

Selenium

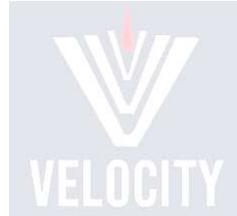
Learn Selenium by Sanjay Sir



Contents

Page No.

1. Introduction to Automation	02
2. Selenium Installation	07
3. Webdriver and Methods (<code>get, close, maximize, getTitle, setPosition</code>)	10
4. Method with return type	16
5. Types of Locator (<code>xPath, Id, name, className, tagName, CSS</code>)	17
6. WebElements Methods (<code>click, sendKeys, clear, isEnabled, isDisplayed, isSelected</code>)	27
7. Collection	32
A) List	33
B) Set	43
8. ListBox	50
9. Screenshot	61
10. Parameterization	64
11. Iframe	76
12. Dynamic WebElement Handling	79



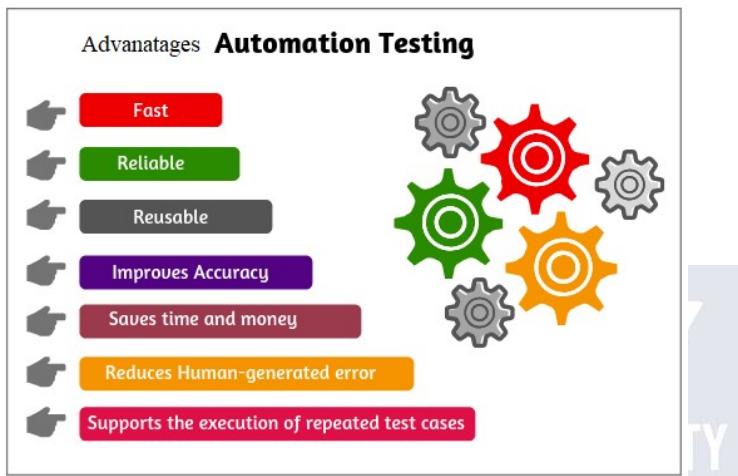
What is Automation Testing?

Automation Testing or Test Automation is a software testing technique that performs using special automated testing software tools to execute a test scripts/test case suite.

Need for Automation Testing

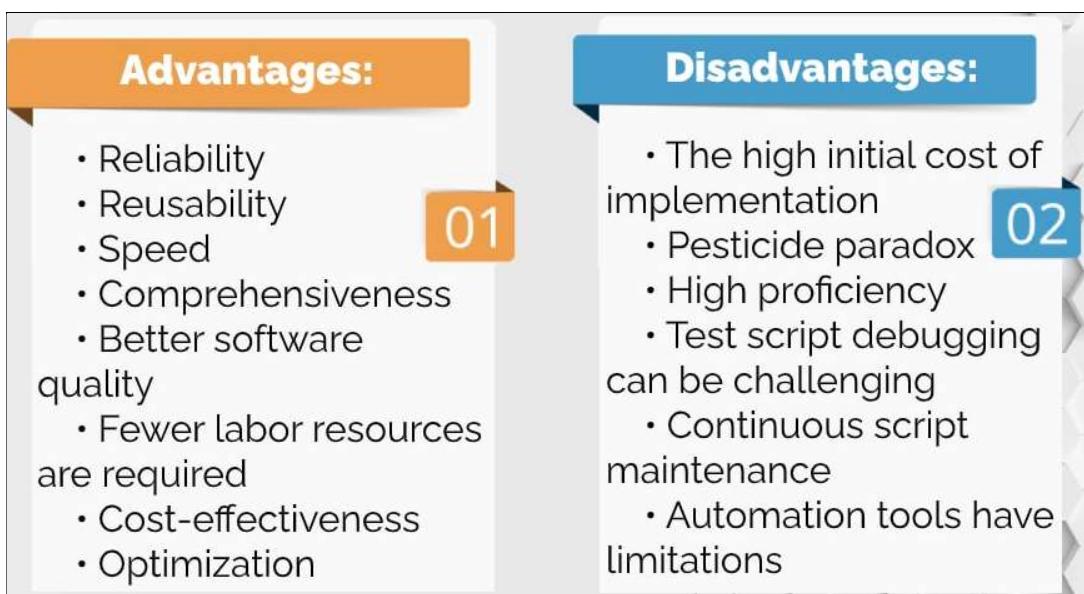
- **Speed** : Automation Scripts are fast when compared to manual testers efforts.
- **Reliable** : Tests perform precisely the same operations each time they are run, thereby eliminating human error.
- **Repeatable**: Tests can be repeated n number of times for execution of the same operation.
- **Coverage**: Automated tests increase coverage.
- **Reusable** : We can reuse tests on different versions of an application, even if the user interface changes.

Advantages of Automation Testing



- Compatibility testing is easy and possible.
- Reusability of Test Scripts.
- Overcomes the drawback of regression testing.

Disadvantages of Automation Testing



Which Test Cases to Automate?

Test cases to be automated can be selected using the following criterion to increase the automation ROI

- High Risk – Business Critical test cases
- Test cases that are repeatedly executed
- Test Cases that are very tedious or difficult to perform manually
- Test Cases which are time-consuming

The following category of test cases are not suitable for automation:

- Test Cases that are newly designed and not executed manually at least once
- Test Cases for which the requirements are frequently changing
- Test cases which are executed on an ad-hoc basis.

Automated Testing Process:

Following steps are followed in an Automation Process

Step 1) Test Tool Selection

Step 2) Define scope of Automation

Step 3) Planning, Design and Development

Step 4) Test Execution

Step 5) Maintenance

Comparison of Manual Vs Automation

Manual Testing	Automated Testing
<ul style="list-style-type: none"> • Manual testing is not accurate at all times due to human error, hence it is less reliable. • Manual testing is time-consuming, taking up human resources. • Investment is required for human resources. • Manual testing is only practical when the test cases are run once or twice, and frequent repetition is not required. • Manual testing allows for human observation, which may be more useful if the goal is user-friendliness or improved customer experience. 	<ul style="list-style-type: none"> • Automated testing is more reliable, as it is performed by tools and/or scripts. • Automated testing is executed by software tools, so it is significantly faster than a manual approach. • Investment is required for testing tools. • Automated testing is a practical option when the test cases are run repeatedly over a long time period. • Automated testing does not entail human observation and cannot guarantee user-friendliness or positive customer experience.

SELENIUM COMPATIBILITY

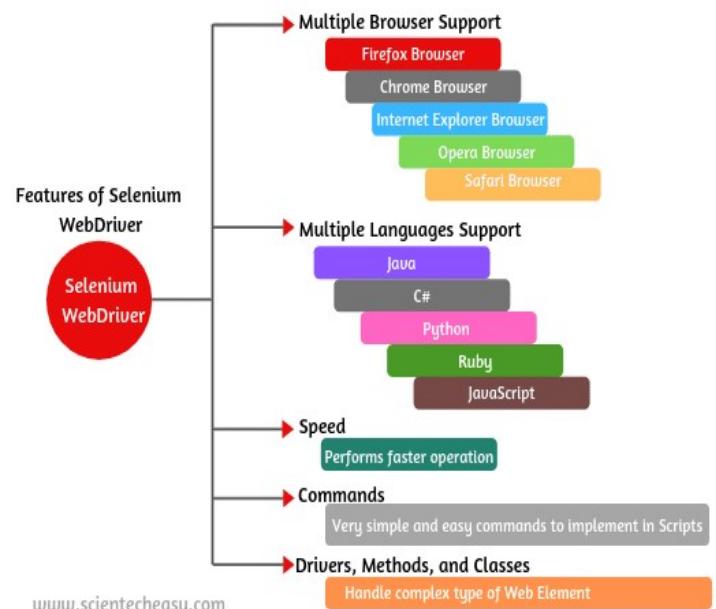
Languages	Platforms	Browsers
Java C# PHP Ruby Perl Python JavaScript Objective-C Haskell R	Windows Linux Mac Android (with Selendroid, Appium, or Robotium) iOS (with ios-driver or Appium)	Google Chrome Internet Explorer Firefox Safari Opera Microsoft Edge HtmlUnitDriver

Different Automation Tool

<ul style="list-style-type: none"> Selenium Appium Katalon Studio Cucumber HPE Unified Functional Testing (UFT) SoapUI TestComplete Worksout IBM Rational Functional Tester (RFT) Telerik Test Studio 	<p>The 10 Best Test Automation Tools</p>
---	--

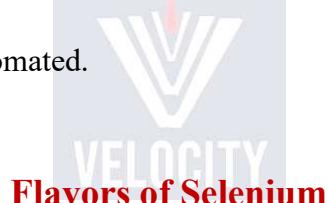
Product	Selenium	Katalon Studio	Unified Functional Testing	TestComplete	watir
Available since	2004	2015	1998	1999	2008
Application Under Test	Web apps	Web (UI & API), Mobile apps	Web (UI & API), Mobile, Desktop, Packaged apps	Web (UI & API), Mobile, Desktop apps	Web apps
Pricing	Free	Free	\$\$\$\$	\$\$	Free
Supported Platforms	Windows Linux OS X	Windows Linux OS X	Windows	Windows	Windows Linux OS X
Scripting languages	Java, C#, Perl, Python, JavaScript, Ruby, PHP	Java/Groovy	VBScript	JavaScript, Python, VBScript, JScript, Delphi, C++ and C#	Ruby
Programming skills	Advanced skills needed to integrate various tools	Not required. Recommended for advanced test scripts	Not required. Recommended for advanced test scripts	Not required. Recommended for advanced test scripts	Advanced skills needed to integrate various tools
Ease of Installation and Use	Require advanced skills to install and use	Easy to setup and use	Complex in installation. Need training to properly use the tool	Easy to setup. Need training to properly use the tool	Advanced skills needed to integrate various tools

Advantages of Selenium



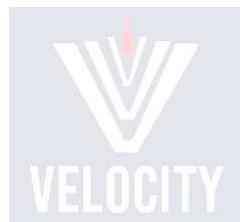
Limitations of Selenium

- We can only automate web-based applications
- Unable to automate captcha
- Selenium doesn't support for file uploading
- Unable to read barcode
- Ad-hoc testing can't be automated.



Compare Selenium IDE, RC and WebDriver

	Selenium IDE	Selenium RC	Selenium WebDriver
Browsers	Only Firefox	IE, Firefox, Chrome, other	IE, Firefox, Chrome, other
Record/Playback	Yes	No	No
Start “server” to run tests	No	Yes	No
Core engine	Java script	Java script	Plugin, interact natively with browser
Object Oriented API	No	Less Object Oriented	Yes
Test on iPhone and Android	No	No	Yes



Selenium installation

step1: Create a new java project

step2: download selenium jar file.

step3: Configure selenium jar file into our selenium project.

1. right click on project
2. select build path
3. select configure build path
4. click on Libraries tab.
5. click on Add external jars.
5. browse selenium jar file.
6. Apply
7. Apply & close

Note: class path & module path ---> select class path option

Step4: download ChromeDriver.exe file

pre-condition: check current chrome browser version

1. unzip downloaded ChromeDriver.exe file

1st selenium program: open browser

step1: set ChromeDriver.exe file path by using System.setProperty(parameter1, parameter2) method

parameter1 - name of browser (**small letter**)

parameter2 - path of browser (**convert \ to **)

Step2: create an object of chromedriver class

note: sessionNotCreated exception--> download lower/upper version ChromeDriver.exe file

```

package sample1;

import org.openqa.selenium.chrome.ChromeDriver;
public class demo1
{
    public static void main(String[] args)
    {
        //step1: set path of chromedriver.exe file
        //parameter1: name of the browser --> small letter
        //parameter2: path of browser
        System.setProperty("webdriver.chrome.driver",
                           "C:\\\\Users\\\\sanjay\\\\Desktop\\\\Study\\\\Selenium_Syllabus\\\\Jul21_Eve_Selenium\\\\chromedriver_win32 (14)\\\\chromedriver.exe");

        //step2: create an object of ChromeDriver class
        ChromeDriver driver=new ChromeDriver();
    }
}

```



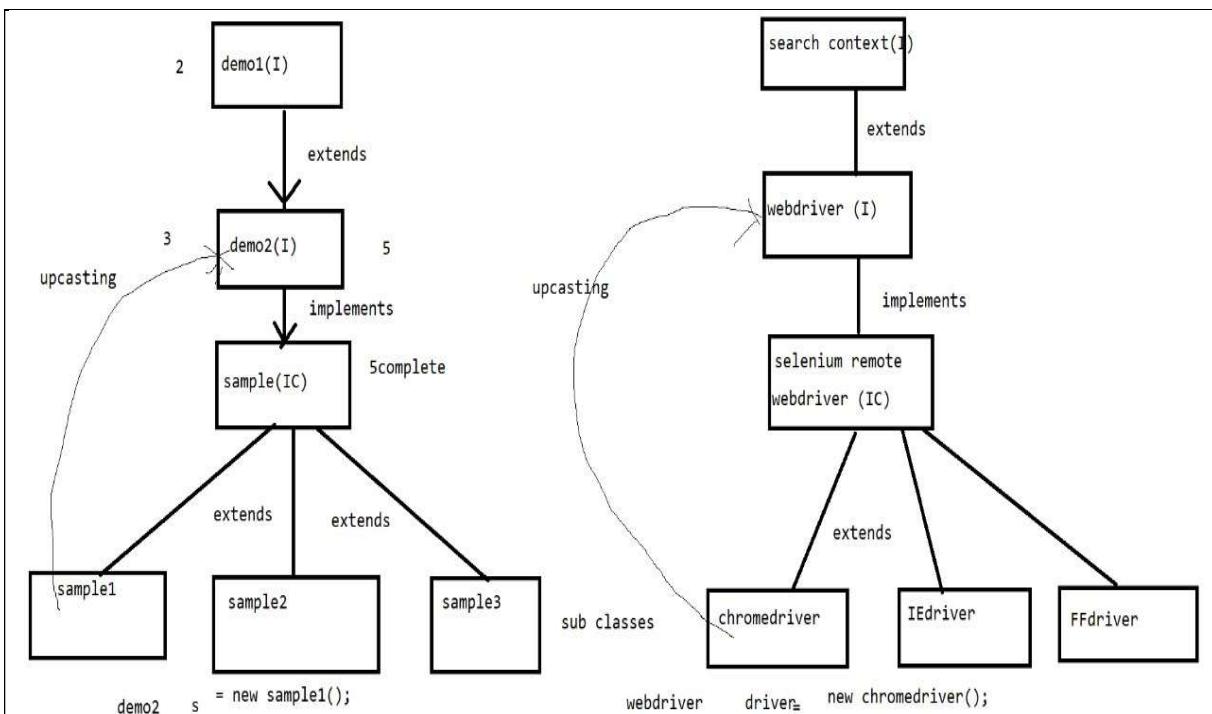
```

package sample1;

import org.openqa.selenium.chrome.ChromeDriver;
public class demo2 {
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver",
                           "C:\\\\Users\\\\sanjay\\\\Desktop\\\\Study\\\\Selenium_Syllabus\\\\Jul21_Eve_Selenium\\\\chromedriver_win32 (14)\\\\chromedriver.exe");

        ChromeDriver driver=new ChromeDriver();
    }
}

```



```

package sample1;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class demo2
{
    public static void main(String[] args)
    {
        System.setProperty("webdriver.chrome.driver",
                           "C:\\\\Users\\\\sanjay\\\\Desktop\\\\Study\\\\Selenium_Syllabus\\\\Jul21_Eve_Selenium\\\\chromedriver_win32 (14)\\\\chromedriver.exe");

        //ChromeDriver driver=new ChromeDriver();
        WebDriver driver=new ChromeDriver();
    }
}

package Installation;

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

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

        ChromeDriver driver1 = new ChromeDriver();

        WebDriver driver = new ChromeDriver();
        driver.get("https://www.google.com/");
        Thread.sleep(10000);
        driver.close();
    }
}
  
```

Webdriver and Its Methods:

- **Webdriver** an interface use to perform actions on browser.
1. **get**- to enter URL in browser or to open an application.
`driver.get("https://www.google.com");`
 2. **close**: to close the current tab of the browser.
`driver.close();`
 3. **quit**: to close the all the tabs present in browser
`driver.quit();`
 4. **maximize** - use to maximize the size/dimention browser
`driver.manage().window().maximize();`
 5. **navigate**: this method is use to open an application, **move forward, backward & refresh** the webpage.
`driver.navigate().to("https://www.google.com");`
 6. **getTitle**: this method is use to get title of a webpage. **return type of getTitle method is String**
`String Title = driver.getTitle();`
 7. **getCurrentURL**: this method is use to get URL of a webpage. **return type of getCurrentURL method is String**
`String URL = driver.getCurrentUrl();`
 8. **setSize**: this method is use to change the dimension or Size
`Dimension d = new Dimension(600, 500);
driver.manage().window().setSize(d);`
 9. **getSize**: this method is use to identify the dimension or Size
`Dimension d2 = driver.manage().window().getSize();
System.out.println(d2);`
 10. **setPosition**: this method is use to set the position of browser on X-Y Axis
`Point p = new Point(100, 50);
driver.manage().window().setPosition(p);`
 11. **getPosition**: this method is use to get the position of browser on X-Y Axis
`System.out.println(driver.manage().window().getPosition());`

Note: can't minimize the browser (we can change size & position of the browser).

```

package Webdriver_Methods;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class A_Get_Method
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver",
        "H:\\Selenium\\chromedriver.exe");

        WebDriver driver = new ChromeDriver();

        //driver.get("https://www.w3schools.com/sql/default.asp");
        driver.get("https://www.google.com/");
        Thread.sleep(3000);
        driver.close(); } // Close is used to close CURRENT TAB only
    }-----
```

```

package Webdriver_Methods;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class B_Close_Method
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver",
        "H:\\Selenium\\chromedriver.exe");

        WebDriver driver = new ChromeDriver();
        driver.get("https://www.google.com/");
        Thread.sleep(3000);
        driver.close();}
```

```

package Webdriver_Methods;

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

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

        WebDriver driver = new ChromeDriver();

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

        Thread.sleep(3000);

        driver.quit(); // Close all the tabs or Close Complete Browser
    }}
```

```

package Webdriver_Methods;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class D_Navigate_Method
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver",
        "H:\\Selenium\\chromedriver.exe");

        WebDriver driver = new ChromeDriver();

        driver.navigate().to("https://www.google.com/");
        Thread.sleep(2000);

        driver.navigate().to("https://www.flipkart.com/");
        Thread.sleep(2000);

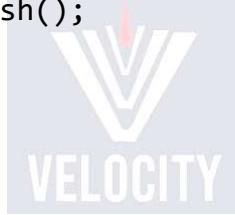
        driver.navigate().back();
        Thread.sleep(2000);

        driver.navigate().forward();
        Thread.sleep(2000);

        driver.navigate().refresh();
        Thread.sleep(2000);

        driver.quit();
    }
}

```



```

package Webdriver_Methods;

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

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

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

        WebDriver driver = new ChromeDriver();
        driver.navigate().to("https://www.flipkart.com/");
        //driver.manage().window().fullscreen();
        driver.manage().window().maximize();
        Thread.sleep(3000);
    }
}

```

```

package Webdriver_Methods;

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

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

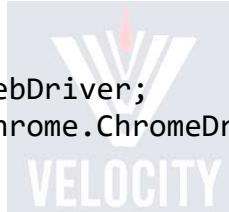
        WebDriver driver = new ChromeDriver();

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

        String Title = driver.getTitle();

        System.out.println(Title);
    }
}

```



```

package Webdriver_Methods;

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

public class F_getTitle1
{
    public static void main(String[] args)
    {
        System.setProperty("webdriver.chrome.driver",
        "H:\\Selenium\\chromedriver.exe");

        WebDriver driver = new ChromeDriver();

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

        String actTitle = driver.getTitle();
        String expTitle = "Online Shopping Site for Mobiles,
Electronics, Furniture, Grocery, Lifestyle, Books & More. Best
Offers!";

        if(actTitle.equalsIgnoreCase(expTitle))
        {
            System.out.println("Navigated to Current WebPage");
        }
        else
        {
            System.out.println("Navigated to Wrong WebPage");
        }
    }
}

```

```

package Webdriver_Methods;

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

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

        WebDriver driver = new ChromeDriver();

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

        String URL = driver.getCurrentUrl();
        System.out.println(URL);
    }
}

```

```

package Webdriver_Methods;

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

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

        WebDriver driver = new ChromeDriver();

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

        Thread.sleep(3000);

        Dimension d = new Dimension(600, 500);
        driver.manage().window().setSize(d);
    }
}

```

```

package Webdriver_Methods;

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

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

        WebDriver driver = new ChromeDriver();

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

        Dimension d1 = driver.manage().window().getSize();

        System.out.println(d1);
        //driver.manage().window().fullscreen();      // (1382, 744)
        driver.manage().window().maximize();         // (1366, 768)

        Dimension d2 = driver.manage().window().getSize();

        System.out.println(d2);
    }
}

```

```

package Webdriver_Methods;

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

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

        WebDriver driver = new ChromeDriver();

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

        Point p = new Point(100, 50);
        driver.manage().window().setPosition(p);

        System.out.println("Position =
"+driver.manage().window().getPosition());
    }
}

```

Method with Return Type

```

package Method_With_Return_Type;

public class MethodWithReturnType
{
    public static void main(String[] args)
    {
        multiplication(10,15);
    }

    public static int multiplication(int a, int b)
    {
        int mul = a*b;
        System.out.println(mul);
        return mul;
    }

}

```

```

package Webdriver_Methods;
public class example_method_with_ReturnType{
    public static void main(String[] args) {
        int num1=100;
        int num2 = addition(10,15); //25

        int num3 = multiplication(num1, num2); //100, 25
        System.out.println(num3);

        String s1="abcd";
        String s2 = convertStringToUpperCase(s1);
        System.out.println(s2);
    }

    public static String convertStringToUpperCase(String str1)
    {
        String str2 = str1.toUpperCase();
        return str2;
    }

    public static int multiplication(int a, int b) //100, 25
    {
        int mul=a*b; //100*25 =2500
        return mul;
    }

    //method with int return type
    public static int addition(int a, int b) //a=10, b=15
    {
        int sum=a+b; //25
        return sum;}}}

```

Locator: Locators are used to find the element present on Webpage.

Types of Locator

- A. X Path (**Slowest locator to find Element**)
 - B. Id (**Fastest locator to find Element**)
 - C. Name
 - D. Class name
 - E. Tag name
 - F. CSS selector
 - G. Link text
 - H. Partial link text
-

A. Locator using X Path: X path is again divided into following sub types

- a. X path by Attributes @
- b. X path by Text
- c. X path by Contains
- d. X path by Index
- e. Absolute X path
- f. Relative X path

xpath Expressions

1. xpath by attribute
`//tagname[@attributeName = 'attribute value']`
2. xpath by text
`//tagname[text() = 'text value']`
3. xpath by contains with attribute
`//tagname[contains (@ attributeName, 'attribute Value')]`
4. xpath by contains with text
`//tagname[contains(text(), 'text value')]`
5. xpath by index
`//tagname[@attributeName = 'attribute value'][1]`
6. AbsoluteX Path /

To find the X path by using absolute method we need to find HTML tree diagram first. In that diagram **we have to move forward from parent to next immediate child**. We need to use single forward slash (/) in between parent and its immediate child. We have to follow this approach until our expected element not found.

7. RelativeX Path //

To find the X path by using absolute method we need to find HTML tree diagram first. In that diagram we have to **move forward from parent to any child class**. We need to use double forward slash (//) in between parent and any child. We are using same example an html tree diagram which is used in absolute X path.

1. xpath by attribute

//tagname[@attributeName = 'attribute value']

```

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

public class E1_XpathByAttribute
{
    public static void main(String[] args)
    {
        System.setProperty("webdriver.chrome.driver", "H:\\Selenium\\chromedriver.exe");

        WebDriver driver = new ChromeDriver();
        driver.get("https://en-gb.facebook.com/");

        driver.findElement(By.xpath("//input[@id='email']")).sendKeys("vaibyendole");

        driver.findElement(By.xpath("//input[@id='pass']")).sendKeys("xyz");

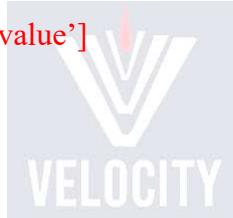
        driver.findElement(By.xpath("//button[@class='_42ft _4jy0 _6lth _4jy6 _4jy1 selected _51sy']")).click();

    }
}

```

2. xpath by text

//tagname[text() = 'text value']



```

package Locators;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class E2_X_PathByText {

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

        driver.get("https://en-gb.facebook.com/login/");      //tagname[text() = 'text value']

        driver.findElement(By.xpath("//a[text()='Forgotten account?']")).click();

        driver.findElement(By.xpath("//input[@type='text']")).sendKeys("9764963423");

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

    }
}

```

3. xpath by contains(text)
4. xpath by contains (attribute)
5. xpath by index

```

package XpathMEthod1;

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

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

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

WebDriver driver = new ChromeDriver();

driver.get("https://en-gb.facebook.com/login/");

//xpath by Contains(text)
driver.findElement(By.xpath("//a[contains(text(),'Forgotten ')]")).click();

//xpath by Contains(attribute)
driver.findElement(By.xpath("//a[contains(@rel,'follow')][1]")).click();

}

```

3. xpath Using Contains(text())

```

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

public class example3_XpathByConatins_UsingText1
{
public static void main(String[] args)
{
    System.setProperty("webdriver.chrome.driver",

"C:\\Users\\sanjay\\Desktop\\Study\\Selenium_Syllabus\\Jul21_Eve_Selenium\\chromed
river_win32 (14)\\chromedriver.exe");
    WebDriver driver=new ChromeDriver();
    driver.get("https://en-gb.facebook.com/");

    //click on forgotten pwd link
    driver.findElement(By.xpath("//a[contains(text(),'password')]")).click();
}
}

```

```
package Locators;
```

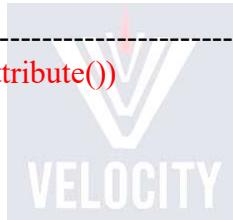
```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class example3_XpathByConatins_UsingText2
{
    public static void main(String[] args)
    {
        System.setProperty("webdriver.chrome.driver",
            "C:\\\\Users\\\\sanjay\\\\Desktop\\\\Study\\\\Selenium_Syllabus\\\\Jul21_Eve_Selenium\\\\chromedriver_win32 (14)\\\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.get("https://en-gb.facebook.com/");

        //click on create new acc link
        driver.findElement(By.xpath("//a[contains(text(),'Account')]")).click();

    }
}
```

4. xpath Using Contains(attribute())



```
package Locators;
```

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

public class example4_XpathByConatins_UsingAttribute1
{
    public static void main(String[] args)
    {
        System.setProperty("webdriver.chrome.driver",
            "C:\\\\Users\\\\sanjay\\\\Desktop\\\\Study\\\\Selenium_Syllabus\\\\Jul21_Eve_Selenium\\\\chromedriver_win32 (14)\\\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.get("https://en-gb.facebook.com/");

        //enter UN
        driver.findElement(By.xpath("//input[contains(@placeholder,'Email address')]")).sendKeys("abc");

    }
}
```

package Locators;

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

public class example4_XpathByConatins_UsingAttribute2
{
    public static void main(String[] args)
    {
        System.setProperty("webdriver.chrome.driver",
        "C:\\\\Users\\\\sanjay\\\\Desktop\\\\Study\\\\Selenium_Syllabus\\\\Jul21_Eve_Selenium\\\\chromed
        river_win32 (14)\\\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.get("https://en-gb.facebook.com/");

        //click on create new acc link
        driver.findElement(By.xpath("//a[contains(@class,'_4jy2 selected _51sy')]")).click();

    }
}
```



- Locator No. 2 : tagName()

- Indexing is not possible with Tagname
- If there are two same Tagname are present (like input, input) then selenium by default perform action only on 1st Tagname. (Both operations/multiple operations perform only on 1st Tagname input)

```
package Locators2;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

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

        WebDriver driver = new ChromeDriver();

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

        driver.findElement(By.tagName("input")).sendKeys("Vaibhav");

        driver.findElement(By.tagName("input")).sendKeys("Yendole");

        // In Above case Tagname "input" is repeated/duplicate so selenium perform
        Both actions on only on 1st Tagname.
    }
}
```

First Name	Vaibhav Yendole
Last Name	

Locator No. 3 : id()

- Indexing is not possible with ID
- If every element has unique ID then we can use ID locator to find element.
- But if there is duplicate ID present then actions will perform on only 1st ID.
- Limitation: If there is no ID locator present we can't use it.
- If ID are unique then we can use ID locator to find element.

```
package Locators2;

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

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

        WebDriver driver = new ChromeDriver();

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

        driver.findElement(By.id("1234")).sendKeys("Ranjit ");

        driver.findElement(By.id("1234")).sendKeys("Walia");

    }
}
```

First Name	Ranjit Walia
Last Name	

Locator No. 4: className()

- Indexing not possible with className() locator.
- If there are unique/different className use then we can use className() locator to find element.
- But if the className are same/ repetitive then selenium will perform action only on 1st element.

```
package Locators2;

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

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

        WebDriver driver = new ChromeDriver();

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

        driver.findElement(By.className("abc")).sendKeys("Virat ");
        driver.findElement(By.className("abc")).sendKeys("Kohli");
    }
}
```

First Name	Virat Kohli
Last Name	

Locator No. 5: name()

```
package Locators2;

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

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

        WebDriver driver = new ChromeDriver();

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

        driver.findElement(By.name("abc123")).sendKeys("Sagar ");
        driver.findElement(By.name("abc123")).sendKeys("Shinde");
    }
}
```

First Name	Sagar Shinde
Last Name	

Locator No. 6: linkText()

- If we want to use linkText locator then text associated with the link can be used to click on the link.
- But if there is duplicate text present then we can't use linkText locator for finding element.

```
<html>
<body>
FN<input type='text' id='1234' class='abc' name='abc123'> <br>
LN<input type='text' id='1234' class='abc' name='abc123'>
</body>
</html>
```

```
package Locators2;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class E_linkText_Locator

{
    public static void main(String[] args)

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

        WebDriver driver =new ChromeDriver();

        driver.get("https://en-gb.facebook.com/");

        driver.findElement(By.linkText("facebook")).click();

    }
}
```



Locator No. 7: partialLinkText()

```
<html>
<body>
<a href="https://www.facebook.com/">facebook</a>
</body>
</html>
```

package Locators;

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

public class example11_PartialLinkText {
    public static void main(String[] args) throws InterruptedException {
        System.setProperty("webdriver.chrome.driver",
                "C:\\\\Users\\\\sanjay\\\\Desktop\\\\Study\\\\Selenium_Syllabus\\\\Jul21_Eve_Selenium\\\\chromedriver_win32 (14)\\\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.get("file:///C:/Users/sanjay/Desktop/Study/Selenium_Syllabus/Jul21_Eve_Selenium/Html%20files/LinkText_PartialLinkText.html");
        Thread.sleep(2000);
        driver.findElement(By.partialLinkText("book")).click();
    }
}
```

WebElement Methods:

WebElement: it is an interface use to perform action on element present on webpage.

1. Sendkeys() :

- This method is use to enter value in the input/text field

2. Clear():

- This method is use to clear value in the text field.

3. Click():

- Click method is use to click on buttons, links also use to select radio buttons & checkboxes.

4. getText:

- This method is use to get text present in a webpage.
- Return type of `getText` function is String.

5. isEnabled():

- This method is use to verify element is enabled or disabled.
- Return type of `isEnabled` function Boolean
- If element is enabled then it returns true otherwise it returns false.

6. isDisplayed():

- This method is use to verify element is present or not.
- Return type of `isDisplayed` function is boolean.
- If element is present then it returns true otherwise it returns false.

7. isSelected():

- This method is used to verify radio/checkbox is selected or not
- Return type of `isSelected` function is boolean.
- If radio button is selected then returns True otherwise False

```
package WebElements_Methods;           1. Sendkeys()          2. Clear():

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

public class E1_sendKeys_and_Clear
{
    public static void main(String[] args)
    {
        System.setProperty("webdriver.chrome.driver", "H:\\Selenium\\chromedriver.exe");

        WebDriver driver = new ChromeDriver();

        driver.get("https://en-gb.facebook.com/");

        WebElement UN = driver.findElement(By.xpath("//input[@name='email']"));

        UN.sendKeys("ABCDEF");

        UN.clear();

        UN.sendKeys("vaibyendole");
    }
}
```

3. Click():

```

package WebElements_Methods;

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

public class E2_Click_WebElement_Method
{
    public static void main(String[] args)
    {
        System.setProperty("webdriver.chrome.driver", "H:\\Selenium\\chromedriver.exe");

        WebDriver driver = new ChromeDriver();

        driver.get("https://en-gb.facebook.com/");

        driver.findElement(By.xpath("//a[text()='Create New Account']")).click();
    }
}
-----
```

```
package WebElements_Methods;
```

4. getText()



```

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

public class E3_getText
{
    public static void main(String[] args)
    {
        System.setProperty("webdriver.chrome.driver", "H:\\Selenium\\chromedriver.exe");

        WebDriver driver = new ChromeDriver();

        driver.get("https://en-gb.facebook.com/");

        String text = driver.findElement(By.xpath("//a[text()='Forgotten password?']")).getText();

        System.out.println(text);
    }
}
```

```
package WebElements_Methods;
```

5. isEnabled()

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

public class E4_isEnabled
{
    public static void main(String[] args)
    {
        System.setProperty("webdriver.chrome.driver", "H:\\Selenium\\chromedriver.exe");

        WebDriver driver = new ChromeDriver();
        driver.get("https://en-gb.facebook.com/");

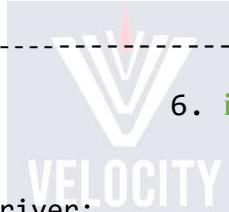
        boolean result = driver.findElement(By.xpath("//button[@name='login']")).isEnabled();

        System.out.println(result);

        if(result==true)
        {
            System.out.println("Element is Enable");
        }
        else
        {
            System.out.println("Element is Disable");
        }
    }
}
```

```
package WebElements_Methods;
```

6. isDisplayed()



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

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

        WebDriver driver = new ChromeDriver();

        driver.get("https://en-gb.facebook.com/");

        boolean result = driver.findElement(By.xpath("//img[@class='fb_logo _8ilh img']")).isDisplayed();

        System.out.println(result);

        if(result==true)
        {
            System.out.println("Element Found");
        }
        else
        {
            System.out.println("Element Not Found");
        }
    }
}
```

7. isSelected()

```

package WebElements_Methods;

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

public class E6_isEnabled
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver",
        "H:\\Selenium\\chromedriver.exe");

        WebDriver driver = new ChromeDriver();

        driver.get("https://en-gb.facebook.com/");

        driver.findElement(By.xpath("//a[text()='Create New
Account']")).click();

        Thread.sleep(3000);

        boolean value =
driver.findElement(By.xpath("//input[@class='_8esa'][1]")).isSelecte
d();
        System.out.println(value);
        if(value==true)
        {
            System.out.println("Radio button selected");
        }
        else
        {
            System.out.println("Radio button is not selected");
        }
    }
}

```

7. isSelected()

```

package WebElements_Methods;

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

public class E6_isEnabled
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver",
        "H:\\Selenium\\chromedriver.exe");

        WebDriver driver = new ChromeDriver();

        driver.get("https://en-gb.facebook.com/");

        driver.findElement(By.xpath("//a[text()='Create New
Account']")).click();

        Thread.sleep(3000);

        WebElement female =
        driver.findElement(By.xpath("//input[@class='_8esa'][1]"));
        boolean result = female.isSelected();
        System.out.println(result);
        if(result==true)
        {
            System.out.println("Radio button already selected");
        }
        else
        {
            System.out.println("Radio button is not selected");
            female.click();
            boolean result1 = female.isSelected();
            if(result1==true)
            {
                System.out.println("Radio Button is Now Selected");
            }
            else
            {
                System.out.println("Radio Button is Not Selected");
            }
        }
    }
}

```

Output:

```

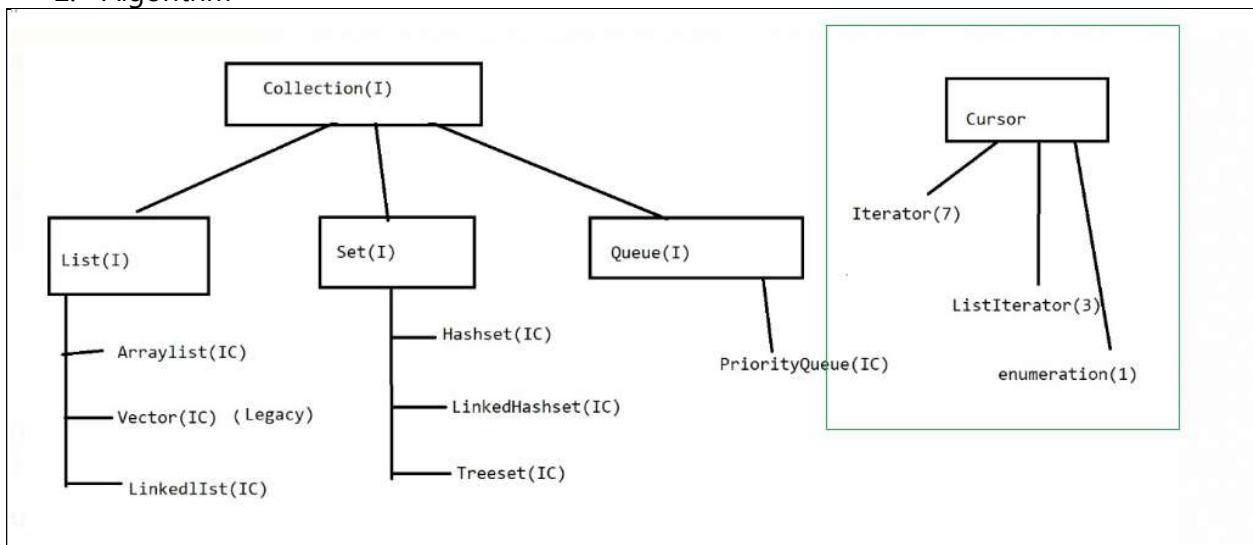
false
Radio button is not selected
Radio Button is Now Selected

```

- Collection

Collection is an interface represents a unified architecture for storing and manipulating a group of objects. It has:

1. Interfaces (I) and its implementations classes (IC)
2. Algorithm



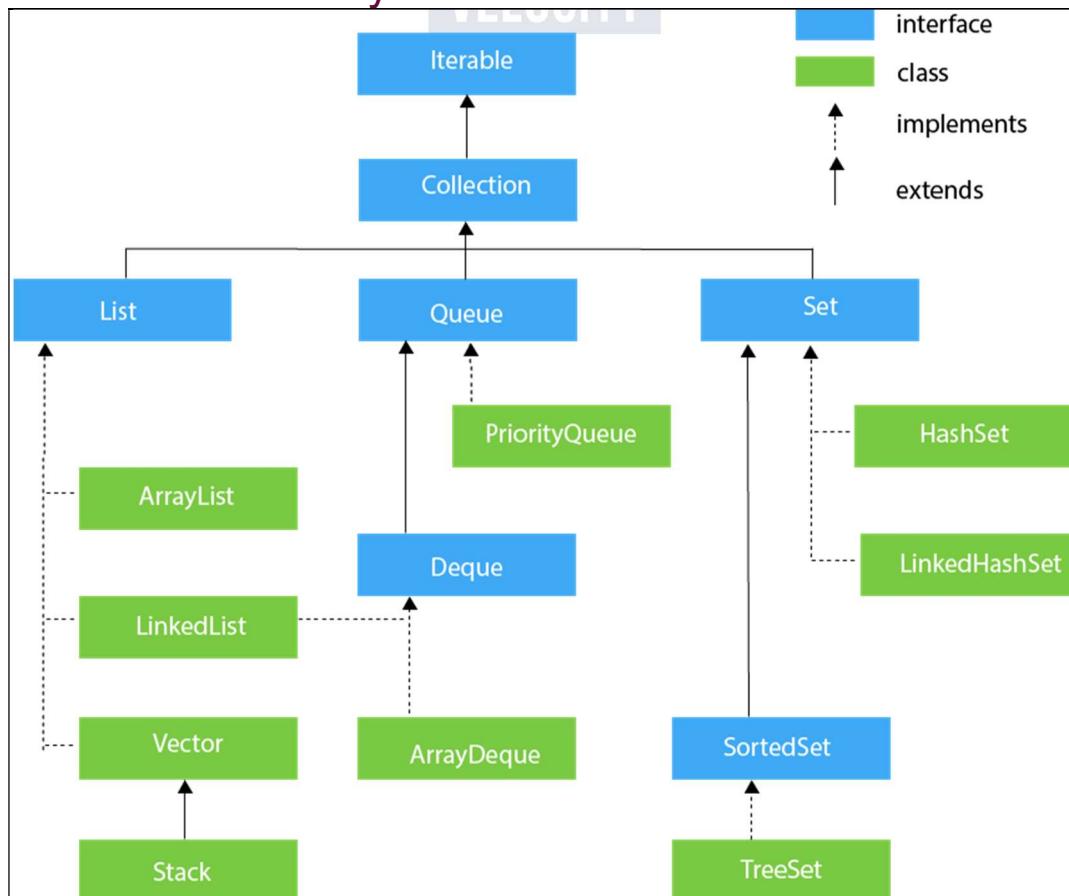
Cursor is used to fetch the data from above mentioned 7 implementation classes

Iterator: Iterator is Universal Cursor i.e. it can able to fetch data from all 7 classes.

ListIterator : can able fetch the data only from List interface (only 3 classes)

Enumeration: can fetch data only of legacy class vector.

Hierarchy of Collection Framework



List (ArrayList, Vector, LinkedList)	Set (HashSet, LinkedHashSet, TreeSet)
1. Duplicate : Allowed 2. Null Values : Allowed 3. Order of Insertion : Maintain 4. Storage Type : Index	1. Duplicate : Not Allowed 2. Null Values : Only one allowed(except treeset) 3. Order of Insertion : Random/ Maintained 4. Storage Type : Hashtable

Parameters	ArrayList	Vector	LinkedList
1. Duplicate	Allowed	Allowed	Allowed
2. Null Values	Allows Any No. of null values	Allows any no. of null values	Allows any no. of null values
3. Storage type:	Index	Index	Index
4. Order of insertion	Maintain(Sequential)	Maintain(Sequential)	Maintain(Sequential)
5. Default capacity	10	10	*No Default capacity in linkedlist
6. Incremental capacity	Current capacity * (3/2) + 1 = 16	Current capacity*2	--
7. Data structure	Resizable	Doubly	Linear
8. Best choice	Retrieval operation (random access interface is implemented in ArrayList & vector)	Retrieval operation (random access interface is implemented in ArrayList & Vector)	Manipulation operation i.e. insertion in between linkedlist or delete ()
9. Worst choice:	Manipulation operation i.e. insertion in between ArrayList or delete ()	Manipulation operation i.e. insertion in between Vector or delete ()	Retrieval operation (random access interface is not implemented)
10.		Vector is legacy class.	

Parameters	HashSet	LinkedHashSet	TreeSet (only Homogenous)
1. Duplicate	Doesn't allow	Doesn't allow	Doesn't allow
2. Null Values	Allow only 1 null value.	Allow only 1 null value	Not Allowed
3. Storage type:	Hashtable	Hashtable	Hashtable
4. Order of insertion	Random insertion	Maintained	Ascending order.
5. Default capacity	No default capacity	No default capacity	No default capacity
6. Incremental capacity	--	--	--
7. Data structure	Hashtable	Hybrid (linear+ Hashtable)	Balanced tree
8. Best choice	To remove duplicate elements when order of insertion is not mandatory.	To remove duplicate elements when order of insertion is mandatory	To remove duplicate elements when order of insertion is Ascending order.

Collection (I)**1. List (I)****2. Set (I)****3. Queue (I)****1. List (I)**

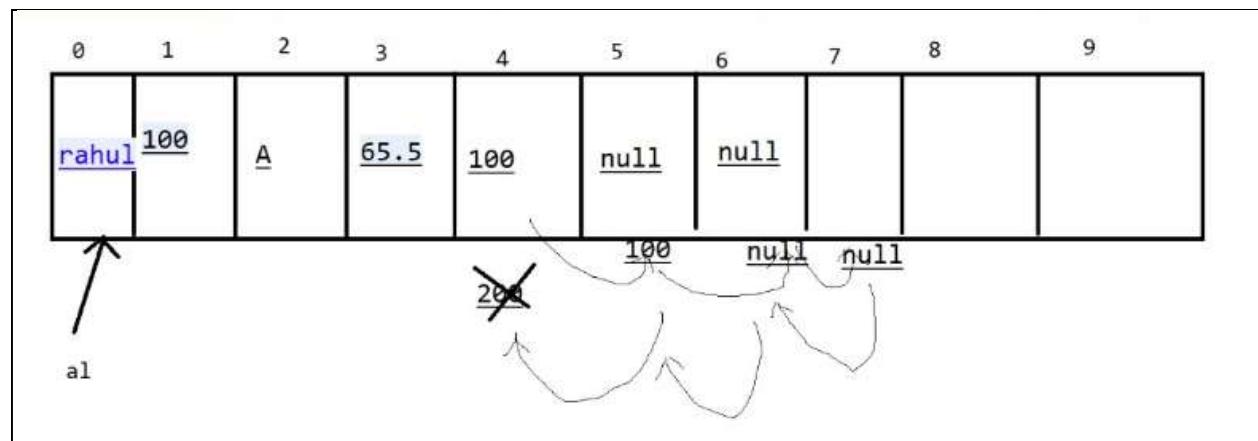
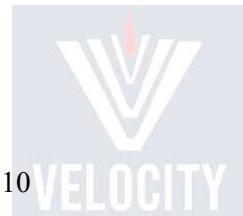
- 1. Duplicate are allowed in list**
- 2. Allows any no. of null values**
- 3. Storage Type: index**
- 4. Order of insertion-maintain**

List (I) further classifies as:

- 1.1 ArrayList(IC)**
- 1.2. Vector(IC) ---->legacy**
- 1.3. LinkedList(IC)**

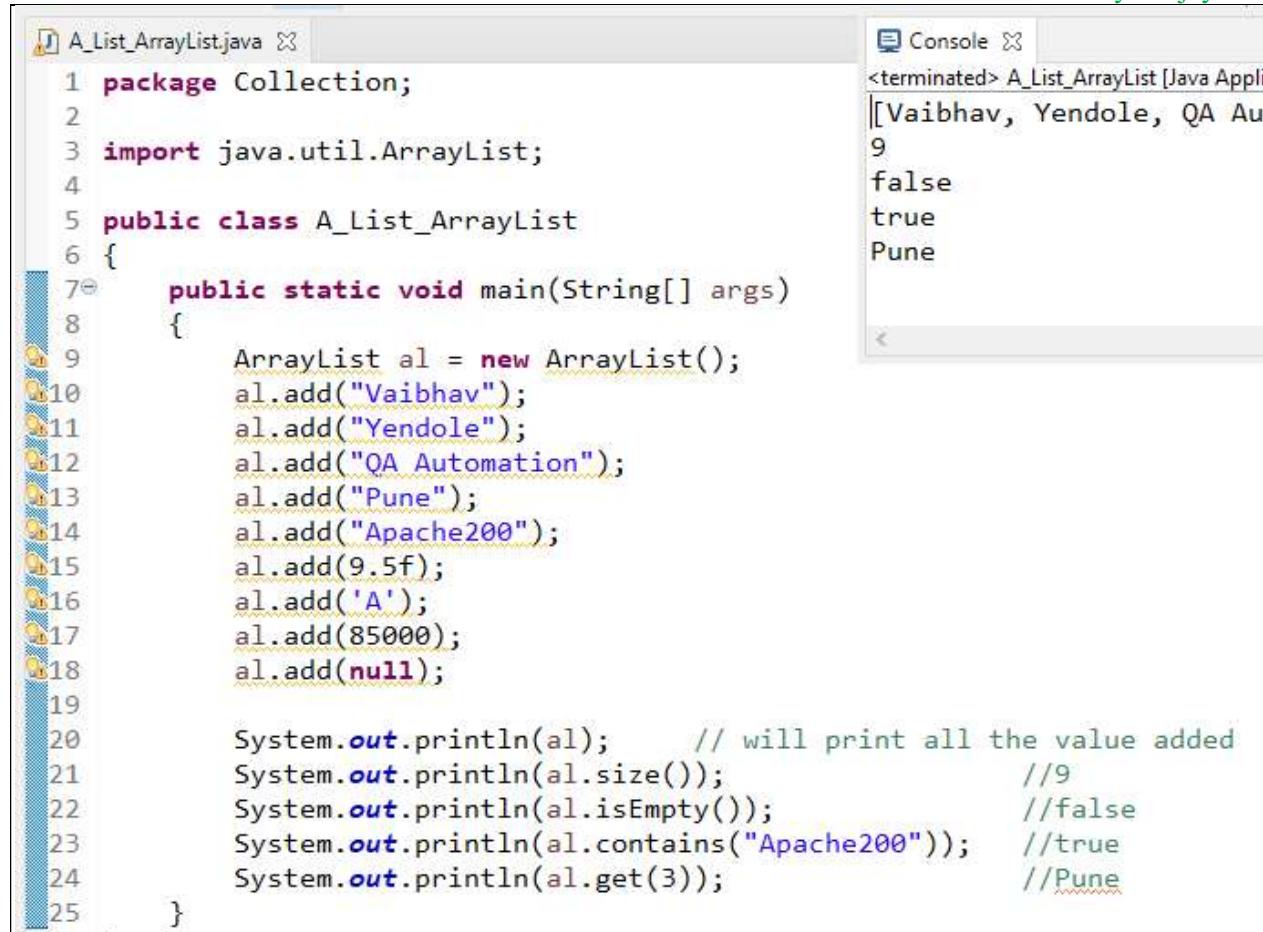
1.1 ArrayList (IC)

1. Duplicate values are allowed in ArrayList
2. Allows any no. of null values
3. Storage type: index
4. Order of insertion-maintain
5. Default capacity for ArrayList is 10
6. Incremental capacity = (current capacity * 3/2) + 1 = 16
7. Data structure: Resizable
8. Best choice: retrieval operation (random access interface is implemented in ArrayList & vector)
9. Worst choice: manipulation operation i.e. insertion in between ArrayList or delete ()



When insertion then perform Right Shift Operation

When Remove any data then performs Left Shift Operation.



The screenshot shows an IDE interface with two panes. The left pane displays the Java code for a class named A_List_ArrayList. The right pane shows the console output of the program's execution.

```

A_List_ArrayList.java
1 package Collection;
2
3 import java.util.ArrayList;
4
5 public class A_List_ArrayList
6 {
7     public static void main(String[] args)
8     {
9         ArrayList al = new ArrayList();
10        al.add("Vaibhav");
11        al.add("Yendole");
12        al.add("QA Automation");
13        al.add("Pune");
14        al.add("Apache200");
15        al.add(9.5f);
16        al.add('A');
17        al.add(85000);
18        al.add(null);
19
20        System.out.println(al);      // will print all the value added
21        System.out.println(al.size()); //9
22        System.out.println(al.isEmpty()); //false
23        System.out.println(al.contains("Apache200")); //true
24        System.out.println(al.get(3)); //Pune
25    }
}

```

Console Output:

```

<terminated> A_List_ArrayList [Java Application]
[Vaibhav, Yendole, QA Automation, Pune, Apache200, 9.5, null, true, false]

```



```

package Collection;

import java.util.ArrayList;

public class A_List_ArrayList
{
    public static void main(String[] args)
    {
        ArrayList al = new ArrayList();           // ArrayList(11)
        al.add("Vaibhav");
        al.add("Yendole");
        al.add("QA Automation");
        al.add("Pune");
        al.add("Apache200");
        al.add(9.5f);
        al.add('A');
        al.add(85000);
        al.add(null);

        System.out.println(al);      // will print all the value added
        System.out.println(al.size()); //9
        System.out.println(al.isEmpty()); //false
        System.out.println(al.contains("Apache200")); //true
        System.out.println(al.get(3)); //Pune
    }
}

```

The screenshot shows a Java IDE interface with two tabs: "A_List_ArrayList.java" and "A_List_ArrayList2.java". The code in "A_List_ArrayList2.java" demonstrates various ArrayList operations. The output window shows the execution of the code, displaying the state of the ArrayList at different points.

```

3 public class A_List_ArrayList2
4 {
5     public static void main(String[] args)
6     {
7         ArrayList al = new ArrayList();
8         al.add(100);
9         al.add(200);
10        al.add(300);
11        al.add(500);
12        al.add(600);
13        al.add(700);
14        al.add(800);
15        al.add(900);    System.out.println("-----ArrayList-----");
16        System.out.println(al);
17        System.out.println(al.size());
18
19        //Insert information in Between ArrayList //Right Shift Operation
20        System.out.println("-----Infromation Added-----");
21        al.add(3, 400);
22        System.out.println(al);
23        System.out.println(al.size());
24        //Output : [100, 200, 300, 400, 500, 600, 700, 800, 900]
25
26        //Remove or Delete Infromation Between ArrayList //Left Shift OPeration
27        al.remove(3);
28        System.out.println("----Remove Infromation from ArrayList-----");
29        System.out.println(al);
30        System.out.println(al.size());
31    }
32 }

```

```

package Collection;
import java.util.ArrayList;
public class A_List_ArrayList2
{
    public static void main(String[] args)
    {
        ArrayList al = new ArrayList();
        al.add(100);
        al.add(200);
        al.add(300);
        al.add(500);
        al.add(600);
        al.add(700);
        al.add(800);
        al.add(900); System.out.println("-----ArrayList-----");
        System.out.println(al);
        System.out.println(al.size());

        //Insert information in Between ArrayList //Right Shift Operation
        System.out.println("-----Infromation Added-----");
        al.add(3, 400);
        System.out.println(al);
        System.out.println(al.size());
        //Output : [100, 200, 300, 400, 500, 600, 700, 800, 900]

        //Remove or Delete Infromation Between ArrayList //Left Shift OPeration
        al.remove(3);
        System.out.println("---Remove Infromation from ArrayList-----");
        System.out.println(al);
        System.out.println(al.size());
    }
}

```

```

package Collection;
import java.util.ArrayList;
public class A_List_ArrayList3
{
    public static void main(String[] args)
    {
        ArrayList ar = new ArrayList();

        ar.add("Nairobi");
        ar.add("Tokyo");
        ar.add("Denver");
        ar.add("Berlin");
        ar.add("Nairobi");

        System.out.println(ar);
        System.out.println(ar.size()); //5

        System.out.println(ar.indexOf("Nairobi")); //0
        System.out.println(ar.lastIndexOf("Nairobi")); //4

        // Modify or Update information
        ar.set(4, "Helsinki");
        System.out.println("Upadted ar = "+ar);
    }
}

```

The screenshot shows an IDE interface with two tabs open: 'A_List_ArrayList.java' and 'A_List_ArrayList2.java'. The 'A_List_ArrayList.java' tab contains the provided Java code. The 'Console' tab shows the execution results:

```

[A_List_ArrayList.java] [A_List_ArrayList2.java] [A_List_ArrayList.java]
Console
<terminated> A_List_ArrayList3 [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe
[Nairobi, Tokyo, Denver, Berlin, Nairobi]
5
0
4
Upadted ar = [Nairobi, Tokyo, Denver, Berlin, Helsinki]

```

```

package Collection;
import java.util.ArrayList;
import java.util.ListIterator;
public class A_List_ArrayList4{
    public static void main(String[] args)
    {
        ArrayList ar = new ArrayList();
        ar.add("Nairobi");

```

```

ar.add("Tokyo");
ar.add("Denver");
ar.add("Berlin");
ar.add("Jarvis");
//1. Print ArrayList info using ListIterator cursor
System.out.println("---Print ArrayList info using ListIterator cursor---");

ListIterator Litr = ar.listIterator();

while(Litr.hasNext())

{
    System.out.println(Litr.next());
}

//2. Print ArrayList info using for loop cursor

System.out.println("---Print ArrayList info using for loop cursor---");

for(int i=0; i<=ar.size()-1; i++)
{
    System.out.println(ar.get(i));
}

//3. Print ArrayList info Using foreach loop
System.out.println("----Print ArrayList info Using foreach loop--");

for(Object s1:ar)

{
    System.out.println(s1);
}
}
}

```

The screenshot shows an IDE interface with multiple tabs open. The active tab contains the Java code for printing ArrayList elements. The code uses three different loops to print the same list of cities: Nairobi, Tokyo, Denver, Berlin, and Jarvis. The first two sections of the code are annotated with yellow arrows pointing to their respective comments. The third section is annotated with a red arrow pointing to its comment. The right side of the interface shows the 'Console' tab with the output of the program, which is a list of the five cities separated by new lines.

```

A_List_ArrayList.java A_List_ArrayList2.java A_List_ArrayList3.java A_List_ArrayList4.java
2* import java.util.ArrayList;
4 public class A_List_ArrayList4{
5*   public static void main(String[] args)
6{   ArrayList ar = new ArrayList();
7  ar.add("Nairobi");
8  ar.add("Tokyo");
9  ar.add("Denver");
10 ar.add("Berlin");
11 ar.add("Jarvis");
12 //1. Print ArrayList info using ListIterator cursor
13 System.out.println("---Print ArrayList info using ListIterator cursor---");
14 ListIterator Litr = ar.listIterator();
15 while(Litr.hasNext())
16 {
17     System.out.println(Litr.next());
18 }
19 //2. Print ArrayList info using for loop cursor
20 System.out.println("----Print ArrayList info using for loop cursor---");
21 for(int i=0; i<=ar.size()-1; i++)
22 {
23     System.out.println(ar.get(i));
24 }
25 //3. Print ArrayList info Using foreach loop
26 System.out.println("----Print ArrayList info Using foreach loop--");
27 for(Object s1:ar)
28 {
29     System.out.println(s1);
30 }
31 }
32 }

---Print ArrayList info
Nairobi using ListIterator
Tokyo cursor---
Denver
Berlin
Jarvis
----Print ArrayList info
Nairobi using for loop
Tokyo cursor---
Denver
Berlin
Jarvis
----Print ArrayList info
Nairobi Using foreach
Tokyo loop--
Denver
Berlin
Jarvis

```

1.2 Vector (IC)

1. Duplicate are allowed in vector
2. Allows any no. of null values
3. Storage type: index
4. Order of insertion-maintain
5. Default capacity for Vector is 10
- *6. Incremental capacity= current capacity*2
- *7. Data structure: doubly
8. Best choice: retrieval operation (random access interface is implemented in arraylist & vector)
9. Worst choice: manipulation operation i.e. insertion in between Vector or delete ()
- *10. Vector is legacy class.

The screenshot shows an IDE interface with two tabs open: 'A_List_Array...' and 'B_List_Vect...'. The 'B_List_Vect...' tab contains the following Java code:

```

1 package Collection;
2 import java.util.Vector;
3 public class B_List_Vector
4 {
5     public static void main(String[] args)
6     {
7         Vector v = new Vector();
8         v.add("Nairobi");
9         v.add("Tokyo");
10        v.add("Denver");
11        v.add("Berlin");
12        v.add("Jarvis");
13        v.add("Nairobi");
14
15        System.out.println(v);
16        System.out.println(v.size());
17        System.out.println(v.isEmpty());
18        System.out.println(v.contains("Tokyo"));
19        System.out.println(v.get(3));
20
21        // Data Manipulation
22        v.add(4, "Helsinki"); //Right Shift
23        System.out.println(v);
24        v.remove(4); //Left Shift
25        System.out.println(v);
26
27        // Data Updation
28        v.set(4, "JarvisPC");
29        System.out.println(v);
30    }
31 }

```

The right side of the interface shows the 'Console' tab with the output of the program:

```

<terminated> B_List_Vector [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (Oct 16)
[Nairobi, Tokyo, Denver, Berlin, Jarvis, Nairobi]
6
false
true
Berlin
[Nairobi, Tokyo, Denver, Berlin, Helsinki, Jarvis, Nairobi]
[Nairobi, Tokyo, Denver, Berlin, Jarvis, Nairobi]
[Nairobi, Tokyo, Denver, Berlin, JarvisPC, Nairobi]

```

The screenshot shows an IDE interface with two tabs: A_List_Array... and B_List_Vect... The code in the B tab prints elements of a Vector named 'v' using two different cursor methods.

```

1 package Collection;
2 import java.util.Iterator;
3 import java.util.ListIterator;
4 import java.util.Vector;
5 public class B_List_Vector2
6 {
7     public static void main(String[] args)
8     { Vector v = new Vector();
9         v.add("Nairobi");
10        v.add("Tokyo");
11        v.add("Denver");
12        v.add("Berlin");
13        v.add("Jarvis");
14        v.add("Nairobi");
15
16        System.out.println("--Print Vector Using Iterator Cursor--");
17        Iterator itr = v.iterator();
18        while(itr.hasNext())
19        {
20            System.out.println(itr.next());
21        }
22
23        System.out.println("--Print Vector Using ListIterator Cursor--");
24        ListIterator litr = v.listIterator();
25        while(litr.hasNext())
26        {
27            System.out.println(litr.next());
28        }
29    }
30 }
31

```

The output window shows two distinct outputs:

- Print Vector Using Iterator Cursor--
Nairobi
Tokyo
Denver
Berlin
Jarvis
Nairobi
- Print Vector Using ListIterator Cursor--
Nairobi
Tokyo
Denver
Berlin
Jarvis
Nairobi

The screenshot shows an IDE interface with two tabs: A_List_Array... and B_List_Vect... The code in the B tab prints elements of a Vector named 'v' using three different iteration methods.

```

1 package Collection;
2 import java.util.Enumeration;
3 import java.util.Vector;
4 public class B_List_Vector3
5 {
6     public static void main(String[] args)
7     { Vector v = new Vector();
8         v.add("Nairobi");
9         v.add("Tokyo");
10        v.add("Denver");
11        v.add("Berlin");
12        v.add("Jarvis");
13        v.add("Nairobi");
14
15        System.out.println("--Print Vector Using For Loop--");
16        for(int i = 0; i < v.size() - 1; i++)
17        {
18            System.out.println(v.get(i));
19        }
20
21        System.out.println("--Print Vector Using For eachLoop--");
22        for(Object s1: v)
23        {
24            System.out.println(s1);
25        }
26
27        System.out.println("--Print Vector Using enumeration cursor--");
28        Enumeration enu = v.elements();
29        while(enu.hasMoreElements())
30        {
31            System.out.println(enu.nextElement());
32        }
33    }
34 }
35

```

The output window shows three distinct outputs:

- Print Vector Using For Loop--
Nairobi
Tokyo
Denver
Berlin
Jarvis
Nairobi
- Print Vector Using For eachLoop--
Nairobi
Tokyo
Denver
Berlin
Jarvis
Nairobi
- Print Vector Using enumeration cursor--
Nairobi
Tokyo
Denver
Berlin
Jarvis
Nairobi

1.3 LinkedList

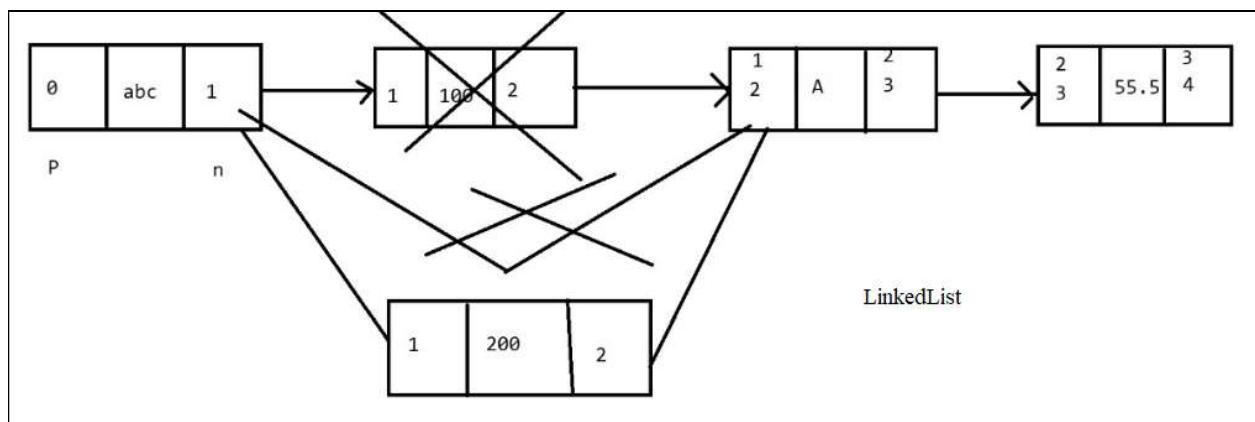
1. Duplicate are allowed in ArrayList
2. Allows any no. of null values
3. Order of insertion-maintain
4. Storage type: index

***5. No Default capacity in linkedlist**

***6. Data structure: linear**

***7. Best choice: manipulation operation i.e. insertion in between linkedlist or delete ()**

***8. Worst choice: retrieval operation (random access interface is not implemented)**



```

A_List_ArrayList.java   A_List_ArrayList2.java   A_List_ArrayList3.java   A_List_ArrayList4.java   B_L  Console  X
9     LinkedList ll = new LinkedList();
10    ll.add("Nairobi");
11    ll.add("Tokyo");
12    ll.add("Denver");
13    ll.add("Berlin");
14    ll.add("Jarvis");
15    ll.add("Nairobi");
16
17    System.out.println("---Print LinkedList Using Iterator Cursor---");
18    Iterator itr = ll.iterator();
19    while(itr.hasNext())
20    {
21        System.out.println(itr.next());
22    }
23    System.out.println("--Print LinkedList Using ListIterator--");
24    ListIterator litr = ll.listIterator();
25    while(litr.hasNext())
26    {
27        System.out.println(litr.next());
28    }
29    System.out.println("--Print LinkedList Using for Loop--");
30    for(int i=0; i<ll.size()-1; i++)
31    {
32        System.out.println(ll.get(i));
33    }
34
35    System.out.println("--Print LinkedList Using foreach Loop--");
36    for(Object s1:ll)
37    {
38        System.out.println(s1);
39    }

```

<terminated> C_List_LinkedList [Java Application] C:\Program Files\J...
Nairobi
Tokyo
Denver
Berlin
Jarvis
Nairobi
--Print LinkedList Using ListIterator--
Nairobi
Tokyo
Denver
Berlin
Jarvis
Nairobi
--Print LinkedList Using for Loop--
Nairobi
Tokyo
Denver
Berlin
Jarvis
Nairobi
--Print LinkedList Using foreach Loop--
Nairobi
Tokyo
Denver
Berlin
Jarvis
Nairobi

```

package Collection;
import java.util.Iterator;
import java.util.LinkedList;
import java.util.ListIterator;
public class C_List_LinkedList
{
    public static void main(String[] args)
    {
        LinkedList ll = new LinkedList();
        ll.add("Nairobi");
        ll.add("Tokyo");
        ll.add("Denver");
        ll.add("Berlin");
        ll.add("Jarvis");
        ll.add("Nairobi");

        System.out.println("--Print LinkedList Using Iterator Cursor--");
        Iterator itr = ll.iterator();
        while(itr.hasNext())
        {
            System.out.println(itr.next());
        }
        System.out.println("--Print LinkedList Using ListIterator--");
        ListIterator litr = ll.listIterator();
        while(litr.hasNext())
        {
            System.out.println(litr.next());
        }
        System.out.println("--Print LinkedList Using for Loop--");
        for(int i=0; i<ll.size()-1; i++)
        {
            System.out.println(ll.get(i));
        }

        System.out.println("--Print LinkedList Using foreach Loop--");
        for(Object s1:ll)
        {
            System.out.println(s1);
        }
    }
}

```

```

package Collections;
import java.util.LinkedList;
import java.util.Iterator;
import java.util.LinkedList;
import java.util.ListIterator;

public class example3_Listlist {
    public static void main(String[] args) {
        LinkedList ll=new LinkedList();
        ll.add("rahul");
        ll.add(100);
        ll.add('A');
        ll.add(65.5f);
    }
}

```

```

ll.add(100);
ll.add(null);
ll.add(null);
System.out.println("-----print LinkedList info using iterator cursor-----");

```

```
Iterator itr = ll.iterator();
```

```

while(itr.hasNext()) //true true fllse
{
    System.out.println(itr.next());
}

```

```
System.out.println("-----print LinkedList info using ListIterator cursor-----");
```

```
ListIterator litr = ll.listIterator();
```

```

while(litr.hasNext())
{
    System.out.println(litr.next());
}

```

```
System.out.println("-----print LinkedList info using for loop-----");
```

```

for(int i=0; i<=ll.size()-1; i++)
{
    System.out.println(ll.get(i));
}

```

```
System.out.println("-----print LinkedList info using foreach loop-----");
```

```

for(Object s1:ll)
{
    System.out.println(s1);
}
}
}

```



The screenshot shows a Java code editor and a terminal window. The code editor displays a file named C_List_LinkedList3.java with the following content:

```

1 package Collection;
2
3 import java.util.LinkedList;
4
5 public class C_List_LinkedList3 {
6     public static void main(String[] args) {
7
8         LinkedList ll = new LinkedList();
9
10        ll.add("Vaibhav");
11        ll.add("Yendole");
12        ll.add(85.86f);
13        ll.add('A');
14        ll.add(9.5f);
15        ll.add("Apche200");
16
17        System.out.println(ll);
18        System.out.println(ll.size());
19        ll.add(3, 'B');
20        System.out.println(ll);
21        ll.remove(3);
22        System.out.println(ll);
23        ll.set(2, 95.86f);
24        System.out.println(ll);
25    }
26
27 }

```

The terminal window to the right shows the output of the program:

```

<terminated> C_List_LinkedList3 [Java Application] C:\Program Files\Java\jdk-16
[Vaibhav, Yendole, 85.86, A, 9.5, Apche200]
6
[Vaibhav, Yendole, 85.86, B, A, 9.5, Apche200]
[Vaibhav, Yendole, 85.86, A, 9.5, Apche200]
[Vaibhav, Yendole, 95.86, A, 9.5, Apche200]

```

2. Set (I)

1. Doesn't allow duplicate
2. Allow only 1 null value (except TreeSet)
3. Order of insertion-random/ascending/maintain insertion
4. Storage type- Hashtable

Set Interface Consist of:

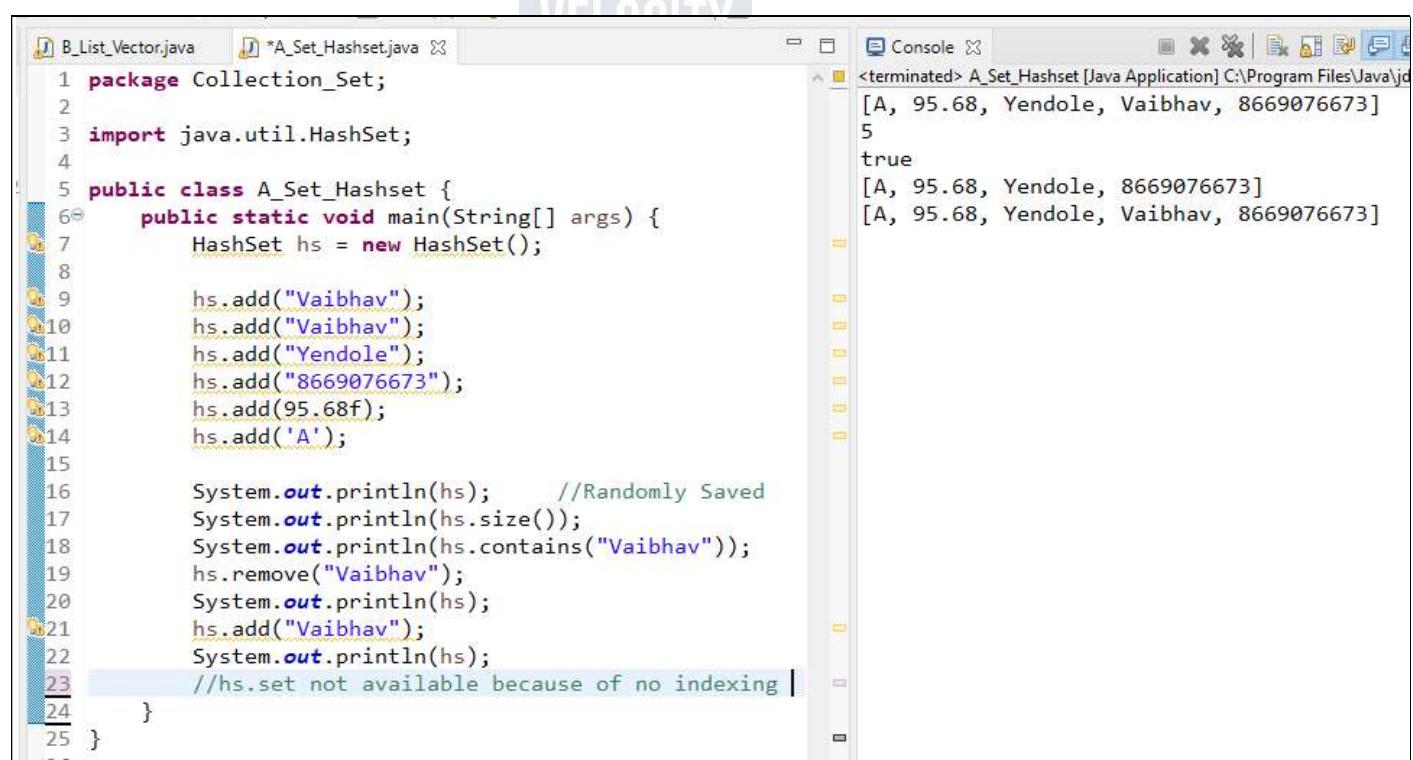
2.1. HashSet(IC)

2.2. LinkedHashSet(IC)

2.3. TreeSet(IC)

2.1. HashSet:

1. Doesn't allow duplicate values
2. Allow only 1 null value.
3. Order of insertion->random insertion
4. Storage type: Hashtable
5. No default capacity
6. DS: Hashtable
7. Best choice: To remove duplicate elements when order of insertion is not mandatory.



The screenshot shows an IDE interface with two files open: `B_List_Vector.java` and `*A_Set_Hashset.java`. The `A_Set_Hashset.java` file contains the following code:

```

1 package Collection_Set;
2
3 import java.util.HashSet;
4
5 public class A_Set_Hashset {
6     public static void main(String[] args) {
7         HashSet hs = new HashSet();
8
9         hs.add("Vaibhav");
10        hs.add("Vaibhav");
11        hs.add("Yendole");
12        hs.add("8669076673");
13        hs.add(95.68f);
14        hs.add('A');
15
16        System.out.println(hs);      //Randomly Saved
17        System.out.println(hs.size());
18        System.out.println(hs.contains("Vaibhav"));
19        hs.remove("Vaibhav");
20        System.out.println(hs);
21        hs.add("Vaibhav");
22        System.out.println(hs);
23        //hs.set not available because of no indexing |
24    }
25 }

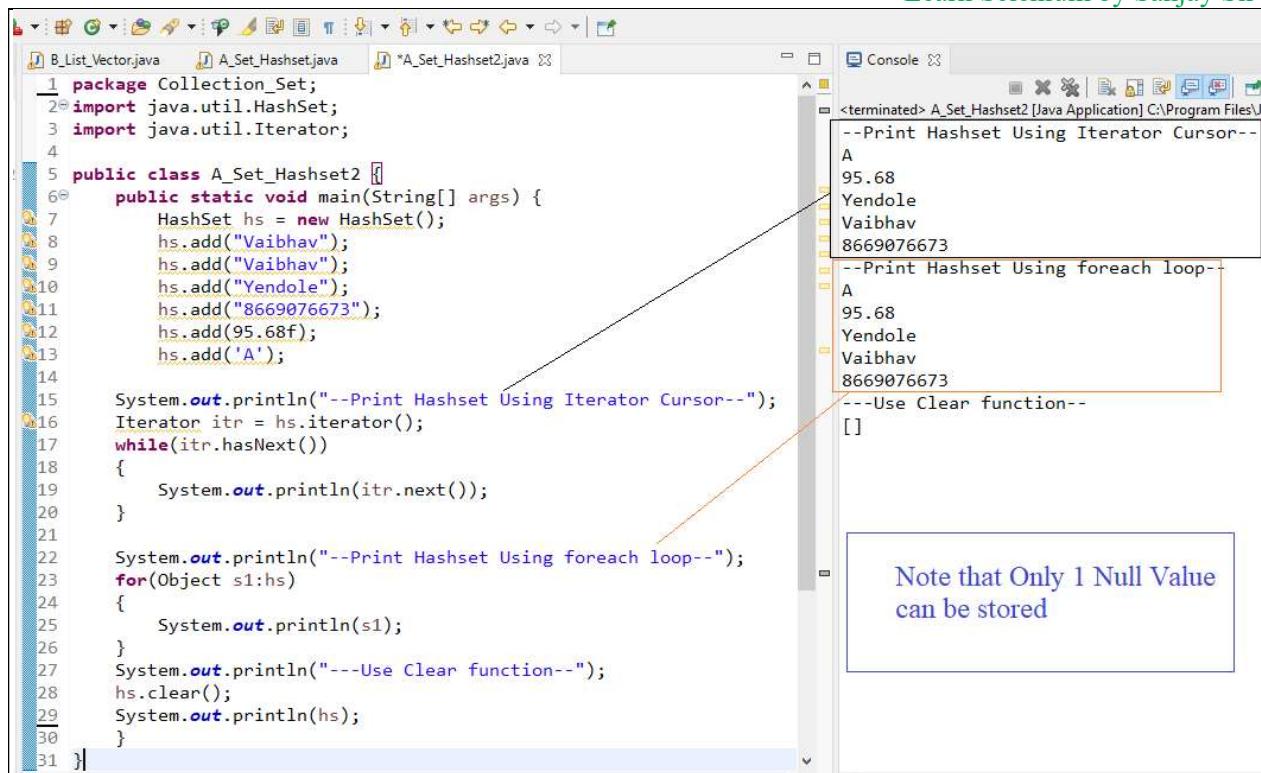
```

The `Console` tab shows the following output:

```

<terminated> A_Set_Hashset [Java Application] C:\Program Files\Java\jde
[A, 95.68, Yendole, Vaibhav, 8669076673]
5
true
[A, 95.68, Yendole, 8669076673]
[A, 95.68, Yendole, Vaibhav, 8669076673]

```



The screenshot shows the Eclipse IDE interface. On the left, the code editor displays `A_Set_Hashset2.java` with the following content:

```

1 package Collection_Set;
2 import java.util.HashSet;
3 import java.util.Iterator;
4
5 public class A_Set_Hashset2 {
6     public static void main(String[] args) {
7         HashSet hs = new HashSet();
8         hs.add("Vaibhav");
9         hs.add("Vaibhav");
10        hs.add("Yendole");
11        hs.add("8669076673");
12        hs.add(95.68f);
13        hs.add('A');
14
15        System.out.println("--Print HashSet Using Iterator Cursor--");
16        Iterator itr = hs.iterator();
17        while(itr.hasNext())
18        {
19            System.out.println(itr.next());
20        }
21
22        System.out.println("--Print HashSet Using foreach loop--");
23        for(Object s1:hs)
24        {
25            System.out.println(s1);
26        }
27        System.out.println("---Use Clear function--");
28        hs.clear();
29        System.out.println(hs);
30    }
31 }

```

The right side shows the `Console` tab with the following output:

```

<terminated> A_Set_Hashset2 [Java Application] C:\Program Files\

--Print HashSet Using Iterator Cursor--
A
95.68
Yendole
Vaibhav
8669076673

--Print HashSet Using foreach loop--
A
95.68
Yendole
Vaibhav
8669076673

---Use Clear function--
[]

```

A callout box points from the text "Note that Only 1 Null Value can be stored" to the empty array output "[]".

**Note that Only 1 Null Value
can be stored**



2.2. LinkedHashSet:

1. Doesn't allow duplicate values
 2. Allow only 1 null value.
 - *3. Order of insertion->maintained
 4. No default capacity
 - *5. DS: Hybrid (Linear+ Hashtable)
 6. Storage type: Hashtable
- 7.*Best choice: To remove duplicate elements when order of insertion is mandatory

The screenshot shows an IDE interface with several tabs at the top: B_List_Vector.java, A_Set_Hashset.java, A_Set_Hashset2.java, and *B_Set_LinkedHashset.java. The code in *B_Set_LinkedHashset.java is as follows:

```

1 package Collection_Set;
2 import java.util.Iterator;
3 import java.util.LinkedHashSet;
4 public class B_Set_LinkedHashSet {
5     public static void main(String[] args) {
6         LinkedHashSet lhs = new LinkedHashSet();
7         lhs.add("Vaibhav");
8         lhs.add("Vaibhav");
9         lhs.add("Yendole");
10        lhs.add('A');
11        lhs.add(98.59f);
12        lhs.add(111);
13        lhs.add(null);
14        lhs.add(null);
15
16        System.out.println(lhs);      // Maintain the Sequence
17        System.out.println(lhs.size()); // 6 . Duplicate not allowed
18
19        System.out.println("--print info in LinkedHashSet using iterator cursor-");
20        Iterator itr = lhs.iterator();
21        while(itr.hasNext())
22        {
23            System.out.println(itr.next());
24        }
25
26        System.out.println("--print info in LinkedHashSet using foreach loop-");
27        for(Object s1:lhs)
28        {
29            System.out.println(s1);
30        }
31    }
}

```

To the right of the code, there are two console windows. The top window shows the output of the iterator method:

```

<terminated> B_Set_LinkedHashSet [Java Application] C:\Program File
[Vaibhav, Yendole, A, 98.59, 111, null]
6
--print info in LinkedHashSet using
Vaibhav           iterator cursor-
Yendole
A
98.59
111
null

```

The bottom window shows the output of the foreach loop:

```

--print info in LinkedHashSet
Vaibhav   using foreach loop-
Yendole
A
98.59
111
null

```

```

package Collection_Set;

import java.util.Iterator;
import java.util.LinkedHashSet;
public class B_Set_LinkedHashset {
    public static void main(String[] args) {
        LinkedHashSet lhs = new LinkedHashSet();
        lhs.add("Vaibhav");
        lhs.add("Vaibhav");
        lhs.add("Yendole");
        lhs.add('A');
        lhs.add(98.59f);
        lhs.add(111);
        lhs.add(null);
        lhs.add(null);

        System.out.println(lhs); // Maintain the Sequence
        System.out.println(lhs.size()); // 6 . Duplicate not allowed

        System.out.println("--print info in LinkedHashSet using
iterator cursor-");
        Iterator itr = lhs.iterator();
        while(itr.hasNext())
        {
            System.out.println(itr.next());
        }

        System.out.println("--print info in LinkedHashSet using
foreach loop-");
        for(Object s1:lhs)
        {
            System.out.println(s1);
        }
        System.out.println(lhs.contains("Vaibhav")); //true
        lhs.remove("Yendole");
        System.out.println(lhs);
        lhs.clear();
        System.out.println(lhs.size());
        System.out.println(lhs);
    }
}

```

2.3. TreeSet:

1. Doesn't allow duplicate
- *2. Null values: not allowed
- *3. Order of insertion- Ascending order.
4. No default capacity.
- *5. DS: Balanced tree
6. Storage type: Hashtable
7. Best Choice: To remove duplicate elements when order of insertion is Ascending order.

Note: we can store only homogeneous data

The screenshot shows an IDE interface with several tabs at the top: B_List_Vector.java, A_Set_Hashset.java, A_Set_Hashset2.java, and B_Set_LinkedHashSet.java. The active tab is C_Set_TreeSet.java. The code in this tab is as follows:

```

1 package Collection_Set;
2 import java.util.TreeSet;
3 public class C_Set_TreeSet {
4     public static void main(String[] args) {
5         TreeSet Tset = new TreeSet();
6         Tset.add("ABC");
7         Tset.add("BCD");
8         Tset.add("CDE");
9         Tset.add("DEF");
10        Tset.add("EFG");
11        Tset.add("FGH");
12        Tset.add("AAA");
13        //Tset.add(9857);    // Stores only homogeneous Value
14        //Tset.add(null);   // Doesn't Accept null Value
15
16        System.out.println(Tset); //Arrange in Ascending order
17        System.out.println(Tset.size()); //
18
19        Tset.pollFirst(); // Remove First Object
20        System.out.println(Tset);
21
22        Tset.pollLast(); // Remove last object
23        System.out.println(Tset);
24
25        Tset.remove("DEF");
26        System.out.println(Tset);
27
28        System.out.println(Tset.first()); //Display info at start position
29        System.out.println(Tset.last()); //Display info at Last position
30    }
31 }

```

To the right of the code editor is a "Console" window titled "<terminated> C_Set_TreeSet [Java Application] C:\Program Files\Java". The console output is:

```

[AAA, ABC, BCD, CDE, DEF, EFG, FGH]
7
[ABC, BCD, CDE, DEF, EFG]
[ABC, BCD, CDE, DEF, EFG]
[ABC, BCD, CDE, EFG]
ABC
EFG

```

```

package Collection_Set;
import java.util.Iterator;
import java.util.TreeSet;
public class C_Set_TreeSet2 {
public static void main(String[] args) {
    TreeSet Tset = new TreeSet();
    Tset.add("ABC");
    Tset.add("BCD");
    Tset.add("CDE");
    Tset.add("DEF");
    Tset.add("EFG");
    Tset.add("FGH");
    Tset.add("AAA");

    System.out.println("--Print info in TreeSet Using Iterator Cursor--");
}

```

```

Iterator itr = Tset.iterator();
while(itr.hasNext())
{
    System.out.println(itr.next());
}
System.out.println("--Print info in TreeSet Using foreach Cursor--");
for(Object s1:Tset)
{
    System.out.println(s1);
}}}

```

The screenshot shows an IDE interface with several tabs at the top: B_List_Vector.java, A_Set_Hashset.java, A_Set_Hashset2.java, B_Set_LinkedHashset.java, C_Set_TreeSet.java, and Console.

The code in C_Set_TreeSet2.java is as follows:

```

1 package Collection_Set;
2 import java.util.Iterator;
3 import java.util.TreeSet;
4 public class C_Set_TreeSet2 {
5     public static void main(String[] args) {
6         TreeSet Tset = new TreeSet();
7         Tset.add("ABC");
8         Tset.add("BCD");
9         Tset.add("CDE");
10        Tset.add("DEF");
11        Tset.add("EFG");
12        Tset.add("FGH");
13        Tset.add("AAA");
14
15        System.out.println("--Print info in TreeSet Using Iterator Cursor--");
16        Iterator itr = Tset.iterator();
17        while(itr.hasNext())
18        {
19            System.out.println(itr.next());
20        }
21
22        System.out.println("--Print info in TreeSet Using foreach Cursor--");
23        for(Object s1:Tset)
24        {
25            System.out.println(s1);
26        }
27    }
28 }

```

The right pane shows the output of the program in the "Console" tab:

```

<terminated> C_Set_TreeSet2 [Java Application] C:\P
--Print info in TreeSet
AAA   Using Iterator Cursor--
ABC
BCD
CDE
DEF
EFG
FGH
--Print info in TreeSet
AAA   Using foreach Cursor--
ABC
BCD
CDE
DEF
EFG
FGH

```

The screenshot shows an IDE interface with several tabs at the top: B_List_Vector.java, A_Set_Hashset.java, A_Set_Hashset2.java, B_Set_LinkedHashset.java, C_Set_TreeSet3.java, and Console.

The code in C_Set_TreeSet3.java is as follows:

```

1 package Collection_Set;
2 import java.util.ArrayList;
3 import java.util.TreeSet;
4 public class C_Set_TreeSet3 {
5     public static void main(String[] args) {
6
7         ArrayList<String> al = new ArrayList<String>();
8         al.add("Yendole");
9         al.add("Vaibhav");
10        al.add("Apache200");
11        //al.add(200);only String allowed
12        System.out.println(al); // Sort result sequential
13
14        TreeSet<Integer> ts = new TreeSet<Integer>();
15        ts.add(50);
16        ts.add(10);
17        System.out.println(ts); // Sort the result ascending
18    }
19

```

The right pane shows the output of the program in the "Console" tab:

```

<terminated> C_Set_TreeSet3 [Java Application] C:\P
[Yendole, Vaibhav, Apache200]
[10, 50]

```

Cursors in Collections

Cursor is used to fetch the data from all implementation classes

Functions of Cursors

- read/get
- remove
- replace
- insert new object

1. Iterator

- All the collection object/implementation classes (7) --> **Universal cursor.**
- Using Enumeration and Iterator we can traverse collection object only in forward direction not a backward --> **Single directional cursor**
- By using iterator we can perform **only read and remove** operation we **cannot** perform replace and addition of new object.

2. ListIterator

- Only applicable for list interface type implementation classes (3) --> **not universal cursor.**
- Using list Iterator we can traverse a List in forward direction and backward direction--> **bidirectional cursor**
- By using ListIterator we can perform read, remove, replace and addition of new object operations.

3. Enumeration

- Only applicable for legacy classes (1) ----> **not universal cursor.**
- Using Enumeration and Iterator we can traverse collection object only in forward direction not a backward --> **Single directional cursor**
- By using enumeration we can get only read access.

Parameter		Iterator	ListIterator	Enumeration
1	Applicable for	All IC Universal cursor.	List IC ArrayList, LinkedList, Vector	Only Legacy Vector Not universal cursor.
2	Traverse List	Single Direction Forward	Bidirectional	Single Direction Forward
3	Functions	Only Read and Remove (2)	Read, Remove, Replace, Addition (4)	Only Read (1)

Q. How to Handle Listbox Using Selenium

Step.1

Identify the element which needs to be handling and store it in reference variable

Data type if this Variable is WebElement.

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

Step.2

Create an object of select class which accept WebElement argument

```
Select s = new Select(month);
```

Step.3

Use select class methods to select desired options

1. SelectByIndex (int index)	s.selectByIndex(9);(Counting is difficult)
2. SelectByValue (String Value)	s.selectByValue("12");
3. SelectByVisibleText (StringText)	s.selectByVisibleText("Oct");

List Methods Present In Select Class.	Return Type
1. SelectByIndex()	void
2. SelectByValue()	void
3. SelectByVisibleText()	void
4. DeselectByIndex()	void
5. DeselectByValue()	void
6. DeselectByVisibleText()	void
7. DeselectAll()	void
8. isMultiple()	Boolean
<ul style="list-style-type: none"> • This method is used to verify list Box is single selected or multiselected. • If it is single selected then return false otherwise true. 	
9. getAllSelectedOptions()	List<WebElement>
<ul style="list-style-type: none"> • This method is used to get only selected options present in the listBox. 	
10. getFirstSelectedOption()	WebElement
<ul style="list-style-type: none"> • This method is used to display first selected option in multiselected listBox. 	
11. getOptions()	List<WebElement>
<ul style="list-style-type: none"> • This method is used to get all the option present in listBox. 	

Q. Write a Script to Insert Year in Facebook signup page

1. SelectByIndex()	void
2. SelectByValue()	void
3. SelectByVisibleText()	void

```

package ListBox;
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.support.ui.Select;

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

    WebDriver driver = new ChromeDriver();
    driver.get("https://en-gb.facebook.com/");

    Thread.sleep(5000);
driver.findElement(By.xpath("a[text()='Create New
Account']]")).click();

//Step.1 Identify ListBox, Find WebElement and Store in reference.
WebElement month =
driver.findElement(By.xpath("//select[@id='month']"));

//Step.2 Create and Object of Select class which accept webelement
args.
Select s = new Select(month);

//Step.3 Use Select class Method to select option
//A) selectByIndex(int value)
//   s.selectByIndex(9);

//B) selectByVisibleText("StringText")
//   s.selectByVisibleText("Oct");

//C) selectByValue("String Value")
s.selectByValue("12");
}

}-----

```

Q. Write a Script to select multiple options in Listbox.

```

package ListBox;
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.support.ui.Select;

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

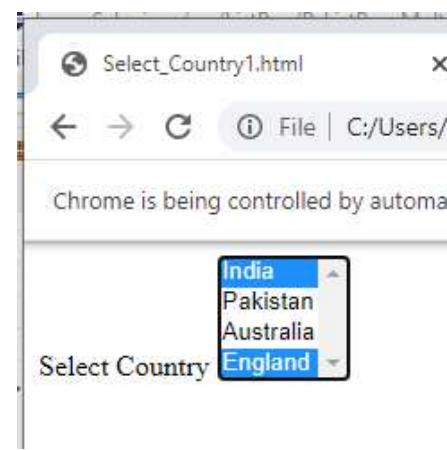
        driver.get("file:///C:/Users/Jordan/Desktop/Velocity%20Testing1/HTML>Select_Country1.html");

        //Step.1 : Identify the webElement which need to be handling and store it in reference variable
        WebElement country =
            driver.findElement(By.xpath("//select[@id='1234']"));

        //Step.2: Create an object of select class which accept WebElement Argument
        Select sm = new Select(country);

        //Step.3 : Use select class method to select options
        sm.selectByIndex(0);
        sm.selectByVisibleText("England");
    }
}

```



Q. Write a Script to Verify Option is multiselectable or not.

8. isMultiple()	Boolean
------------------------	----------------

```

package ListBox;
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.support.ui.Select;
public class D_ListBox_IsMultiSelectable {
    public static void main(String[] args) throws
InterruptedException {
        System.setProperty("webdriver.chrome.driver",
        "H:\\Selenium\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://en-gb.facebook.com/");
    }
}

```

```

        driver.findElement(By.xpath("//a[text()='Create New
Account']")).click();

        Thread.sleep(6000);
        //Step.1
        WebElement day =
driver.findElement(By.xpath("//select[@title='Day']"));

        Thread.sleep(3000);
        //Step.2
        Select dd = new Select(day);

        // Use of isMultiple();
        Thread.sleep(3000);
        boolean result = dd.isMultiple();
        System.out.println(result);           //false
        if(result==true)      // WebElement is Not MultiSelectable
        {
            System.out.println("WebElement is MultiSelectable");
        }
        else
        {
            System.out.println("WebElement is Not
MultiSelectable");
        }
    }
}

```



The screenshot shows a Java IDE interface with two tabs open: D_ListBox_IsMultiSelectable.java and *D_ListBox_IsMultiSelectable1.java. The code in D_ListBox_IsMultiSelectable.java is as follows:

```

1 package ListBox;
2 import org.openqa.selenium.By;
3 import org.openqa.selenium.WebDriver;
4 import org.openqa.selenium.WebElement;
5 import org.openqa.selenium.chrome.ChromeDriver;
6 import org.openqa.selenium.support.ui.Select;
7 public class D_ListBox_IsMultiSelectable1 {
8     public static void main(String[] args) throws InterruptedException {
9         System.setProperty("webdriver.chrome.driver", "H:\\Selenium\\chromedriver.exe");
10        WebDriver driver = new ChromeDriver();
11        driver.get("file:///C:/Users/Jordan/Desktop/Velocity%20Testing1/HTML>Select_Coun
12        Thread.sleep(2000);
13        //Step.1
14        WebElement country = driver.findElement(By.xpath("//select[@id='1234']"));
15
16        Thread.sleep(2000);
17        //Step.2
18        Select cc = new Select(country);
19
20        Thread.sleep(2000);
21        boolean result = cc.isMultiple();
22        System.out.println(result);
23
24        if(result==true)
25        {
26            System.out.println("Option is multiSelectable");
27        }
28        else
29        {
30            System.out.println("Option is Not multiSelectable");
31        } } }

```

The right side of the interface shows the Eclipse IDE's integrated terminal window. The output is:

```

<terminated> D_ListBox_IsMultiSelectable1 [Java Application] C:\Program Files\Java\jdk-11.0.11\bin\java.exe -Dfile.encoding=UTF-8 -jar C:\Program Files\Java\jdk-11.0.11\bin\chromedriver.exe
Starting ChromeDriver 94.0.4606.41 (333e85df3c9b656d3b32a0c3d00a8b5) on port 5473
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations#Network-Isolation
ChromeDriver was started successfully.
[1634815991.937][WARNING]: This version of ChromeDriver is old.
Oct 21, 2021 5:03:13 PM org.openqa.selenium.remote
INFO: Detected dialect: W3C
true
Option is multiSelectable

```

4. DeselectByIndex()	void
5. DeselectByValue()	void
6. DeselectByVisibleText()	void

```

package ListBox;
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.support.ui.Select;

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

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

    driver.get("file:///C:/Users/Jordan/Desktop/Velocity%20Testing1/HTML>Select_Country1.html");

    Thread.sleep(3000);
    //Step.1
    WebElement country =
driver.findElement(By.xpath("//select[@id='1234']"));

    //Step.2
    Select cc = new Select(country);

    Thread.sleep(2000);
    //Step.3
    cc.selectByIndex(0);
    cc.selectByVisibleText("England");

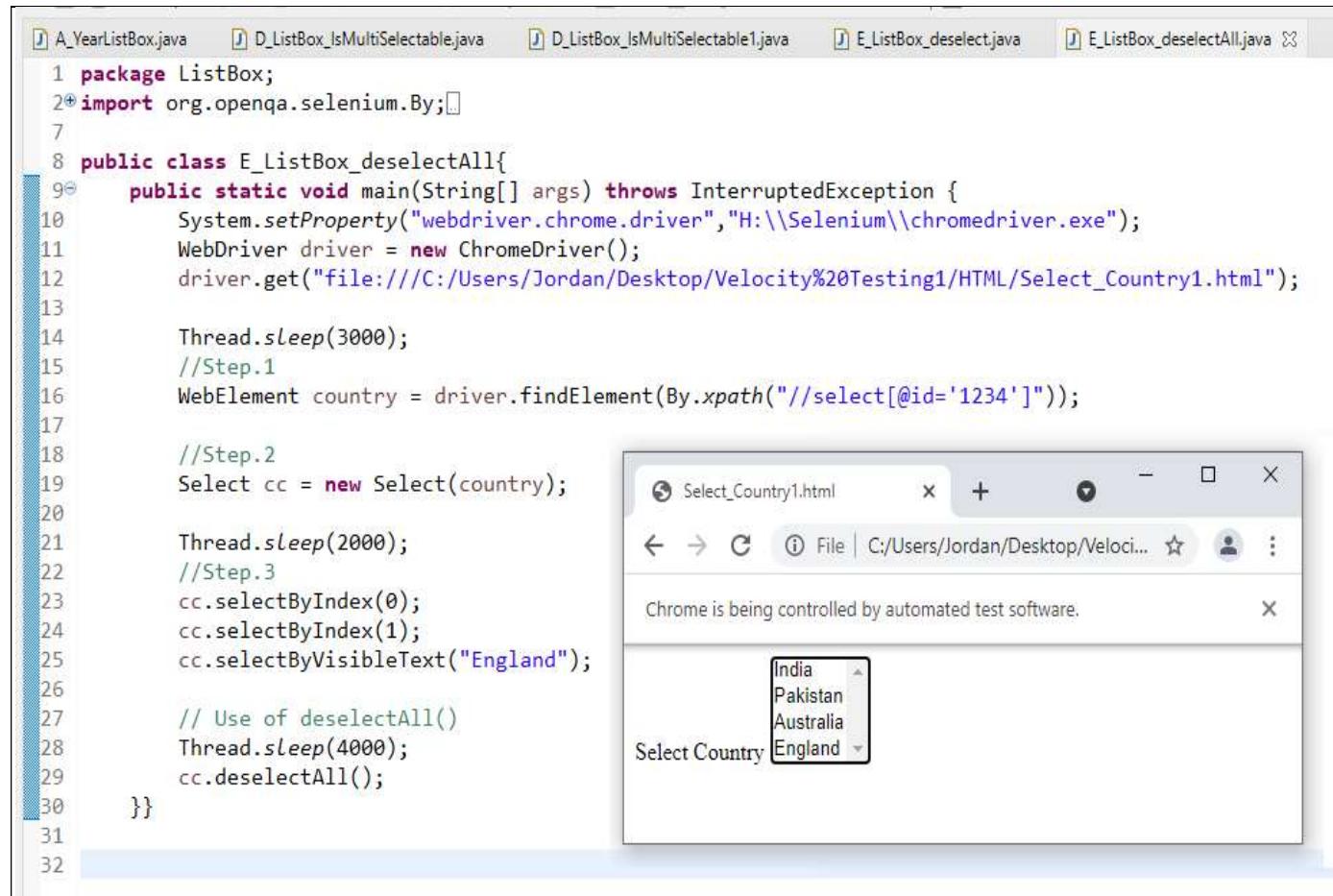
    // Use of deselect
    Thread.sleep(4000);
    cc.deselectByIndex(0);
    cc.deselectByVisibleText("England");
    cc.selectByIndex(1);

}
}
-----
```

7. DeselectAll()**void**

But we can't use deselect option for single selectable list box.

If we try to do this then we will get exception **Unsupported Operation Exception**.



```

1 package ListBox;
2 import org.openqa.selenium.By;
3
4 public class E_ListBox_deselectAll{
5     public static void main(String[] args) throws InterruptedException {
6         System.setProperty("webdriver.chrome.driver","H:\\Selenium\\chromedriver.exe");
7         WebDriver driver = new ChromeDriver();
8         driver.get("file:///C:/Users/Jordan/Desktop/Velocity%20Testing1/HTML>Select_Country1.html");
9
10        Thread.sleep(3000);
11        //Step.1
12        WebElement country = driver.findElement(By.xpath("//select[@id='1234']"));
13
14        //Step.2
15        Select cc = new Select(country);
16
17        Thread.sleep(2000);
18        //Step.3
19        cc.selectByIndex(0);
20        cc.selectByIndex(1);
21        cc.selectByVisibleText("England");
22
23        // Use of deselectAll()
24        Thread.sleep(4000);
25        cc.deselectAll();
26
27    }
28
29
30 }
31
32

```

10. getFirstSelectedOption ()**WebElement**

```

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

        driver.get("file:///C:/Users/Jordan/Desktop/Velocity%20Testing1/H
TML>Select_Country1.html");

```

```

//Step.1 Select the webElement which need to be handled then stored it
in reference Variable
    WebElement country =
driver.findElement(By.xpath("//select[@id='1234']"));

//Step.2 create an Object of the Select class which accept the
argument of WebElement
    Select cc = new Select(country);

    Thread.sleep(4000);
//Step.3 Use Select class methods to Select desire options

    cc.selectByVisibleText("Pakistan");
        Thread.sleep(3000);
    cc.selectByVisibleText("India");

//Method.1
//WebElement selectOption = cc.getFirstSelectedOption();
//String text = selectOption.getText();
//System.out.println(text);      //India

//Method.2
//String text1 = cc.getFirstSelectedOption().getText();
//System.out.println(text1);      //India

//Method.3
System.out.println(cc.getFirstSelectedOption().getText());
}
}

```

The screenshot shows a Java development environment with multiple tabs open. The active tab contains the code provided above. To the right is the 'Console' window, which displays the output of the Selenium script. Below the code editor is a screenshot of a web browser window titled 'Select_Country1.html', showing a dropdown menu with 'India' selected.

```

1 package ListBox;
2
3 import org.openqa.selenium.By;
4 import org.openqa.selenium.WebDriver;
5 import org.openqa.selenium.WebElement;
6 import org.openqa.selenium.chrome.ChromeDriver;
7 import org.openqa.selenium.support.ui.Select;
8
9 public class E_getFirstSelectedOptions {
10     public static void main(String[] args) throws InterruptedException {
11         System.setProperty("webdriver.chrome.driver", "H:\\Selenium\\chromedriver.exe");
12         WebDriver driver = new ChromeDriver();
13         driver.get("file:///C:/Users/Jordan/Desktop/Velocity%20Testing1/HTML>Select_Country1.html");
14
15         //Step.1 Select the webElement which need to be handled then stored it in reference Variable
16         WebElement country = driver.findElement(By.xpath("//select[@id='1234']"));
17
18         //Step.2 create an Object of the Select class which accept the argument of WebElement
19         Select cc = new Select(country);
20
21         Thread.sleep(4000);
22         //Step.3 Use Select class methods to Select desire options
23         cc.selectByIndex(1);
24         cc.selectByVisibleText("India");
25
26         //Method.1
27         WebElement selectOption = cc.getFirstSelectedOption();
28         String text = selectOption.getText();
29         System.out.println(text);
30
31

```

```

package Listbox;

import java.util.ArrayList;
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;
import org.openqa.selenium.support.ui.Select;

public class example5_getAllSelectedOption_MultiSelectableListbox
{
    public static void main(String[] args) throws InterruptedException {
        System.setProperty("webdriver.chrome.driver",
            "C:\\\\Users\\\\sanjay\\\\Desktop\\\\Study\\\\Selenium_Syllabus\\\\Jul21_Eve_Selenium\\\\chromedriver_w
in32 (14)\\\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("file:///C:/Users/sanjay/Desktop/Study/Selenium_Syllabus/Jul21_Eve_Selenium/Html
%20files/sample4_ListBox.html");
        WebElement SC = driver.findElement(By.xpath("//select[@id='1234']"));
        Select s=new Select(SC);
        s.selectByIndex(0);
        s.selectByIndex(2);
        s.selectByIndex(3);
        List<WebElement> allSelectedOptions = s.getAllSelectedOptions();
        System.out.println(allSelectedOptions.size());
        for( WebElement s1: allSelectedOptions)
        {
            System.out.println(s1.getText());
        }
        // ArrayList<String> al=new ArrayList<String>();
        // al.add("abc");
        // al.add("dkjhdqa");
        // System.out.println(al.size());
        // for(String s1: al)
        // {
        //     System.out.println(s1);
        // }
    }
}

```

```

package Listbox;

import java.util.ArrayList;
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;
import org.openqa.selenium.support.ui.Select;

public class example6_getAllOptionsInListbox2
{
    public static void main(String[] args) throws InterruptedException {
        System.setProperty("webdriver.chrome.driver",
            "C:\\Users\\sanjay\\Desktop\\Study\\Selenium_Syllabus\\Jul21_Eve_Selenium\\chromedriver_w
in32 (14)\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("file:///C:/Users/sanjay/Desktop/Study/Selenium_Syllabus/Jul21_Eve_Selenium/Html
%20files/sample4_ListBox.html");
        //step1:
        WebElement SC = driver.findElement(By.xpath("//select[@id='1234']"));

        Select s=new Select(SC);

        List<WebElement> allElements = s.getOptions();

        System.out.println(allElements.size());

        for( WebElement element : allElements)
        {
            System.out.println(element.getText());
        }

    }
}
=====
```

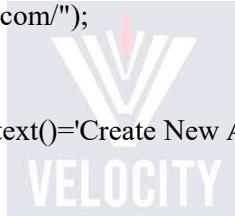
```

package Listbox;

import java.util.ArrayList;
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;
import org.openqa.selenium.support.ui.Select;

public class example7_getAllOptionsInListbox1
{
    public static void main(String[] args) throws InterruptedException {
        System.setProperty("webdriver.chrome.driver",
            "C:\\\\Users\\\\sanjay\\\\Desktop\\\\Study\\\\Selenium_Syllabus\\\\Jul21_Eve_Selenium\\\\chromedriver_w
            in32 (14)\\\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://en-gb.facebook.com/");
        //click on create new acc link
        driver.findElement(By.xpath("//a[text()='Create New Account']")).click();
        Thread.sleep(2000);
        //step1:
        WebElement month = driver.findElement(By.xpath("//select[@id='month']"));
        //step2:
        Select s=new Select(month);
        List<WebElement> allOptions = s.getOptions();
        System.out.println(allOptions.size());
        for(WebElement s1:allOptions) {
            System.out.println(s1.getText());
        }
        int sizeOfListbox = s.getOptions().size();
    }
}

```



```

package Listbox;

import java.util.ArrayList;
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;
import org.openqa.selenium.support.ui.Select;

public class example7_getSizeListbox
{
    public static void main(String[] args) throws InterruptedException {
        System.setProperty("webdriver.chrome.driver",
            "C:\\Users\\sanjay\\Desktop\\Study\\Selenium_Syllabus\\Jul21_Eve_Selenium\\chromedriver_win32 (14)\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://en-gb.facebook.com/");
        //click on create new acc link
        driver.findElement(By.xpath("//a[text()='Create New Account']")).click();

        Thread.sleep(2000);

        //step1:
        WebElement month = driver.findElement(By.xpath("//select[@id='month']"));

        //step2:
        Select s=new Select(month);

        // List<WebElement> allOptions = s.getOptions();
        // System.out.println(allOptions.size());

        int sizeOfListbox = s.getOptions().size();
        System.out.println(sizeOfListbox);
    }
}

```

Screenshot

Q. How to capture Screenshot Using Selenium Webdriver

1. To capture screenshot using selenium Webdriver,
typecast Webdriver object into **TakesScreenshot()** Interface
2. Then call the method **getScreenshotAs()** and pass the argument **OutputType.FILE**

```
((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
```

3. This method return object of type **File**

```
File src = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
```

4. After Execution, it captures the Screenshot to local memory, to move screenshot from local memory to destination folder call the static method **copy (__, __)** which is present in **FileHandler class** which accepts two parameters as path of source and path of destination.

```
FileHandler.copy(src, dest);
```

Optional

5. To create path of destination, create object of File class which accept the String destination path as input.

```
File dest = new File("path");
File dest = new File("H:\\Velocity Testing\\Screenshot\\amazon"+RS+".jpg");
```

```

package Screenshot;
import java.io.File;
import java.io.IOException;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.io.FileHandler;

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

//Step.1 typecast Webdriver object into TakesScreenshot() Interface
//Step.2 call the method getScreenshotAs() and pass the argument
OutputType.FILE

File src = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);

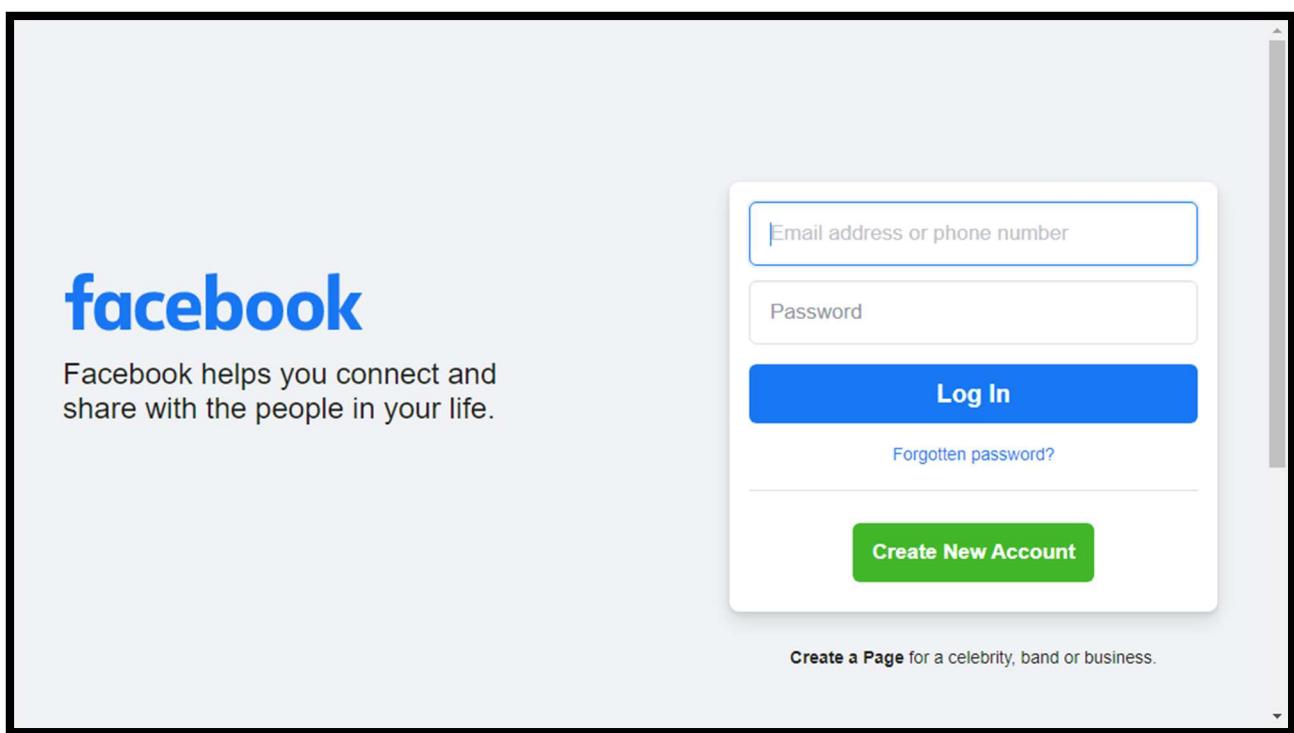
File dest=new File("H:\\Velocity Testing\\Screenshot\\facebook.jpg");

//Step.3 After Execution it captures the Screenshot to local memory,
to move screenshot from local memory to destination folder call the
static method copy (, ) which is present in FileHandler class which
accepts two parameters as path of source and path of destination.

FileHandler.copy(src, dest);

    }
}

```



```

package Screenshot;
import java.io.File;
import java.io.IOException;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.io.FileHandler;
import net.bytebuddy.utility.RandomString;

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

        WebDriver driver = new ChromeDriver();
        driver.get("https://www.amazon.in/");

        String RS = RandomString.make(3); // Use to add random character in name

        File src =
        ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);

        File dest = new File("H:\\Velocity
Testing\\Screenshot\\amazon"+RS+".jpg");

        FileHandler.copy(src, dest);
    }
}

```



The screenshot shows a Java development environment with two tabs open: A_YearListBox.java and A_CaptureScreenshot.java. The A_CaptureScreenshot.java tab is active, displaying the code for the screenshot capture process. To the right of the code editor, a screenshot of a web browser is visible, showing the Amazon.in homepage. The browser status bar indicates 'Chrome is being controlled by automated test software'. The main content of the browser shows a festive banner for 'GREAT INDIAN FESTIVAL Extra Happiness Days' and promotional offers for mobile phones.

```

1 package Screenshot;
2 import java.io.File;
3 import java.io.IOException;
4 import org.openqa.selenium.OutputType;
5 import org.openqa.selenium.TakesScreenshot;
6 import org.openqa.selenium.WebDriver;
7 import org.openqa.selenium.chrome.ChromeDriver;
8 import org.openqa.selenium.io.FileHandler;
9 import net.bytebuddy.utility.RandomString;
10
11 public class B_CaptureScreenShot_Amazon {
12     public static void main(String[] args) throws IOException {
13         System.setProperty("webdriver.chrome.driver", "H:\\Selenium\\chromedriver.exe");
14
15         WebDriver driver = new ChromeDriver();
16         driver.get("https://www.amazon.in/");
17
18         String RS = RandomString.make(3); // Use to add random character in name
19
20         File src = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
21
22         File dest = new File("H:\\Velocity Testing\\Screenshot\\amazon"+RS+".jpg");
23
24         FileHandler.copy(src, dest);
25     }
26 }

```

Parameterization

- The process of fetching the data from external source and use this data in selenium script, is called as parameterization.
- These external sources can be Excel sheet, CSV file, Test NG data provider etc.

Q. How to fetch data from excel sheet

- Step 1:** First Configure the “Apache Poi Jar File” into project
- Step 2:** Create an Excel sheet with data and save it to desired location.
- Step 3:** Create and object of “**FileInputStream**” class **with excel sheet path** as an input in constructor.
- Step 4:** To open the excel sheet use static method **create()** present in “**workBookFactory**” class
- Step 5:** To open specific sheet in excel use **getSheet()** method
- Step 6:** To identify the desired row use method **getRow()**
- Step 7:** To identify the desired cell use **getCell()** method.
- Step 8:**
- A) To fetch string type of information call **getStringCellValue()** method.
- B) To fetch numeric type of information call **getNumericCellValue()** method.
- C) To fetch boolean type of information call **getBooleanCellValue()** method.

```

package Parameterization;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import org.apache.poi.EncryptedDocumentException;
import org.apache.poi.sl.usermodel.Sheet;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.ss.usermodel.WorkbookFactory;

public class A_getString {
    public static void main(String[] args) throws
EncryptedDocumentException, IOException {
        //Step.3 Create and object of FileInputStream class with
        //excel sheet path as input in constructor
        FileInputStream file = new FileInputStream("H:\\Velocity
Testing\\ExcelFile\\Sample_File.xlsx");
    }
}

```

```

//Step.4 To open the excel sheet use static method create()
from Workbookfactory Class
    Workbook book = WorkbookFactory.create(file);

    //Step.5 To open specific sheet in excel used getSheet()
method
        org.apache.poi.ss.usermodel.Sheet sh =
book.getSheet("Sheet1");

    //Step.6 To identify the desired row use method getRow();
        org.apache.poi.ss.usermodel.Row rw= sh.getRow(9);

    //Step.7 To identify the desired cell use method getCell();
        Cell cl = rw.getCell(7);

        String Value = cl.getStringCellValue();
        System.out.println(Value);                                //Vaibhav

```

//ShortCut Steps

```

Cell value =
WorkbookFactory.create(file).getSheet("Sheet1").getRow(5).getCell(9);

OR
String value =
WorkbookFactory.create(file).getSheet("Sheet1").getRow(9).getCell(7).g
etStringCellValue();

System.out.println(value);
}
}-----
```

```

package Parameterization;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import org.apache.poi.EncryptedDocumentException;
import org.apache.poi.ss.usermodel.WorkbookFactory;

public class B_getNumericData {
    public static void main(String[] args) throws
EncryptedDocumentException, IOException {

FileInputStream file = new FileInputStream("H:\\Velocity
Testing\\ExcelFile\\Sample_File.xlsx");

double value =
WorkbookFactory.create(file).getSheet("Sheet1").getRow(5).getCell(9).g
etNumericCellValue();

        System.out.println(value);          //123.0
    }
}-----
```

```

package Parameterization;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import org.apache.poi.EncryptedDocumentException;
import org.apache.poi.ss.usermodel.WorkbookFactory;

public class C_getBooleanData {
    public static void main(String[] args) throws
EncryptedDocumentException, IOException {

        FileInputStream file = new FileInputStream("H:\\Velocity
Testing\\ExcelFile\\Sample_File.xlsx");

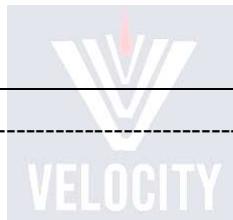
        boolean value =
WorkbookFactory.create(file).getSheet("Sheet1").getRow(7).getCell(11).
getBooleanCellValue();

        System.out.println(value); //true
    }
}

```

Index to Size → +1

Size to Index → -1



```

package Parameterization;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;

import org.apache.poi.EncryptedDocumentException;
import org.apache.poi.ss.usermodel.WorkbookFactory;

public class D_getRowSize {
    public static void main(String[] args) throws
EncryptedDocumentException, IOException {
        FileInputStream file = new FileInputStream("H:\\Velocity
Testing\\ExcelFile\\Sample_File.xlsx");

        //int rowSize =
WorkbookFactory.create(file).getSheet("Sheet1").getLastRowNum()+1;
        int rowSize2 =
WorkbookFactory.create(file).getSheet("Sheet1").getLastRowNum();
        // getLastRowNum(); --> returns Index of Last Row
        // getLastRowNum()+1;-->returns Size of Last Row
        //System.out.println(rowSize); //13
        System.out.println(rowSize2); //12
    } // (Above both operation can't be perform simultaneously.
}

```

```

package Parameterization;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import org.apache.poi.EncryptedDocumentException;
import org.apache.poi.ss.usermodel.WorkbookFactory;

public class E_getColSize_inRow {
    public static void main(String[] args) throws
EncryptedDocumentException, IOException {

        FileInputStream file = new FileInputStream("H:\\Velocity
Testing\\ExcelFile\\Sample_File.xlsx");

        int ColSize =
WorkbookFactory.create(file).getSheet("Sheet1").getRow(0).getLastCellN
um();

        System.out.println(ColSize);      // Returns Actual No of
Columns Means Size. //13
    }
}
-----
```

```

package Parameterization;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import org.apache.poi.EncryptedDocumentException;
import org.apache.poi.ss.usermodel.WorkbookFactory;

public class F_getNumericData_AsString {
    public static void main(String[] args) throws
EncryptedDocumentException, IOException {
        FileInputStream file = new FileInputStream("H:\\Velocity
Testing\\ExcelFile\\Sample_File.xlsx");

        String value =
WorkbookFactory.create(file).getSheet("Sheet1").getRow(6).getCell(9).g
etStringCellValue();

        System.out.println(value);      //456
        // Note : To use this method Make sure that the Number
Stored in Excel file
        // must poses single inverted comma before the number ('456)
    }
}
-----
```

```

package Parameterization;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import org.apache.poi.EncryptedDocumentException;
import org.apache.poi.ss.usermodel.Row;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.WorkbookFactory;

public class H_printAll_Data_In_Col {
    public static void main(String[] args) throws
EncryptedDocumentException, IOException {

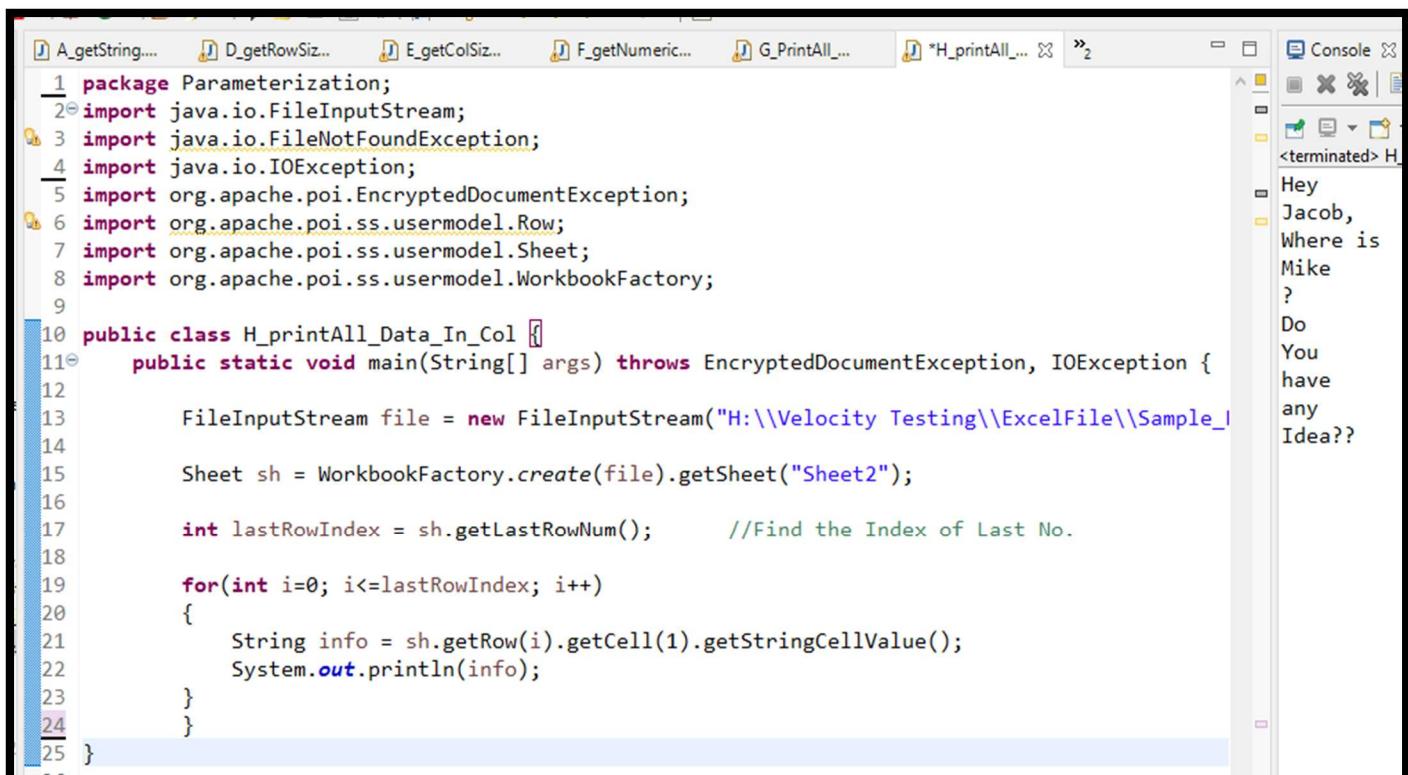
        FileInputStream file = new FileInputStream("H:\\Velocity
Testing\\ExcelFile\\Sample_File.xlsx");

        Sheet sh = WorkbookFactory.create(file).getSheet("Sheet2");

        int lastRowIndex = sh.getLastRowNum();           //Find the Index
of Last No.

        for(int i=0; i<=lastRowIndex; i++)
        {
            String info =
sh.getRow(i).getCell(1).getStringCellValue();
            System.out.println(info);
        }
    }
}

```



The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** Shows various tabs like A_getString..., D_getRowSiz..., E_getColSiz..., F_getNumeric..., G_PrintAll..., and H_printAll... along with a close button.
- Left Side:** A package explorer window showing a single package named "Parameterization" containing several source files (A, B, C, D, E, F, G, H).
- Right Side:** A "Console" window showing the output of the Java application. The output includes:
 - "Hey"
 - "Jacob,"
 - "Where is"
 - "Mike"
 - "?"
 - "Do"
 - "You"
 - "have"
 - "any"
 - "Idea??"
- Code Editor:** The main editor area displays the Java code for class H_printAll_Data_In_Col. The code reads an Excel file named "Sample_File.xlsx" and prints all data from the second column of the second sheet ("Sheet2"). The code uses Apache POI libraries for file handling.

```

package Parameterization;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import org.apache.poi.EncryptedDocumentException;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.WorkbookFactory;

public class G_PrintAll_Data_In_Sheet {
    public static void main(String[] args) throws
EncryptedDocumentException, IOException {

FileInputStream file = new FileInputStream("H:\\Velocity
Testing\\ExcelFile\\Sample_File.xlsx");

Sheet sh = WorkbookFactory.create(file).getSheet("Sheet3");

int lastRowIndex = sh.getLastRowNum();
//Step.1 Take outer for loop for Rows
for(int i=0; i<=lastRowIndex;i++)
{
    //Step.2 Inner for loop used for Columns
    int lastCellIndex = sh.getRow(i).getLastCellNum()-1;
    for(int j=0; j<=lastCellIndex; j++)
    {
        String info = sh.getRow(i).getCell(j).getStringCellValue();
        System.out.print(info+" ");
    }
    System.out.println();
}
}
}

```

The screenshot shows the Eclipse IDE interface with the Java code for reading an Excel file. The code uses Apache POI to handle the file. The 'Console' view on the right shows the output of the program, which is the content of the 'Sheet3' from the 'Sample_File.xlsx' file.

```

<terminated> G_PrintAll_Data_In_Sheet [Java Application] C:\Pr
Hey Mike are you There?? There??

```

```

package Parameterization;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;

import org.apache.poi.EncryptedDocumentException;
import org.apache.poi.ss.usermodel.CellType;
import org.apache.poi.ss.usermodel.WorkbookFactory;

public class I_Get_Type_Of_Cell {
    public static void main(String[] args) throws
EncryptedDocumentException, IOException {

FileInputStream file = new FileInputStream("H:\\Velocity
Testing\\ExcelFile\\Sample_File.xlsx");

CellType findtype =
WorkbookFactory.create(file).getSheet("Sheet1").getRow(9).getCell(7).g
etCellType();

        System.out.println(findtype); //STRING
    }
}

```

```

1 package Parameterization;
2
3 import java.io.FileInputStream;
4 import java.io.FileNotFoundException;
5 import java.io.IOException;
6
7 import org.apache.poi.EncryptedDocumentException;
8 import org.apache.poi.ss.usermodel.CellType;
9 import org.apache.poi.ss.usermodel.WorkbookFactory;
10
11 public class I_Get_Type_Of_Cell {
12     public static void main(String[] args) throws EncryptedDocumentException, IOException {
13
14         FileInputStream file = new FileInputStream("H:\\Velocity Testing\\ExcelFile\\Sample_File.xlsx");
15
16         CellType findtype = WorkbookFactory.create(file).getSheet("Sheet1").getRow(9).getCell(7).getCellType();
17
18         System.out.println(findtype); //STRING
19     }
20 }

```

```

package Parameterization;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import org.apache.poi.EncryptedDocumentException;
import org.apache.poi.ss.usermodel.CellType;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.WorkbookFactory;

public class J_Verify_Type_of_Cell {
    public static void main(String[] args) throws
EncryptedDocumentException, IOException {

        FileInputStream file = new FileInputStream("H:\\Velocity
Testing\\ExcelFile\\Sample_File.xlsx");

```

```

Sheet sh = WorkbookFactory.create(file).getSheet("Sheet1");

CellType dataType = sh.getRow(9).getCell(7).getCellType();

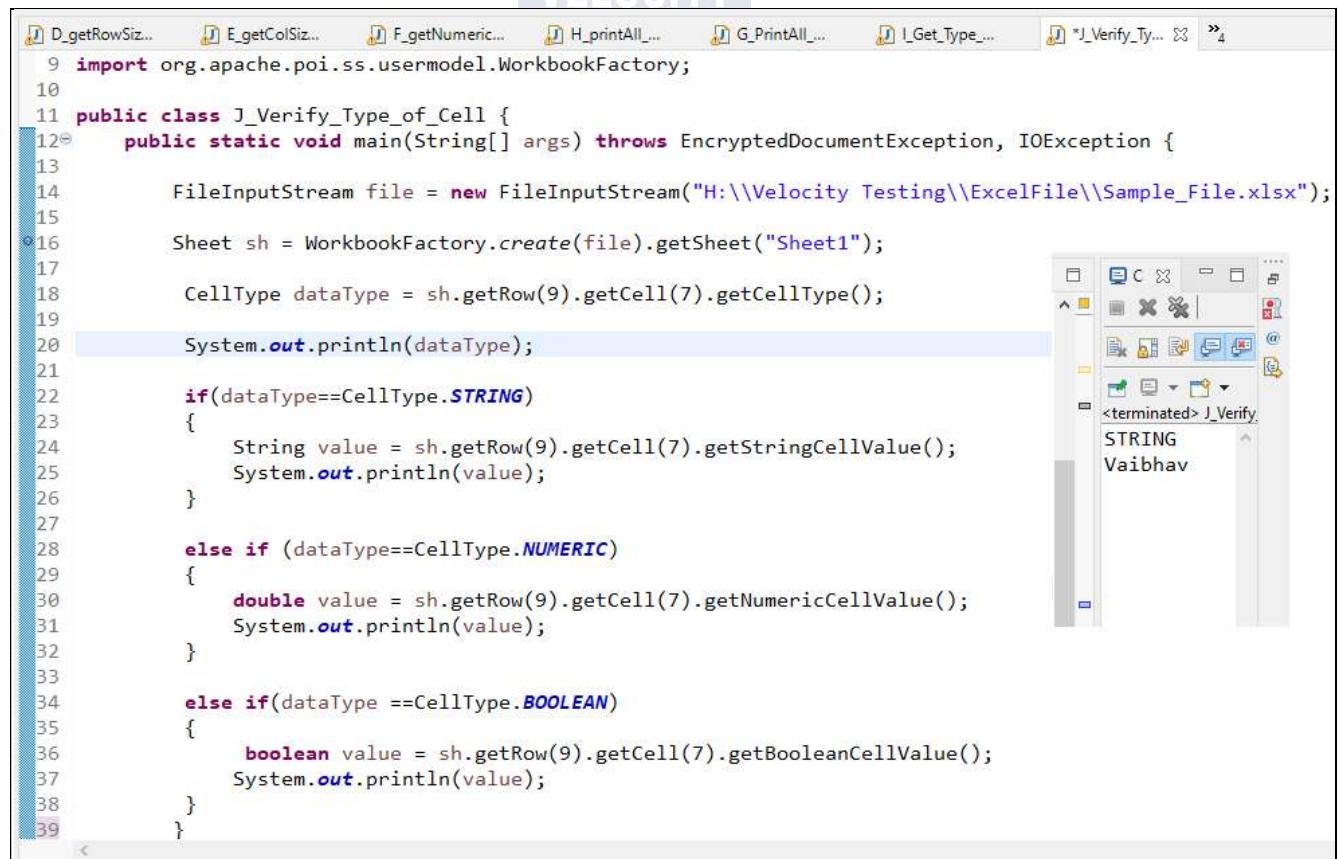
System.out.println(dataType);

if(dataType==CellType.STRING)
{
    String value =
sh.getRow(9).getCell(7).getStringCellValue();
    System.out.println(value);
}

else if (dataType==CellType.NUMERIC)
{
    double value =
sh.getRow(9).getCell(7).getNumericCellValue();
    System.out.println(value);
}

else if(dataType ==CellType.BOOLEAN)
{
    boolean value =
sh.getRow(9).getCell(7).getBooleanCellValue();
    System.out.println(value);
}
}

```



The screenshot shows the Eclipse IDE interface with the following details:

- Toolbar:** Shows various icons for file operations like D_getRowSiz..., E_getColSiz..., F_getNumeric..., H_printAll..., G_PrintAll..., I_Get_Type_..., J_Verify_Ty..., and >4.
- Code Editor:** Displays the Java code for reading cell types. Line 20 highlights the line `System.out.println(dataType);`.
- Output View:** On the right, the output window shows the result of the execution: `<terminated> J_Verify STRING Vaibhav`.

```

9 import org.apache.poi.ss.usermodel.WorkbookFactory;
10
11 public class J_Verify_Type_of_Cell {
12     public static void main(String[] args) throws EncryptedDocumentException, IOException {
13
14         FileInputStream file = new FileInputStream("H:\\Velocity Testing\\ExcelFile\\Sample_File.xlsx");
15
16         Sheet sh = WorkbookFactory.create(file).getSheet("Sheet1");
17
18         CellType dataType = sh.getRow(9).getCell(7).getCellType();
19
20         System.out.println(dataType);
21
22         if(dataType==CellType.STRING)
23         {
24             String value = sh.getRow(9).getCell(7).getStringCellValue();
25             System.out.println(value);
26         }
27
28         else if (dataType==CellType.NUMERIC)
29         {
30             double value = sh.getRow(9).getCell(7).getNumericCellValue();
31             System.out.println(value);
32         }
33
34         else if(dataType ==CellType.BOOLEAN)
35         {
36             boolean value = sh.getRow(9).getCell(7).getBooleanCellValue();
37             System.out.println(value);
38         }
39

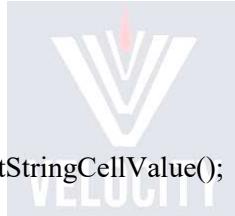
```

```

package Excel_Sheet;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import org.apache.poi.EncryptedDocumentException;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.CellType;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.WorkbookFactory;
public class example11_printData_verifyTypeOfCell2 {
    public static void main(String[] args) throws EncryptedDocumentException, IOException {
        FileInputStream file=new
FileInputStream("C:\\\\Users\\\\sanjay\\\\Desktop\\\\Study\\\\Selenium_Syllabus\\\\Jul21_Eve_Selenium\\\\July21_ Eve.xlsx");
        Cell CellInfo = WorkbookFactory.create(file).getSheet("Sheet6").getRow(0).getCell(0);
        CellType s1 = CellInfo.getCellType();
        if(s1==CellType.STRING)
        {
            String value = CellInfo.getStringCellValue();
            System.out.println(value);
        }
        else if (s1==CellType.NUMERIC)
        {
            double value = CellInfo.getNumericCellValue();
            System.out.println(value);
        }
        else if (s1==CellType.BOOLEAN)
        {
            boolean value = CellInfo.getBooleanCellValue();
            System.out.println(value);
        }
    }
}

```



```

package Excel_Sheet;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import org.apache.poi.EncryptedDocumentException;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.CellType;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.WorkbookFactory;

public class example12_PrintAllDataInARow_VerifyTypeOfCell {
    public static void main(String[] args) throws EncryptedDocumentException, IOException {
        FileInputStream file=new
FileInputStream("C:\\\\Users\\\\sanjay\\\\Desktop\\\\Study\\\\Selenium_Syllabus\\\\Jul21_Eve_Selenium\\\\July21_ Eve.xlsx");
        Sheet sh = WorkbookFactory.create(file).getSheet("Sheet6");
        int lastCellIndex = sh.getRow(0).getLastCellNum()-1;
        for(int i=0; i<=lastCellIndex; i++)
        {
            Cell cellInfo = sh.getRow(0).getCell(i);
            CellType s1 = cellInfo.getCellType();
            if(s1==CellType.STRING)
            {
                String value =cellInfo.getStringCellValue();
                System.out.print(value+" ");
            }
            else if (s1==CellType.NUMERIC)
            {
                double value = cellInfo.getNumericCellValue();
                System.out.print(value+" ");
            }
            else if (s1==CellType.BOOLEAN)
            {
                boolean value = cellInfo.getBooleanCellValue();
                System.out.print(value+" ");
            }
        }
    }
}

```

```

package Excel_Sheet;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import org.apache.poi.EncryptedDocumentException;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.CellType;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.WorkbookFactory;

public class example13_PrintAllDataInACol_VerifyTypeOfCell {
    public static void main(String[] args) throws EncryptedDocumentException, IOException {
        FileInputStream file=new
FileInputStream("C:\\\\Users\\\\sanjay\\\\Desktop\\\\Study\\\\Selenium_Syllabus\\\\Jul21_Eve_Selenium\\\\July21_ Eve.xlsx");
        Sheet sh = WorkbookFactory.create(file).getSheet("Sheet6");
        int lastRowIndex = sh.getLastRowNum();
        for(int i=0; i<=lastRowIndex; i++)
        {
            Cell CellInfo = sh.getRow(i).getCell(0);
            CellType s2 = CellInfo.getCellType();
            if(s2==CellType.STRING)
            {
                System.out.println(CellInfo.getStringCellValue());
            }
            else if (s2==CellType.NUMERIC)
            {
                System.out.println(CellInfo.getNumericCellValue());
            }
            else if (s2==CellType.BOOLEAN) {
                System.out.println(CellInfo.getBooleanCellValue());
            }
        }
    }
}

```

```

package Excel_Sheet;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import org.apache.poi.EncryptedDocumentException;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.CellType;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.WorkbookFactory;

public class example14_PrintAllDataInASheet_VerifyTypeOfCell {
    public static void main(String[] args) throws EncryptedDocumentException, IOException {
        FileInputStream file=new
FileInputStream("C:\\\\Users\\\\sanjay\\\\Desktop\\\\Study\\\\Selenium_Syllabus\\\\Jul21_Eve_Selenium\\\\July21_ Eve.xlsx");
        Sheet sh = WorkbookFactory.create(file).getSheet("Sheet6");
        int lastRowIndex = sh.getLastRowNum();
        for (int i = 0; i<=lastRowIndex; i++) //outer for loop --> rows
        {
            int lastCellIndex = sh.getRow(i).getLastCellNum()-1;
            for (int j = 0; j <=lastCellIndex; j++)
            {
                Cell cellInfo = sh.getRow(i).getCell(j);
                CellType s2 = cellInfo.getCellType();
                if(s2==CellType.STRING)
                {
                    System.out.print(cellInfo.getStringCellValue()+" ");
                }
                else if (s2==CellType.NUMERIC)
                {
                    System.out.print(cellInfo.getNumericCellValue()+" ");
                }
                else if (s2==CellType.BOOLEAN) {
                    System.out.print(cellInfo.getBooleanCellValue()+" ");
                }
            }
            System.out.println();
        }
    }
}

```

Iframe

- Displaying webpage which is a part of another webpage is known as IFrame.
- IFrame is created by using Tagname IFrame.

How to Handle Iframe using Selenium Webdriver

To handle Iframe using selenium Webdriver we need to **switch selenium focus** from main webpage to **iframe**.

Use `driver.switchTo().frame()`

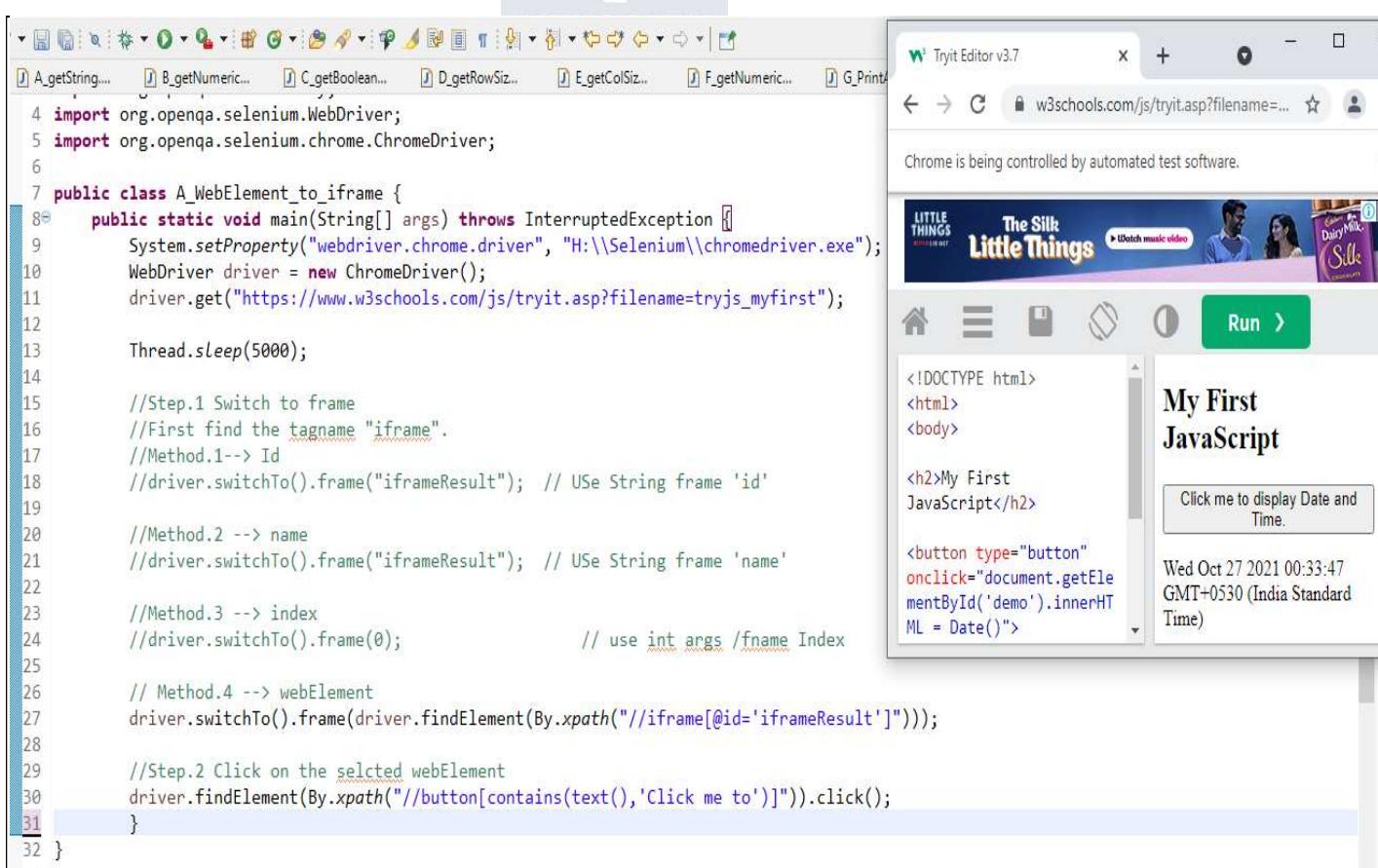
We can switch to Iframe using 4 different ways as

1. **id** `driver.switchTo().frame("iframeResult")`
1. **name** `driver.switchTo().frame("iframeResult")`
2. **index** `driver.switchTo().frame(0)`
3. **WebElement :**

```
driver.switchTo().frame(driver.findElement(By.xpath("//iframe[@id='iframeResult']")));
```

4. If we want to go back to one frame only then use function **switch to parent frame**.

5. If we want to go directly to main / default page then use **switch to default content function**.



The screenshot shows a Java IDE interface with a code editor and a browser window. The code editor contains a Java class named `A_WebElement_to_iframe` with a `main` method. The code demonstrates how to switch to an IFrame using four different methods: `switchTo().frame("iframeResult")`, `switchTo().frame("iframeResult")`, `switchTo().frame(0)`, and `switchTo().frame(driver.findElement(By.xpath("//iframe[@id='iframeResult']")))`. The browser window displays the output of the script, showing the text "My First JavaScript" and a button labeled "Click me to display Date and Time". The browser status bar indicates "Chrome is being controlled by automated test software".

```

4 import org.openqa.selenium.WebDriver;
5 import org.openqa.selenium.chrome.ChromeDriver;
6
7 public class A_WebElement_to_iframe {
8     public static void main(String[] args) throws InterruptedException {
9         System.setProperty("webdriver.chrome.driver", "H:\\Selenium\\chromedriver.exe");
10        WebDriver driver = new ChromeDriver();
11        driver.get("https://www.w3schools.com/js/tryit.asp?filename=tryjs_myfirst");
12
13        Thread.sleep(5000);
14
15        //Step.1 Switch to frame
16        //First find the tagname "iframe".
17        //Method.1--> Id
18        //driver.switchTo().frame("iframeResult"); // Use String frame 'id'
19
20        //Method.2 --> name
21        //driver.switchTo().frame("iframeResult"); // Use String frame 'name'
22
23        //Method.3 --> index
24        //driver.switchTo().frame(0); // use int args /fname Index
25
26        // Method.4 --> webElement
27        driver.switchTo().frame(driver.findElement(By.xpath("//iframe[@id='iframeResult']")));
28
29        //Step.2 Click on the selected webElement
30        driver.findElement(By.xpath("//button[contains(text(),'Click me to')]")).click();
31    }
32 }

```

```

package iframe;

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

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

driver.get("https://www.w3schools.com/js/tryit.asp?filename=tryjs_myfi
rst");

        Thread.sleep(5000);

//Step.1 Switch to frame
//First find the tagname "iframe".

//Method.1--> Id
//driver.switchTo().frame("iframeResult"); // USE String frame 'id'

//Method.2 --> name
//driver.switchTo().frame("iframeResult"); // USE String frame 'name'

//Method.3 --> index
//driver.switchTo().frame(0); // use int args /fname Index

// Method.4 --> webElement
driver.switchTo().frame(driver.findElement(By.xpath("//iframe[@id='ifr
ameResult']")));
}

//Step.2 Click on the selcted webElement

        driver.findElement(By.xpath("//button[contains(text(),'Click
me to')]")).click();
    }
}-----
```

```

package iframe;

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

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

    driver.get("https://www.w3schools.com/js/tryit.asp?filename=tryjs
_myfirst");

    //switch to frame
    driver.switchTo().frame("iframeResult");

    //click on date & time btn
    driver.findElement(By.xpath("//button[contains(text(),'Click
me to')]")).click();

    //switch to main page
    //driver.switchTo().parentFrame();
    driver.switchTo().defaultContent();

    //click on home Icon
    driver.findElement(By.xpath("//a[@id='tryhome']")).click();

}
}

```

Dynamic Handling

```

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

public class DynamicElement_Handling1 {
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver",
        "H:\\Selenium\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.flipkart.com/");

        //Implicitly Wait ( Selenium Wait Method)

        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

        // in implicitly, it will wait for 10,20,etc sec to find the element
        But if element found before 10 sec it will proceed.

        driver.findElement(By.xpath("//button[@class='_2KpZ6l
        _2doB4z']")).click();

        driver.findElement(By.xpath("//input[@class='_3704LK']")).sendKeys("Ip
        hone11");

        driver.findElement(By.xpath("//button[@class='L0Z3Pu']")).click();

        String Rating =
        driver.findElement(By.xpath("((//div[@class='_2kHMtA'])[1]//span)[6]")
        ).getText();

        System.out.println(Rating);      // 76,710 Ratings
    }
}

```