage(WebDriver driver)
14 {
15     PageFactory.initElements(driver, this);
16 }
17
18 //Step.3 Utilize within the method with access level public
19 public void verifyID()
20 {
21     String actId = UserID.getText();
22     String expId = "DV1510";
23
24     if(actId.equalsIgnoreCase(expId))
25     {
26         System.out.println("Pass");
27     }
28     else
29     {
30         System.out.println("Fail");
31     }
32 }
33

```

3

```

1 package POM_With_PageFactory;
2
3 import java.util.concurrent.TimeUnit;
4 import org.openqa.selenium.WebDriver;
5 import org.openqa.selenium.chrome.ChromeDriver;
6
7 public class A_TestClassKite {
8     public static void main(String[] args) {
9         System.setProperty("webdriver.chrome.driver", "H:\\Selenium\\chromedriver.exe");
10        WebDriver driver = new ChromeDriver();
11        driver.manage().window().maximize();
12        driver.get("https://kite.zerodha.com/");
13        driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);
14
15        A_kiteLoginPage page1 = new A_kiteLoginPage(driver);
16        page1.enterUN();
17        page1.enterPass();
18        page1.clickButton();
19
20        A_kiteLogin2Page page2 = new A_kiteLogin2Page(driver);
21        page2.enterPin();
22        page2.clickButton2();
23
24        A_kiteLogin3HomePage homePage = new A_kiteLogin3HomePage(driver);
25        homePage.verifyID();
26    }
27

```

4

Page Object Module with Data Driven Frame

Note : First three Classes i.e. LoginPage1, LoginPage2, HomePage3, make Some changes : Remove the Data Provided like UserId, Pass, Pin. (Changes have to made displayed on next page)

This is Test Class Only



```

1 package POM_With_DDF;
2 import java.io.FileInputStream;
3 import java.io.FileNotFoundException;
4 import java.io.IOException;
5 import java.util.concurrent.TimeUnit;
6 import org.apache.poi.EncryptedDocumentException;
7 import org.apache.poi.ss.usermodel.Sheet;
8 import org.apache.poi.ss.usermodel.WorkbookFactory;
9 import org.openqa.selenium.WebDriver;
10 import org.openqa.selenium.chrome.ChromeDriver;
11
12 public class A_TestClass {
13     public static void main(String[] args) throws EncryptedDocumentException, IOException {
14         //Open eXcel Sheet and Get Data
15         FileInputStream file = new FileInputStream("C:\\\\Users\\\\Jordan\\\\Desktop\\\\Velocity Testing1\\\\XcelSheet\\\\Kite Login.xlsx");
16
17
18         Sheet sh = WorkbookFactory.create(file).getSheet("LoginSheet");
19
20         System.setProperty("webdriver.chrome.driver", "H:\\\\Selenium\\\\chromedriver.exe");
21         WebDriver driver = new ChromeDriver();
22         driver.manage().window().maximize();
23         driver.get("https://kite.zerodha.com/");
24         driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);
25
26         A_loginPage1 page1 = new A_loginPage1(driver);
27         String TID = sh.getRow(0).getCell(0).getStringCellValue();
28         page1.setEnterID(TID);
29
30         String TPass = sh.getRow(0).getCell(1).getStringCellValue();
31         page1.setEnterPass(TPass);
32
33         page1.clickLoginButton();
34
35         A_loginPage2 page2 =new A_loginPage2(driver);
36         String TPIN = sh.getRow(0).getCell(2).getStringCellValue();
37         page2.setEnterPin(TPIN);
38         page2.clickContinue();
39
40         A_HomePage3 page3 =new A_HomePage3(driver);
41         String tVerify = sh.getRow(0).getCell(3).getStringCellValue();
42         page3.VerifyKiteHomePageUserID(TID);
43     }
44 }
```

Method : 1

```

A_loginPage1.java A_loginPage2.java A_HomePage3.java *A_TestClass.java *A_TestClass2.java
Selenium src POM_With_DDF A_TestClass2 main(String[]) : void

1 package POM_With_DDF;
2
3 import java.io.FileInputStream;
4 import java.io.FileNotFoundException;
5 import java.io.IOException;
6 import java.util.concurrent.TimeUnit;
7
8 import org.apache.poi.EncryptedDocumentException;
9 import org.apache.poi.ss.usermodel.Sheet;
10 import org.apache.poi.ss.usermodel.WorkbookFactory;
11 import org.openqa.selenium.WebDriver;
12 import org.openqa.selenium.chrome.ChromeDriver;
13
14 public class A_TestClass2
15 {
16     public static void main(String[] args) throws EncryptedDocumentException, IOException {
17
18         FileInputStream file = new FileInputStream("C:\\\\Users\\\\Jordan\\\\Desktop\\\\Velocity"
19             + " Testing\\\\XcelSheet\\\\Kite Login.xlsx");
20         Sheet sh = WorkbookFactory.create(file).getSheet("LoginSheet");
21
22         System.setProperty("webdriver.chrome.driver", "H:\\\\Selenium\\\\chromedriver.exe");
23         WebDriver driver = new ChromeDriver();
24         driver.manage().window().maximize();
25         driver.get("https://kite.zerodha.com/");
26         driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);
27
28         A_loginPage1 page1 = new A_loginPage1(driver);
29         page1.setEnterID(sh.getRow(0).getCell(0).getStringCellValue());
30         page1.setEnterPass(sh.getRow(0).getCell(1).getStringCellValue());
31         page1.clickLoginButton();
32
33         A_loginPage2 page2 = new A_loginPage2(driver);
34         page2.setEnterPin(sh.getRow(0).getCell(2).getStringCellValue());
35
36         A_HomePage3 page3 = new A_HomePage3(driver);
37         page3.VerifyKiteHomePageUserID(sh.getRow(0).getCell(3).getStringCellValue());
38     }
39 }
40
41     String tVerify = sh.getRow(0).getCell(3).getStringCellValue();
42     page3.VerifyKiteHomePageUserID(TID);
43 }
44

```

Method : 2

```

//Step.2 Initialize

A_loginPage1(WebDriver driver)
{
    PageFactory.initElements(driver, this);
}

public void setEnterID(String userID)
{
    ID.sendKeys(userID);
}

public void setEnterPass(String EntPass)
{
    Pass.sendKeys(EntPass);
}

public void clickLoginButton()
{
    lgnBtn.click();
}

```

```

//Step.3 Usage

public void setEnterPin(String entPin)
{
    PIN.sendKeys(entPin);
}

public void clickContinue()
{
    Continue.click();
}

```

```

//Usage
public void VerifyKiteHomePageUserID(String expID)
{
    String actualID = UserID.getText();
    if(actualID.equals(expID))
    {
        System.out.println("Pass");
    }
    else
    {
        System.out.println("fail");
    }
}

```

• TestNG

- TestNG is a java unit framework used for writing/designing of Test classes.
- Main class is not required to execute the program, Use annotation @Test for the execution.

When @Test is called then

1. click on add testNG library
 2. click on import Test(org.testng.annotations)
-

Features of TestNG

1. Emailable Report
 - 2.
 - 3.
 - 4.
 - 5.
-

1. Emailable Report

- Report generation is very important when you are doing the Automation Testing as well as for Manual Testing.
- By looking at the result, you can easily identify how many test cases are passed, failed and skipped.
- By looking at the report, you will come to know what the status of the project is.
- Selenium web driver is used for automating the web-application, but it won't generate any reports.

The TestNG will generate the default report.

---Steps to generate Emailable report---

1. Execute Test class and refresh the project.
2. You will get test-output folder.
3. In That folder Right click on the "emailable-report.html" and select the option Open with the web browser or double click on it.

Note:

1. If we use **sop()** to display text as a output then result will be displayed in console not in Emailable report. **[sop() → Display only in Console]**
 2. To display text in Emailable report we need to use static method log present in Reporter class.
eg. Reporter.log("String msg", true)
 - **sop () → console**
 - **Reporter.log () → Emailable report**
 - **Reporter.log ("msg", true) → Console+ Emailable report**
 - **Reporter.log ("msg", false)/Reporter.log ("msg") → Emailable report**
-

2. TestNG Annotation

1. **@BeforeTest (1)**: Always Execute First (i.e before all the methods/classes)
2. **@BeforeClass(2)**: - It is use for execution of test method before execution of test class.
3. **@BeforeMethod(3)**: - It is use for execution of test method before execution of every test method with an annotation **@Test**.
4. **@Test(4)**: - Use for execution of test method/TC.
5. **@AfterMethod(5)**: - It is use for execution of test method after execution of every test method with an annotation **@Test**.
6. **@AfterClass: (6)** - It is use for execution of test method after execution of test class.
7. **@AfterTest (7)**: Always Execute Last (i.e after all the methods/classes)
8. **@BeforeSuit()**
9. **@AfterSuit()**

Note : **@Test** : Methods are execute in alphabetically default (if priority not mentioned)

```
package TestNG;                                A_Normal_Test_Class

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class A_Normal_Test_Class {
    public static void main(String[] args) {
        // Normal Test Class consist of Main Method
        System.setProperty("webdriver.chrome.driver",
"H:\\\\Selenium\\\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://kite.zerodha.com/");
    }
}
```

```
package TestNG;                                TestNG Class

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class B_TestNG_TestClass {
    // In TestNG class Main method is not required.
    // Call Any method and add Annotation @Test before the class

    public void OpenKite()
    {
        System.setProperty("webdriver.chrome.driver",
"H:\\\\Selenium\\\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://kite.zerodha.com/");
    }
}
```

```
package TestNG;
import org.testng.Reporter;
import org.testng.annotations.Test;
public class C_Emailable_Report {
    // E-Mailable Report is Possible only for @Test
    //Scenario 1 : Use of SOP();
    @Test
    public void TC0()
    {
        System.out.println("Running Test Case 0");
    }

    //Scenario 2 : Use of Reporter.log(, true);
    @Test
    public void TC1()
    {
        Reporter.Log("Running TC1", true);
    }

    //Scenario 3 : Use of Reporter.log(, false/ Null);
    @Test
    public void TC2()
    {
        Reporter.Log("Running TC2", false);
    }

    @Test
    public void TC3()
    {
        Reporter.Log("Running TC3", true);
    }

    @Test
    public void TC4()
    {
        Reporter.Log("Running TC4" );
    }
```



Console Output

```
<terminated> C_Emailable_Report [TestNG] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe
[RemoteTestNG] detected TestNG version 7.4.0
Running Test Case 0
Running TC1
Running TC3
PASSED: TC2
PASSED: TC4
PASSED: TC0
PASSED: TC1
PASSED: TC3

=====
Default test
Tests run: 5, Failures: 0, Skips: 0
=====

=====
Default suite
Total tests run: 5, Passes: 5, Failures: 0, Skips: 0
=====
```

Test	# Passed	# Skipped	# Retried	# Failed	Time (ms)	Included Groups	Excluded Groups
Default suite							
Default test	5	0	0	0	196		
Default suite							
Default test — passed							
TestNG C_Emailable_Report	TC0	1637203389383	26				
	TC1	1637203389415	4				
	TC2	1637203389423	4				
	TC3	1637203389431	5				
	TC4	1637203389441	28				

Default test

TestNG.C_Emailable_Report#TC0

[back to summary](#)

TestNG.C_Emailable_Report#TC1

[back to summary](#)

TestNG.C_Emailable_Report#TC1

[back to summary](#)

TestNG.C_Emailable_Report#TC2

[back to summary](#)

TestNG.C_Emailable_Report#TC3

[back to summary](#)

TestNG.C_Emailable_Report#TC4

[back to summary](#)

```

package TestNG;
import org.testng.Reporter;
import org.testng.annotations.AfterClass;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.AfterTest;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.BeforeTest;
import org.testng.annotations.Test;

public class D_Annotations {

```

Annotations

```

@BeforeClass
public void OpenBrowser()
{
    Reporter.Log("Open Browser", true);
}

@BeforeMethod
public void loginToApp()
{
    Reporter.Log("Login App", true);
}

@Test
public void T2VerifyUserId()
{
    Reporter.Log("Verify User ID 2", true);
}

@Test
public void T1VerifyUserId()
{
    Reporter.Log("Verify User ID 1", true);
}

@AfterMethod
public void logout()
{
    Reporter.Log("Click on Logout", true);
}

@AfterTest
public void closeBrowser()
{
    Reporter.Log("Close Browser", true);
}
}

```



```

Console X
<terminated> D_Annotations [TestNG] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (Nov 14, 2022, 10:45:20)
[RemoteTestNG] detected TestNG version 7.4.0
Open Browser
Login App
Verify User ID 1
Click on Logout
Login App
Verify User ID 2
Click on Logout
Close Browser
PASSED: T1VerifyUserId
PASSED: T2VerifyUserId

=====
Default test
Tests run: 2, Failures: 0, Skips: 0
=====

=====
Default suite
Total tests run: 2, Passes: 2, Failures: 0, Skips: 0
=====
```

Results of running class D_Annotations

Test	# Passed	# Skipped	# Retried	# Failed	Time (ms)	Included Groups	Excluded Groups
Default test	2	0	0	0	154		

Default suite

Default test — passed

Class	Method	Start	Time (ms)
TestNG D_Annotations	T1VerifyUserId	1637220166577	14
	T2VerifyUserId	1637220166604	7

Default test

TestNG.D_Annotations#T1VerifyUserId

Messages
Verify User ID 1

[back to summary](#)

TestNG.D_Annotations#T2VerifyUserId

Messages
Verify User ID 2

[back to summary](#)

3. TestNG Keywords

1. invocationCount	: Same Test Case Run Multiple Time	invocationCount=5;
2. priority	: Set Execution Method Order	priority=1
3. enabled	: use to disable Test Method	enable = false
4. TimeOut	: Fail the Test Case if Time Consuming	@Test(timeOut=8000)
5. dependsOnMethods:	If 1 TC depend upon other	dependsOnMethods= {"TC name"}

1. invocationCount:

- Sometimes same test method/TC need to be executed multiple which can be possible by using TestNG keyword "invocationCount"
- eg. invocationCount=5;

2. priority:

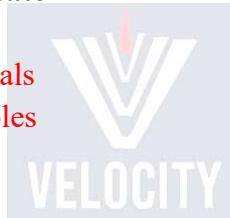
- To change test method/TC execution order we need to use TestNG keyword "priority".
- eg. priority=1

Note: priority can be

1. by **default=0**
2. +ve integer
3. -ve integer
4. **Duplicate**

Priority can't be

1. Decimals
2. Variables



3. enabled:

- Disabling a test method/TC in TestNG can be achieved by setting the enabled attribute of the @Test annotation to false.
- eg. enabled=false

4. TimeOut:

- If test class contains multiple test methods if one of the test method is time consuming to execute then TestNG by default fail that TC & executes the other TC.
- eg. @Test(timeOut=8000)

5. dependsOnMethods:

- If 1 TC execution depends on multiple TC then we need to use "dependsOnMethods" TestNG keyword.
- eg. dependsOnMethods= {"TC name"}

```

package TestNG_Keywords;           //Priority
import org.testng.Reporter;
import org.testng.annotations.Test;
public class A_Priority1 {
    @Test(priority = 2)
    public void TC1()
    {
        Reporter.log("Running TC1", true);
    }

    @Test(priority = 1)
    public void TC2()
    {
        Reporter.log("Running TC2", true);
    }

    @Test      //Bydefault priority = 0
    public void TC3()
    {
        Reporter.log("Running TC3", true);
    }
}

```

```

@Test(priority = -7)
public void TC4()
{
    Reporter.log("Running TC4", true);
}

```

```

@Test(priority = -3)
public void TC5()
{
    Reporter.log("Running TC5", true);
}

```

//Note : Method will execute according to lowest value of priority
//In Above Method Test case Execute in Order = -7, -3, 0, 1, 2
// If Two default Priority=0 then execute alphabetically

```

package TestNG_Keywords;           //InvocationCount

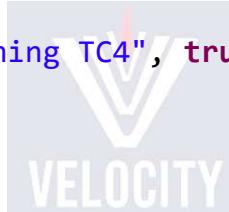
import org.testng.Reporter;
import org.testng.annotations.Test;

public class B_Invocation_Count {

    @Test(invocationCount = 4)
    public void TC1()
    {
        Reporter.log("Running TC1", true);
    }
}

/invocationCount = 4 means Execute TC1 Four Times
}-

```

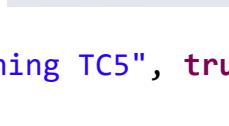


```

Console × TestNG Report
<terminated> A_Priority1 [TestNG] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (Nov 1)
[RemoteTestNG] detected TestNG version 7.4.0
Running TC4
Running TC5
Running TC3
Running TC2
Running TC1
PASSED: TC5
PASSED: TC2
PASSED: TC4
PASSED: TC3
PASSED: TC1
=====
Default test
Tests run: 5, Failures: 0, Skips: 0
=====

=====
Default suite
Total tests run: 5, Passes: 5, Failures: 0, Skips: 0
=====


```



```

Console × TestNG Report
<terminated> B_Invocation_Count [TestNG] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (Nov 1)
[RemoteTestNG] detected TestNG version 7.4.0
Running TC1
Running TC1
Running TC1
Running TC1
PASSED: TC1
PASSED: TC1
PASSED: TC1
PASSED: TC1
=====
Default test
Tests run: 1, Failures: 0, Skips: 0
=====

=====
Default suite
Total tests run: 4, Passes: 4, Failures: 0, Skips: 0
=====


```

```
package TestNG_Keywords;
```

```
import org.testng.Reporter;
import org.testng.annotations.Test;
```

```
public class C_enabled {
```

```
 @Test
```

```
 public void TC1()
```

```
{
```

```
 Reporter.Log("Running TC1", true);
```

```
}
```

```
@Test(enabled = false)
```

```
 public void TC2()
```

```
{
```

```
 Reporter.Log("Running TC2", true);
```

```
}
```

```
 @Test
```

```
 public void TC3()
```

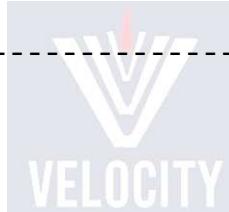
```
{
```

```
 Reporter.Log("Running TC3", true);
```

```
}
```

Here TC2 is Enabled = false i.e Disabled so only TC1 and TC3 will Execute

```
}
```



```
package TestNG_Keywords;
```

```
import org.testng.Reporter;
```

```
import org.testng.annotations.Test;
```

```
public class D_timeOut {
```

```
 @Test(timeOut = 5000)
```

```
 public void TC1() throws InterruptedException
```

```
{
```

```
 Thread.sleep(7000);
```

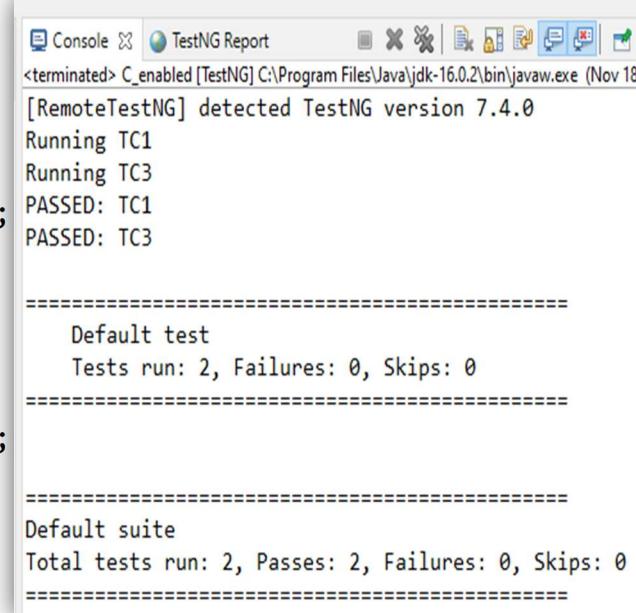
```
 Reporter.Log("Running TC1", true);
```

```
}
```

//Use to Fail the Test Case if Time Required for execution is more.

Test	# Passed	# Skipped	# Retried	# Failed	Time (ms)	Included Groups	Excluded Groups
Default suite							
Default test	0	0	0	1	5,125		
Default suite							
TestNG_Keywords.D_timeOut#TC1							
Default test							
TestNG_Keywords.D_timeOut#TC1							

```
}
```



The screenshot shows the TestNG Report window with the following output:

```

Console TestNG Report
<terminated> C_enabled [TestNG] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (Nov 18, 2023, 10:45:45 AM)
[RemoteTestNG] detected TestNG version 7.4.0
Running TC1
Running TC3
PASSED: TC1
PASSED: TC3

=====
Default test
Tests run: 2, Failures: 0, Skips: 0
=====

=====
Default suite
Total tests run: 2, Passes: 2, Failures: 0, Skips: 0
=====
```

```

package TestNG_Keywords;

import org.testng.Reporter;
import org.testng.annotations.Test;
import org.testng.Assert;

public class E_dependsOnMethods {
    @Test
    public void login()
    {
        Assert.fail(); // Forcefully fail the method
        Reporter.log("running login TC...",true);
    }

    @Test(dependsOnMethods= {"login"})
    public void logout()
    {
        Reporter.log("running logout TC...",true);
    }
    //Logout is depends upon login Method
}

```

Test	# Passed	# Skipped	# Retried	# Failed	Time (ms)	Included Groups	Excluded Groups
Default suite							
Default test	0	1	0	1	169		
Default suite							
Default test — failed							
TestNG_Keywords.E_dependsOnMethods		login		1637259953860	23		
Default test — skipped							
TestNG_Keywords.E_dependsOnMethods		logout		1637259953898	0		

Default test

TestNG_Keywords.E_dependsOnMethods#login

```

}-->
package TestNG_Keywords;


```

```

import org.testng.Reporter;
import org.testng.annotations.Test;

public class E_dependsOnMethods2 {
    @Test
    public void login1()
    {
        Reporter.log("running login1 TC...",true);
    }

    @Test(timeOut=5000)
    public void login2() throws InterruptedException
    {
        Thread.sleep(7000);
        Reporter.log("running login2 TC...",true);
    }
}

```

```

@Test(dependsOnMethods= {"login1","login2"})
public void logout()
{
    Reporter.log("running logout TC...",true);
}

```

Test	# Passed	# Skipped	# Retried	# Failed	Time (ms)	Included Groups	Excluded Groups
Default suite							
Default test	1	1	0	1	5,130		
Class Method Start Time (ms)							
Default suite							
Default test — failed							
TestNG_Keywords.E_dependsOnMethods2	login2		1637260299872	5011			
Default test — skipped							
TestNG_Keywords.E_dependsOnMethods2	logout		1637260304908	0			
Default test — passed							
TestNG_Keywords.E_dependsOnMethods2	login1		1637260299851	15			

Default test

TestNG_Keywords.E_dependsOnMethods2#login2

IF

```

@Test(timeout=5000)
public void login2() throws InterruptedException
{
    Thread.sleep(1000);
    Reporter.log("running login2 TC...",true);
}

```

Test	# Passed	# Skipped	# Retried	# Failed	Time (ms)	Included Gro
Default suite						
Default test	3	0	0	0	1,137	
Class Method Start Time (ms)						
Default suite						
Default test — passed						
TestNG_Keywords.E_dependsOnMethods2	login1		1637260442498	19		
	login2		1637260442526	1017		
	logout		1637260443549	2		

Default test

TestNG_Keywords.E_dependsOnMethods2#login1

Messages
running login1 TC...

```

Console TestNG Report
<terminated> E_dependsOnMethods2 [TestNG] C:\Program Files\Java\jdk-16.0.2\bin\javav
[RemoteTestNG] detected TestNG version 7.4.0
running login1 TC...
running login2 TC...
running logout TC...
PASSED: login2
PASSED: login1
PASSED: logout

=====
Default test
Tests run: 3, Failures: 0, Skips: 0
=====

=====
Default suite
Total tests run: 3, Passes: 3, Failures: 0, Skips: 0
=====


```

Test-Suite:

- It is xml file which contains all the test classes' name which need to be executed.
- It is use to execute all/multiple Test classes.

```

package TestNG_TestSuit;
import org.testng.Reporter;
import org.testng.annotations.Test;
public class A_SuitClass1 {
    @Test
    public void TC1()
    {
        Reporter.Log("running TC1...",true);
    }
    @Test(priority =1)
    public void TC2()
    {
        Reporter.Log("running TC2...",true);
    }-----
}

package TestNG_TestSuit;
import org.testng.Reporter;
import org.testng.annotations.Test;
public class A_SuitClass2 {
    @Test(priority =2)
    public void TC3()
    {
        Reporter.Log("running TC3...",true);
    }

    @Test(priority =3)
    public void TC4()
    {
        Reporter.Log("running TC4...",true);
    }-----
}

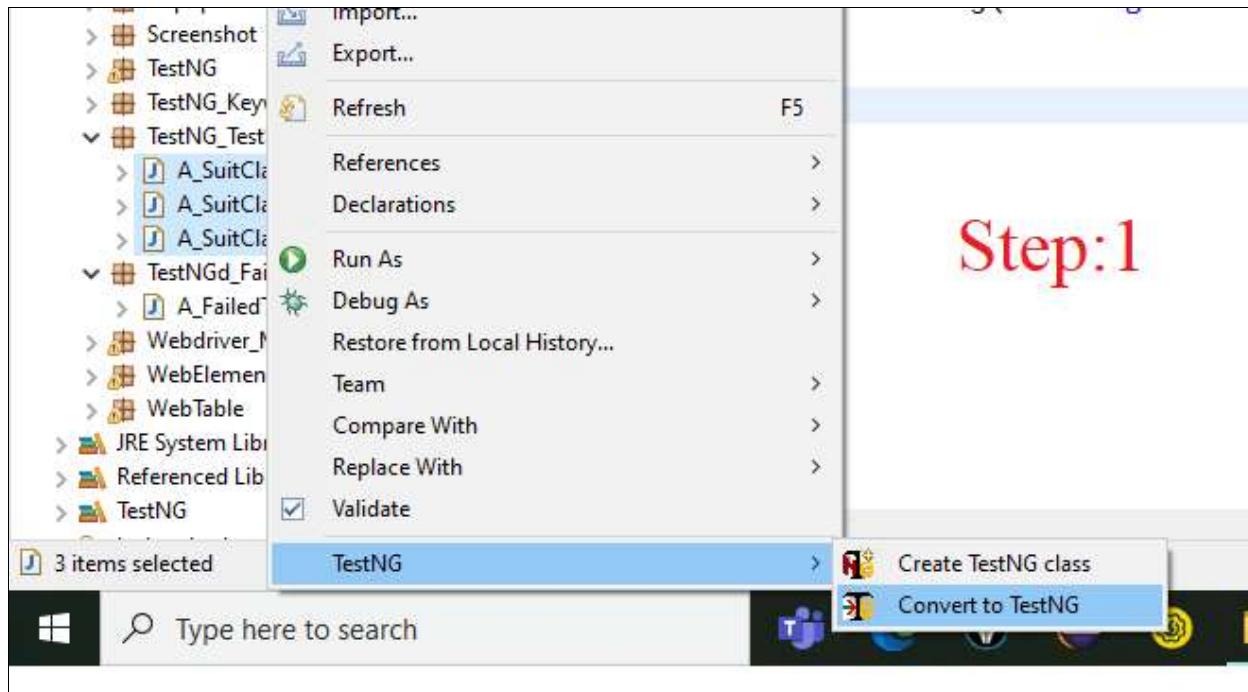
package TestNG_TestSuit;
import org.testng.Reporter;
import org.testng.annotations.Test;
public class A_SuitClass3 {
    @Test (priority =4)
    public void TC5()
    {
        Reporter.Log("running TC5...",true);
    }
    @Test (priority =5)
    public void TC6()
    {
        Reporter.Log("running TC6...",true);
    }-----
}

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite">
    <test thread-count="5" name="Test">
        <classes>
            <class name="TestNG_TestSuit.A_SuitClass2"/>
            <class name="TestNG_TestSuit.A_SuitClass3"/>
            <class name="TestNG_TestSuit.A_SuitClass1"/>
        </classes>
    </test> <!-- Test -->
</suite> <!-- Suite -->

```

Steps to Create Test Suit

1. Click>Select the Classes (Multiple).
2. Right Click on selected classes Click on TestNG then Click Convert to TestNG
3. New Popup will open Rename the XML file and Click on Finish.
4. Refresh the Project (Click F5)
5. At the Bottom of Project List there is test-output folder.



Step:2

Step:3

Failed.xml

- While executing the automation scripts, test cases may fail for several reasons.
- To optimize our next runs, we need to re-run only failed test cases

Reasons for fail TC

1. Environment issue
2. Script error
3. Bug

Steps to execute failed.xml file

1. Create testng.xml file under project folder.
2. Execute testng.xml file
3. Refresh the project
4. In the test-output folder >> testng-failed.xml file will be created.
5. execute "testng-failed.xml"

In this way we can execute fail test cases in TestNG class.

Java - Eclipse IDE

Results of running class A_FailedTestCases

Passed: 5 Failed: 1 Skipped: 0

All Tests | Failed Tests | Summary

Default suite (5/1/0/0) (0.046 s)

Default test (0.046 s)

TestNGd_FailedTestCases.A_FailedTestCases

- TC1 (0.031 s)
- TC2 (0.003 s)
- TC3 (0.003 s)
- TC4 (0.005 s)
- TC5 (0.002 s)
- TC6 (0.002 s)

Failure Exception

Default test
Tests run: 6, Failures: 1, Skips: 0

Default suite
Total tests run: 6, Passes: 5, Failures: 1, Skips: 0

Console

```
<terminated> A_FailedTestCases [TestNG] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (Nov 21, 2021, 10:08:26 AM)
at org.testng.TestRunner.run(TestRunner.java:596)
at org.testng.SuiteRunner.runTest(SuiteRunner.java:377)
at org.testng.SuiteRunner.runSequentially(SuiteRunner.java:326)
at org.testng.SuiteRunner.privateRun(SuiteRunner.java:324)
at org.testng.SuiteRunner.run(SuiteRunner.java:276)
at org.testng.SuiteRunnerWorker.runSuite(SuiteRunnerWorker.java:103)
at org.testng.SuiteRunnerWorker.run(SuiteRunnerWorker.java:95)
at org.testng.TestNG.runSuitesSequentially(TestNG.java:125)
at org.testng.TestNG.runSuitesLocally(TestNG.java:1134)
at org.testng.TestNG.runSuites(TestNG.java:1063)
at org.testng.TestNG.run(TestNG.java:1031)
at org.testng.remote.AbstractRemoteTestNG.run(AbstractRemoteTestNG.java:115)
at org.testng.remote.RemoteTestNG.initAndRun(RemoteTestNG.java:49)
at org.testng.remote.RemoteTestNG.main(RemoteTestNG.java:45)

=====
Default test
Tests run: 6, Failures: 1, Skips: 0
=====

=====
Default suite
Total tests run: 6, Passes: 5, Failures: 1, Skips: 0
=====
```

```
package TestNGd_FailedTestCases;
import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.Test;

public class A_FailedTestCases {
    @Test
    public void TC1()
    {
        Reporter.Log("Running TC1",true);
    }
    @Test
    public void TC2()
    {
        Reporter.Log("Running TC2",true);
    }
    @Test
    public void TC3()
    {
        Reporter.Log("Running TC3",true);
    }
    @Test
    public void TC4()
    {
        Assert.fail();
        Reporter.Log("Running TC4",true);
    }
    @Test
    public void TC5()
    {
        Reporter.Log("Running TC5",true);
    }
    @Test
    public void TC6()
    {
        Reporter.Log("Running TC6",true);
    }
}-----
```



● Verification Using TestNG

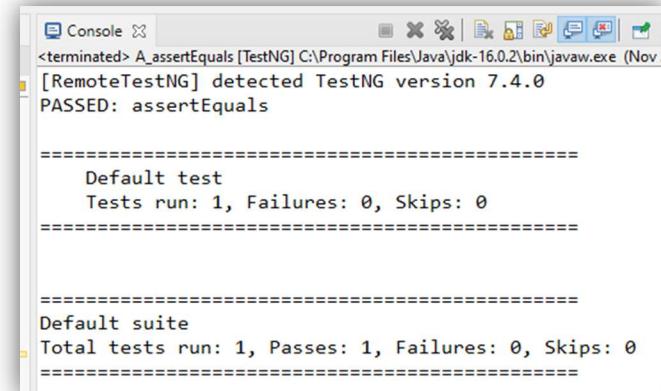
- Asserts helps us to verify the conditions of the test and decide whether test has failed or passed.
- `assertEquals()` : If both results are equal then Pass.
 - `assertNotEquals()` : If both results are equal then Fail.
 - `assertTrue()` : TC Pass if result is true.
 - `assertFalse()` : TC Pass if result is false.
 - `AssertNull()` : TC Pass if result is null.
 - `assertNotNull()`

```
package TestNGe_Verification;

import org.testng.Assert;
import org.testng.annotations.Test;

public class A_assertEquals {
    @Test
    public void assertEquals()
    {
        String actResult = "Velocity";
        String expResult = "Velocity";
        String FailedResult2 ="City";
        Assert.assertEquals(actResult, expResult, "Failed: Both Result are Different");
        //Note : "Failed: Both Result are Different" this message will display only if test case Fail.
    }
}
```

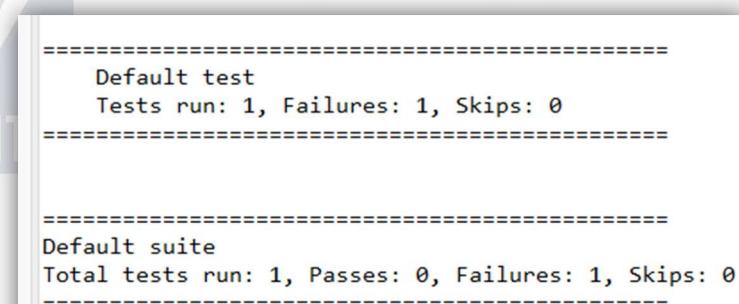
```
package TestNGe_Verification;
import org.testng.Assert;
import org.testng.annotations.Test;
public class A_assertEquals {
    @Test
    public void assertEquals()
    {
        String actResult = "Velocity";
        String expResult = "Velocity";
        String FailedResult2 ="City";
        Assert.assertEquals(actResult, FailedResult2, "Failed: Both Result are Different");
    }
}
```



```
<terminated> A_assertEquals [TestNG] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (Nov 21, 2021, 10:34:40 AM - 10:34:43 AM)
[RemoteTestNG] detected TestNG version 7.4.0
PASSED: assertEquals

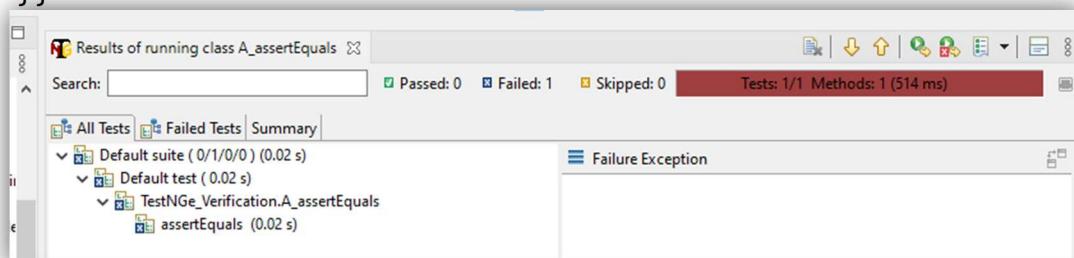
=====
Default test
Tests run: 1, Failures: 0, Skips: 0
=====

=====
Default suite
Total tests run: 1, Passes: 1, Failures: 0, Skips: 0
=====
```



```
=====
Default test
Tests run: 1, Failures: 1, Skips: 0
=====

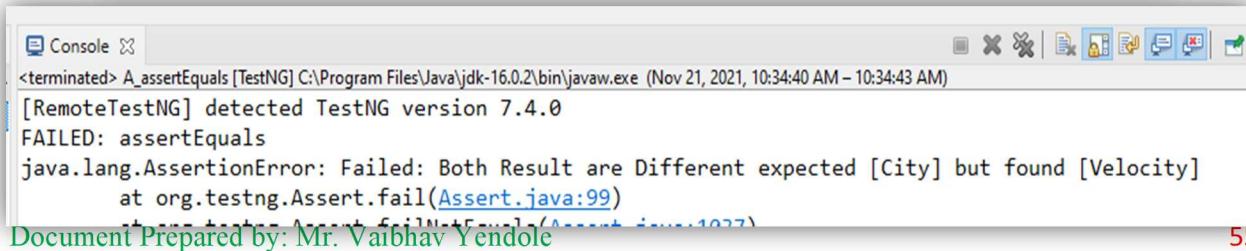
=====
Default suite
Total tests run: 1, Passes: 0, Failures: 1, Skips: 0
=====
```



Results of running class A_assertEquals

Passed: 0 Failed: 1 Skipped: 0 Tests: 1/1 Methods: 1 (514 ms)

All Tests	Failed Tests	Summary
Default suite (0/1/0/0) (0.02 s)	Default test (0.02 s)	Failure Exception
TestNGe_Verification.A_assertEquals	assertEquals (0.02 s)	



```
<terminated> A_assertEquals [TestNG] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (Nov 21, 2021, 10:34:40 AM - 10:34:43 AM)
[RemoteTestNG] detected TestNG version 7.4.0
FAILED: assertEquals
java.lang.AssertionError: Failed: Both Result are Different expected [City] but found [Velocity]
    at org.testng.Assert.fail(Assert.java:99)
    at org.testng.Assert.assertEquals(Assert.java:141)
    at TestNGe_Verification.A_assertEquals(A_assertEquals.java:22)
```

```

package TestNGe_Verification;

import org.testng.Assert;
import org.testng.annotations.Test;

public class B_AssertNotEquals {
    @Test
    public void AssertNotEquals()
    {
        String actResult = "Vaibhavi";
        String expResult = "Vaibhav";
        Assert.assertEquals(actResult, expResult, "Failed:
Results are Equal");
    }
    //If Both Result are Equal : Test Case Will Fail;
    // If Both Results are Not Equal : Test Case Will Pass;
}-----
```



```

package TestNGe_Verification;

import org.testng.Assert;
import org.testng.annotations.Test;

public class C_ASSERTTrue {
    @Test
    public void ASSERTTrue()
    {
        boolean actResult = true;
        Assert.assertTrue(actResult);
    }
    //Test Case will Pass if result = true
    // Test Case Will Fail if result = false
}-----
```



```

package TestNGe_Verification;

import org.testng.Assert;
import org.testng.annotations.Test;

public class D_ASSERTFalse {
    @Test
    public void ASserFalse()
    {
        boolean actResult = false;
        Assert.assertFalse(actResult);
    }
    //Test Case will Pass if result = false
    // Test Case Will Fail if result = true
}-----
```

```

package TestNGe_Verification;

import org.testng.Assert;
import org.testng.annotations.Test;

public class F_AssertNull {
    @Test
    public void assrtNull()
    {
        String actNull = "Automation"; //Fail
        Assert.assertNull(actNull);
    }
    // If Result is Null then TC Pass
    //If Result contains value then TC Fail
}

```

```

-----
package TestNGe_Verification;

import org.testng.Assert;
import org.testng.annotations.Test;

public class G_AssertNotNull {
    @Test
    public void assertNotNull()
    {
        String name = "Manul QA";
        Assert.assertNotNull(name);
    }
    // If Result is Null then TC Fail
    // If Result contains value then TC Pass
}
-----
```

Disadvantage of Assert Classes

If test class consist of multiple `@Test` methods, and in One of the Test Method Multiple verification are preferred and While executing one of verification failed then rest of the verification will not be verified and testNG will execute next method by failing verification failed method.

To Overcome the above drawback SoftAssert() class is used.

SoftAssert() Classes

- It is a class which contains non static methods use to do verification.
- SoftAssert will do verification if any verification is failed, notifies and executes the rest of verification in @Test method

```
SoftAssert soft = new SoftAssert();
soft.assertAll();
```

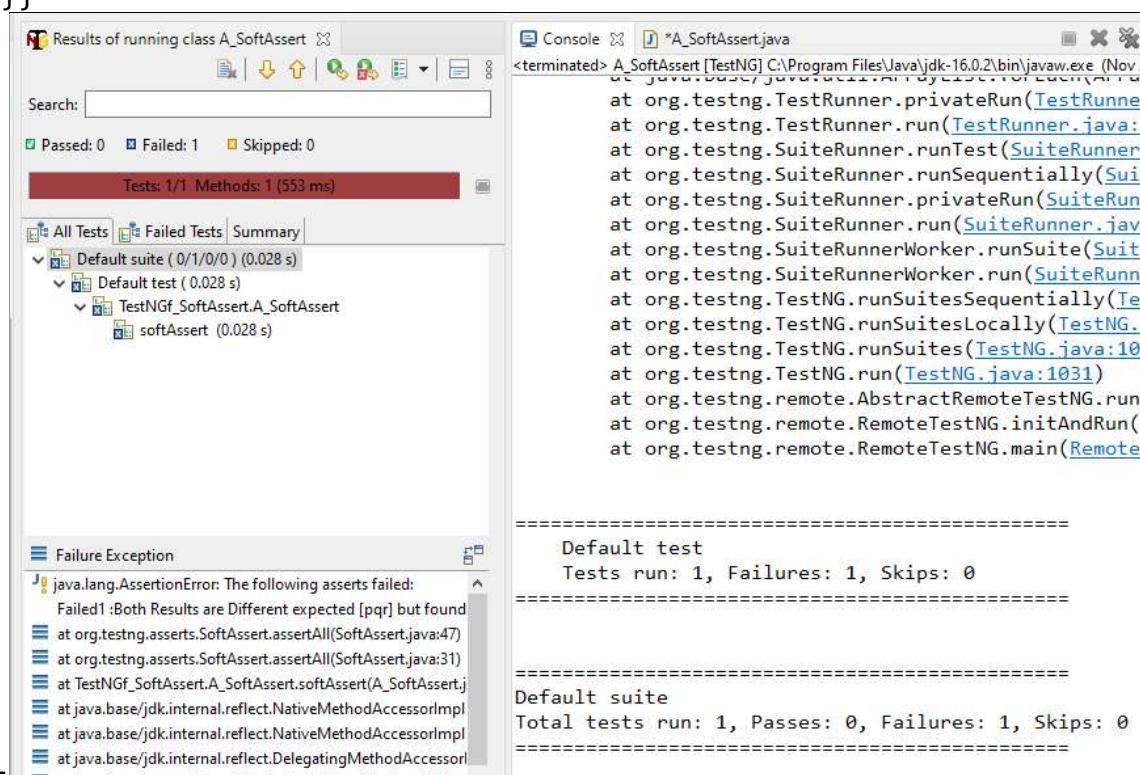
```
package TestNGf_SoftAssert;
import org.testng.annotations.Test;
import org.testng.asserts.SoftAssert;
public class A_SoftAssert {
    @Test
    public void softAssert()
    {
        SoftAssert soft = new SoftAssert();

        String s1 = "abc";
        String s2 = "pqr";
        soft.assertEquals(s1, s2, "Failed1 :Both Results are Different");

        String s3 ="Velocity";
        String s4 ="Pune";
        soft.assertNotEquals(s3, s4,"Failed2 : Both Results are Same");

        boolean s5 = true;
        soft.assertTrue(s5);

        soft.assertAll(); //This one is mandatory to mention at the end.
    }--
```



Disable Test Case Execution

Q. How to Disable Test Case Execution?

1. From Test Class Using “enable = false” keyword
 - 2.1 Using `Exclude` Keyword
 - 2.2 Using `include` keyword

- When you have 10 Methods and if you want only 2 methods to execute then follow approach of include keyword But
- When you have 10 Methods and if you want 8 methods to execute then follow approach of exclude keyword

```
package TestNGg_DisableTestCaseExecution;
import org.testng.Reporter;
import org.testng.annotations.Test;

public class A_DisableTCE_enableFalse{
    @Test
    public void TC1()
    {
        Reporter.Log("Running TC1", true);
    }

    @Test
    public void TC2()
    {
        Reporter.Log("Running TC2", true);
    }

    @Test(enabled=false) //Approach 1 : enabled = false
    public void TC3()
    {
        Reporter.Log("running TC3...",true);
    }
}
// In the above case only TC1 and TC2 executed, TC3 is not considered
```

The screenshot shows the Eclipse IDE interface with two panes. The left pane displays the test structure:

- Results of running class `A_DisableTCE_enableFalse`
- Search bar
- Statistics: Passed: 2, Failed: 0, Skipped: 0
- Summary: Tests: 1/1 Methods: 2 (1173 ms)
- Test tree:
 - All Tests
 - Failed Tests
 - Summary
 - Default suite (2/0/0/0) (0.024 s)
 - Default test (0.024 s)
 - TestNGg_DisableTestCaseExecution.A_DisableTCE_enableFalse
 - TC1 (0.02 s)
 - TC2 (0.004 s)
- Failure Exception

The right pane shows the console output:

```
<terminated> A_DisableTCE_enableFalse [TestNG] C:\Program Files\Java\jdk-16.0.2\bin\java
[RemoteTestNG] detected TestNG version 7.4.0
Running TC1
Running TC2
PASSED: TC1
PASSED: TC2
=====
Default test
Tests run: 2, Failures: 0, Skips: 0
=====

=====
Default suite
Total tests run: 2, Passes: 2, Failures: 0, Skips: 0
=====
```

```

package TestNGg_DisableTestCaseExecution;
import org.testng.Reporter;
import org.testng.annotations.Test;
public class B_DisableTCE_with_include_exclude
{
    @Test()
    public void TC1()
    {
        Reporter.Log("running TC1...",true);
    }

    @Test
    public void TC2()
    {
        Reporter.Log("running TC2...",true);
    }

    @Test
    public void TC3()
    {
        Reporter.Log("running TC3...",true);
    }

    @Test
    public void TC4()
    {
        Reporter.Log("running TC4...",true);
    }
}

```



1. Using `include` Keyword

- A) Right click on class, click on TestNG, Select ConvertToTestNG

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite">
    <test thread-count="5" name="Test">
        <classes>

            <class name="TestNGg_DisableTestCaseExecution.B_DisableTCE_with_include_exclude"/>

                <methods>
                    <include name="TC1"></include>
                    <include name="TC2"></include>

                </methods>
            </classes>
        </test> <!-- Test -->
    </suite> <!-- Suite -->

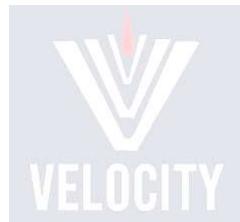
```

// Only TC1 and TC2 Will Run

2. Using `exclude` keyword

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite">
  <test thread-count="5" name="Test">
    <classes>
      <class
name="TestNGg_DisableTestCaseExecution.B_DisableTCE_with_include_exclude"/>
        <methods>
          <exclude name="TC1"></exclude>
          <exclude name="TC2"></exclude>
        </methods>
    </classes>
  </test> <!-- Test -->
</suite> <!-- Suite -->
```

//Only TC3 and TC4 will Run



• Grouping

- TestNG groups allow you to perform grouping of different tests methods.
-

```

package TestNGh_Grouping;

import javax.swing.GroupLayout.Group;

import org.testng.Reporter;
import org.testng.annotations.Test;

public class A_Grouping_test_groups_run_include_exclude_
{
    @Test(groups = "login")
    public void TC1()
    {
        Reporter.log("Running TC1", true);
    }

    @Test (groups = "payment")
    public void TC2()
    {
        Reporter.log("Running TC2", true);
    }

    @Test (groups = "login")
    public void TC3()
    {
        Reporter.log("Running TC3", true);
    }

    @Test
    public void TC4()
    {
        Reporter.log("Running TC4", true);
    }

    @Test (groups = "select plan")
    public void TC5()
    {
        Reporter.log("Running TC5", true);
    }

    @Test(groups = "payment")
    public void TC6()
    {
        Reporter.log("Running TC6", true);
    }
}

```

The screenshot shows the Eclipse IDE interface with three panes:

- Left pane (grouping.xml):** Displays the XML configuration for the suite. It includes sections for <groups>, <run>, <include name="Login">, and <classes>.
- Middle pane (Console):** Shows the TestNG console output. It starts with "Selenium_grouping1.xml [TestNG] detected TestNG version 7.4.0", followed by "Running TC1" and "Running TC3". The output ends with "Total tests run: 2, Passes: 2, Failures: 0, Skips: 0".
- Right pane (Results of running suite):** Shows the results of the suite execution. It indicates 2 passed tests, 0 failed, and 0 skipped. The summary shows 1 test and 2 methods (TC1 and TC3) with their execution times.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite">
    <test thread-count="5" name="Test">
        <groups>
            <run>
                <include name = "Login">
                </include>
            </run>
        </groups>
        <classes>
            <class name="TestNGh_Grouping.A_Grouping_test_groups_run_include_exclude_" />
        </classes>
    </test> <!-- Test -->
</suite> <!-- Suite -->
```

- <exclude name = "Login"> (Use for run other group and exclude this one)

The screenshot shows the Eclipse IDE interface with three panes:

- Left pane (grouping1.xml):** Displays the XML configuration for the suite. It includes sections for <groups>, <run>, <exclude name="Login">, and <classes>.
- Middle pane (Console):** Shows the TestNG console output. It starts with "Selenium_grouping1.xml [TestNG] detected TestNG version 7.4.0", followed by "Running TC2", "Running TC4", "Running TC5", and "Running TC6". The output ends with "Total tests run: 4, Passes: 4, Failures: 0, Skips: 0".
- Right pane (Results of running suite):** Shows the results of the suite execution. It indicates 4 passed tests, 0 failed, and 0 skipped. The summary shows 1 test and 4 methods (TC2, TC4, TC5, TC6) with their execution times.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite">
    <test thread-count="5" name="Test">
        <groups>
            <run>
                <exclude name = "Login">
                </exclude>
            </run>
        </groups>
        <classes>
            <class name="TestNGh_Grouping.A_Grouping_test_groups_run_include_exclude_" />
        </classes>
    </test> <!-- Test -->
</suite> <!-- Suite -->
```

- Parallel

- Parallel testing or parallel execution is the process of running test cases parallel rather than one after another
 - Parallel testing is done to save time required for execution of TC one after another execution.
-

```
package TestNGi_Parallel;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;
public class A_Parallel1 {
    @Test
    public void OpenKite()
    {
        System.setProperty("webdriver.chrome.driver",
"H:\\Selenium\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://kite.zerodha.com/");
    }
}-----
```

```
package TestNGi_Parallel;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;
public class B_Parallel2
{
    @Test
    public void openGoogle()
    {
System.setProperty("webdriver.chrome.driver",
"H:\\Selenium\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.google.com/");
    }
}-----
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite">
    <test thread-count="5" name="Test1">
        <classes>
            <class name="TestNGi_Parallel.A_Parallel1"/>
        </classes>
    </test> <!-- Test -->
    -----
    <test thread-count="5" name="Test2">
        <classes>
            <class name="TestNGi_Parallel.B_Parallel2"/>
        </classes>
    </test> <!-- Test -->
</suite> <!-- Suite -->
```

Console X parallel.xml B_parallel2.java

```

<terminated> Selenium_parallel.xml [TestNG] C:\Program Files\Java\jdk-16.0.2
[RemoteTestNG] detected TestNG version 7.4.0^
Starting ChromeDriver 96.0.4664.45 (76e4c1bb
Only local connections are allowed.
Please see https://chromedriver.chromium.org
ChromeDriver was started successfully.
Nov 23, 2021 8:28:14 AM org.openqa.selenium.
INFO: Detected dialect: W3C
Starting ChromeDriver 96.0.4664.45 (76e4c1bb
Only local connections are allowed.
Please see https://chromedriver.chromium.org
ChromeDriver was started successfully.
Nov 23, 2021 8:28:21 AM org.openqa.selenium.
INFO: Detected dialect: W3C
=====
Suite
Total tests run: 2, Passes: 2, Failures: 0,
=====
```

A_Parallel1.java

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0-
3<suite name="Suite">
4<test thread-count="5" name="Test1">
5<classes>
6<class name="TestNGi_Parallel.A_Parallel1"/>
7</classes>
8</test> <!-- Test -->
9
10<test thread-count="5" name="Test2">
11<classes>
12<class name="TestNGi_Parallel.B_Parallel2"/>
13</classes>
14</test> <!-- Test -->
15
16
17</suite> <!-- Suite -->
```

Results of running suite

Search:

Passed: 2 Failed: 0 Skipped: 0

Tests: 2/2 Methods: 2 (14913 ms)

All Tests Failed Tests Summary

Suite (2/0/0) (14.425 s)
 Test1 (7.457 s)
 TestNGi_Parallel.A_Parallel1
 OpenKite (7.457 s)
 Test2 (6.968 s)
 TestNGi_Parallel.B_Parallel2
 openGoogle (6.968 s)

Kite - Zerodha's fast and elegant X +

← → C kite.zerodha.com

Chrome is being controlled by automated test software.

Google

← → C google.com

Chrome is being controlled by automated test software.

Gmail Images Sign in

Google

18:1:456 149M of 317M



Compatibility Testing

1. Compatibility Testing (Running One After Another)

```

package TestNGj_Compatibility_Testing;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.edge.EdgeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.testng.annotations.Parameters;
import org.testng.annotations.Test;
import com.graphbuilder.curve.Point;

public class A_Multibrowser_Testing {
    @Parameters("browserName")
    @Test
    public void TC(String browserName)
    {
        WebDriver driver = null;
        if(browserName.equals("Chrome"))
        {
            System.setProperty("webdriver.chrome.driver","H:\\Selenium\\chromedriver.exe");
            driver = new ChromeDriver();
        }
        org.openqa.selenium.Point p = new org.openqa.selenium.Point(500, 700);
        driver.manage().window().setPosition(p);
    }
    else if(browserName.equals("Firefox"))
    {
        System.setProperty("webdriver.gecko.driver","H:\\Selenium\\geckodriver.exe");
        driver = new FirefoxDriver();
    }
    else if(browserName.equals("Edge"))
    {
        System.setProperty("webdriver.edge.driver","H:\\Selenium\\msedgedriver.exe");
        driver = new EdgeDriver();
    }
    driver.get("https://www.facebook.com/");
}
}

```

Results of running suite

Search:

Passed: 3 Failed: 0 Skipped: 0

Tests: 3/3 Methods: 3 (37626 ms)

All Tests Failed Tests Summary

- Suite (3/0/0/0) (84.351 s)
 - FirefoxTest (22.59 s)
 - TestNGj_Compatibility_Testing.A_Multibrowser_Testing
 - TC (22.59 s)
 - "Firefox" (22.59 s)
 - EdgeTest (24.54 s)
 - TestNGj_Compatibility_Testing.A_Multibrowser_Testing
 - TC (24.54 s)
 - "Edge" (24.54 s)
 - ChromeTest (37.221 s)
 - TestNGj_Compatibility_Testing.A_Multibrowser_Testing
 - TC (37.221 s)
 - "Chrome" (37.221 s)

Console

*A_Compatibility_Testing.xml

<terminated> Selenium_A_Compatibility_Testing.xml [TestNG] C:\Program Files\Java\jdk-16.0.2\bin\java

1637737965159 mozrunner::runner INFO Running command
Starting ChromeDriver 96.0.4664.45 (76e4c1bb2ab4671b8beba3444)
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations#Local_Connections
Starting MSEdgeDriver 94.0.992.31 (fd65e6bbbedc86a22f2393b7cd8)
Only local connections are allowed.
Nov 24, 2021 12:42:54 PM org.openqa.selenium.remote.ProtocolHandshake INFO: Detected dialect: W3C
[1637737977.895][WARNING]: This version of MSEdgeDriver has no certificate error handling logic.
Nov 24, 2021 12:42:58 PM org.openqa.selenium.remote.ProtocolHandshake INFO: Detected dialect: W3C
JavaScript warning: https://static.xx.fbcdn.net/rsrc.php/v3iX

=====
Suite
Total tests run: 3, Passes: 3, Failures: 0, Skips: 0

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite" parallel="tests">
    <test name="EdgeTest">
        <parameter name="browserName" value="Edge"></parameter>
        <classes>
            <class
name="TestNGj_Comaptibility_Testing.A_Multibrowser_Testing"/>
        </classes>
    </test> <!-- Test -->

    <test name="ChromeTest">
        <parameter name="browserName" value="Chrome"></parameter>
        <classes>
            <class
name="TestNGj_Comaptibility_Testing.A_Multibrowser_Testing"/>
        </classes>
    </test> <!-- Test -->

    <test name="FirefoxTest">
        <parameter name="browserName" value="Firefox"></parameter>
        <classes>
            <class
name="TestNGj_Comaptibility_Testing.A_Multibrowser_Testing"/>
        </classes>
    </test> <!-- Test -->

</suite> <!-- Suite -->

```

```

package TestNGj_Comaptibility_Testing;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.edge.EdgeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.testng.annotations.Parameters;
import org.testng.annotations.Test;
public class B_MultibrowserTesting_With_parallel {
    @Parameters("browserName")
    @Test
    public void TC(String browserName)
    {
        WebDriver driver = null;
        if(browserName.equals("Chrome"))
        {
System.setProperty("webdriver.chrome.driver","H:\\Selenium\\chromedriver.exe");
            driver = new ChromeDriver();
        }
        else if(browserName.equals("Firefox"))
        {
System.setProperty("webdriver.gecko.driver","H:\\Selenium\\geckodriver.exe");

```

```

        driver = new FirefoxDriver();
    }
    else if(browserName.equals("Edge"))
    {
System.setProperty("webdriver.edge.driver","H:\\Selenium\\msedgedriver.exe");
        driver = new EdgeDriver();
    }
driver.get("https://wazirx.com/");
driver.findElement(By.xpath("//span[text()='Log in']")).click();
driver.findElement(By.xpath("//input[@placeholder='Enter your
email']")).sendKeys("v123@gmail.com");
driver.findElement(By.xpath("//input[@placeholder='Enter your
password']")).sendKeys("Pass@23");
driver.findElement(By.xpath("//span[text()='Login']")).click();
}
}-----

```

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Compatibility Testing" parallel="tests">

<test name="ChromeTest">
<parameter name="browserName" value="Chrome"></parameter>
<classes>
    <class
name="TestNGj_Comaptibility_Testing.B_MultibrowserTesting_With_parallel"/>
</classes>
</test> <!-- Test -->

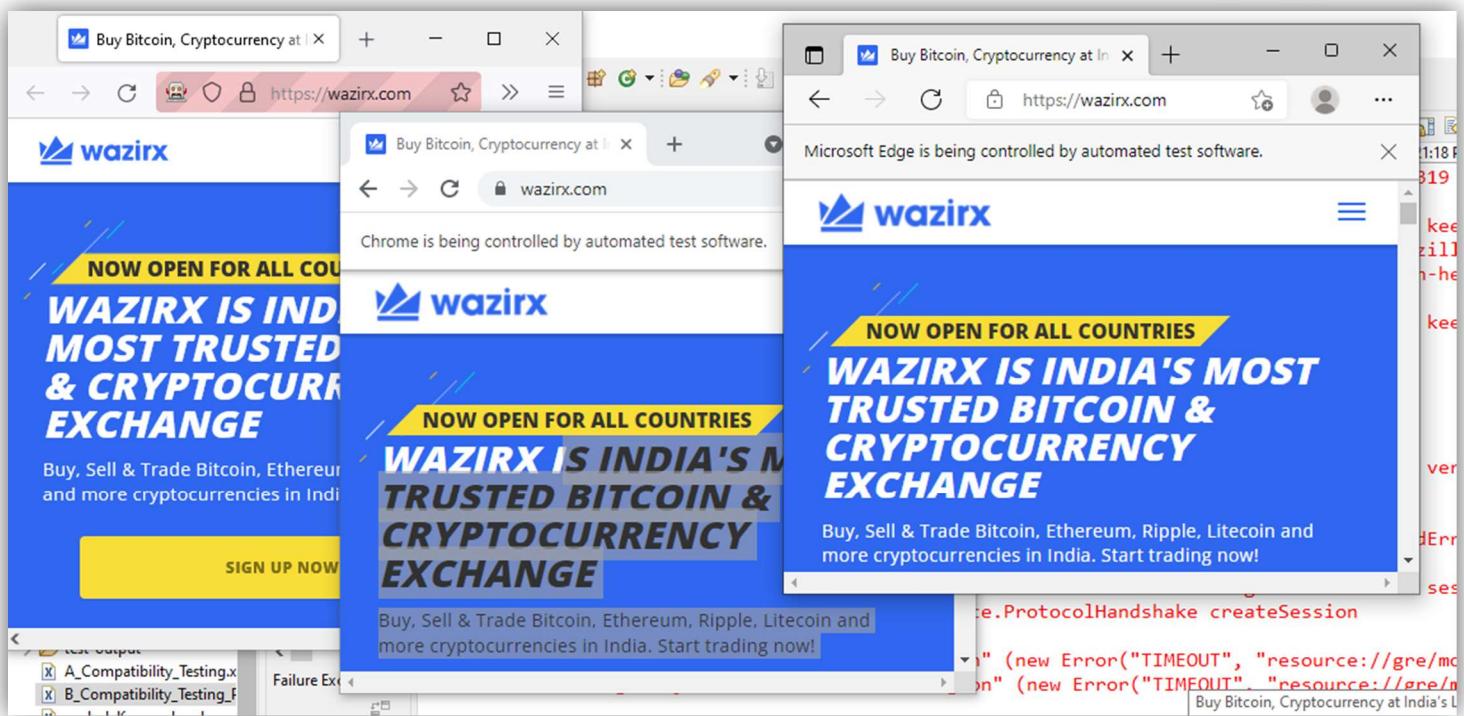
<test name="FirefoxTest">
<parameter name="browserName" value="Firefox"></parameter>
<classes>
    <class
name="TestNGj_Comaptibility_Testing.B_MultibrowserTesting_With_parallel"/>
</classes>
</test> <!-- Test -->

<test name="EdgeTest">
<parameter name="browserName" value="Edge"></parameter>
<classes>
    <class
name="TestNGj_Comaptibility_Testing.B_MultibrowserTesting_With_parallel"/>
</classes>
</test> <!-- Test -->

</suite> <!-- Suite -->

```





Comparison of Junit vs TestNG

	Junit	TestNG
Parallel execution	JUnit does not support to run parallel tests.	TestNG can run parallel tests.
Supports Annotation	It does not support advanced annotation.	It supports advanced annotation.
Dependency tests	The dependency tests are missing in JUnit.	Dependency tests are present in TestNG
Grouping tests	Grouping tests together is not possible in JUnit.	Tests can be grouped together and run parallel.

TestNG advantages

- It has different assertions that helps in checking the expected and actual results.
- It allows to assign priority to test methods
- It allows to define dependency of one test method over other method
- It provide Detailed (HTML) reports/ Emailable report
- It allows grouping of test methods into test class
- TestNG provides parallel execution of test methods
- TestNG provides multibrowser/CT testing of test methods

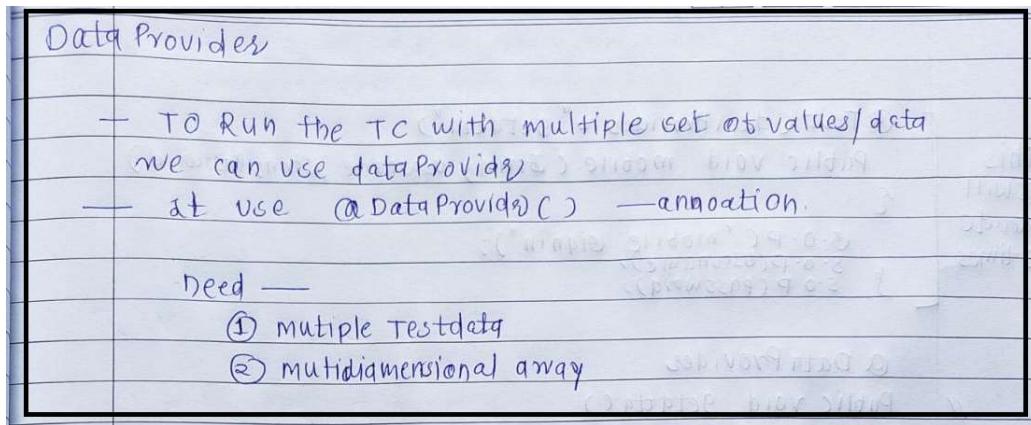
Data provider

Similar to TestNG Parameters, DataProviders are a means to pass data to test scripts in TestNG. Using DataProvider in TestNG, we can easily inject multiple values into the same test case. It comes inbuilt in TestNG and is popularly used in data-driven frameworks.

```

1 @DataProvider(name = "name of the data provider")
2 public Object[][] dataProviderfunc(){
3     Return new Object[][]{
4         values
5     }
6 }
```

- The DataProvider annotation has a single attribute called name, which you can select as per your convenience.
- DataProviders are separate methods used in test functions, which means that this annotation is not used on test functions like the testNG parameters.
- The DataProvider method returns a 2D list of objects.
- In case you do not define a name for the DataProvider, the DataProvider method name is considered its default name. So, the name of the DataProvider calls the DataProvider method.



```

Example9_dataprovider.java
1 package TestNG;
2 import org.testng.annotations.DataProvider;
3 import org.testng.annotations.Test;
4 public class Example9_dataprovider {
5
6     @Test(dataProvider="getdata")
7     public void login(String usnm, String passwd)
8     {
9         System.out.println("this is loginpage");
10        System.out.println(usnm);
11        System.out.println(usnm);
12    }
13    @DataProvider()
14    public Object[][] getdata()
15    {
16
17        Object[][] data=new Object[3][2];
18        data[0][0]="1stusername";
19        data[0][1]="1stpassword";
20        data[1][0]="2ndusername";
21        data[1][1]="2ndpassword";
22        data[2][0]="3rdusername";
23        data[1][1]="3rdpassword";
24        return data;
25    }
26 }
```

POM DDF with TestNG

Consider the same Problem discuss in POM (Page No.37)

```

package POM_With_DDF;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.sql.Driver;
import java.util.concurrent.TimeUnit;

import org.apache.poi.EncryptedDocumentException;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.WorkbookFactory;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.AfterClass;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;
}

public class A_KiteLoginTest {

    //declare Globally objects
    A_loginPage1 page1;
    A_loginPage2 page2;
    A_HomePage3 page3;
    Sheet sh;
    WebDriver driver;
}

@BeforeClass
public void openbrowser() throws EncryptedDocumentException, IOException
{
    //All the methods Run only one Time so declare in BeforeClass
    FileInputStream file = new
    FileInputStream("C:\\\\Users\\\\Jordan\\\\Desktop\\\\Velocity
    Testing1\\\\XcelSheet\\\\Kite Login.xlsx");

    sh = WorkbookFactory.create(file).getSheet("LoginSheet");

    System.setProperty("webdriver.chrome.driver","H:\\\\Selenium\\\\chromedriver.exe");

        driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://kite.zerodha.com/");
    driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);

    //POM Class Object Creation
    page1 = new A_loginPage1(driver);
    page2 =new A_loginPage2(driver);
    page3 =new A_HomePage3(driver);
}

```

```

@BeforeMethod           //login Method
public void loginToApp()
{
    //Enter ID Pass
page1.setEnterID(sh.getRow(1).getCell(0).getStringCellValue());

page1.setEnterPass(sh.getRow(1).getCell(1).getStringCellValue());

page1.clickLoginButton();

    //Enter Pin

page2.setEnterPin(sh.getRow(1).getCell(2).getStringCellValue());

page2.clickContinue();
}

@Test                  //Actual Test Case
public void verifyPN()
{
String actPN=page3.getKiteHomePageProfileName(); //get POM page 3
String expPN=sh.getRow(1).getCell(3).getStringCellValue(); // Get
from Excel

Assert.assertEquals(actPN, expPN, "Failed: Both Results are
Different");

//Don't Use below Method
//page3.VerifyKiteHomePageUserID(sh.getRow(1).getCell(3).getStringCell
Value());
}

@AfterMethod          //Logout method

public void logoutFromApp()
{
    Reporter.Log("Logout",true);
}

@AfterClass
public void closeBrowser()
{
    driver.close();

    page1 =null;
    page2 = null;
    page3 = null;
    sh=null;
    driver=null;
}
}

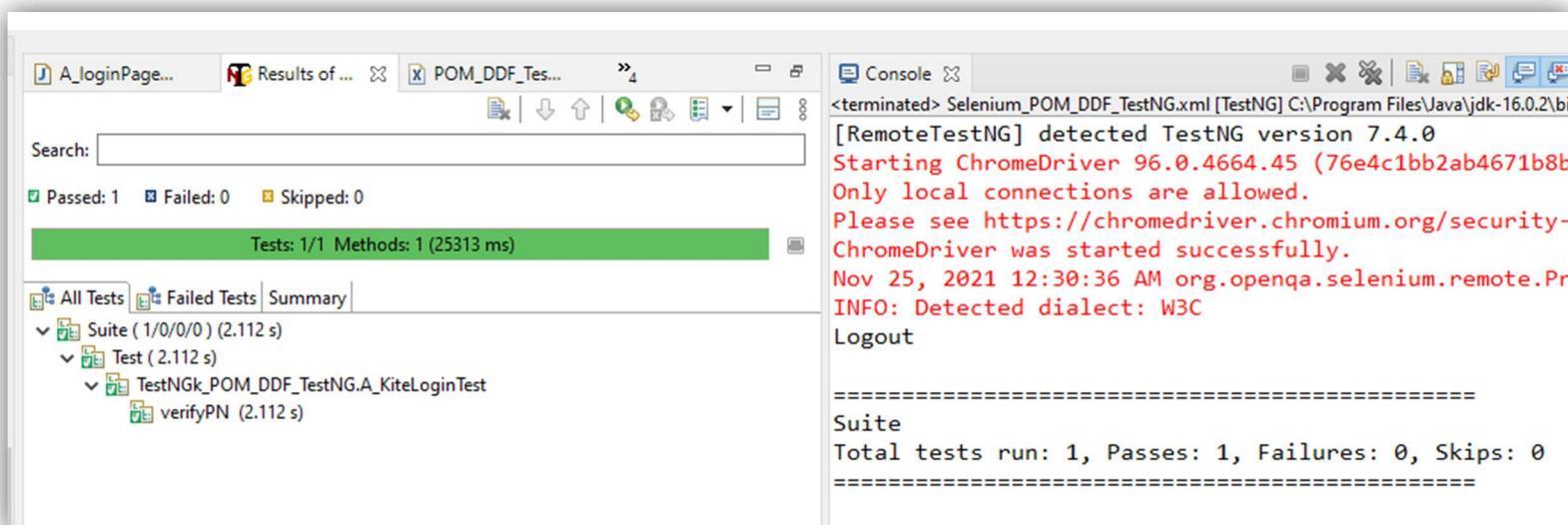
-----

```

Changes made in the A_HomePage3 (page3)

```
public String getKiteHomePageProfileName()
{
    String PN = UserID.getText();
    return PN;
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite">
    <test thread-count="5" name="Test">
        <classes>
            <class name="TestNGk_POM_DDF_TestNG.A_KiteLoginTest"/>
        </classes>
    </test> <!-- Test -->
</suite> <!-- Suite -->
```



Q. What is Use of Finalize Keyword? OR

Q. What is Need of Declare object name as null lastly as already while garbage collection it will call finalized Method?

- When you run class in suit at the time of each and every class execution, garbage collector does not call.
- Garbage collection will call after executing all classes (at the time of report publish).
- When number of classes are declare and executed at that time memory allocated to different objects and when garbage collections calls finalize there may be chances of data losses so object declare as null lastly. (For safe side)

POM_DDF_TestNG_BaseClass_Utility_Class

Consider A_loginPage1, A_loginPage2, A_HomePage3 Constant as mentioned in above previous example.

1. Base Class

```

package TestNG1_POM_DDF_TestNG_BaseClass_UtilityClass;

import java.util.concurrent.TimeUnit;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class A_BaseClass {
    // Base class consist of Mandatory code that run everytime like
open browser
    // Use extends keyword to inherited Base class
    //Declared Globally
    WebDriver driver;
    public void initializeBrowser()
    {
        System.setProperty("webdriver.chrome.driver",
"H:\\Java\\\\Selenium\\\\Browsers\\\\chromedriver.exe");
        driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://kite.zerodha.com/");
        driver.manage().timeouts().implicitlyWait(20,
TimeUnit.SECONDS);
    }
}

```

2. Utility Class

```

package TestNG1_POM_DDF_TestNG_BaseClass_UtilityClass;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import org.apache.poi.EncryptedDocumentException;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.WorkbookFactory;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.io.FileHandler;

public class A_Utility_Class {
    //In the Utility class consist of Selenium Common Functions
    //like Take Screenshot, Parameterization i.e used sometimes if needed

    //Static Method is used          // Use Return Type as String instead
    of void
}

```

```

//Author Name : Y. Vaibhav
//Description : This method is use to get test data from excel sheet
//Need to pass Two parameters Row Index the Col Index
public static String getTD(intRowIndex, intCellIndex) throws
EncryptedDocumentException, IOException
{
    FileInputStream file = new
FileInputStream("H:\\Java\\Selenium\\TestData_Excel\\Kite
Login.xlsx");

Sheet sh = WorkbookFactory.create(file).getSheet("LoginSheet");
String value =
sh.getRow(rowIndex).getCell(cellIndex).getStringCellValue();
    return value;
}

//Author Name : Y. Vaibhav
//This method is use to capture the Screenshot
//Need to pass two parameters 1.driver 2.test case ID
public static void captureScreenshot(WebDriver driver, int
testCaseID) throws IOException
{
    File src =
((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);

    File dest = new
File("H:\\Java\\Selenium\\Screenshot\\TCID"+testCaseID+".jpg");
    FileHandler.copy(src, dest);
}
}-----

```

3. Test Class

```

package TestNG1_POM_DDF_TestNG_BaseClass_UtilityClass;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.sql.Driver;
import java.util.concurrent.TimeUnit;
import org.apache.poi.EncryptedDocumentException;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.WorkbookFactory;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.AfterClass;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeClass;

```

```

import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;

public class A_KiteLoginTest extends A_BaseClass{
    //declare Globally objects
    A_loginPage1 page1;
    A_loginPage2 page2;
    A_HomePage3 page3;
    Sheet sh;

    @BeforeClass
    public void openbrowser() throws EncryptedDocumentException,
    IOException
    {
    //This class is Not static and initializeBrowser() is also an Non
    Static Method
        //call methodeName(); to open browser
        initializeBrowser();

        //POM Class Object Creation
        page1 = new A_loginPage1(driver);
        page2 =new A_loginPage2(driver);
        page3 =new A_HomePage3(driver);
    }

    @BeforeMethod           //login Method
    public void loginToApp() throws EncryptedDocumentException,
    IOException
    {
        //Enter ID Pass
        page1.setEnterID(A.Utility_Class.getTD(2, 0));
        page1.setEnterPass(A.Utility_Class.getTD(2, 1));
        page1.clickLoginButton();

        //Enter Pin
        page2.setEnterPin(A.Utility_Class.getTD(2, 2));
        page2.clickContinue();
    }

    @Test                  //Actual Test Case
    public void verifyPN() throws EncryptedDocumentException,
    IOException
    {
        String actPN=page3.getKiteHomePageProfileName();
        //get POM page 3
        String expPN=A.Utility_Class.getTD(2, 3);          // Get
        from Excel
        Assert.assertEquals(actPN, expPN, "Failed: Both Results are
        Different");

        //Don't Use this Method

        //page3.VerifyKiteHomePageUserID(sh.getRow(1).getCell(3).getStrin
        gCellValue());
    }
}

```

```

@AfterMethod          //Logout method
public void logoutFromApp()
{
    Reporter.log("Logout",true);
}

@AfterClass
public void closeBrowser()
{
    driver.close();

    page1 =null;
    page2 = null;
    page3 = null;
    sh=null;
    driver=null;
}
}-----

```

Console X

<terminated> A_KiteLoginTest [TestNG] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (Nov 26)

[RemoteTestNG] detected TestNG version 7.4.0

Starting ChromeDriver 96.0.4664.45 (76e4c1bb2ab4671b8b...

Logout

PASSED: verifyPN

=====

Default test

Tests run: 1, Failures: 0, Skips: 0

=====

=====

Default suite

Total tests run: 1, Passes: 1, Failures: 0, Skips: 0

=====

Results of running class A_KiteLoginTest X

Search: Passed: 1 Failed: 0 Skipped: 0 1/1 Methods: 1 (18)

All Tests	Failed Tests	Summary
Default suite (1/0/0/0) (4.518 s)		Failure Exception
Default test (4.518 s)		
TestNGI_POM_DDF_TestNG_BaseCl		
verifyPN (4.518 s)		

Capture Screenshot of only Failed TC

- In the @AfterMethod use `listner(interface)` and get the current status of executed TC
- If the status of TC is failed then we use code for capture screenshot.
- In the `ItestResult` using `getStatus()` pass or fail status can obtain.

To get test case status we need to use `listener(interface)` in TestNG.

Property file selenium:

- To store imp login credentials of application eg. UN, PWD, PIN, URL

Step 1) Creating a properties file in eclipse

1. Right-click on the main project folder and Select New-> Other->select General -> File and click on 'Next' button->
2. Provide a valid file name with the extension '`.properties`' on the new file resource window and click on 'Finish' button

Step 2) Storing data into properties file

1. Data is stored in properties file in the form of key-value pairs, with the key being unique across the file.
2. Open file in Eclipse and store some data eg- URL= <https://kite.zerodha.com/>

Step 3) Reading data from properties file

```

FileInputStream file = new FileInputStream ("H:\\Java\\Selenium
\\PropertyFile\\PropertyFile.properties");

Properties p = new Properties();
p.load(file);

String value = p.getProperty(key);

Properties obj = new Properties();
obj.load(objfile);

String value= obj.getProperty("URL");

return value;

```

Extent Reporter

- ExtentReports is an open-source reporting library used in selenium test automation.
- Extent reports become the first choice of Selenium Automation Testers, even though Selenium comes with inbuilt reports using frameworks like JUnit and TestNG

Capture Screenshot Using Property file

```

package TestNGm_Screenshot_Failed_TC;
import java.io.IOException;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class A_BaseClass {
// Base class consist of Mandatory code that run every time like open browser
    // Use extends keyword to inherited Base class
    //Declared Globally
    WebDriver driver;
    public void initializeBrowser() throws IOException
    {
        System.setProperty("webdriver.chrome.driver", "H:\\Java\\Selenium\\Browsers\\chromedriver.exe");
        driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get(A_Utility_Class.getDataFromPropertyFile("URL"));
        driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);
    }
}
-----
```

```

package TestNGm_Screenshot_Failed_TC;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.Properties;
import org.apache.poi.EncryptedDocumentException;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.WorkbookFactory;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.io.FileHandler;

public class A_Utility_Class {
//In the Utility class consist of Selenium Common Functions
//like Take Screenshot, Parameterization i.e used sometimes if needed

/Static Method is used// Use Return Type as String instead of void

//Author Name : Y. Vaibhav
//Description : This method is use to get test data from excel sheet
//Need to pass Two parameters Row Index the Col Index
public static String getTD(int RowIndex, int CellIndex) throws
EncryptedDocumentException, IOException
{
    FileInputStream file = new
FileInputStream("H:\\Java\\Selenium\\TestData_Excel\\Kite
Login.xlsx");
```

```

Sheet sh = WorkbookFactory.create(file).getSheet("LoginSheet");
String value =
sh.getRow(rowIndex).getCell(cellIndex).getStringCellValue();
    return value;
}

//Author Name : Y. Vaibhav
//This method is use to capture the Screenshot
//Need to pass two parameters 1.driver 2.test case ID
    public static void captureScreenshot(WebDriver driver, int
TestCaseID) throws IOException
{
    File src =
((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);

    File dest = new
File("H:\\Java\\Selenium\\Screenshot\\TCID"+TestCaseID+".jpg");

    FileHandler.copy(src, dest);
}

```

```

//Author Name : Y. Vaibhav
//This method is use to access imp credentials stores in property file
    public static String getDataFromPropertyFile(String key) throws
IOException
{
    FileInputStream file = new
FileInputStream("H:\\Java\\Selenium\\PropertyFile\\PropertyFile.proper
ties");

    Properties p = new Properties();
    p.load(file);

    String value = p.getProperty(key);
    return value;
}
-----
```

```

package TestNGm_Screenshot_Failed_TC;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.sql.Driver;
import java.util.concurrent.TimeUnit;
import org.apache.poi.EncryptedDocumentException;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.WorkbookFactory;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.Assert;
import org.testng.ITestResult;
import org.testng.Reporter;
import org.testng.annotations.AfterClass;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;
```

```

public class A_KiteLoginTest extends A_BaseClass{
    //declare Globally objects
    A_loginPage1 page1;
    A_loginPage2 page2;
    A_HomePage3 page3;
    Sheet sh;
    int TestCaseID; //Provided in Excel Sheet in manual TC

    @BeforeClass
    public void openbrowser() throws EncryptedDocumentException,
    IOException
    {
        //This class is Not static and initializeBrowser() is also an Non
        Static Method
        //call methodeName(); to open browser
        initializeBrowser();

        //POM Class Object Creation
        page1 = new A_loginPage1(driver);
        page2 =new A_loginPage2(driver);
        page3 =new A_HomePage3(driver);
    }

    @BeforeMethod           //login Method
    public void loginToApp() throws EncryptedDocumentException,
    IOException
    {
        //Enter ID Pass
        page1.setEnterID(A.Utility_Class.getDataFromPropertyFile("UN"));
        page1.setEnterPass(A.Utility_Class.getDataFromPropertyFile("PWD"));
        page1.clickLoginButton();

        //Enter Pin
        page2.setEnterPin(A.Utility_Class.getDataFromPropertyFile("PIN"));
        page2.clickContinue();
    }

    @Test                  //Actual Test Case
    public void verifyPN() throws EncryptedDocumentException,
    IOException
    {
        TestCaseID =200;
        String actPN=page3.getKiteHomePageProfileName();
        //get POM page 3
        String expPN=A.Utility_Class.getTD(2, 3);          // Get
        from Excel
        Assert.assertEquals(actPN, expPN, "Failed: Both Results are
        Different");

        //Don't Use this Method
        //page3.VerifyKiteHomePageUserID(sh.getRow(1).getCell(3).getStringCell
        Value());
    }
}

```

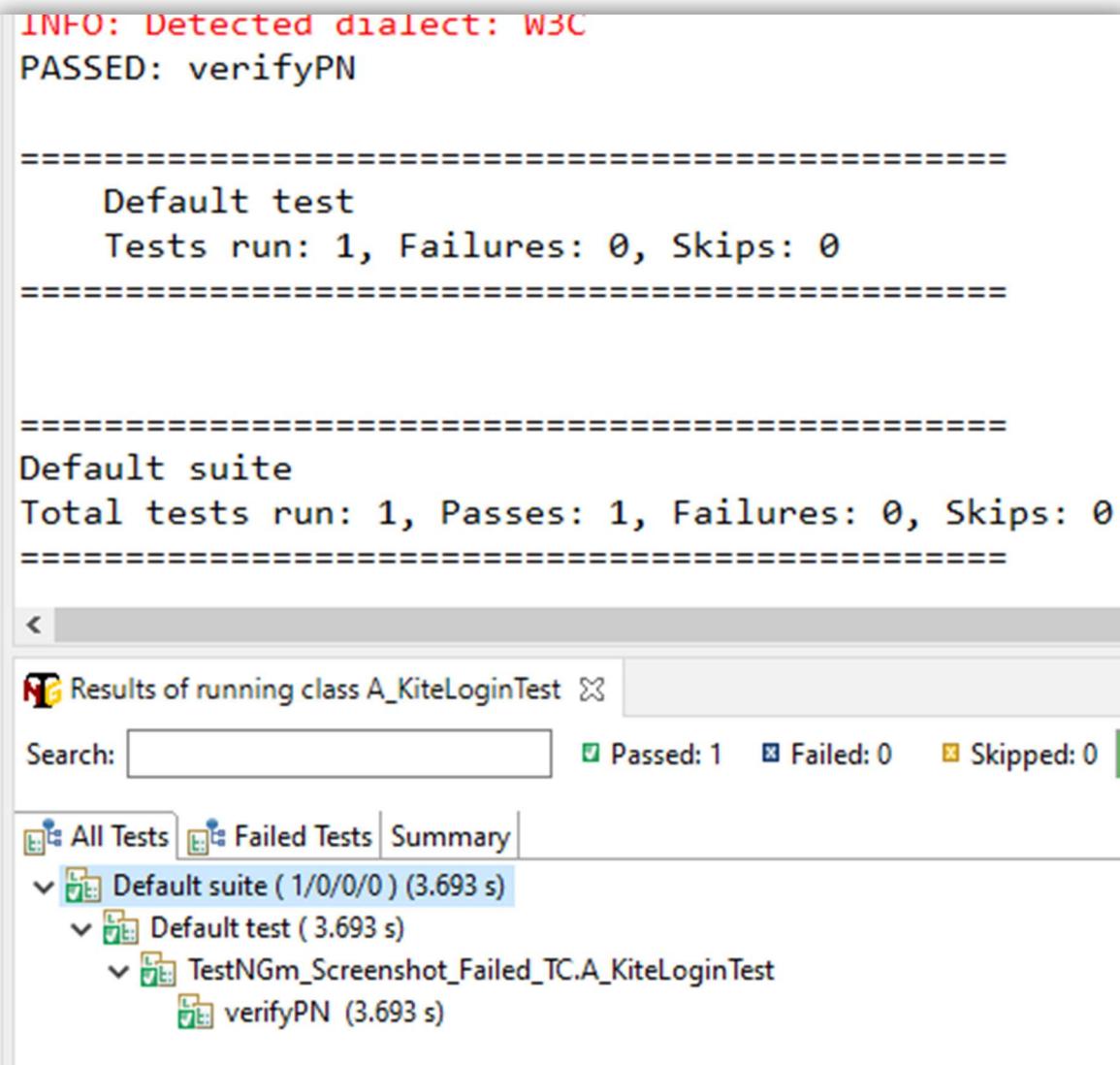
```

@AfterMethod           //ITestResult - interface
public void logoutFromApp(ITestResult result) throws IOException
{
    if(result.getStatus()==ITestResult.FAILURE)
    {
        A_Utility_Class.captureScreenshot(driver, TestCaseID);
    }
}

@AfterClass
public void closeBrowser()
{
    driver.close();

    page1 =null;
    page2 = null;
    page3 = null;
    sh=null;
    driver=null;
}
}-----

```



Maven Project

Steps

1. Create New Project (Alter+Shift+N) → Click on Project → Click on Maven → Expand and Click on Maven Project → Click on Next → Click on Next → Enter Group Id(Package Name) and Artifact ID (Project Name) → Finish
2. IF we wan to add jar file, then Click on pom.xml file where we can add dependencies to add jar file. (4 line code – dependency can obtain from <https://mvnrepository.com/>)
3. Open pom.xml then Insert Selenium Jar (Selenium Java),

```
<!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
<dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>4.0.0</version>
</dependency>
```

Apache POI File dependency

```
<!-- https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml -->
<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi-ooxml</artifactId>
    <version>5.0.0</version>
</dependency>
```



The screenshot shows the Eclipse IDE interface. On the left, there's a tree view of the project structure under 'Kite_Maven'. It includes 'src/main/java', 'src/test/java', 'JRE System Library [JavaSE-1.7]', 'Maven Dependencies', 'src', 'target', and 'pom.xml'. The main workspace shows the XML code for the pom.xml file. The code lists two dependencies: one for Selenium Java version 4.0.0 and one for Apache POI poi-ooxml version 5.0.0. Below the code, there's a 'Velocity' watermark.

```

<dependencies>
    <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
    <dependency>
        <groupId>org.seleniumhq.selenium</groupId>
        <artifactId>selenium-java</artifactId>
        <version>4.0.0</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml -->
    <dependency>
        <groupId>org.apache.poi</groupId>
        <artifactId>poi-ooxml</artifactId>
        <version>5.0.0</version>
    </dependency>

    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>4.11</version>
        <scope>test</scope>
    </dependency>
</dependencies>

```

4. In the SRC/Main/Java: Store all POM Classes (Every different module create new Package like saperate module for login, wishlist etc)
5. In the SRC/Test/Java: Store all POM Classes
6. Create one Folder **Library file** in which Base Class and Utility Class is Store
7. Create folder for Screenshot, TestData , Browsers.
8. Create a Property File and store in a folder.
9. Update/set the path for Excel File, Screenshot, Property File.

Q. Which OOPs concept in Selenium Framework

1. **Interface:** WebDriver driver = new ChromeDriver();

In this statement WebDriver is nothing but interface in selenium.

2. **UPCASTING:** WebDriver driver = new ChromeDriver();

Above statement is nothing but UPCASTING in selenium.

3. INHERITANCE

- We create a Base Class in the Framework to initialize WebDriver interface, WebDriver waits, Property files etc., in the Base Class.
- We extend the Base Class in Tests Class. That is nothing but Inheritance in Selenium Framework.

4. POLYMORPHISM

Combination of overloading and overriding is known as Polymorphism.

5. METHOD OVERLOADING

- We use implicit wait in Selenium. Implicit wait is an example of overloading. In Implicit wait we use different time stamps such as SECONDS, MINUTES, HOURS etc.,
- A class having multiple methods with same name but different parameters is called Method Overloading.

eg. driver.switchTo().frame(): - String name, int index

6. METHOD OVERRIDING

- Declaring a method in child class which is already present in the parent class is called Method Overriding.
- Examples are get and navigate methods of different drivers in Selenium.

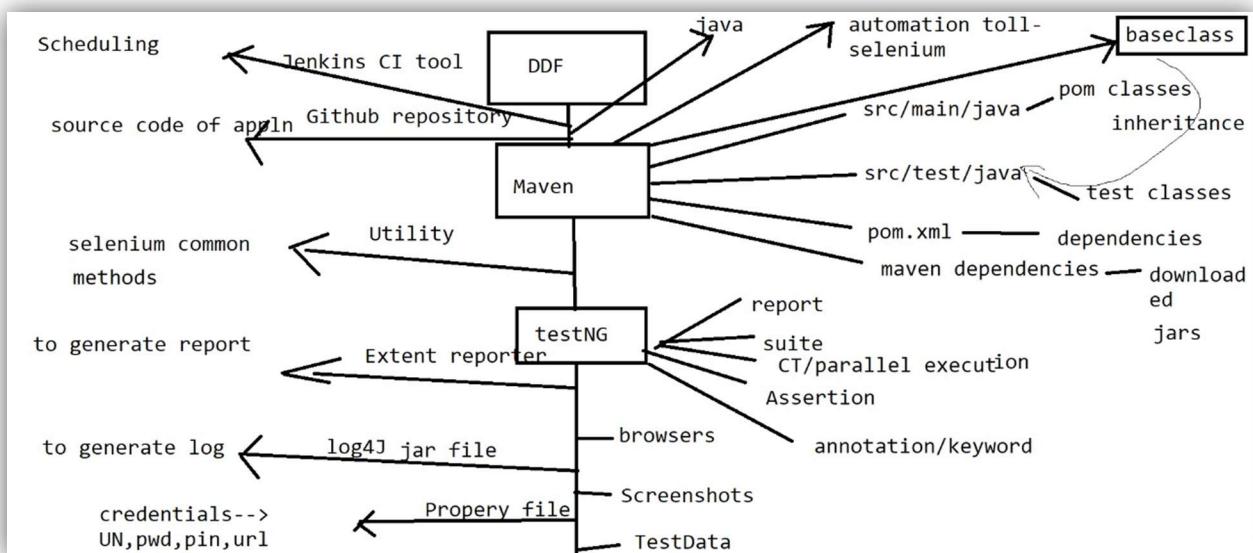
7. ENCAPSULATION

- All the POM classes in a framework are an example of Encapsulation. In POM classes, we declare the data members using @FindBy and initialization of data members will be done using Constructor to utilize those in methods.
- Encapsulation is a mechanism of binding code and data together in a single unit.
- Encapsulation is the process of wrapping up code and data together in a single unit. It is used to hide the data of a class from another class.
- Encapsulation can be achieved when you declare all variables as private and a public method in a class to get the values of the variable.

8. ABSTRACTION

- In Page Object Model design pattern, we write locators (such as id, name, xpath etc.,) in a Page Class.
- We utilize these locators in pom class but we can't see these locators in the tests. Literally we hide the locators from the tests.
- Abstraction is the methodology of hiding the implementation of internal details and showing the functionality to the users.

Q. Explain Selenium Framework.



- In my current project we are implementing Data driven framework using java programming language and selenium is used as automation tool
- In the framework Maven project is using, in the maven project, src main java consist of POM Classes and src Test java consist of Test Classes. In Pom.xml added the dependencies, downloaded jar are present in maven dependencies folder.
- Base class consist of the code related to opening the browsers which are extended to different test classes using extends keywords.
- To design the test class we are using TestNG by TestNg we can generate the Emailable reports, we can create test suit of different classes also execute the classes parallel, compatibility is also possible.
- It provides different assertions and also can use different annotation like before method, after method, before test, @Test,
- In my project we also created some folders like Browser folder where exe files of chrome, Firefox, edge browsers are stores, in the screenshots, Failed test cases captured screenshot saves also in the Test Data Consist of Xcel file.
- Property file is created to store the important credentials data like UN, Password, URL, PIN.
- Extent reporter use to generate report and log4j jar file use to generate log.
- In the Utility classes, frequently used methods like parameterization, screenshots are included.
- Daily developed code/source code is stores in the Github repository.
- For the continuous integration (CI) we use Jenkins for scheduling. If login time is 11.00 AM and If I have to run the script at 10.00 am in certain situation by scheduling in Jenkins task can complete.

Automation Test Cases

Investment Banking(Kite Zeroda) Project Scenarios

1. Quantity field should be by default 1
2. when we click on stock it should show buy ,sell,view chart and other details
26. wishlist--add/remove
3. while selecting order if we select CNC (cash and carry) order ,we can hold the stock for long term --manual
4. While Selecting order,if we select MIS(margine intraday square) it should automatically exit at 3-20 pm.--manual
5. if you try to place order ,but you don't have fund in your account then it should display insufficient fund msg
6. if you forget to logout account,at end of every day it should automatically logged out --manual/automation
7. when we placed the order by selecting market price ,price field should be disabled
8. when we placed limit order price field should be editable
11. price field should accept value in multiple of 0.05 (eg 55.05,55,56.15 etc accepted)
(55.06,55.07 not accepted)
12. after placing order ,in orders tab it will show Executed/pending/rejected orders
1. valid 2. invalid
13. in portfolio tab , should show holdings and position of stock. --> mobile
14. after clicking on user id tab it should show customer details
15. in options quantity field should accept in Lott eg (eg. 25 for banknifty,75 for nifty, depends on stock)
16. while adding fund in your demat account ,after clicking on add fund button , different payment options should enabled
17. when we click on withdraw button , withdrawal amount field should be enabled and proceed button should be disabled.
18. when we click on recent withdrawal option,it should show our withdrawal activities
19. when we click on AMO (after market order) option, we should able to placed order after market timing--manual
20. at time of order placing, BSE Or NSE only one should enabled
21. when we click on chart it should show different formats of chart (eg 1d,5d,1M etc) d-day,M- month
22. application should show same price of stock as per NSE or bse ,if we open an application in different platforms or machines, mobile etc ---CT
24. if we placed withdrawal request, ammount should be credit next day before 1 pm (if no hollyday) otherwise it should credit next of hollyday--manual
25. while withdrawing funds ,it should show withdrawable ammount.

Modules in Project: (Total around 16 Modules)				
Profile	Fund	Orders	wish list	Buy
Sell	Position	Holding	Dashboard	Login

Profile	Fund	Orders	wish list	Buy
Sell	Position	Holding	Dashboard	Login

Q. What is difference between implicit wait and explicit wait?

Synchronization: matching selenium test script speed with browser speed

1. Implicit wait:

Applicable: complete webpage

1 parameter: time value (seconds)

2. Explicit wait:

Applicable: single element in a webpage

1 parameter: time value (seconds)

2 parameter: Condition (isselected, isdisplayed, isenabled)

3. Fluent wait:

Applicable: single element

1 parameter: time value(seconds) --10

2 parameter: Condition (isselected, isdisplayed, isenabled)

3 frequency: time(time in sec)- 5 s

* Bitbucket :-
Bitbucket is web based version control system repository used for maintaining the source code. That uses either mercurial or Git.

* Version Control System :- (VCS)
VCS are system that track changes made to digital assets over time.

Any digital assets (eg. File, image, video) entity that can be changed over time can be tracked by using VCS.

* TOP version Control System.

* 1. Git (Bitbucket internally uses Git)

2. CVS

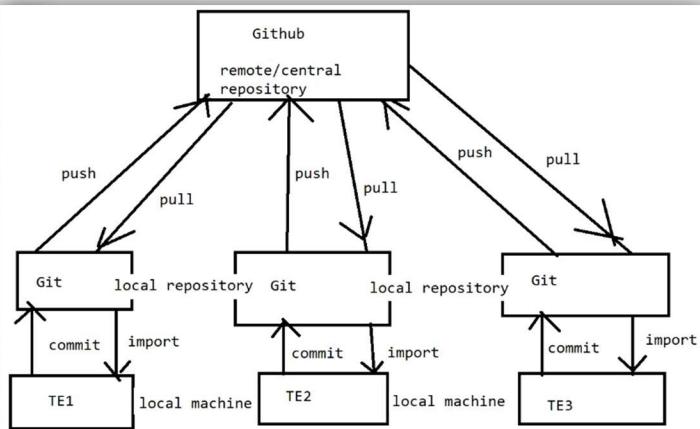
3. SVN

4. Assembly

* 5. mercurial

6. Bazaar

group of
 * Git allows people to work on the same documents (code) at the same time.
 * It eliminates confusion when more than 1 person is making changes to same code



Git commands cheatsheet

@JSCRIPTERS

• git init	>Create a new Git repository	• git checkout [branch]	Switch to a branch
• git clone	Clone a repository	• git branch -d	Delete a Branch
• git status	Show state of current directory	• git branch -m	Rename a branch
• git log	List the commit history	• git merge [branch]	Merge the specified branch
• git diff	diff b/w working directory & index	• git add [file]	Stage changes
• git show	Display the content and metadata	• git add .	Stage everything
• git branch	List all branches in the repository	• git revert [file]	Undo changes
		• git clean -n	Show untracked files
		• git commit --amend	Replace the last commit
		• git remote add	connection to a remote repository
		• git pull	Fetch a repository
		• git push	Push a branch
		• git config --global user.name	Define the author name to be used
		• git reset	Reset staging area to match most recent commit
		• git fetch <remote> <branch>	Fetches a specific , from the repo.

GitHub Repository

1 github acc--> multiple repository

 1 repository--> multiple branches (Master)

Steps To Create Repository In Github Server

1. create a project in eclipse
2. create acc in github
3. create repository in github
4. Create local Repository(git) : right click on project-->Team-->Share project—select checkbox -->select checkbox-->Create repository-->Finish
5. open local repository window : click on quick search-->search git repository --> git repositories(git)
6. commit source code from local machine to local repository: right click on project-->Team -->Add to index--> right click on project-->Team-->Commit-->add commit message-->commit
7. push source code from local repository(git) to remote repository(bitbucket/github) right click on project-->Team-->remote-->push-->Enter URL--> Enter UN & PWD-->next-->source ref-->master-->add Specification-->finish

Clone repository

1. open git repository window--> clone a git repository--> Enter repository URL,UN & Pwd--> Next-->select branch & Next--> Finish
2. Move Project from local repo to working directory (import project) : File-->import-->Git-->Project from git-->Existing local repo-->select project-->Finish

<https://github.com/SanjayChetlure/Sept20.git>

UN: SanjayChetlure

pWD:SanjayC@1992

gitHub--> multiple repository

 1 repository--> multiple branches (master)

Q. how to solve conflict issue

master--> pre-defined

 --> user defined

multiple branches-->

1. master

2. Nikhil

3. Prasad

 --> pull request to TL from Nikhil & Prasad

 TL --> code review of Nikhil & Prasad

ok-->move Nikhil & Prasad code to master branch correction--> rpl code change--> code modify--> code push--> TL pull request-->master brnach

next day--> pull latest code from master branch

Q. Difference between String & StringBuffer

String	StringBuffer/Stringbuilder
<ul style="list-style-type: none"> String class is immutable. 	<ul style="list-style-type: none"> StringBuffer class is mutable.
<ul style="list-style-type: none"> String is slow and consumes more memory when you concat too many strings because every time it creates new instance. 	<ul style="list-style-type: none"> StringBuffer is fast and consumes less memory when you concat strings.

Q. Difference between StringBuffer and StringBuilder

StringBuffer	StringBuilder
<ul style="list-style-type: none"> StringBuffer was introduced in Java 1.0(Legacy) 	<ul style="list-style-type: none"> StringBuilder was introduced in Java 1.5 (not Legacy)
<ul style="list-style-type: none"> performance: low 	<ul style="list-style-type: none"> performance: High
<ul style="list-style-type: none"> synchronized & Thread safe 	<ul style="list-style-type: none"> not synchronized & not Thread safe

Q. What is Final, Finally and finalize.

Final: is a keyword

Finally: finally is the block in Java Exception Handling to execute the important code whether the exception occurs or not

Finalize: finalize is the method in Java which is used to perform clean up processing just before object is garbage collected.

Q. Difference between Hashmap & HashTable

HashMap	Hashtable
<ul style="list-style-type: none"> not legacy class 	<ul style="list-style-type: none"> legacy class
<ul style="list-style-type: none"> performance: high 	<ul style="list-style-type: none"> performance: low
<ul style="list-style-type: none"> HashMap is non-synchronized 	<ul style="list-style-type: none"> HashTable is synchronized
<ul style="list-style-type: none"> It is not-thread safe 	<ul style="list-style-type: none"> It is thread safe