Selenium

Selenium Webdriver Architecture

1.HSSF & XSSF

XSSF = XML spreadsheet format(xlsx) & HSSF = Horrible spreadsheet format(xls)

Reading and writing in excel

(Ex.1)

```
public class ReadandWriteExcel {
 public static void main(String []args){
   try {
   // Specify the file path which you want to create or write
   File src=new file(file path+file name) ;
   // Load the file
   FileInputStream fis=new FileInputStream(src);
    // load the workbook
    XSSFWorkbook wb=new XSSFWorkbook(fis)
    XSSFSheet sh1= wb.getSheetAt(0);
  // getRow specify which row we want to read and getCell which column
  System.out.println(sh1.getRow(0).getCell(0).getStringCellValue());
  System.out.println(sh1.getRow(0).getCell(1).getStringCellValue());
  System.out.println(sh1.getRow(1).getCell(0).getStringCellValue());
  System.out.println(sh1.getRow(1).getCell(1).getStringCellValue());
System.out.println(sh1.getRow(2).getCell(0).getStringCellValue());
  @;
```

```
// here createCell will create column
 // and setCellvalue will set the value
  sh1.getRow(0).createCell(2).setCellValue("2.41.0");
  sh1.getRow(1).createCell(2).setCellValue("2.5");
  sh1.getRow(2).createCell(2).setCellValue("2.39");

// here we need to specify where you want to save file
  FileOutputStream fout=new FileOutputStream(new File("location of file/filename.xlsx"));
 // finally write content
  wb.write(fout);

// close the file
  fout.close();

  } catch (Exception e) {

   System.out.println(e.getMessage());
 }}}
```

(Ex.2) **Using loop for the whole file**

```
public class ReadExcelFile {
	public void readExcel() throws BiffException, IOException {
		String FilePath = "D:\\sampledoc.xls";
		FileInputStream fs = new FileInputStream(FilePath);
		Workbook wb = Workbook.getWorkbook(fs);

		// TO get the access to the sheet
		Sheet sh = wb.getSheet("Sheet1");

		// To get the number of rows present in sheet
		int totalNoOfRows = sh.getRows();

		// To get the number of columns present in sheet
		int totalNoOfCols = sh.getColumns();

		for (int row = 0; row < totalNoOfRows; row++) {

			for (int col = 0; col < totalNoOfCols; col++) {
				System.out.print(sh.getCell(col, row).getContents() + "\t");
			}
			System.out.println();
		}
```

```
	}
	public static void main(String args[]) throws BiffException, IOException {a
		ReadExcelFile DT = new ReadExcelFile();
		DT.readExcel();
	}
}
```

The output of the below program is:

Username	password
testuser1	testpassword1
testuser2	testpassword2
testuser3	testpassword3
testuser4	testpassword4

Browser code

```
public static void main(String[] args) {
System.setProperty("webdriver.chrome.driver", "path of the exe file\\chromedriver.exe");
// Initialize browser
WebDriver driver=new ChromeDriver();
// Open facebook
driver.get("http://www.facebook.com");
 // Maximize browser
driver.manage().window().maximize();
}
}
```

Here some code to iterate through the list and only do something with the displayed items.

```
List<WebElement> elements = driver.findElements(By.className("vfmThumbnail"));
java.util.Iterator<WebElement> i = elements.iterator();

while(i.hasNext())

{ WebElement element = i.next();

if (element.isDisplayed())

{ // Do something with the element } }
```

To store the screenshot with time format

```
public void screenShot() throws IOException, InterruptedException {
```

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

File dest= new File("filPath/screenshot_"+timestamp()+".png"); FileUtils.copyFile(scr,
dest); Thread.sleep(3000); }

public string timestamp() {

return new SimpleDateFormat("yyyy-MM-dd HH-mm-ss").format(new Date());

}
```

To verify the specific message from alert box

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

String str= alert.getText();

//"This is a String contains Example";

System.out.println("Contains sequence 'ing': " + str.contains("ing"));
```

Scroll upto bottom

```
//Find element by link text and store in variable "Element"

WebElement Element = driver.findElement(By.linkText("Linux"));

//This will scroll the page until the element is found

js.executeScript("arguments[0].scrollIntoView();", Element);
```

Scroll upto bottom of the page

```
js.executeScript("window.scrollTo(0, document.body.scrollHeight)");
```

Send input without using send keys

```
// Initialize JS object
JavascriptExecutor JS = (JavascriptExecutor)webdriver;
// Enter username
JS.executeScript("document.getElementById('User').value='Abha_Rathour'");
// Enter password
JS.executeScript("document.getElementById('Password').value='password123'");
```

Handling iframes

https://www.guru99.com/handling-iframes-selenium.html

Java Script executor

```
JavascriptExecutor js = (JavascriptExecutor) driver;
   js.executeScript(Script,Arguments);
(ex.)
JavascriptExecutor js = (JavascriptExecutor) driver;
js.executeScript("window.scrollBy(0,1000)");
```

Desired capabilities

https://www.guru99.com/desired-capabilities-selenium.html

Broken Links

https://www.guru99.com/find-broken-links-selenium-webdriver.html

Ways to refresh

1.Get Method

```
driver.get("https://www.guru99.com");
driver.get(driver.getCurrentURL());
```

2.Navigate Method

```
driver.get("https://www.guru99.com");
driver.navigate.to(driver.getCurrentURL());
```

3.**F5-sendkeys**

```
driver.get("https://www.guru99.com");
driver. findElement(By.id("username")).sendKeys(Keys.F5);
```

4.**ASCII code**

```
driver.get("https://www.guru99.com");
driver. findElement(By.id("username")).sendKeys("\uE035");
```

Right Click

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

WebElement elementLocator = driver.findElement(By.id("ID"));

actions.contextClick(elementLocator).perform();
```

Double Click

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

WebElement elementLocator = driver.findElement(By.id("ID"));

actions.doubleClick(elementLocator).perform();
```

Why can't we create object creation for chrome driver

https://sqa.stackexchange.com/questions/32305/why-this-webdriver-driver-new-chromedriver-not-this-chromedriver-driver

How to handle state element exception?

https://www.softwaretestingmaterial.com/stale-element-reference-exception-selenium-webdriver

An **IFrame** (Inline Frame) is an HTML document embedded inside another HTML document on a website. The IFrame HTML element is often used to insert content from another source, such as an advertisement, into a Web page

Sequence of Execution of the annotations:
BeforeSuite.
BeforeTest.
BeforeClass.
BeforeMethod.
Test.
AfterMethod.
AfterClass.
AfterTest.
AfterSuite.

What is web table?
A table is made of rows and columns. When we create a table for a web page, that is called as a web table. In HTML, table is created using <table> tag. Web table is a HTML structure for creating rows and columns on a Web page.

A web table can consists below parts:
Header(s): It is created using <th> tag.
Row(s):It is created using <tr> tag.
Columns(s):It is created using <td> tag.

Types of web tables:
We can categorized web tables in two parts:

Static web table: Number of rows and columns will be definite. Eg. Table of months, Table of days etc.
Dynamic table: Number of rows and columns will be dynamic. It will be keep on increasing or decreasing based on data. For Eg: Sales table, Student table.

To print the all cells data

```
// Grab the table
WebElement table = driver.findElement(By.id("divListView"));
// Now get all the TR elements from the table
List<WebElement> allRows = table.findElements(By.tagName("tr"));
// And iterate over them, getting the cells
for (WebElement row : allRows) {
   List<WebElement> cells = row.findElements(By.tagName("td"));
   // Print the contents of each cell
   for (WebElement cell : cells) {
      System.out.println(cell.getText());
   }
}
```

// Retrieve cell value by providing row and column number

```
WebElement colValue= driver.findElement(By.xpath("//table[@name='BookTable']/tbody/tr[2]/td[3]"));
System.out.println(colValue.getText());
```

TestNG grouping

```
@Test(groups={“Smoke”})
```

In testNG xml file

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Loan Department">

    <test name="Regression">
        <groups>
            <run>
                <include name ="Smoke"/>
            </run>
        </groups>
    <classes>
      <class name="test.day1"/>
      <class name="test.day2"/>
      <class name ="test.day4"/>
        <class name="test.day3/">

    </classes>
  </test>

</suite>
```

Different Types of Exception

- **ElementNotVisibleException**: Although an element is present in the DOM, it is not visible (cannot be interacted with). E.g. Hidden Elements – defined in HTML using type=”hidden”.
- **ElementNotSelectableException**: Although an element is present in the DOM, it may be disabled (cannot be clicked/selected).
- InvalidSelectorException: Selector used to find an element does not return a WebElement. Say XPath expression is used which is either syntactically invalid or does not select WebElement.
- **NoSuchElementException**: WebDriver is unable to identify the elements during run time, i.e. FindBy method can’t find the element.
- NoSuchFrameException: WebDriver is switching to an invalid frame, which is not available.
- NoAlertPresentException: WebDriver is switching to an invalid alert, which is not available.
- NoSuchWindowException: WebDriver is switching to an invalid window, which is not available.
- **StaleElementReferenceException**: The referenced element is no longer present on the DOM page (reference to an element is now Stale). E.g. The Element belongs to a different frame than the current one OR the user has navigated away to another page.
- SessionNotFoundException: The WebDriver is performing the action immediately after ‘quitting’ the browser.
- **TimeoutException**: Thrown when there is not enough time for a command to be completed. E.g. the element didn’t display in the specified time.
- **FileNotFoundException**. This exception is thrown during a failed attempt to open the file denoted by a specified pathname
- Reading local file which is not available on Run Time.//**IO Exception**(means that the input of your code does not seem to be aligning with the output)

ExtentReports = object is used to generate an HTML **report** in the specified location.(under test folder)

- **WebDriver** is the Interface which contains all the Selenium Method signatures (Eg findElement(),switchTo(),get() etc.)
- WebDriver defines common methods which all browser classes (such as Firefox, Chrome etc.,) use. All these class methods are derived from WebDriver interface.

Exceptions

- When the bad user input is given to the program, then it will get halt abruptly. Halting a program abruptly is called as exception.
- In java to handle exception try & catch block will be used.
- Try block will create exception object and the catch block will handle the exception and address of object.
- When ever an exception happens inside the try block, try will create an exception object and reference of that object will given to catch block, so now catch will handle the exception then program will not halt abruptly.

- An Exception is an event, which occurs during the execution of a program, that disrupts the normal flow of the program's instructions or in simple words, any issue which makes your test case stop in between the execution is an Exception.

There are main two types of Exceptions in Selenium WebDriver

- Checked Exceptions – Checked Exceptions are handled during the process of writing codes.These Exceptions can be handled during compile time. If they are not handled, it gives compile time error. Example- FileNotFoundException, IOException etc.
- Unchecked Exceptions – These exceptions can not be handled during compile time & they got caught at run time. Example – ArrayIndexOutOfBoundException.

Locating by **DOM** (Document Object Model) The Document Object Model (DOM), in simple terms, is the way by which HTML elements are structured. Selenium IDE is able to use the DOM in accessing page elements.

ROBOT class

Robot Class

Why Robot Class?
In certain Selenium Automation Tests, there is a need to control keyboard or mouse to interact with OS windows like Download pop-up, Alerts, Print Pop-ups, etc. or native operation System applications like Notepad, Skype, Calculator, etc.

Benefits of Robot Class

- Robot Class can simulate Keyboard and Mouse Event
- Robot Class can help in upload/download of files when using selenium web driver
- Robot Class can easily be integrated with current automation framework (keyword, data-driven or hybrid)

Understanding Robot Class internal methods and usage
Robot Class methods can be used to interact with keyboard/mouse events while doing browser automation. Alternatively AutoIT can be used, but its drawback is that it generates an executable file (exe) which will only work on windows, so it is not a good option to use.

Some commonly and popular used methods of Robot Class during web automation:

- keyPress(): Example: robot.keyPress(KeyEvent.VK_DOWN) : This method with press down arrow key of Keyboard
- mousePress() : Example : robot.mousePress(InputEvent.BUTTON3_DOWN_MASK) : This method will press the right click of your mouse.

- mouseMove() : Example: robot.mouseMove(point.getX(), point.getY()) : This will move mouse pointer to the specified X and Y coordinates.
- keyRelease() : Example: robot.keyRelease(KeyEvent.VK_DOWN) : This method with release down arrow key of Keyboard
- mouseRelease() : Example: robot.mouseRelease(InputEvent.BUTTON3_DOWN_MASK) : This method will release the right click of your mouse

Some of the popular methods under Robot Class are:

- keyPress();
- .mousePress();
- .mouseMove();
- .keyRelease();
- .mouseRelease();

Example:
First of all create the object of the Robot Class as following:
Robot robot=new Robot();

1. .keyPress()
robot.keyPress(KeyEvent.VK_ESC);
This will press Escape key on keyboard.

2. .keyRelease()
robot.keyRelease(KeyEvent.VK_CAPS_LOCK);
This will release the CAPS_LOCK key.

3. .mousePress()
robot.mousePress(InputEvent.BUTTON1_MASK);
This will press Left mouse button.

4. .mouseRelease()
robot.mouseRelease(InputEvent.BUTTON1_MASK);
This will release Left mouse button.

5. .mouseMove()
robot.mouseMove(coordinates.getX(), coordinates.getY());
This will move the mouse pointer to X and Y co-ordinates.
Coordinates is the object reference

Sample code to automate common use cases using Robot Class

Lets take example of web site http://spreadsheetpage.com/index.php/file/C35/P10/ wherein after you click on a web element (.//a[@href=contains(text(),'yearly-calendar.xls']) a O.S download pop-up appears.

To handle this we use Robot class (by creating an instance of Robot Class in your code say Robot robot = new Robot()) . Robot class us present in AWT package of JDK.

- To press down arrow key of Keyboard we use (robot.keyPress(KeyEvent.VK_DOWN))
- To press TAB key of keyboard (we use robot.keyPress(KeyEvent.VK_TAB))
- To press Enter key we use (robot.keyPress(KeyEvent.VK_ENTER)).

Here is a sample code

```
class Excercise1 {

    public static void main(String[] args) throws AWTException, InterruptedException {
        WebDriver driver = new FirefoxDriver();
        driver.get("http://spreadsheetpage.com/index.php/file/C35/P10/"); // sample url
        driver.findElement(By.xpath(".//a[@href=contains(text(),'yearly-calendar.xls')]")).click();
        Robot robot = new Robot();  // Robot class throws AWT Exception
        Thread.sleep(2000); // Thread.sleep throws InterruptedException
        robot.keyPress(KeyEvent.VK_DOWN);  // press arrow down key of keyboard to navigate and select
Save radio button
        Thread.sleep(2000);  // sleep has only been used to showcase each event separately
        robot.keyPress(KeyEvent.VK_TAB);
        Thread.sleep(2000);
        robot.keyPress(KeyEvent.VK_TAB);
        Thread.sleep(2000);
        robot.keyPress(KeyEvent.VK_TAB);
        Thread.sleep(2000);
        robot.keyPress(KeyEvent.VK_ENTER);
     // press enter key of keyboard to perform above selected action
    }
}
```

Example : 02

```
public class Excersise1 {

   @Test
   public static void  execution() throws InterruptedException, AWTException {
      WebDriver driver = new FirefoxDriver();
      driver.manage().window().maximize();
      driver.get("http://spreadsheetpage.com/index.php/file/C35/P10/"); // sample url
      Robot robot = new Robot();
      robot.mouseMove(630, 420); // move mouse point to specific location
```

```
        robot.delay(1500);        // delay is to make code wait for mentioned milliseconds before executing next
step
        robot.mousePress(InputEvent.BUTTON1_DOWN_MASK); // press left click
        robot.mouseRelease(InputEvent.BUTTON1_DOWN_MASK); // release left click
        robot.delay(1500);
        robot.keyPress(KeyEvent.VK_DOWN); // press keyboard arrow key to select Save radio button
        Thread.sleep(2000);
        robot.keyPress(KeyEvent.VK_ENTER);
        // press enter key of keyboard to perform above selected action
    }
}
```

Example : 3 to click enter

```
public class TestLogin {
 @Test
 public void TestRobo()throws Exception
 {
  // Open Firefox
  WebDriver driver=new FirefoxDriver();
  // Maximize the window
  driver.manage().window().maximize();
  // Open facebook
  driver.get("http://www.facebook.com");
  // Enter Username
  driver.findElement(By.id("email")).sendKeys("Selenium@gmail.com");
  // Enter password
  driver.findElement(By.id("pass")).sendKeys("mukesh");
  // Create object of Robot class
  Robot r=new Robot();
   // Press Enter
   r.keyPress(KeyEvent.VK_ENTER);
   // Release Enter
   r.keyRelease(KeyEvent.VK_ENTER);
}
}
```

Example : 04 to upload a file

```
//Store the location of the file in clipboard
                    //Clipboard
                     StringSelection strSel = new StringSelection("C:\\SeleniumResume.doc");
                    Toolkit.getDefaultToolkit().getSystemClipboard().setContents(strSel, null);
```

```
                    // Create object of Robot class
 Robot robot = new Robot();
 Thread.sleep(1000);

  // Press Enter
 robot.keyPress(KeyEvent.VK_ENTER);

// Release Enter
 robot.keyRelease(KeyEvent.VK_ENTER);

  // Press CTRL+V
 robot.keyPress(KeyEvent.VK_CONTROL);
 robot.keyPress(KeyEvent.VK_V);
 // Release CTRL+V
 robot.keyRelease(KeyEvent.VK_CONTROL);
 robot.keyRelease(KeyEvent.VK_V);
 Thread.sleep(1000);

 // Press Enter
 robot.keyPress(KeyEvent.VK_ENTER);
 robot.keyRelease(KeyEvent.VK_ENTER);
```

Disadvantages of Robot Class

Robot framework has few disadvantages mentioned below:

- Keyword/mouse event will only works on current instance of Window. E.g. suppose a code is performing any robot class event, and during the code execution user has moved to some other screen then keyword/mouse event will occur on that screen.
- Most of the methods like mouseMove is screen resolution dependent so there might be a chance that code working on one machine might not work on other.

Summary

Robot class in AWT package is used to generate keyboard/mouse events to interact with OS windows and native apps.

The primary purpose of Robot is to support selenium automated tests project build in Java platform

<u>Difference between action and robot class</u>

With the help of Robot & Actions class,perform Enter :

```
Robot r=new Robot();
r.keyPress(KeyEvent.VK_ENTER);
                                    (or)
Actions action = new Actions(driver);
action.sendKeys(Keys.ENTER).build().perform();
```

Actions Class

- Selenium has the built-in ability to handle various types of keyboard and mouse events. In this post, we'll teach you about the Selenium Actions class which enables user interaction with the web applications.
- To perform mouse operations on a website, first of all, we've to launch the browser and then navigate to the URL of the site. The entire control of the browser as well as the application is in the reference variable of type WebDriver.

How To Use Selenium Actions Class?

First, we'll initialize the WebDriver instance followed by creating the action builder object. Please check out from the below code.

```
WebDriver driver = new FirefoxDriver();
WebElement ele = driver.findElement(By.xpath(……));
Actions ref = new Actions (driver);
ref.movetoElement(ele);
ref.click().build().perform();
```

In the above code, we are executing more than one actions. So we need to use <build()> method to compile all these into a single step.

Selenium Actions Class Methods

Following is the list of most commonly used keyboard and mouse events provided by the Selenium Actions class.

Method **Description**

- <clickAndHold()> - Clicks at the present mouse location (without releasing).
- <contextClick()> - Performs a context-click at the current mouse location.
- <doubleClick()> - It performs a double-click at the existing mouse location.
- <dragAndDrop(source, target)> - Invokes click-and-hold at the source location and moves to the location of the target element before releasing the mouse.

Parameters:
source– element to grab.
target– element to release.
(ex)

```
// Create object of actions class
Actions act=new Actions(driver);
```

```
// find element which we need to drag
WebElement drag=driver.findElement(By.xpath(".//*[@id='draggable']"));

// find element which we need to drop
WebElement drop=driver.findElement(By.xpath(".//*[@id='droppable']"));

// this will drag element to destination
act.dragAndDrop(drag, drop).build().perform();
```

- <dragAndDropBy(source, x-offset, y-offset)>Performs click-and-hold at the source location, shifts by a given offset, then frees the mouse.

Parameters:
source- element to grab.
<xOffset>- to shift horizontally.
<yOffset>- to shift vertically.
(ex)

Drag And Drop Element In Horizontal Direction By 100 Pixel

If you wants to drag and drop element by 100 pixel offset In horizontal direction then you can use syntax like bellow.

```
new Actions(driver).dragAndDropBy(dragElementFrom, 100, 0).build() .perform();
```

Drag And Drop Element In Vertical Direction By 100 Pixel

To drag and drop element by 100 pixel offset In vertical direction, You can use bellow given syntax In your software test case.

```
new Actions(driver).dragAndDropBy(dragElementFrom, 0, 100).build() .perform();
```

Drag And Drop Element In Horizontal And Vertical Direction By -100 Pixel

To drag and drop element by -100 pixel offset In horizontal and vertical direction, You can write your software test case code as bellow. Here -X offset represents right to left direction and -Y offset represents bottom to top direction of web page.

```
new Actions(driver).dragAndDropBy(dragElementFrom, -100, -100).build() .perform();
```

- <moveByOffset(x-offset, y-offset)> Shifts the mouse from its current position (or 0,0) by the given offset.

Parameters:
<x-offset> – Sets the horizontal offset. A (-ve) value means shifting the mouse to the left.
<y-offset> – Sets the vertical offset. A (-ve) value means shifting the mouse to the up.

- <moveToElement(toElement)> - It shifts the mouse to the center of the element.

Parameters:
<toElement> – target element.

(ex.1)
Actions action = new Actions(driver);
WebElement mainMenu = driver.findElement(By.linkText("MainMenu"));
action.moveToElement(mainMenu).moveToElement(driver.findElement(By.xpath("submenuxpath"))).click().build().perform();

(ex.2)
Below is the example to perform mouse hover

WebElement searchBtn = driver.findElement(By.id("searchbtn"));
Actions action = new Actions(driver);
action.moveToElement(searchBtn).perform();

- <release()> Frees up the depressed left mouse button at the existing mouse location.
- <sendKeys(onElement, charsequence)> Transmits a series of keystrokes onto the element.

Parameters:
<onElement> – an element that will receive the keystrokes, usually a text field.
<charsequence> – any string value representing the sequence of keystrokes to be sent.

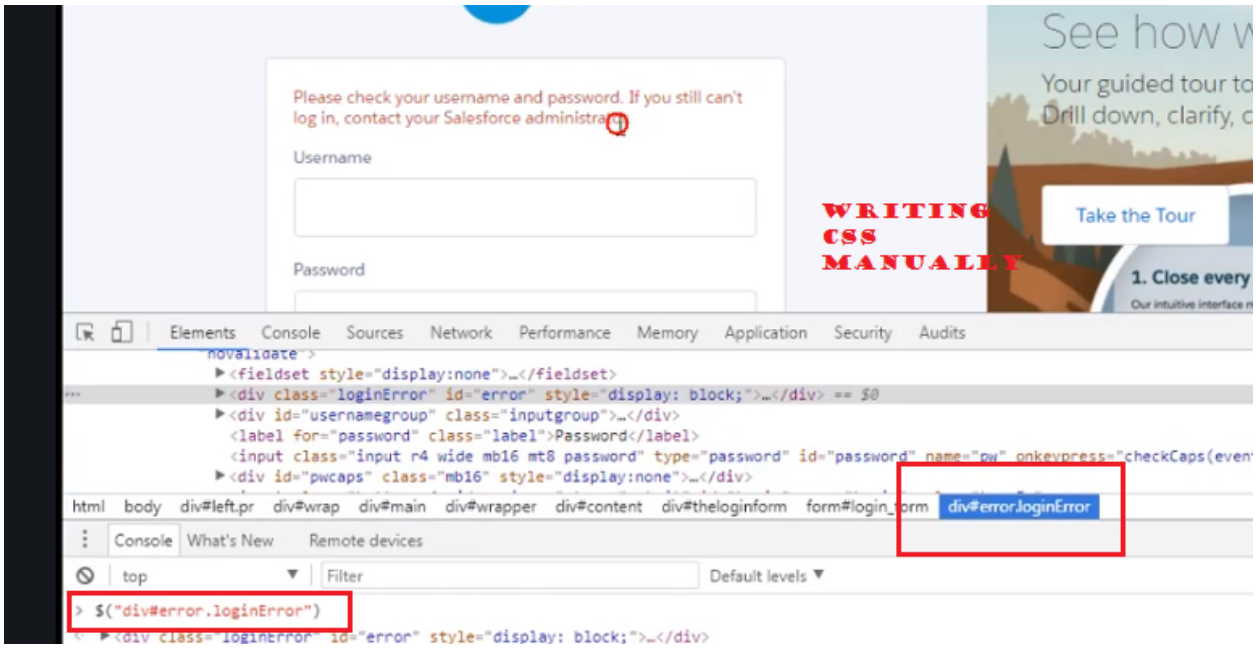
Xpath and CSS

- <input id="u_0_2" here input is the tagname,Id is the attribute
- In the above example id has a alphanumeric value so it may change any time(every refresh) so we always prefer xpath & CSS
- When we see the anchor or a tag you can use the link text attribute

- Need to give as driver.findelements(By.llinktext("Forgot account"));
- If space is there in the class name then we can't use this for inspecting(compound classes will not be accepted error will come)
- Double codes inside double codes not valid and we can write xpath in N number of ways
- Firepath is available with only firefox browser
- When xpath start with html then it's not reliable(so at that time use the chrome browser)
- When there is a no direct way to find css then use the bottom toolbar from chrome to identify
- Derived xpath = $X("locator into goes here") and Derived css = $("locator into goes here") --- to verify the given xpath and css are correct or not(use console)

- $ means driver.findelements

Writing css manually



xpath

We can write as follow

- **//tagname[@attribute='value']** or **//*[@attribute='value']** ---customized **xpath** from html attributes
- * to refer tag name

Xpath syntax: Regular expression

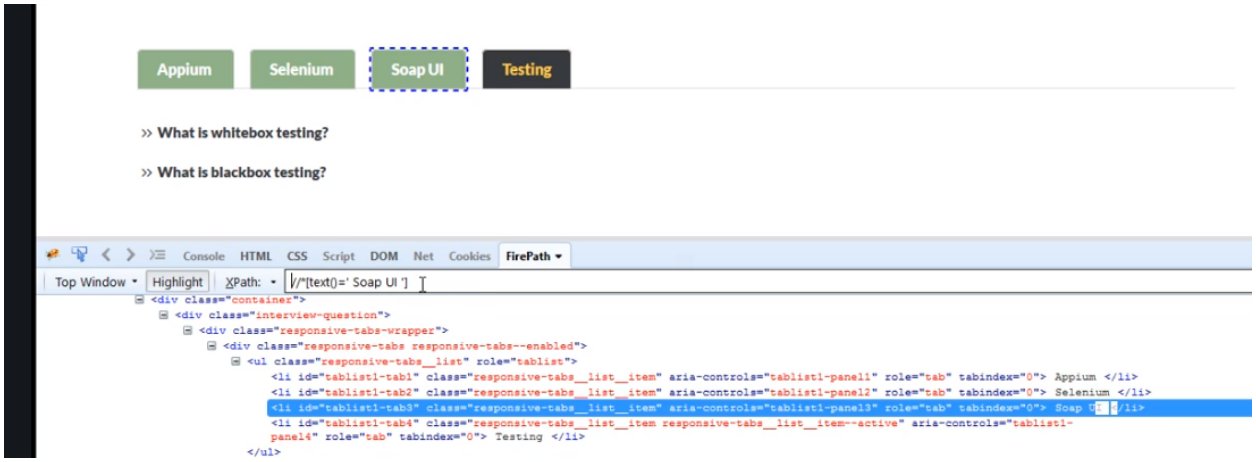
//tagname[contains(@attribute,'value')]--(value not needed a exact match partial value is enough)

Ex. <input name="username123">-------//input[contains(@name,'username')]

- When don't have the id to inspect use "text" to locate as follows

//*[text()=' Selenium ']

Ex.



<u>css</u>

- **tagname[attribute='value']** or **[attribute='value']** ---customized **css** from html attributes Css in another way **"tagname#idvalue","tagname.classvalue"**(we can skip tagname)

CSS syntax: Regular expression
tagname[attribute*='value'] --(value not needed a exact match partial value is enough)

Ex. <input name="username123">-------//input[name*='username']

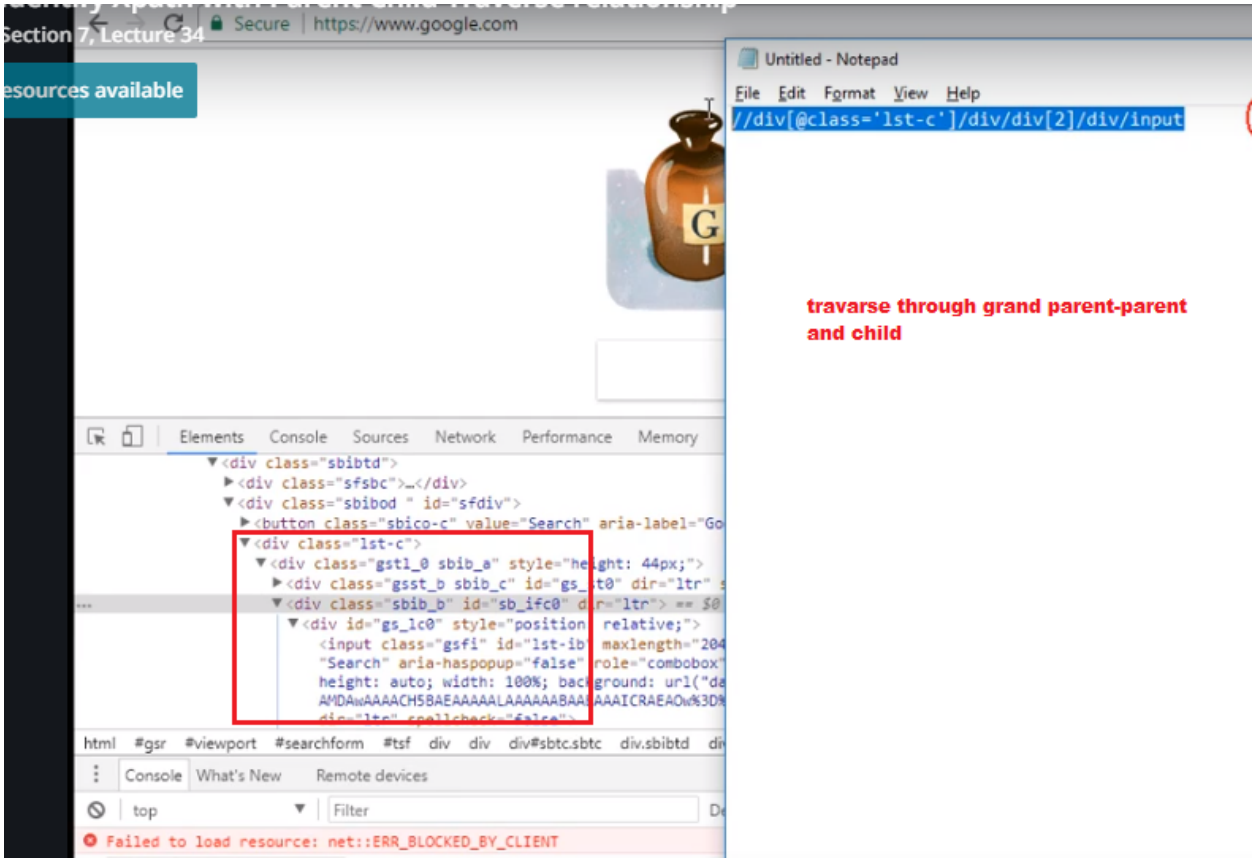
'# hash' represent id
'.dot represent classname

- If space is there between the class value then remove the space and put dot as follows

.input identityinput then we can write as **.input.identityinput**

<u>Parent to child relation</u>

- When we don't have unique or static attribute then we should use the parent to child relation will come into picture



Sharp arrow is the parent

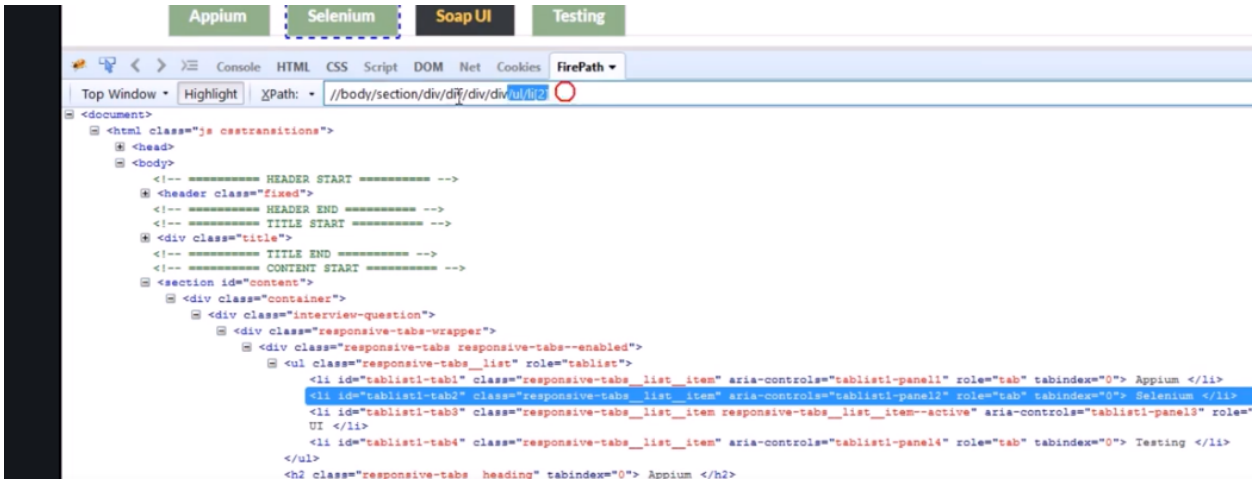
Absolute and Relative Xpath

Absolute xpath:It uses complete path from the root element to the desired element

Relative xpath:You can simply start by referencing the element you want and go from there

- Always relative xpaths are preferred as they are not the complete path from the root element because in future any of the web element when added or removed then absolute xpath changes

Absolute xpath is given below.(traversing to child from parent)

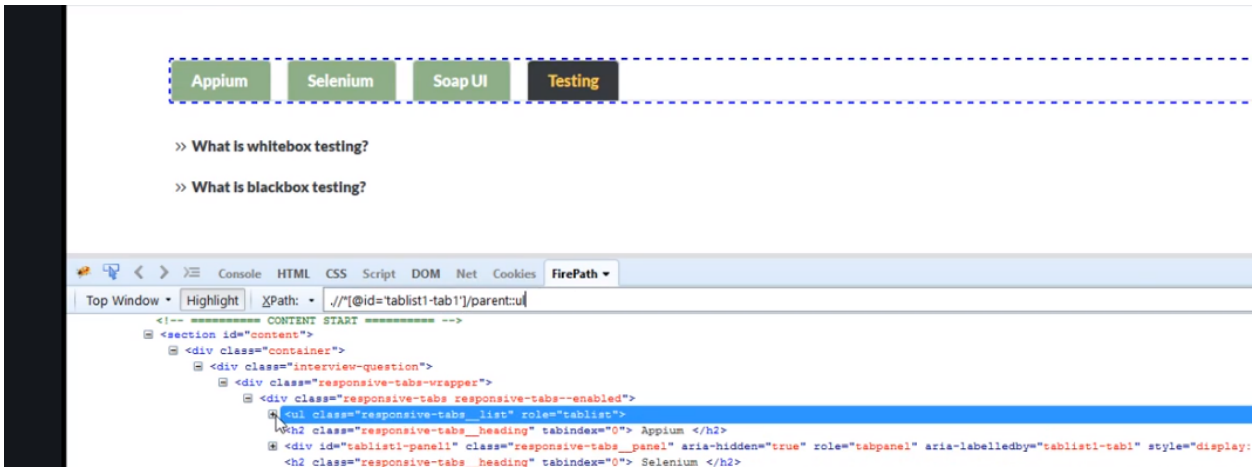


If you want traverse back and forward then use xpath but css can't traverse back so xpath is best

Parent:

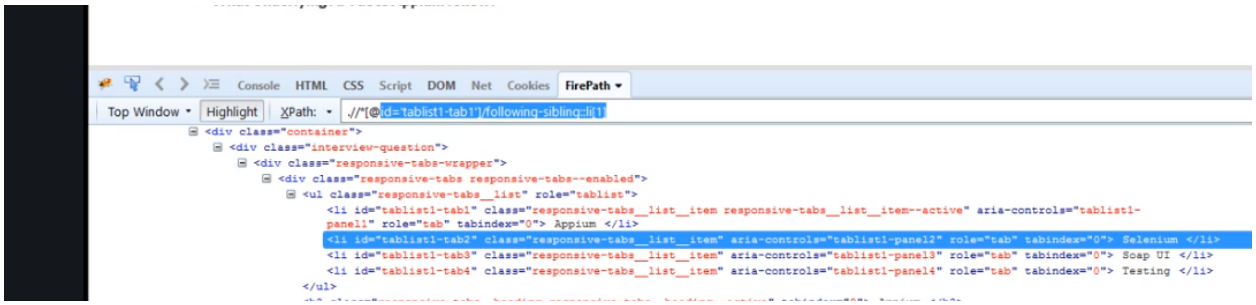
Selects the parent of the current node as shown in the below screen.
Xpath=//*[@id='rt-feature']//parent::div

There are 65 "div" nodes matching by using "parent" axis. If you want to focus on any particular element then you can use the below XPath:
Xpath=//*[@id='rt-feature']//parent::div[1]



Following-sibling:

Select the following siblings of the context node. Siblings are at the same level of the current node as shown in the below screen. It will find the element after the current node.
xpath=//*[@type='submit']//following-sibling::input
Ex.



Preceding:

Select all nodes that come before the current node as shown in the below screen.

In the below expression, it identifies all the input elements before "LOGIN" button that is Userid and password input element.

Xpath=//*[@type='submit']//preceding::input

There are 2 "input" nodes matching by using "preceding" axis. If you want to focus on any particular element then you can use the below XPath:

Xpath=//*[@type='submit']//preceding::input[1]
You can change the xpath according to the requirement by putting [1],[2]……and so on.

Child:
Selects all children elements of the current node (Java) as shown in the below screen.

Xpath=//*[@id='java_technologies']/child::li

There are 71 "li" nodes matching by using "child" axis. If you want to focus on any particular element then you can use the below xpath:

Xpath=//*[@id='java_technologies']/child::li[1]
You can change the xpath according to the requirement by putting [1],[2]……and so on.

Ancestor:
The ancestor axis selects all ancestors element (grandparent, parent, etc.) of the current node as shown in the below screen.

In the below expression, we are finding ancestors element of the current node("ENTERPRISE TESTING" node).

Xpath=//*[text()='Enterprise Testing']//ancestor::div

Dynamic XPath: Dynamic XPath is also called as custom XPath and it is one way to locate element uniquely. Dynamic XPath is used to locate exact attribute or decrease the number of matching nodes/result from a webpage and following XPath expressions can be used for the same:

- Contains
- Sibling
- Ancestor

Contains - Contains is used to locate web element who matches the specific text from multiple blocks. As per below image, if in your web page, there are sections which have the same element for all row, then you can find the specific element by using Contains to select text in that row.

Example:

.//*[@class='product']//h4[contains(text(),'Text')]//ancestor::div[@class='table-good']
.//*[@class='product']//h4[contains(.,'Text')]//ancestor::div[@class='table-good']

Sibling - As the meaning of sibling, we can use this to find an element which is related to some other element. There are basically two types of sibling function which are used in XPath. Preceding Sibling - If we select one sibling from given list, the “preceding sibling” function takes the preceding options of the selected one.

Example:

//ul[@class="OPENCR_flex_containerleft"]//li[@class="OPENCR_flex_box-other"]//select[@name="sort"]//option[contains (text(), 'by title')]/preceding-sibling::option

B) Following Siblings - If we select one sibling from given list, the “following sibling” function takes the following options of selected one. Example:

//ul[@class="OPENCR_flex_containerleft"]//li[@class="OPENCR_flex_box-other"]//select[@name="sort"]//option[contains (text(), 'by title')]/following-sibling::option
Ancestor - We can use this to find an element on basis of the parent element.
As per below image, if in your web page there is a section which has a same element for all the rows then you can find one row element by using ancestor.

Example: In this example, we are trying to find the first row on the basis of text in the row.

//*[text()=’Text']//ancestor::div[@class='table-goods']

Data Driven Framework - is one of the popular Automation Testing Framework in the current market. Data Driven automated testing is a method in which the test data set is created in the excel sheet, and is then imported into automation testing tools to feed to the software under test

Keyword Driven Framework is a type of Functional Automation Testing Framework which is also known as Table-Driven testing or Action Word based testing. The basic working of the Keyword Driven Framework is to divide the Test Case in to four different parts. First is called as Test Step, second is Object of Test Step, third is Action on Test Object and fourth is Data for Test Object.

We can create three types of test framework using Selenium WebDriver. ... In Keyword driven framework, keywords are written in some external files like excel file and java code will call this file and execute test cases. The **hybrid framework** is a mix of keyword driven and data driven framework.

Difference between getWindowHandle and getWindowHandles in selenium

GetWindowHandle Command :- To get the window handle of the current window.
String handle= driver.getWindowHandle();
//Return a string of alphanumeric window handle

GetWindowHandles Command :- To get the window handle of all the current windows.

```
Set<String> handle= driver.getWindowHandles();
//Return a set of window handle
```

(Ex.1) Switching into the child window from parent window

```
 public static void main(String[] args) {
 // Create a new instance of the Firefox driver
        driver = new FirefoxDriver();
// Put an Implicit wait, this means that any search for elements on the page could take the time the implicit wait is set for before throwing exception
        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
        // Launch the URL
        driver.get("http://toolsqa.com/automation-practice-switch-windows/");
        // Store and Print the name of the First window on the console
        String handle= driver.getWindowHandle();
        System.out.println(handle);
        // Click on the Button "New Message Window"
        driver.findElement(By.name("New Message Window")).click();
        // Store and Print the name of all the windows open
         Set handles = driver.getWindowHandles();
         System.out.println(handles);
        // Pass a window handle to the other window
        for (String handle1 : driver.getWindowHandles()) {
         System.out.println(handle1);
         driver.switchTo().window(handle1);
          }
         // Closing Pop Up window
         driver.close();
         // Close Original window
         driver.quit();
 }
}
```

(Ex.2) Switching into the child window from parent window and closing after the last child window

```
 public class MultipleWindowsClass{
 @Test
 public void testMultipleWindows() throws InterruptedException{
 System.setProperty("webdriver.gecko.driver",
System.getProperty("user.dir")+"\\src\\test\\java\\drivers\\geckodriver.exe");
 // To open browser
 WebDriver driver=new FirefoxDriver();
 // To maximize browser
 driver.manage().window().maximize();
```

```
 // To open Naukri website with multiple windows
 driver.get("http://www.naukri.com/");
 // It will return the parent window name as a String
 String mainWindow=driver.getWindowHandle();
 // It returns no. of windows opened by WebDriver and will return Set of Strings
 Set<String> set =driver.getWindowHandles();
 // Using Iterator to iterate with in windows
 Iterator<String> itr= set.iterator();
 while(itr.hasNext()){
 String childWindow=itr.next();
    // Compare whether the main windows is not equal to child window. If not equal, we will close.
 if(!mainWindow.equals(childWindow)){
 driver.switchTo().window(childWindow);
 System.out.println(driver.switchTo().window(childWindow).getTitle());
 driver.close();
 }
 }
 // This is to switch to the main window
 driver.switchTo().window(mainWindow);
 }
}
```

TestNG

Using @Test(enabled = false) this method. We are able to skip / ignore method.
Using @Test(enabled = false) method we can not run the method it simply skip / ignore method and execute next one.

If we want to execute / run method then simply add true above the method e.g: @Test(enabled = true) .

Ex.

```
@Test(enabled = true)
   public void testMethodOne() {
   System.out.println("Test method one.");
   }
   @Test(enabled = false)
   public void testMethodTwo() {
   System.out.println("Test method two.");
   }
```

Using “exclude” parameter in testng.xml:

TestNg provides an option to include or exclude for Groups, Test Methods, Classes and Packages using include and exclude tags by defining in testng.xml.
Only include methods will be Execute / Run other methods will be skip / ignore.

Using <exclude name=”method_name” /> this method. We are able to skip / ignore method.
// Skip / Ignore Test Methods.
<exclude name="method_name" />

// Skip / Ignore Test Methods.
<exclude name="method_name" />

Using < include name=”method_name” /> this method. Execute method properly.
// Execute Test Methods.
<include name="method_name" />

// Execute Test Methods.
<include name="method_name" />

Different types of wait

Implicit Wait

- Selenium Webdriver has borrowed the idea of implicit waits from Watir.
- The implicit wait will tell to the web driver to wait for certain amount of time before it throws a "No Such Element Exception". The default setting is 0. Once we set the time, web driver will wait for that time before throwing an exception.
- In the below example we have declared an implicit wait with the time frame of 10 seconds. It means that if the element is not located on the web page within that time frame, it will throw an exception.

To declare implicit wait:
Syntax:
driver.manage().timeouts().implicitlyWait(TimeOut, TimeUnit.SECONDS);
driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS) ;
Implicit wait will accept 2 parameters, the first parameter will accept the time as an integer value and the second parameter will accept the time measurement in terms of SECONDS, MINUTES, MILISECOND, MICROSECONDS, NANOSECONDS, DAYS, HOURS, etc.

Explicit Wait

- Explicit Wait are used to halt the execution until the time a particular condition is met or the maximum time has elapsed. Unlike Implicit waits, Explicit waits are applied for a particular instance only.
- WebDriver introduces classes like WebDriverWait and ExpectedConditions to enforce Explicit waits

WebDriverWait wait = new WebDriverWait(WebDriverRefrence,TimeOut);
(ex)
WebDriverWait wait = new WebDriverWait(drv,30);
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//div[contains(text(),'COMPOSE')]")));
drv.findElement(By.xpath("//div[contains(text(),'COMPOSE')]")).click();

The following are the Expected Conditions that can be used in Explicit Wait

alertIsPresent()
elementSelectionStateToBe()
elementToBeClickable()
elementToBeSelected()
frameToBeAvaliableAndSwitchToIt()
invisibilityOfTheElementLocated()
invisibilityOfElementWithText()
presenceOfAllElementsLocatedBy()
presenceOfElementLocated()
textToBePresentInElement()
textToBePresentInElementLocated()
textToBePresentInElementValue()
titleIs()
titleContains()
visibilityOf()
visibilityOfAllElements()
visibilityOfAllElementsLocatedBy()
visibilityOfElementLocated()

Fluent Wait

- The fluent wait is used to tell the webdriver to wait for a condition, as well as the frequency with which we want to check the condition before throwing an "ElementNotVisibleException" exception.
- Frequency: Setting up a repeat cycle with the time frame to verify/check the condition at the regular interval of time
- Let's consider a scenario where an element is loaded at different intervals of time. The element might load within 10 seconds, 20 seconds or even more than that if we declare an explicit wait of 20 seconds. It will wait till the specified time before throwing an exception. In such scenarios, the fluentwait is the ideal wait to use as this will try to find the element at different frequency until it finds it or the final timer runs out.

Syntax:
Wait wait = new FluentWait(WebDriver reference)

```
.withTimeout(timeout, SECONDS)
.pollingEvery(timeout, SECONDS)
.ignoring(Exception.class);

(Ex)
Wait<WebDriver> wait = new FluentWait<WebDriver>(driver)
        .withTimeout(30, TimeUnit.SECONDS)
                        .pollingEvery(5, TimeUnit.SECONDS)
                        .ignoring(NoSuchElementException.class);
```

In the above example, we are declaring a fluent wait with the timeout of 30 seconds and the frequency is set to 5 seconds by ignoring "NoSuchElementException"

Synchronization in Selenium Webdriver:

Synchronization is a mechanism which involves two or more components working parallel with each other. Usually, in test automation, there will be two components such as application under test and the test automation tool.

Both of them will have specified speeds and the test scripts should be written in a way such that both these components will work with same speed. This will help to avoid “Element Not Found” error which otherwise will consume more time to clear off. Here the synchronization will come for the help.

Generally, there are two different categories of synchronization in test automation and here is a brief about them.

Unconditional Synchronization

In this case, only the timeout value to be specified. The tool will wait till certain time before proceeding.

Examples: **Wait(), Thread.Sleep()**

The main advantage of this method is that it will come for help when we interact with a third party system such as an interface. Here, it is not possible to write condition or check for a condition. In such cases, the application can be made to wait for a specific period using this type of synchronization. The major disadvantage is that at some times, the tool will be made to wait unnecessarily even when the application is ready.

Conditional Synchronization

In this case, a condition also will be specified along with the timeout value. The tool will wait to check the condition and will come out if nothing happens. However, it is important to set a timeout value also in conditional synchronization so that the tool will proceed even if the condition is not met. There two different types of conditional statements in selenium webdriver and they are an implicit wait,explicit wait and fluent wait.

How to handle Alert in Selenium WebDriver

Alert interface provides the below few methods which are widely used in Selenium Webdriver.

1) void dismiss() // To click on the 'Cancel' button of the alert.
driver.switchTo().alert().dismiss();

2) void accept() // To click on the 'OK' button of the alert.
driver.switchTo().alert().accept();

3) String getText() // To capture the alert message.
driver.switchTo().alert().getText();

4) void sendKeys(String stringToSend) // To send some data to alert box.
driver.switchTo().alert().sendKeys("Text");

Drop down selection code

Select oSelect = new Select(driver.findElement(By.id("yy_date_8")));
oSelect.selectByIndex(4);

Java program

Code to split the given input

```
public class SeparateAlphabetsAndDigitsFromString {
	public static void main(String[] args) {
	String input = "1a2b3c4defg&^$h567jkl";
	separateDigitsAndAlphabets(input);
	}
	public static void separateDigitsAndAlphabets(String str) {
	String number = "";
	String letter = "";
	String special = "";
	for (int i = 0; i < str.length(); i++) {
		char a = str.charAt(i);
		if (Character.isDigit(a)) {
		number = number + a;
		} else if(Character.isAlphabetic(str.charAt(i))){
		  letter=letter+a;
		}
		else {
		special = special + a;
		}
	}
	System.out.println("Alphates in string:"+letter);
```

```
        System.out.println("Numbers in String:"+number);
        System.out.println("Special in String:"+special);
        }
}
```

Output

```
Alphates in string:&^$
Numbers in String:1234567
Special in String:abcdefghjkl
```

Java codes to read

----------Java code ----------

Palindrome or not

---String palindrome-----

```
public class Palindrome
{
   public static void main(String args[])
   {
      String a, b = "";
      Scanner s = new Scanner(System.in);
      System.out.print("Enter the string you want to check:");
      a = s.nextLine();
      int n = a.length();
      for(int i = n - 1; i >= 0; i--)
      {
         b = b + a.charAt(i);
      }
      if(a.equalsIgnoreCase(b))
      {
         System.out.println("The string is a palindrome.");
      }
      else
      {
         System.out.println("The string is not a palindrome.");
      }
   }
}
```

---**Number Palindrome**------

```
class PalindromeExample{
```

```
 public static void main(String args[]){
  int r,sum=0,temp;
  int n=454;//It is the number variable to be checked for palindrome

  temp=n;
  while(n>0){
   r=n%10;  //getting remainder
   sum=(sum*10)+r;
   n=n/10;
  }      if(temp==sum)   {
   System.out.println("palindrome number ");
  else
   System.out.println("not palindrome");
}
}
```

----------**Sort number using array**-----

```
import java.util.Arrays;
public class SortExample
{
        public static void main(String[] args)
        {          // Our arr contains 8 elements
        int[] arr = {13, 7, 6, 45, 21, 9, 101, 102};
        Arrays.sort(arr);
        System.out.printf("Modified arr[] : %s",Arrays.toString(arr));
        }
}
```

Output

```
Modified arr[] : [6, 7, 9, 13, 21, 45, 101, 102]

class SortArrayExample {

  public static void main(String[] args) {
        // int Array
        Integer[] intArray = new Integer[] {
        new Integer(15),
        new Integer(9),
        new Integer(16),
        new Integer(2),
        new Integer(30)
        };

        // Sorting int Array in descending order
```

```
        Arrays.sort(intArray, Collections.reverseOrder());

        // Displaying elements of int Array
        System.out.println("Int Array Elements in reverse order:");
        for (int i = 0; i < intArray.length; i++)
        System.out.println(intArray[i]);

        // String Array
        String[] stringArray =
        new String[] { "FF", "PP", "AA", "OO", "DD" };

        // Sorting String Array in descending order
        Arrays.sort(stringArray, Collections.reverseOrder());

        // Displaying elements of String Array
        System.out.println("String Array Elements in reverse order:");
        for (int i = 0; i < stringArray.length; i++)
        System.out.println(stringArray[i]);
  }
}
```

Output

```
Int Array Elements in reverse order:
30
16
15
9
2
String Array Elements in reverse order:
PP
OO
FF
DD
AA
```

-------split the input by space-------

```
public class GFG {
   public static void main(String args[])
   {
      String str = "Geeks for Geeks";
      String[] arrOfStr = str.split(" ");  or String[] arrOfStr = str.split("for ");

      for (String a : arrOfStr)
```

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

Output
Geeks
for
Geeks

If we use **for** then output will be like
Geeks
Geeks

Code to swap two numbers

```
class demo
{
    public static void main(string arg[])
    {
        System.out.println("Before swapping");
        int x=10;
        int y=20;
        System.out.println("value of x:"+x);
        System.out.println("value of y:"+y);
        system.out.println("After swapping");
        x=x+y;//without third variable                  temp = x;   ///using temp
        y=x-y;                  or                          x = y;
        x=x-y;                                              y = temp;



        System.out.println("value of x:"+x);
        System.out.println("value of y:"+y);
    }
}
```

--------**To find the duplicate value**-------

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

        //Initialize array
        int [] arr = new int [] {1, 2, 3, 4, 2, 7, 8, 8, 3};
```

```
        System.out.println("Duplicate elements in given array: ");
        //Searches for duplicate element
        for(int i = 0; i < arr.length; i++) {
            for(int j = i + 1; j < arr.length; j++) {
                if(arr[i] == arr[j])
                    System.out.println(arr[j]);
            }
        }
    }
}
```

Output

Duplicate elements in given array:
2
3
8

Anagram or not?
https://www.javatpoint.com/java-program-to-check-whether-two-strings-are-anagram-or-not

Sort number
https://www.javatpoint.com/java-program-to-sort-the-elements-of-an-array-in-descending-order

Program to find the second largest number

```
import java.util.Arrays;
public class LargestNumberSample {
   public static void main(String args[]){
      int array[] = {10, 20, 25, 63, 96, 57};
      int size = array.length;
      Arrays.sort(array);
      System.out.println("sorted Array ::"+Arrays.toString(array));
      int res = array[size-2];
      System.out.println("2nd largest element is ::"+res);
   }
}
```

Output

sorted Array ::[10, 20, 25, 57, 63, 96]
2nd largest element is ::63

Armstrong

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

```
   int c=0,a, temp;
   int n=153;//It is the number to check armstrong
   temp=n;
   while(n>0)
   {
   a=n%10;
   n=n/10;
   c=c+(a*a*a);
   }
   if(temp==c)
   System.out.println("armstrong number");
   else
      System.out.println("Not armstrong number");
  }
}
```

Output

ex:153

Reverse a string using string buffer

```
public class StringReverseExample{
   public static void main(String[] args) {
      String string = "abcdef";
      String reverse = new StringBuffer(string).reverse().toString();
      System.out.println("\nString before reverse: "+string);
      System.out.println("String after reverse: "+reverse);
   }
}
```

Output

The above code sample will produce the following result.

```
String before reverse:abcdef
String after reverse:fedcba
```

Code to find the duplicate character

```
public class DuplStr {
 public static void main(String argu[]) {
  String str = "w3schools";
  int cnt = 0;
  char[] inp = str.toCharArray();
  System.out.println("Duplicate Characters are:");
```

```
  for (int i = 0; i < str.length(); i++) {
   for (int j = i + 1; j < str.length(); j++) {
    if (inp[i] == inp[j]) {
     System.out.println(inp[j]);
     cnt++;
     break;
    }
   }
  }
 }
}
```

Program Output:

```
Duplicate Characters are: s o
```

Remove duplicate character from string

```
class GFG
{
	static String removeDuplicate(char str[], int n)
	{
		// Used as index in the modified string
		int index = 0;
		// Traverse through all characters
		for (int i = 0; i < n; i++)
		{
			// Check if str[i] is present before it
			int j;
			for (j = 0; j < i; j++)
			{
				if (str[i] == str[j])
				{
					break;
				}
			}
			// If not present, then add it to
			// result.
			if (j == i)
			{
				str[index++] = str[i];
			}
		}
		return String.valueOf(Arrays.copyOf(str, index));
```

```
        }

        // Driver code
        public static void main(String[] args)
        {
                char str[] = "geeksforgeeks".toCharArray();
                int n = str.length;
                System.out.println(removeDuplicate(str, n));
        }
}
```

Output Geksfor

code to get the tomorrow's date as mentioned format (dd-mm-yyyy)

```
import java.text.SimpleDateFormat;
import java.util.Date;
class TestDates_Format {
 public static void main(String args[]) {
  Date objDate = new Date(); // Current System Date and time is assigned to objDate
  System.out.println(objDate);
  String strDateFormat = "hh:mm:ss a dd-MMM-yyyy"; //Date format is Specified
  SimpleDateFormat objSDF = new SimpleDateFormat(strDateFormat); //Date format string is
passed as an argument to the Date format object
  System.out.println(objSDF.format(objDate)); //Date formatting is applied to the current
date
 }
}
```

Hasp Map to store the two set of values

```
public class SingleKeyMultipleValueUsingList {

   public static void main(String[] args) {

      // create map to store

      Map<String, List<String>> map = new HashMap<String, List<String>>();

      // create list one and store values

      List<String> valSetOne = new ArrayList<String>();

      valSetOne.add("Apple");

      valSetOne.add("Aeroplane");

      // create list two and store values

      List<String> valSetTwo = new ArrayList<String>();
```

```
        valSetTwo.add("Bat");
        valSetTwo.add("Banana");
        // create list three and store values
        List<String> valSetThree = new ArrayList<String>();
        valSetThree.add("Cat");
        valSetThree.add("Car");
        // put values into map
        map.put("A", valSetOne);
        map.put("B", valSetTwo);
        map.put("C", valSetThree);
        // iterate and display values
        System.out.println("Fetching Keys and corresponding [Multiple] Values n");
        for (Map.Entry<String, List<String>> entry : map.entrySet()) {
            String key = entry.getKey();
            List<String> values = entry.getValue();
            System.out.println("Key = " + key);
            System.out.println("Values = " + values + "n");
        }
    }
}
```

Print highest two numbers

```
public class TwoMaxNumbers {
        public void printTwoMaxNumbers(int[] nums){
        int maxOne = 0;
        int maxTwo = 0;
        for(int n:nums){
        if(maxOne < n){
                maxTwo = maxOne;
```

```
                maxOne =n;
        } else if(maxTwo < n){
                maxTwo = n;
        }
        }
        System.out.println("First Max Number: "+maxOne);
        System.out.println("Second Max Number: "+maxTwo);
        ------Add here
int total=maxOne+maxTwo;
System.out.println("addition is: "+total); ------
}
        public static void main(String a[]){
        int num[] = {5,34,78,2,45,1,99,23};
TwoMaxNumbers tmn = new TwoMaxNumbers();
        tmn.printTwoMaxNumbers(num);
        }
}
```

Remove duplicate string

String reverse

```
public class String_Get_Small_Letters {
public static void main(String[] args) {
        // TODO Auto-generated method stub
        String text = "This IS My TEXT StrinG";
        //String text = "45))=:&TestG";
        String test = text.replaceAll("[^a-z]","");
        System.out.println(test);
        StringBuilder sb = new StringBuilder(text);
        sb.reverse();
        System.out.println(sb);
        }
}
```

Output
Output for Uppercase removed string - hisytrin
Output for reverse string - GnirtS TXET yM SI sihT

Program to remove duplicate

```
public class Remove_duplicate_Values_fromArray {
        public static void main(String[] args) {
                String[] array = {"test","Abc1","test","Abc","Abc"};
                LinkedHashSet<String> linkedHashSet = new LinkedHashSet<String>(Arrays.asList(array));
                String[] newArray = linkedHashSet.toArray(new String[linkedHashSet.size()]);
                System.out.println("Array after removing duplicates: "
           + Arrays.toString(newArray));
        }
}
```

OUTPUT : Array after removing duplicates: [test, Abc1, Abc]

```
public class ReverseWords {

   public static void main(String[] args)
   {
      String s[] = "i like this program very much".split(" ");
      String ans = "";
      for (int i = s.length - 1; i >= 0; i--) {
         ans += s[i] + " ";
      }
      System.out.println("Reversed String:");
      System.out.println(ans.substring(0, ans.length() - 1));
   }
}
```

Output:
Reversed String:
much very program this like i

JAVA

Java constructor inheritance?

https://stackoverflow.com/questions/15721764/java-constructor-inheritance

String predefined methods

https://courses.cs.washington.edu/courses/cse341/98au/java/jdk1.2beta4/docs/api/java/lang/String.html

finding a div with a class/id and verifying text inside
https://stackoverflow.com/questions/18570622/selenium-and-xpath-finding-a-div-with-a-class-id-and-verifying-text-inside

Verifying table content

https://stackoverflow.com/questions/36848352/how-to-select-value-from-dropdown-without-using-select-class-becuase-in-have-dr

As in **encapsulation**, the data in a class is hidden from other classes, so it is also known as data-hiding. Encapsulation can be achieved by: Declaring all the variables in the class as private and writing public methods in the class to set and get the values of variables.

```
// Java program to demonstrate encapsulation
public class Encapsulate
{
        // private variables declared
        // these can only be accessed by
        // public methods of class
        private String geekName;
        private int geekRoll;
        private int geekAge;

        // get method for age to access
        // private variable geekAge
        public int getAge()
        {
        return geekAge;
        }

        // get method for name to access
```

```
public class TestEncapsulation
{
        public static void main (String[] args)
        {
        Encapsulate obj = new Encapsulate();

        // setting values of the variables
        obj.setName("Harsh");
        obj.setAge(19);
        obj.setRoll(51);

        // Displaying values of the variables
        System.out.println("Geek's name: " +
obj.getName());
        System.out.println("Geek's age: " +
obj.getAge());
        System.out.println("Geek's roll: " +
obj.getRoll());
```

```
        // private variable geekName
        public String getName()
        {
        return geekName;
        }

        // get method for roll to access
        // private variable geekRoll
        public int getRoll()
        {
        return geekRoll;
        }

        // set method for age to access
        // private variable geekage
        public void setAge( int newAge)
        {
        geekAge = newAge;
        }

        // set method for name to access
        // private variable geekName
        public void setName(String newName)
        {
        geekName = newName;
        }

        // set method for roll to access
        // private variable geekRoll
        public void setRoll( int newRoll)
        {
        geekRoll = newRoll;
        }
}
```

```
        // Direct access of geekRoll is not possible
        // due to encapsulation
        // System.out.println("Geek's roll: " +
obj.geekName);
        }
}
```

<u>This Keyword</u>

- "This" keyword used to point the current object executing
- It is a keyword created by JVM
- We can't use this in the static context or method
- Using this keyword we can access both the static & non-static member
- We can call static method using this keyword but we can't use the this keyword inside the static method

Ex .

```
class Qy {
int i=10;
public static void main(String[] args) {
Qy r1=new Qy();
System.out.println(r1.i);
r1.test();
}
public void test(){
System.out.println(this.i);
}
}
```

Output

```
10
10
```

- Using this keyword we can call the constructor of same class, to do that 'this keyword' should placed inside the another constructor as a very first statement

Ex 85. Passing the value through this keyword

```
class Qy {
int i;
Qy(int i){
   System.out.println(i);
}
Qy(){
   this(100);
   System.out.println("one");
}
public static void main(String[] args) {
new Qy();
}
}
```

Super Keyword

- Using the "super" keyword we can access the members of parent class
- Using super keyword we can access both static & Non static member
- Super keyword can't be used inside the static method or context

Ex. Accessing variables

class A {
int i=8;
}
	class B extends A{
	public static void main(String[] args) {
	B r1=new B();
	r1.test();

Output
8
	}
	public void test() {
	System.out.println(super.i);
	}
	}

Ex . Accessing Methods

class A {
public void test(){
System.out.println("print");
}
}
 	
Output
print
	class B extends A{
	public static void main(String[] args) {
	B r1=new B();
	r1.xyz();
	}
	public void xyz() {
	super.test();
	}
	}

- Using the super keyword we can call the constructor of parent class but we should use super keyword in child class constructor as a very first statement as above
- If we didn't keep the super keyword, compiler will automatically keep the keyword but it will only call the no args constructor of parent class
- If you didn't create the child class constructor with no args, then that block also will kept by compiler
- Super keyword kept inside the child constructor with or without arguments as follows

Ex 90.

class T {
T(){
System.out.println("Test");
}
}
 	
Output
Test
1
	class D extends T{
	D(int i){
	//super();
	System.out.println(i);
	}
	public static void main(String[] args) {
	new D(1);
	}
	}

- If in the parent class only one constructor available with argument means as a programmer we should explicitly write the super keyword in the child class constructor (as super keyword with argument)
- Super keyword will not placed automatically if parent class has the only constructor with arguments & compiler will place the super keyword in child class when there is a constructor without arguments in parent class

Object class in Java. Object class is present in java.lang package. Every class in Java is directly or indirectly derived from the Object class. If a Class does not extend any other class then it is direct child class of Object and if extends other class then it is an indirectly derived.

protected Object **clone**() throws CloneNotSupportedException
Creates and returns a copy of this object.

public **boolean equals**(Object obj)
Indicates whether some other object is "equal to" this one.

protected void **finalize**() throws Throwable
Called by the garbage collector on an object when garbage collection determines that there are no more references to the object

public int **hashCode**()
Returns a hash value that is used to search object in a collection

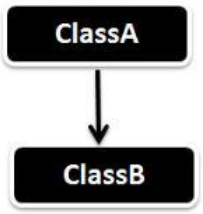
public String **toString**()
toString() provides String representation of an Object and used to convert an object to String.

Inheritance

- By using the inheritance we reuse the non static members of parent class in child class by creating object
- Static members of parent class will not get inherited but it will give the feel of inheritance(conversion by JVM)
- Inheritance means moving the value of member into child class object memory

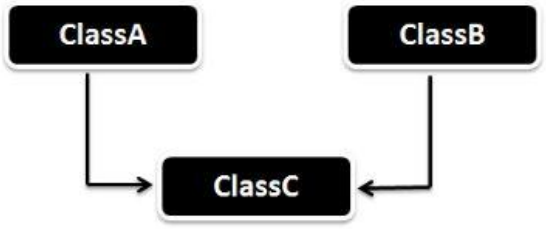
class T {
  int i=5;
  static int j=10;
}
 	
Output
5
	class D extends T{
	  public static void main(String[] args) {
	  D r=new D();
	  System.out.println(r.i); or (r.j) JVM = T.j
	  }
	}

Below are the **different types of inheritance** which is supported by Java.
Single Inheritance

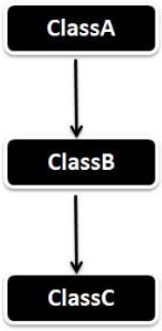


Multiple Inheritance (Through Interface)

But you can achieve multiple inheritance in Java using **Interfaces.**

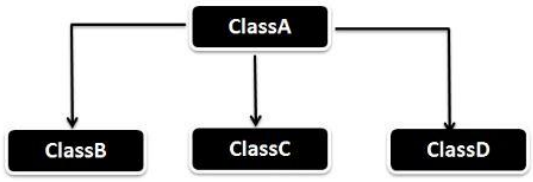


Multilevel Inheritance

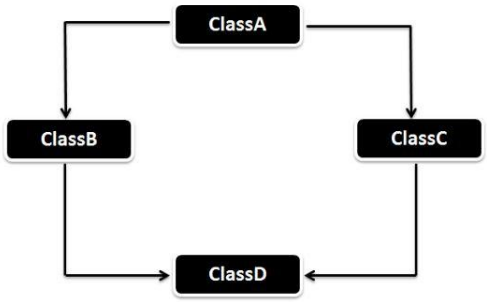


Hierarchical Inheritance

ClassA will be acting as a parent class for ClassB, ClassC and ClassD.



Hybrid Inheritance (Through Interface)



Surefire plugin: It generates reports in two different file formats: Plain text files (*.txt) XML files (*.xml)

Data providers

```
@DataProvider
 public Object[][] getData(){
        Object[][] data=new Object[1][2];
        data[0][0]="username";
        data[0][1]="Password";
        //data[0][2]="Non restricted user";
        return data;
        }

@Test(dataProvider="getData")
        public void initialize(String username,String password) throws IOException{

}
```

Final

1.Final is used to apply restrictions on class, method and variable. Final class can't be inherited, final method can't be overridden and final variable value can't be changed.
2.Final is a keyword.

Finally

1.Finally is used to place important code, it will be executed whether exception is handled or not.
2.Finally is a block.

finalize

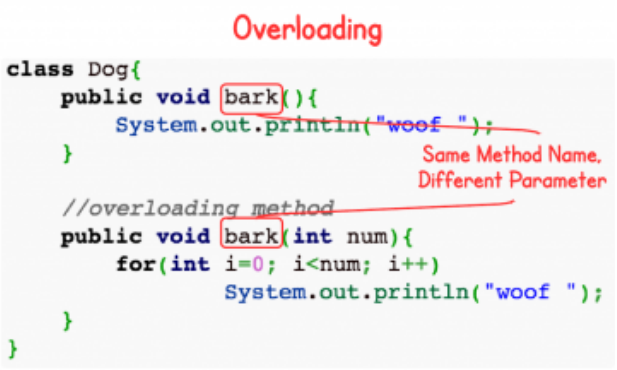
1.Finalize is used to perform clean up processing just before object is garbage collected.
2.Finalize is a method.

```
class FinalizeExample{
```

```
public void finalize(){System.out.println("finalize called");}
public static void main(String[] args){
FinalizeExample f1=new FinalizeExample();
FinalizeExample f2=new FinalizeExample();
f1=null;
f2=null;
System.gc();
}}
```

Polymorphism

Polymorphism - is the ability of an object to take on many forms.
Overloading occurs when two or more methods in one class have the same method name but different parameters. **Overriding** means having two methods with the same method name and parameters (i.e., method signature). One of the methods is in the parent class and the other is in the child class.



Overriding

- After inheriting method from parent class, if you want to modify that inherited method, then we use the concept of overriding
- Inheritance is mandatory for overriding
- Overriding is not at all possible for variable, we can only able to override a method
- Whenever we need to modify a inherited method we should use override

Ex 106.

package T;
public class Q1 {
  public void test() {
  System.out.println("a");
  }
}
 	
Output
b
a
	package T;
	public class Test extends Q1 {
	  public void test(){
	  System.out.println("b");
	  }
	  public static void main(String[] args) {
	  Test r=new Test();
	  r.test();
	  Q1 r1=new Q1();
	  r1.test();
	  }
	}

- Static member of overriding is not possible
- While overriding if you are increasing the scope of the access specifier then it will not give any error i.e if class is a private one, then inheritance will not happen so overriding also not possible

Overriding = Runtime polymorphism
Overloading = Compile time polymorphism
Overriding will happen while running the dot class file
Overloading will happen while generating the dot class file

Overloading

- In overloading we can create multiple methods with the same name but then they are differentiated based on the type of arguments

Ex 108.

```
public class Q1 {
  public void  test() {
  System.out.println("Test");
  }
  public void  test(int a,int b) {
  System.out.println(a);
  System.out.println(b);
}
  public static void main(String[] args) {
  Q1 r=new Q1();
  r.test(10,20);
  r.test();
  }
}
```

Output

10,20,Test

Access specifier

	Private	**Default**	**Protected**	**Public**
Same class	Yes	Yes	Yes	Yes
Same package & subclass	No	Yes	Yes	Yes
Same package & non subclass	No	Yes	Yes	Yes
Different package & subclass	No	No	Yes	Yes
Different package & non subclass	No	No	No	Yes

Key Difference Between String and StringBuffer.
The length of String object is fixed but the length of an object of StringBuffer can be increased when required. String object is immutable i.e. it's object can't be reassigned again whereas, the object of StringBuffer is mutable.

1.The length of String object is fixed but the length of an object of StringBuffer can be increased when required.
2.String object is immutable i.e. it’s object can’t be reassigned again whereas, the object of StringBuffer is mutable.
3.String object is slower in performance whereas, the StringBuffer object is faster.
4.String object consumes more memory whereas, StringBuffer objects consumes less memory.
(Ex.)StringBuffer Sb= new StringBuffer ("Teckpix");
Sb.append("Solution");
system.out.println(Sb);
// Output : Teckpix Solution

Abstract class vs Interface

Interface:

- Interfaces are 100% abstract or 100% incomplete(will not have complete method)
- Incomplete method inherited from an interface and should be completed in a class
- Abstract is a keyword in java which is used to specify the incomplete method
- In an interface every method by default is abstract,hence it's not mandatory to add abstract keyword and also by default method is public in interface
- Reference variable of an interface can be created but object can't be created

  a=interface,b=class

  A r= new b();

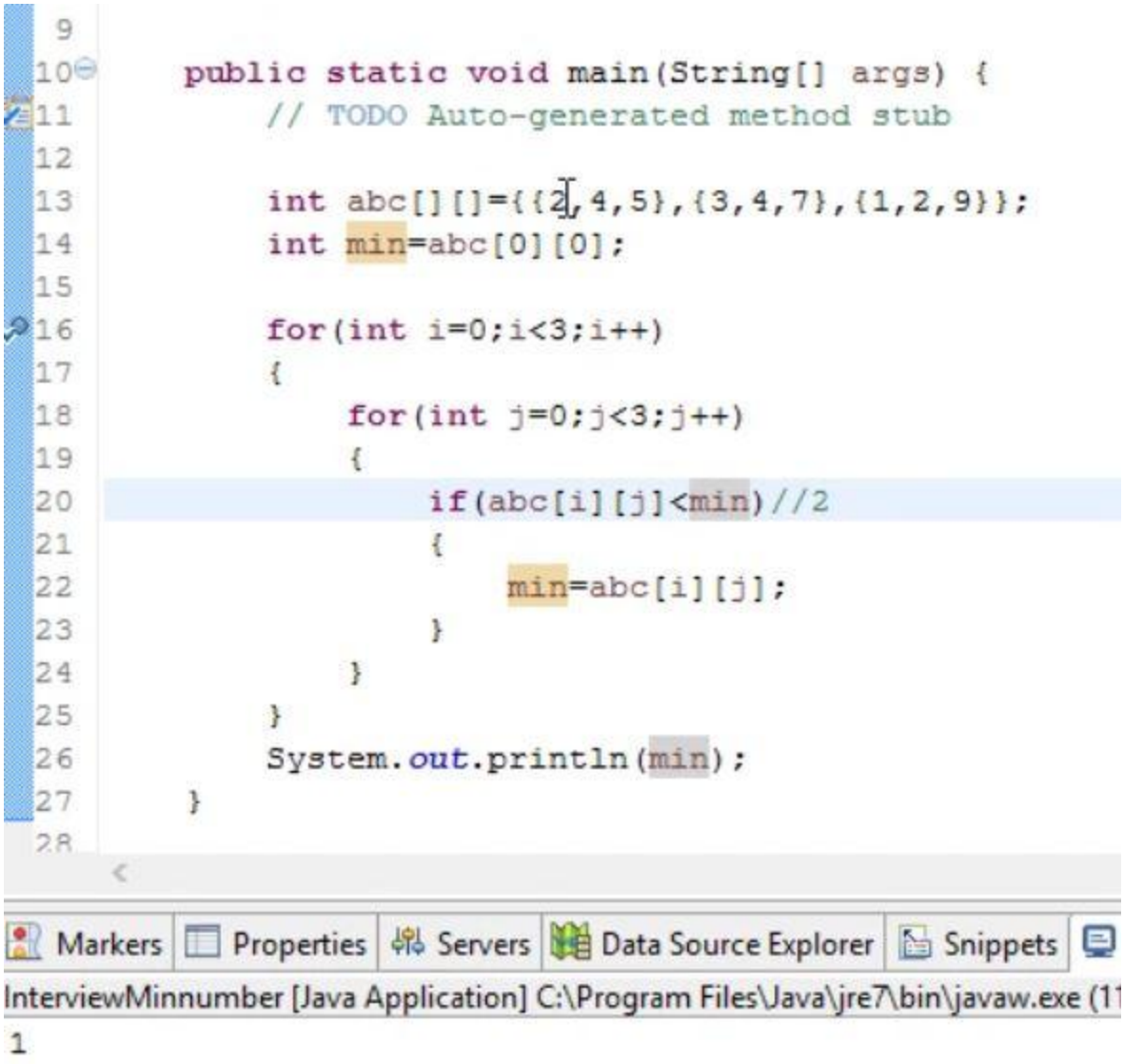
- We can't create the static method in an interface as they can't be inherited & if they are not inherited then we can't override it

Abstract:

- Abstract class is 0 to 100% incomplete(can have both complete & incomplete method)
- Every method should have abstract keyword to specify it's incomplete
- We can create the object of abstract class
- We can't create the static method in abstract class

Interface	**Abstract**
Interface can have only abstract methods.	Abstract class can have abstract and non-abstract methods
Variables declared in a Java interface are by default **final**.	An abstract class may contain non-final variables.
Interface has only **static** and final variables.	Abstract class can have final, non-final, static and non-static variables.
A Java interface can be implemented using keyword "implements"	abstract class can be extended using keyword "extends".
Members of a Java interface are **public** by default.	A Java abstract class can have class members like private, protected,
We can't keep the main method inside the interface	We can keep the main method inside the abstract class
We can't keep the constructor inside the interface	We can keep the constructor inside the abstract class

Array code to find min value



```
9
10      public static void main(String[] args) {
11          // TODO Auto-generated method stub
12
13          int abc[][]={{2,4,5},{3,4,7},{1,2,9}};
14          int min=abc[0][0];
15
16          for(int i=0;i<3;i++)
17          {
18              for(int j=0;j<3;j++)
19              {
20                  if(abc[i][j]<min)//2
21                  {
22                      min=abc[i][j];
23                  }
24              }
25          }
26          System.out.println(min);
27      }
28
```

Markers | Properties | Servers | Data Source Explorer | Snippets

InterviewMinnumber [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (11

1

What is a Collection?

A collection is an object that represents a group of individual objects represented as a single unit. Collection is a dynamic array where we can add elements and reduce elements.

Arrays vs Collections:

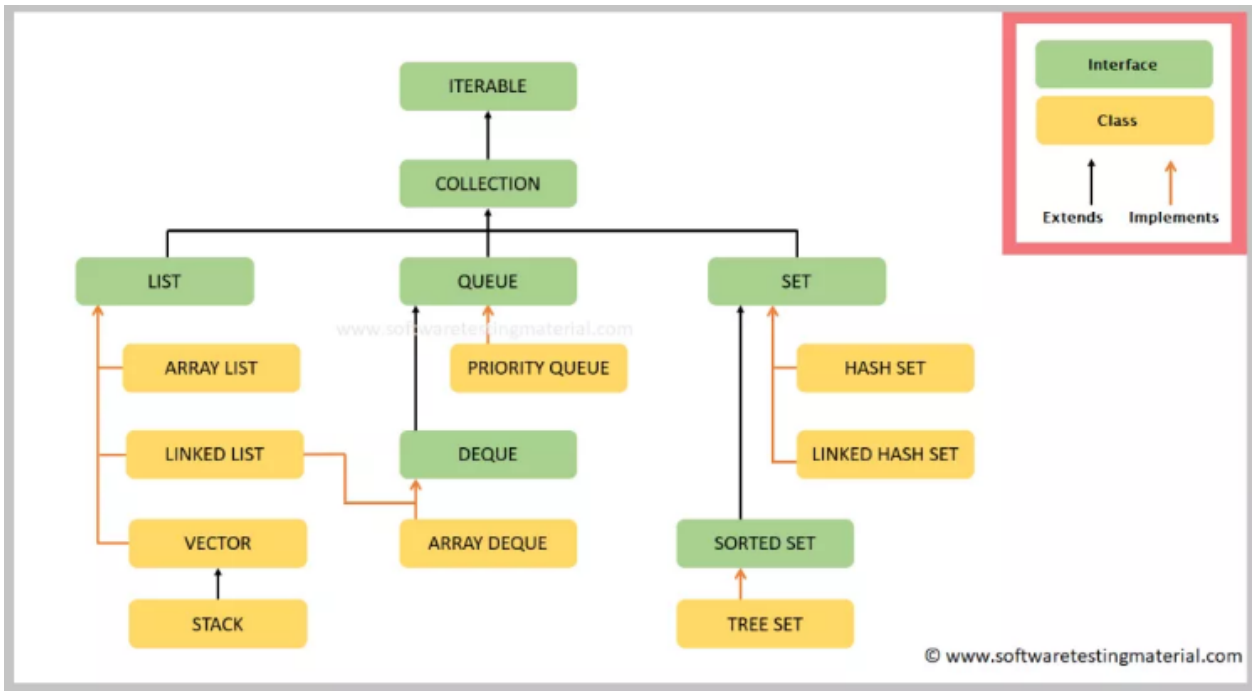
Both Collections and Arrays hold references to Objects and can be managed as a group.

Collections do not need to be assigned a certain capacity when instantiated where as Arrays need to be assigned certain capacity when instantiated.

Collections can grow and shrink in size automatically when objects are added or removed where as Arrays contain fixed size.

Collections cannot hold basic data type elements (primitive types such as int, long, or double; instead, they hold Wrapper Classes such as Integer, Long, or Double) where as Arrays hold basic data type elements.

Types of collections:



The extends keyword is mainly used to extend a class i.e. to create a subclass in Java, while implements keyword is used to **implement** an interface in Java. The extends keyword can also be used by an interface for extending another interface.

Collections is nothing but a backpack, guess what are the things you can store in a backpack.

Probably you must have guessed everything that we can store in backpack, but let list few of them : 1. Laptop, Mouse, HeadPhones, earPhones, Mouse (computer's and animal too), Book, Notepad, pen, pencil you can write so on.
Almost you can store everything, you cannot store elephant though.

Collections in java is a framework that provides an architecture to store and manipulate the group of objects Collections in java

List

List is an ordered collection and can contain duplicate elements. You can access any element from its index.

List is more like array with dynamic length, List is one of the most used Collection type.

ex.Library having racks with index

List is a interface, and it just provides what should be present if somebody implements the interface, so the list interface will not have any implementations.

ArrayList and LinkedList classes provide implementation to the List interface methods, the methods present in the ArrayList and LinkedList are same with little changes

Find the text of radio buttons and print it onto the output console

```
package com.sample.stepdefinitions;
import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

public class NameDemo {
public static void main(String[] args) {
   System.setProperty("webdriver.chrome.driver", "X://chromedriver.exe");
   WebDriver driver = new ChromeDriver();
   driver.get("http://demo.guru99.com/test/ajax.html");
   List<WebElement> elements = driver.findElements(By.name("name"));
   System.out.println("Number of elements:" +elements.size());
   for (int i=0; i<elements.size();i++){
     System.out.println("Radio button text:" + elements.get(i).getAttribute("value"));
   } } }
```

Methods in List

- size() - Returns the number of elements in this List. ex.public int size()
- isEmpty - public boolean isEmpty()
  Returns true if this List contains no elements.
- Contains - public boolean contains(Object o)
  Returns true if this List contains the specified element.

clear():
clear() method is used for removing all the elements of the list in one go, this will not delete the list but elements in the list.

SET
The Set interface extends the Collection interface. It will make sure that an instance of Set contains no duplicate elements.

Set is nothing but a sack, which can contains any objects inside the sack

Because sack does not maintain any order, so when you pick the pebble it will lead to the confusion that which one you want to access(intention), so due to this issue Set type of collection in java will not allow you put duplicates in set.
Set stores the values in random order and accessing the elements happens through element Value rather than index.

HashSet, LinkedHashSet, TreeSet are the classes which implement the Set interface

When to use List and Set
We cannot use list and set interchangeably unless it is very required, let's see when to use list and when to use set.

1. If You want a Collection that does not store duplicate values, then we use a Set based collection.
2. If You want to frequently access elements operations based on an index value then we use a List based collection. E.g. ArrayList
3. If You want to maintain the insertion order of elements in a collection then we use a List based collection.
4. For fast search operation based on a key, value pair, we use a HashMap based collection.
5. If You want to maintain the elements in a sorted order, then we use a TreeSet based collection

Map in Java

- Maps are not part of collection but built based on the collection concepts
- Map is nothing but a key value pair mappings, every in the map a has a value and every value has a key.
- Map keys follows Set interface and allows only unique values, Map Values are following List interface and allows duplicates
- It is similar to the attendance system, the roll number is Keys, and names are Values. Roll Number is unique but the names could be duplicates.

Generics in java

In my Way : Collection is like road, any vehicle can go on the road. Consider that it is peak time and you have heavy traffic(2, 4, 6 wheelers) on the road, now you feel like you want to make all the vehicles to into two wheelers(I'm assuming that you are not having super powers), is it possible ? No. If you force to convert, you may have issue* in converting

Consider that there is road which allows only two wheelers, now it has traffic, will you able convert them into two wheelers ?, Yes, because we allowed only two wheelers on the road.

Coming to the Generic, Generic is nothing but the road which accepts everything, Generic provides way to make it accept only one kind of value.

Below code shows that the ArrayList accepts only the type of String values into the List.
Common syntax:
List< Generic > = new ArrayList();
ArrayList <String> products = new ArrayList();
When we did not mention any generic type then collection will accept everything but if we mention some type then collection will accept only that types of data.

Why do need Generic :
Consider in a scenario you want to store boolean values in a list without specifying the generic type, so you got 8 values as boolean and one value as float, and another values as int.

Do you think, we can convert int to boolean, or float to boolean ?, No, you cannot convert.

This is what happens when you do not specify any generic type in the collection and this throws ClassCastException in java.

But if we make a list to accept only boolean values while accepting if it identifies the int value then it will not only reject but also gives you an error during compile time itself.

(Ex.) findElements() method in selenium

Iterator in Java

Often you will want to traverse through the elements of a collection, say to display each element.

The better way is to employ an iterator, which is an object that implements either the Iterator or the ListIterator interface.

Iterator enables you to traverse through the elements, add or remove elements. This is unidirectional. ListIterator is Bi-directional in traversing.

List iterator :

An iterator for lists that allows one to traverse the list in either direction, modify the list during iteration, and obtain the iterator current position in the list.

The iterator position of a ListIterator is never placed at an element rather it is placed between two elements in a list. Because of this List Iterator has capability to make it traverse in both directions, i.e., forward and backward.

List iterator provides methods to traverse forward using next() and traverse backward using previous()

Differences between Collection and Collections
Below are the few difference between the Collection(framework), Collections class.
1. Collection is an interface in Java. But Collections is a class in Java.
2. Collection is a base interface. Collections is a utility class in Java.
3. Collection defines methods that are used for data structures that contain the objects. Collections defines the methods that are used for operations like access, find etc. on a Collection.
4. Collection Interface has only abstract methods but Collections consists of only static methods which are used to operate on objects of type Collection.

Arrays vs Collection
There are so many difference but I am highlighting whatever i know, so there is chance that more than these differences to be present between them.

1. Arrays are fixed in size but Collections are dynamic in size.
2. With respect to memory arrays are not good to use but with respect to memory Collections are better to use.
3. With respect to performance; it is better to use arrays but with respect to performance collection are not good to use.
4. Arrays can hold only homogeneous elements but collections can hold both homogeneous and heterogeneous elements
5. Arrays don't have readymade methods but collections have readymade data structures and methods, which makes collections more user friendly
6. Arrays can hold both primitives and wrapper objects but collections can hold only objects.
7. ClassCasException occurs in Collection but not in Arrays
8. We have iterators for going through the collection but we donot have them for arrays
9. Java returns Arrays rather than Collection in most of the operations due to performance issues.

Difference between Array and Array list

Difference between **Array** and **ArrayList** are following:

1. Implementation of array is simple fixed sized array but Implementation of ArrayList is dynamic sized array.
2. Array can contain both primitives and objects but ArrayList can contain only object elements
3. You can't use generics along with array but ArrayList allows us to use generics to ensure type safety.
4. You can use *length* variable to calculate the length of an array but *size()* method to calculate size of ArrayList.
5. Array use assignment operator to store elements but ArrayList use *add()* to insert elements.

Array: Simple fixed sized arrays that we create in Java, like below
int arr[] = new int[10]

ArrayList : Dynamic sized arrays in Java that implement List interface.
ArrayList<Type> arrL = new ArrayList<Type>();
Here Type is the type of elements in ArrayList to
be created

(Ex.)

```
class Test
{
   public static void main(String args[])
   {
      /* ..... Normal Array..... */
      int[] arr = new int[2];
      arr[0] = 1;
      arr[1] = 2;
      System.out.println(arr[0]);

            /*.....ArrayList.....*/
      // Create an arrayList with initial capacity 2
      ArrayList<Integer> arrL = new ArrayList<Integer>(2);

      // Add elements to ArrayList
      arrL.add(1);
      arrL.add(2);

      // Access elements of ArrayList
```

```
        System.out.println(arrL.get(0));
    }
}
```

How Java multi threading tackles this problem?

To avoid polling, Java uses three methods, namely, wait(), notify() and notifyAll().
All these methods belong to object class as final so that all classes have them. They must be used within a synchronized block only.

- wait()-It tells the calling thread to give up the lock and go to sleep until some other thread enters the same monitor and calls notify().
- notify()-It wakes up one single thread that called wait() on the same object. It should be noted that calling notify() does not actually give up a lock on a resource.
- notifyAll()-It wakes up all the threads that called wait() on the same object.

HashMap in Java

HashMap is a part of Java's collection since Java 1.2. It provides the basic implementation of Map interface of Java. It stores the data in (Key, Value) pairs. To access a value one must know its key. HashMap is known as HashMap because it uses a technique called Hashing. Hashing is a technique of converting a large String to small String that represents the same String. A shorter value helps in indexing and faster searches. HashSet also uses HashMap internally. It internally uses a link list to store key-value pairs already explained in HashSet in detail and further articles.

Few important features of HashMap are:

- HashMap is a part of java.util package.
- HashMap extends an abstract class AbstractMap which also provides an incomplete implementation of Map interface.
- It also implements Cloneable and Serializable interface. K and V in the above definition represent Key and Value respectively.
- HashMap doesn't allow duplicate keys but allows duplicate values. That means A single key can't contain more than 1 value but more than 1 key can contain a single value.
- HashMap allows null key also but only once and multiple null values.
- This class makes no guarantees as to the order of the map; in particular, it does not guarantee that the order will remain constant over time. It is roughly similar to HashTable but is unsynchronized.

Multiple choice questions

Question and answer

1) The following codes both print:

"Welcome to Testing recruitment"

System.out.println(driver.getTitle());

System.out.println(driver.findElement(By.tagName(“title”)).getText())

a) **True**
b) False

2) In webdriver, which methods navigates to a URL?

a) goToUrl("url")
b) navigate.to("url")
c) getUrl("url")
d) **get("url")**

3) Consider the following html snippet

Firefox

Google Chrome

Internet Explorer

Opera

Safari

Which CSS selector is a valid statement to select Opera?

a) css = li.contains("Opera")
b) css = ul.li(4)
c) **css = ul > li:nth-of-type(4)**
d) css = ul > li:nth-of-type(3)
e) css = ul.li:nth-child(4)

4) In webdriver, which of the following is a valid select statement that selects a value from a drop down element?

a) selectByIndex()

b) selectByVisibleText()
c) selectByValue()
d) **All above**
e) none of above

5) Which WebDriver method is used to change focus to an alert, a frame or a browser window?

a) changeFocus()
b) setFocus()
c) **switchTo()**
d) changeTo()

6) The Selenium IDE is used

a) To create customized test results.
b) To deploy your tests across multiple environments using Selenium Grid
c) To test with HTML Unit
d) **To test a web application against Firefox only.**

7) Select the command which is NOT a type of assertion in Selenium IDE.

a) Asser t
b) Verify
c) Waitfor
d) **Wait**

8) Select the command which is used to check the presence of a certain element.

a) verifyTable
b) verifyTitlePresent
c) verifyTextPresent
d) **verifyElementPresent**

9) Select the command which is used to print a string value or a variable in Selenium IDE.

a) The 'display' command
b) **The 'echo' command**
c) The 'print' command
d) The 'printr' command

10) Select the command which is used to compare the contents of a table with expected values.

a) VerifyTables
b) verifyTableData
c) **verifyTable**
d) verifyTableCell

11. In webdriver, deselectAll() is a valid command.

a) **True**
b) False

12. In WebDriver, which command can be used to enter values onto text boxes? Select the best answer.

a) type()
b) selenium.type()
c) driver.type("text")
d) sendKeys()
e) **sendKeys("text")**

13. In webdriver, which command takes you forward by one page on the browser's history?

a) navigate.forward()
b) Navigate.forward()
c) **navigate().forward()**
d) Navigate.forward
e) navigate_forward()

14. In webdriver, which method closes the open browser?

a) quit()
b) terminate()
c) shutdown()
d) **close()**

15. In webdriver, what is the method that counts the number of elements?

a) driver.getCountOfElements()
b) driver.findElement(By.id("search")).getCount()
c) **driver.findElements(By.id("search")).size()**
d) driver.findElements(By.id("search")).length()

16) Select the command which is used to pause execution until the specified element becomes present.

a) **waitForElementPresent**
b) waitForPagePresent
c) waitForTablePresent
d) waitForFieldPresent

17) Select the command which is used to pause execution until the page is loaded completely.

a) **waitForPageToLoad**
b) waitForElementPresent
c) waitForPage
d) waitForLoad

18)Select the command which is NOT used in verifying page elements .

a) verifyElementPresent
b) **verifyElementRight**
c) verifyElementNotPresent
d) verifyElementPositionLeft

19) Select the variation which locates elements by the value of their "id" attribute in Web Driver Selenium

a) By.id
b) By.idno
c) By.id_no
d) By.tag_id

20) Which Component is used to run multiple tests simultaneously in different browsers and platforms?

a) Selenium Grid
b) Selenium IDE
c) Selenium RC
d) Selenium Webdriver

Database

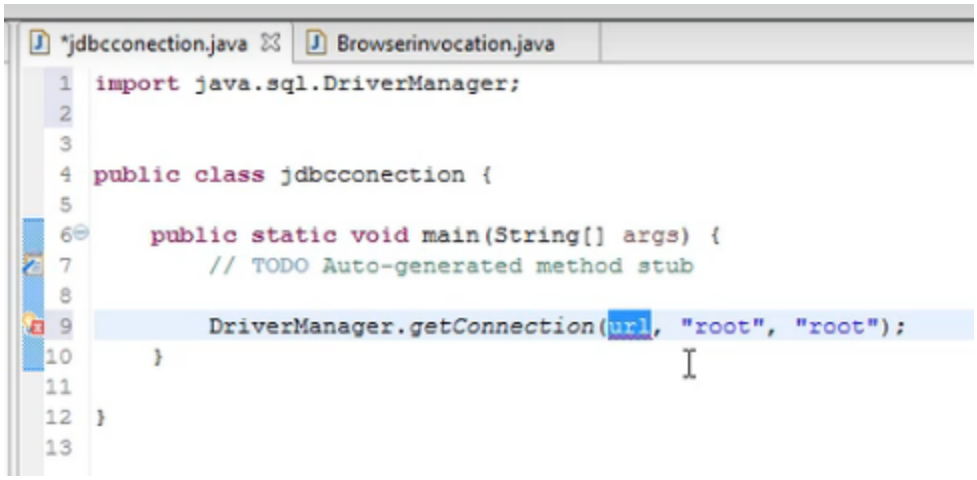
Database import notes

- Download mysql from dev.mysql website
- Set credentials for DB and install the necessary items
- Integrate JDBC with Database
  a.Download mysql java connector jar and add to our build path in eclipse
  b.Configure device manager as below

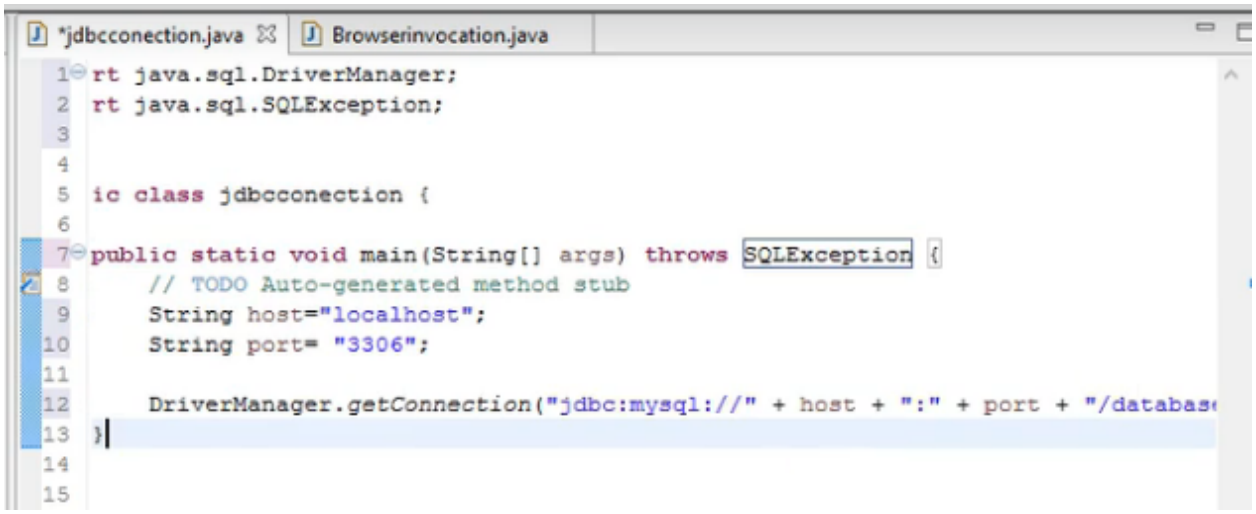
Connection URL

```
"jdbc:mysql://" + host + ":" + port + "/databasename";
```

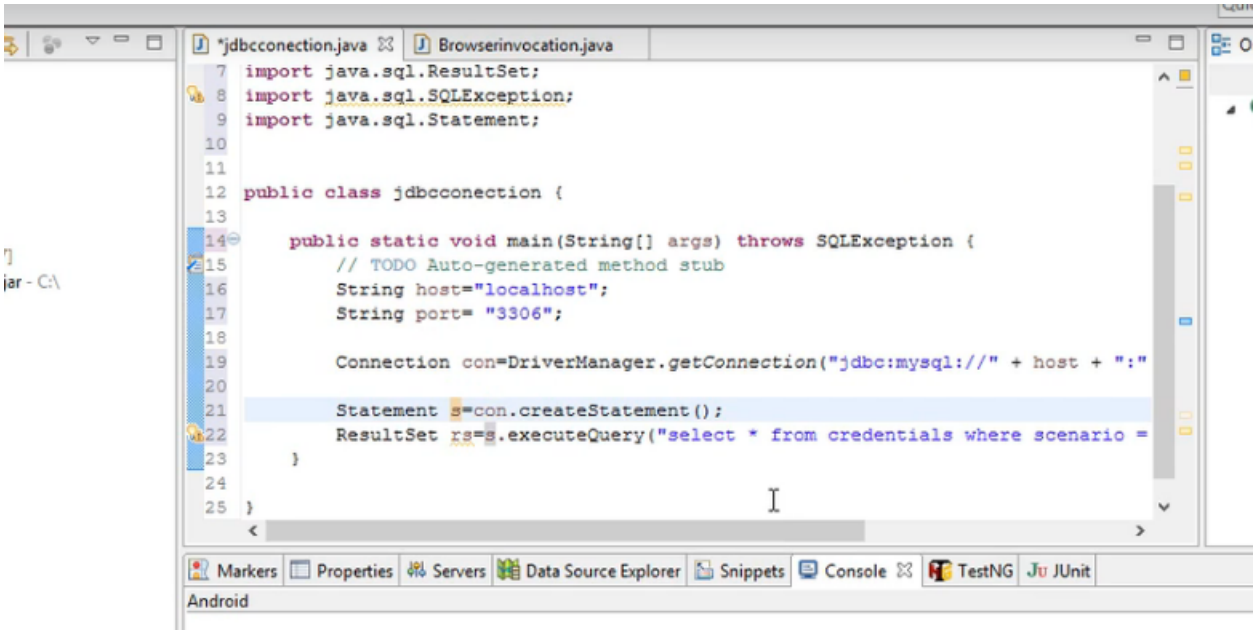
```
jdbc:mysql://" + localhost + ":" + 3306 + "/demo";
```



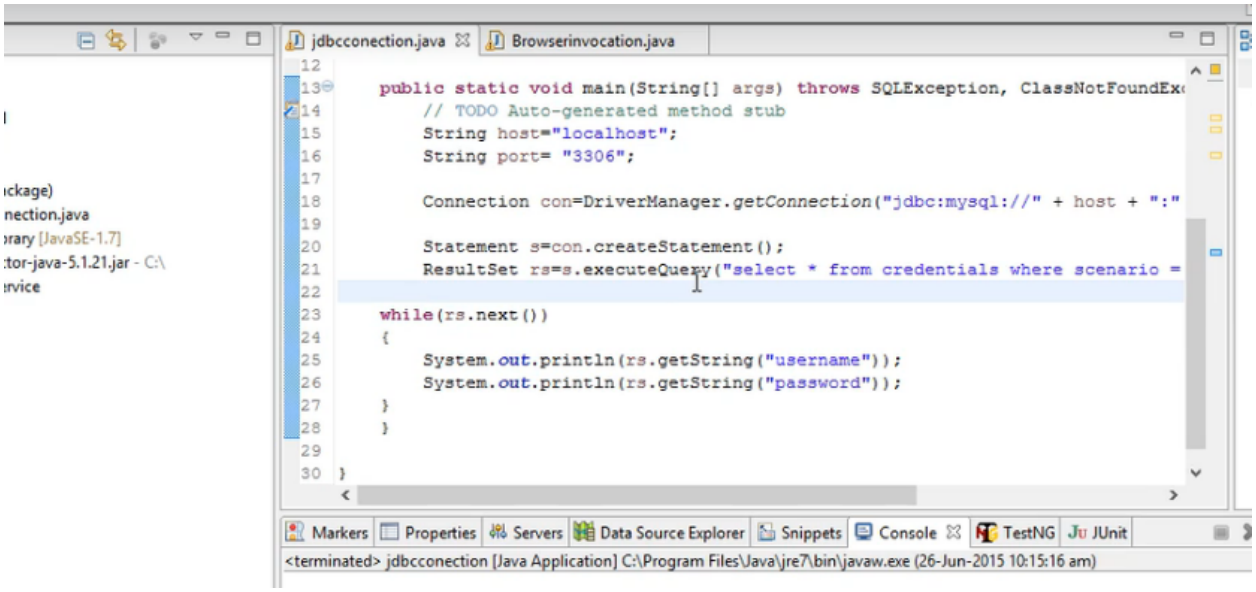
1st root is DB user name,2nd one is password



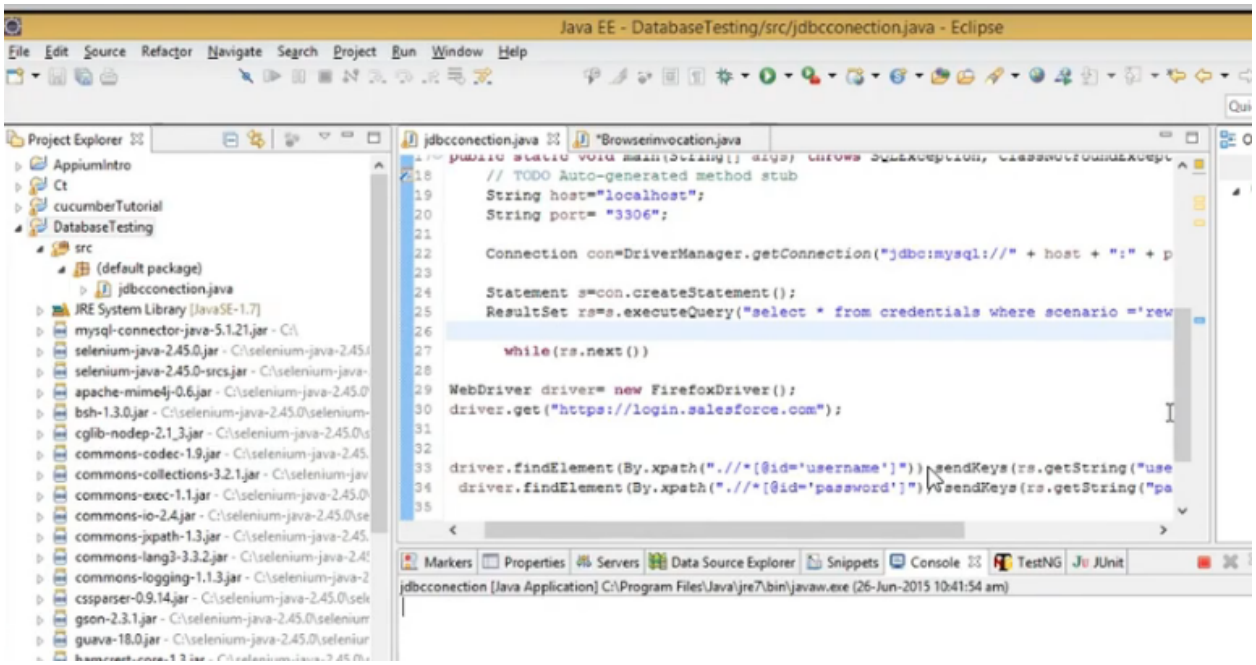
Write query as below



Query to print the result set in console and use the next in while loop to check values in all the index
Initially the result set will be in the base index only so we need to move from base to first index by using while loop as below



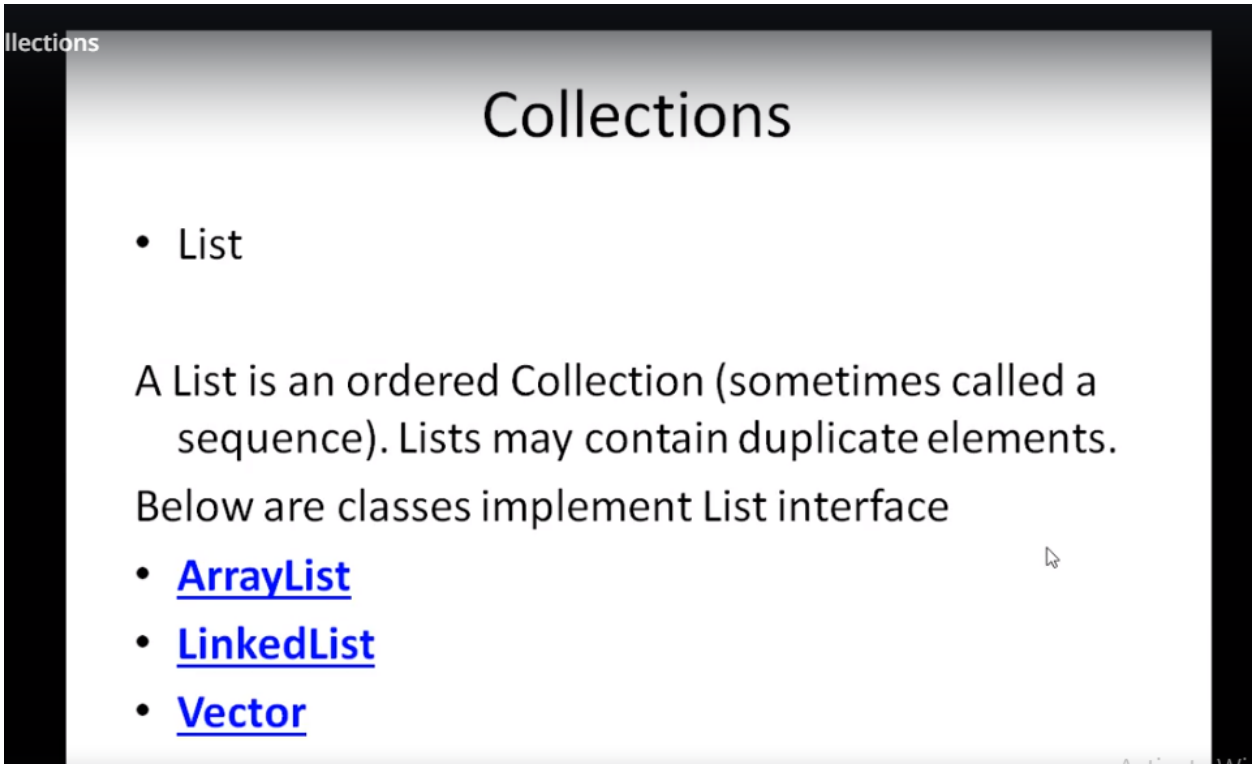
Code to get the username and password from DB and giving input to the web



---

ons

The **Java Collections Framework** is a collection of interfaces and classes which helps in storing and processing the data efficiently. This framework has several useful classes which have tons of useful functions which makes a programmer task super easy


llections
Collections
• List
A List is an ordered Collection (sometimes called a sequence). Lists may contain duplicate elements.
Below are classes implement List interface
• ArrayList
• LinkedList
• Vector

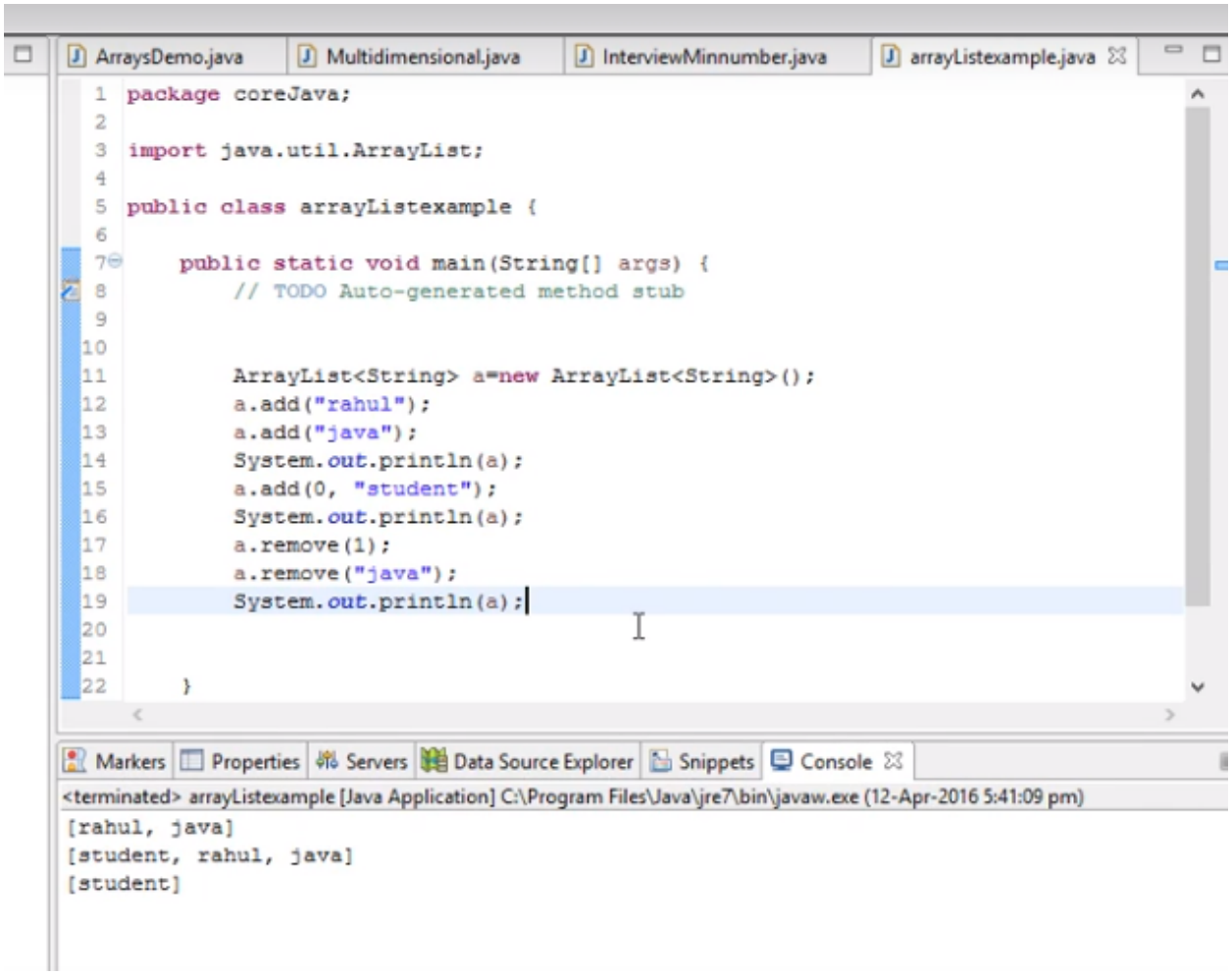
- **Set**
- A Set is a Collection that cannot contain duplicate elements.
- however it makes no guarantees concerning the order of iteration.
- Below are classes implement List interface
- HashSet
- LinkedHashSet
- TreeSet

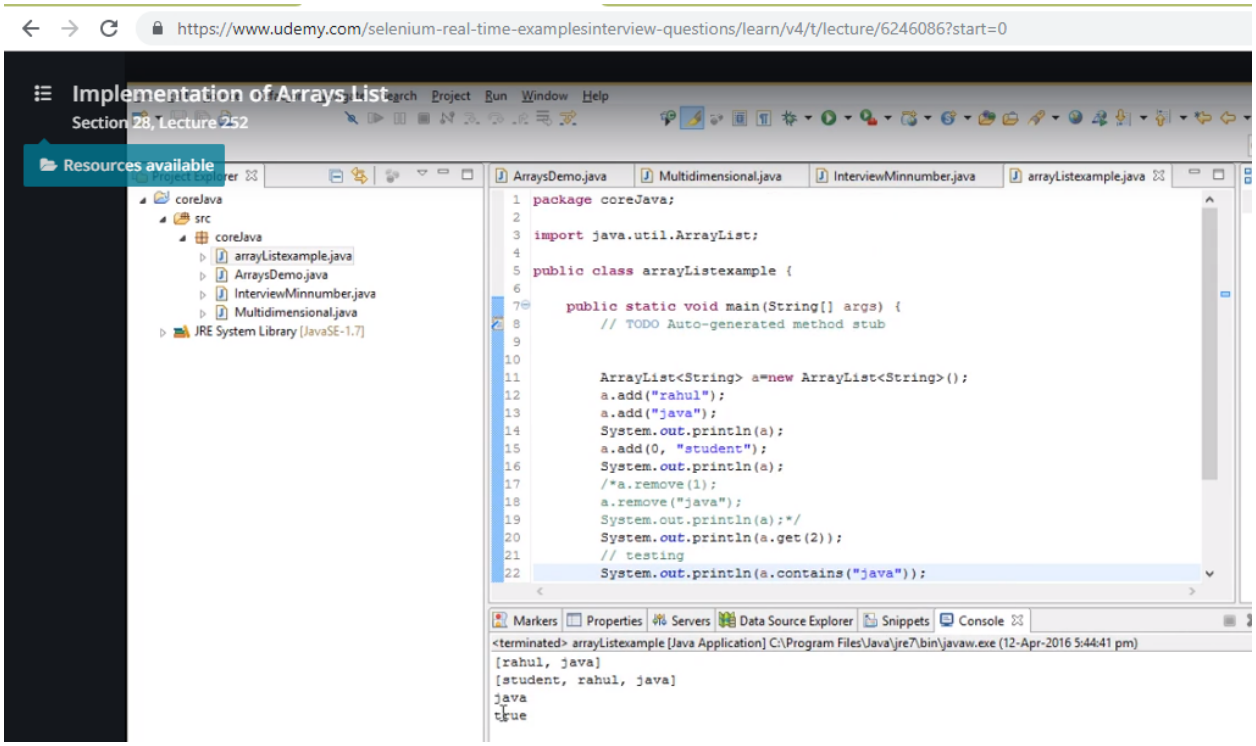
37 people bookmarked this moment.

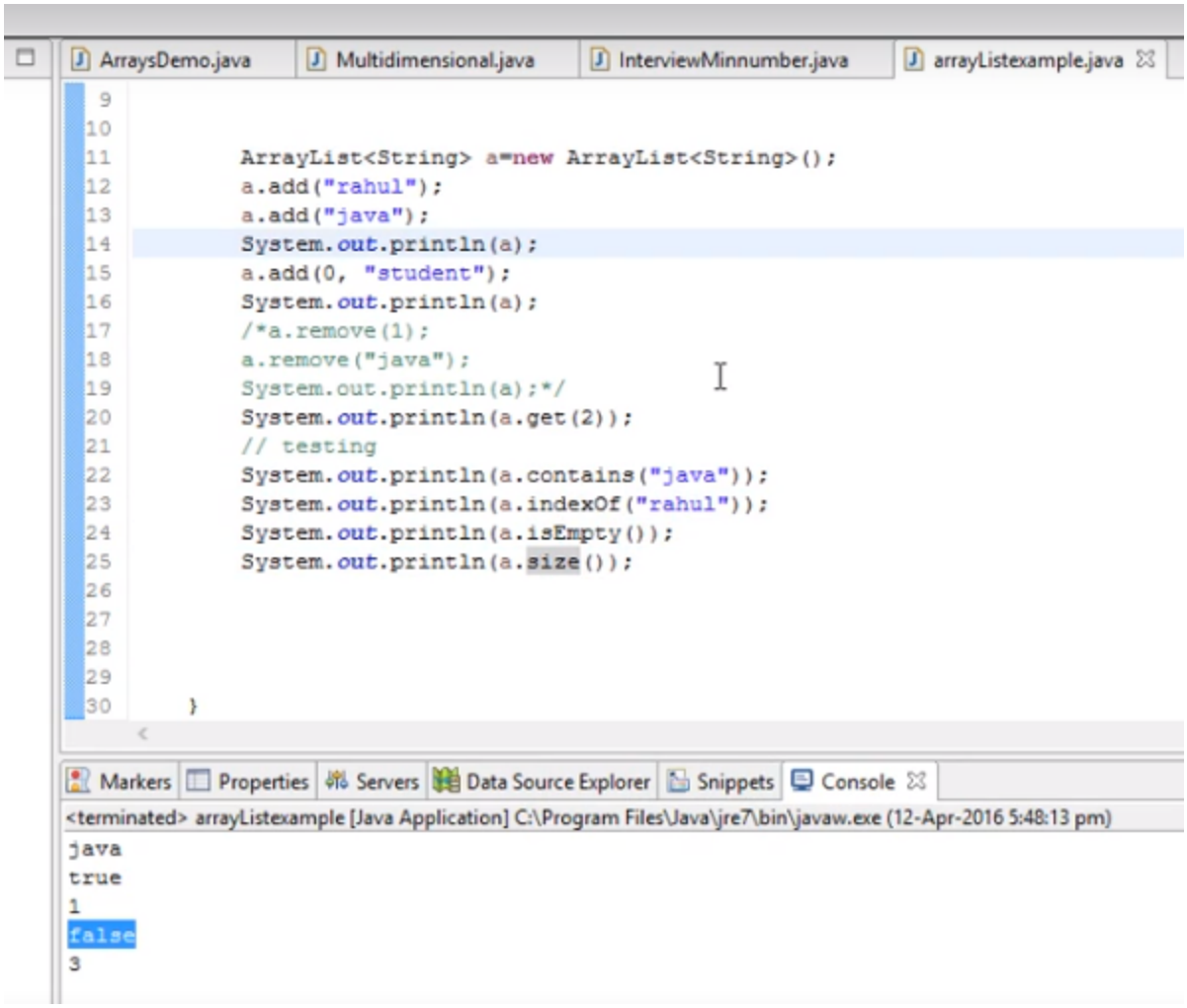
tions

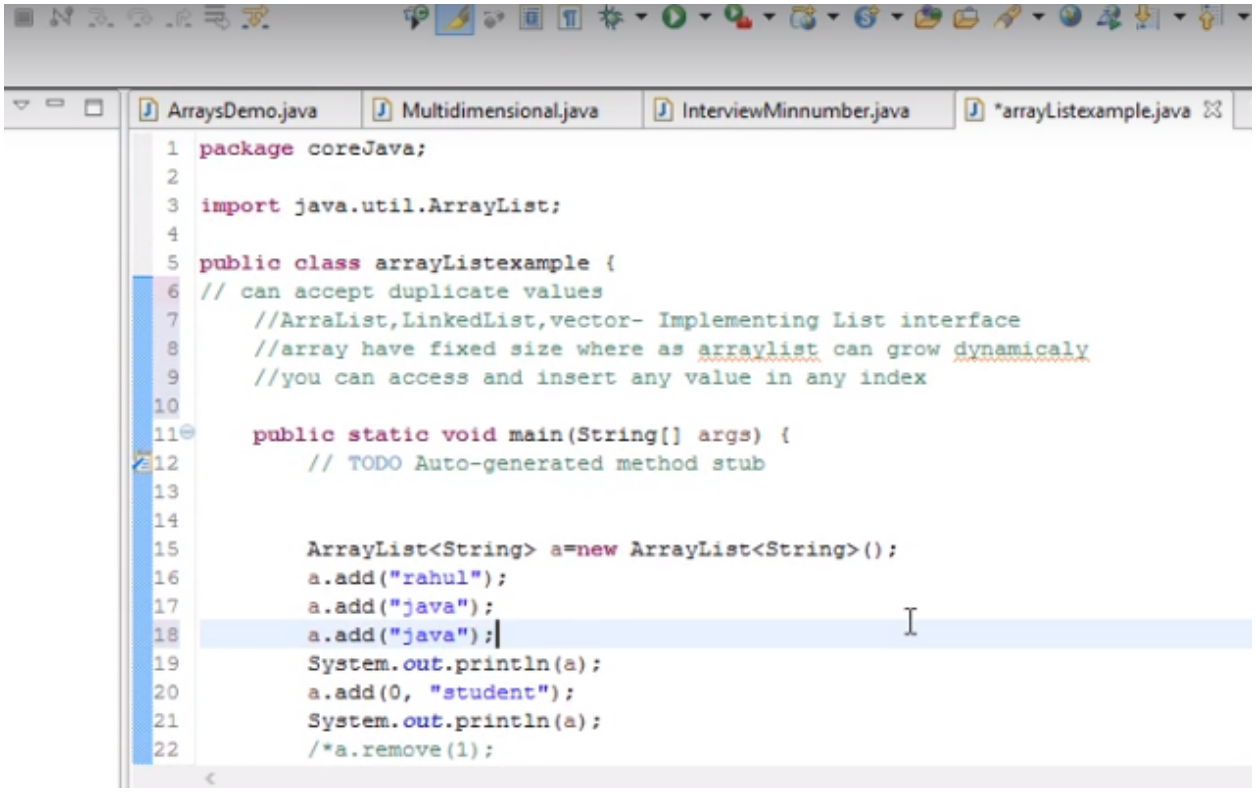
- **Map**
- A Map is an object that maps keys to values. A map cannot contain duplicate keys.
- Below ae main implementations of Map interfaces
- HashMap
- TreeMap
- LinkedHash

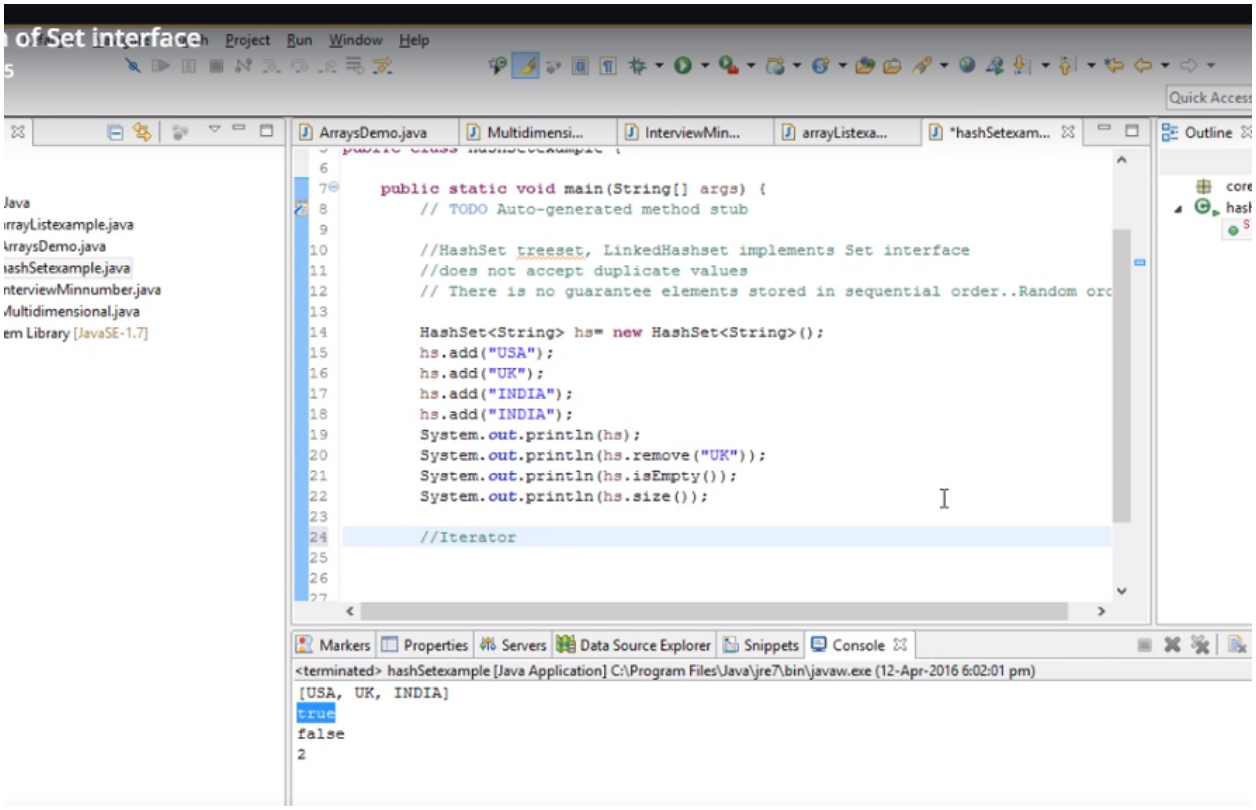
37 people bookmarked this moment.

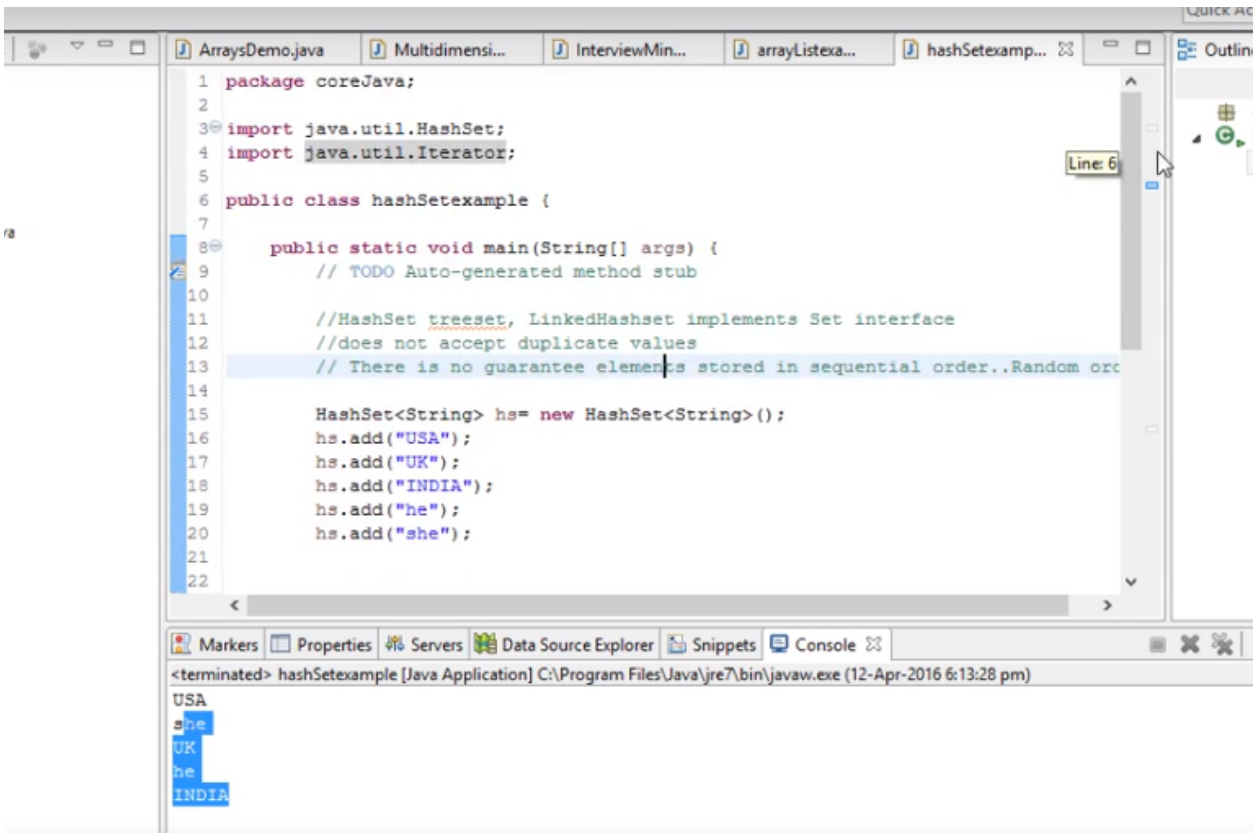

ArraysDemo.java
Multidimensional.java
InterviewMinnumber.java
arrayListexample.java
1 package coreJava;
2
3 import java.util.ArrayList;
4
5 public class arrayListexample {
6
7 public static void main(String[] args) {
8 // TODO Auto-generated method stub
9
10
11 ArrayList<String> a=new ArrayList<String>();
12 a.add("rahul");
13 a.add("java");
14 System.out.println(a);
15 a.add(0, "student");
16 System.out.println(a);
17 a.remove(1);
18 a.remove("java");
19 System.out.println(a);
20
21
22 }
Markers
Properties
Servers
Data Source Explorer
Snippets
Console
<terminated> arrayListexample [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12-Apr-2016 5:41:09 pm)
[rahul, java]
[student, rahul, java]
[student]

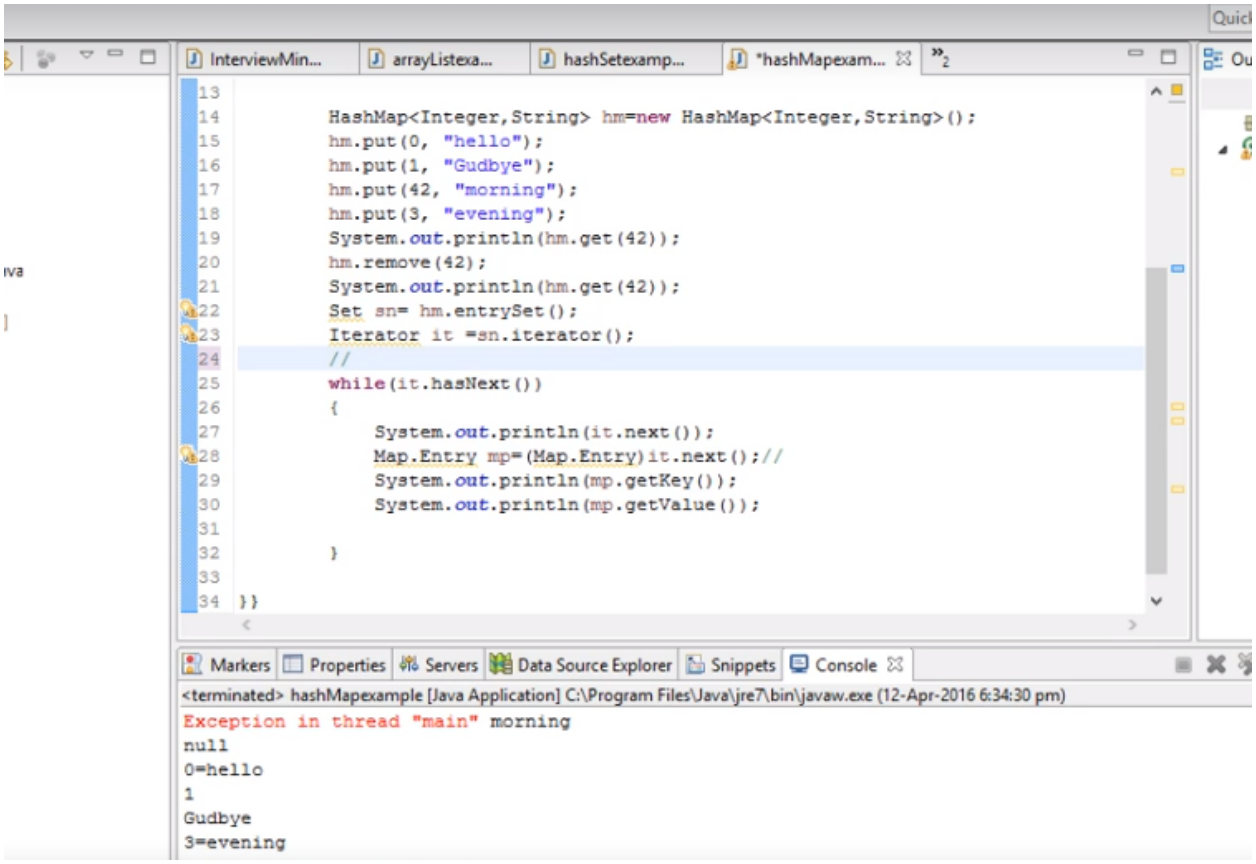

https://www.udemy.com/selenium-real-time-examplesinterview-questions/learn/v4/t/lecture/6246086?start=0
Implementation of Arrays List
Section 28, Lecture 252
Resources available
Project Explorer
coreJava
src
coreJava
arrayListexample.java
ArraysDemo.java
InterviewMinnumber.java
Multidimensional.java
JRE System Library [JavaSE-1.7]
ArraysDemo.java
Multidimensional.java
InterviewMinnumber.java
arrayListexample.java
1 package coreJava;
2
3 import java.util.ArrayList;
4
5 public class arrayListexample {
6
7 public static void main(String[] args) {
8 // TODO Auto-generated method stub
9
10
11 ArrayList<String> a=new ArrayList<String>();
12 a.add("rahul");
13 a.add("java");
14 System.out.println(a);
15 a.add(0, "student");
16 System.out.println(a);
17 /*a.remove(1);
18 a.remove("java");
19 System.out.println(a);*/
20 System.out.println(a.get(2));
21 // testing
22 System.out.println(a.contains("java"));
Markers
Properties
Servers
Data Source Explorer
Snippets
Console
<terminated> arrayListexample [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12-Apr-2016 5:44:41 pm)
[rahul, java]
[student, rahul, java]
java
true

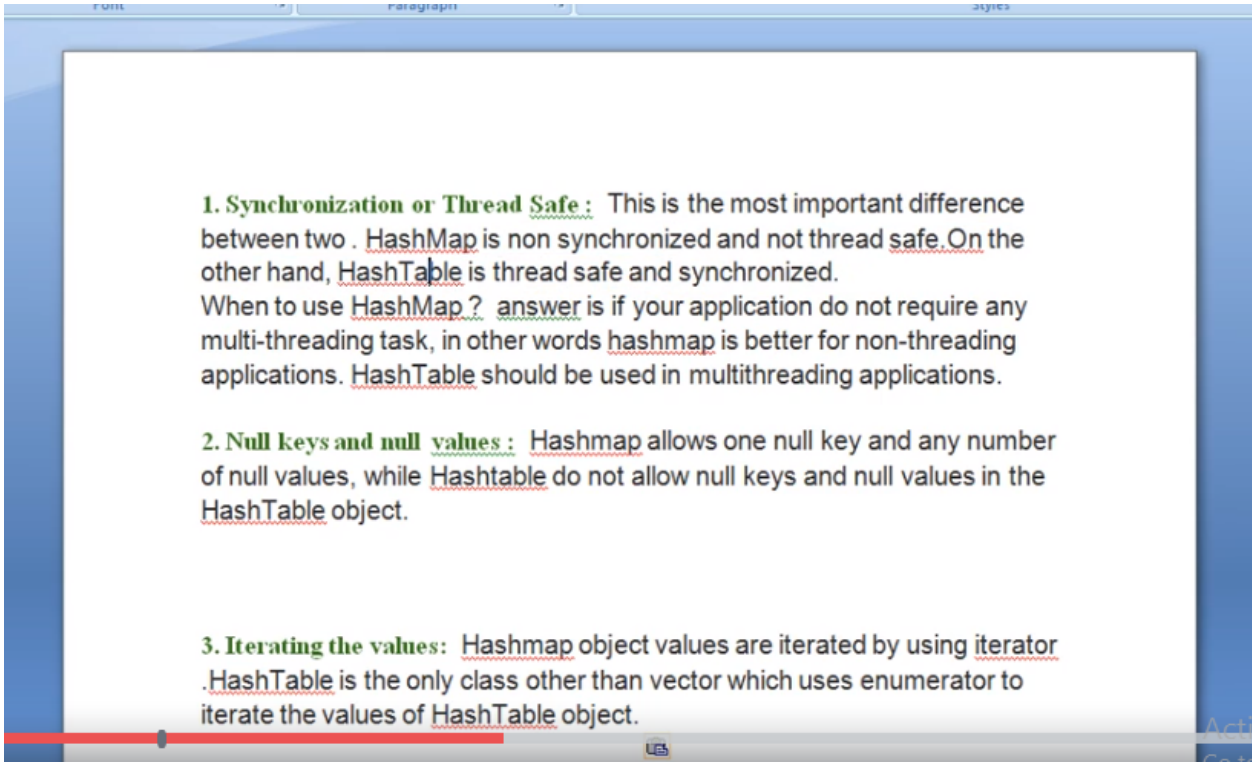

ArraysDemo.java
Multidimensional.java
InterviewMinnumber.java
arrayListexample.java
ArrayList<String> a=new ArrayList<String>();
a.add("rahul");
a.add("java");
System.out.println(a);
a.add(0, "student");
System.out.println(a);
/*a.remove(1);
a.remove("java");
System.out.println(a);*/
System.out.println(a.get(2));
// testing
System.out.println(a.contains("java"));
System.out.println(a.indexOf("rahul"));
System.out.println(a.isEmpty());
System.out.println(a.size());
}
Markers
Properties
Servers
Data Source Explorer
Snippets
Console
<terminated> arrayListexample [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12-Apr-2016 5:48:13 pm)
java
true
1
false
3


ArraysDemo.java
Multidimensional.java
InterviewMinnumber.java
*arrayListexample.java
1 package coreJava;
2
3 import java.util.ArrayList;
4
5 public class arrayListexample {
6 // can accept duplicate values
7 //ArraList,LinkedList,vector- Implementing List interface
8 //array have fixed size where as arraylist can grow dynamicaly
9 //you can access and insert any value in any index
10
11 public static void main(String[] args) {
12 // TODO Auto-generated method stub
13
14
15 ArrayList<String> a=new ArrayList<String>();
16 a.add("rahul");
17 a.add("java");
18 a.add("java");
19 System.out.println(a);
20 a.add(0, "student");
21 System.out.println(a);
22 /*a.remove(1);


of Set interface
Project Run Window Help
ArraysDemo.java
Multidimensi...
InterviewMin...
arrayListexa...
*hashSetexam...
Outline
Java
rrayListexample.java
rraysDemo.java
ashSetexample.java
nterviewMinnumber.java
Aultidimensional.java
em Library [JavaSE-1.7]
public static void main(String[] args) {
// TODO Auto-generated method stub
//HashSet treeset, LinkedHashset implements Set interface
//does not accept duplicate values
// There is no guarantee elements stored in sequential order..Random orc
HashSet<String> hs= new HashSet<String>();
hs.add("USA");
hs.add("UK");
hs.add("INDIA");
hs.add("INDIA");
System.out.println(hs);
System.out.println(hs.remove("UK"));
System.out.println(hs.isEmpty());
System.out.println(hs.size());
//Iterator
Markers
Properties
Servers
Data Source Explorer
Snippets
Console
<terminated> hashSetexample [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12-Apr-2016 6:02:01 pm)
[USA, UK, INDIA]
true
false
2


ArraysDemo.java
Multidimensi...
InterviewMin...
arrayListexa...
hashSetexamp...
Outlin
package coreJava;
import java.util.HashSet;
import java.util.Iterator;
Line: 6
public class hashSetexample {
public static void main(String[] args) {
// TODO Auto-generated method stub
//HashSet treeset, LinkedHashset implements Set interface
//does not accept duplicate values
// There is no guarantee elements stored in sequential order..Random orc
HashSet<String> hs= new HashSet<String>();
hs.add("USA");
hs.add("UK");
hs.add("INDIA");
hs.add("he");
hs.add("she");
Markers
Properties
Servers
Data Source Explorer
Snippets
Console
<terminated> hashSetexample [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12-Apr-2016 6:13:28 pm)
USA
she
UK
he
INDIA


InterviewMin...
arrayListexa...
hashSetexamp...
*hashMapexam...
HashMap<Integer,String> hm=new HashMap<Integer,String>();
hm.put(0, "hello");
hm.put(1, "Gudbye");
hm.put(42, "morning");
hm.put(3, "evening");
System.out.println(hm.get(42));
hm.remove(42);
System.out.println(hm.get(42));
Set sn= hm.entrySet();
Iterator it =sn.iterator();
//
while(it.hasNext())
{
System.out.println(it.next());
Map.Entry mp=(Map.Entry)it.next();//
System.out.println(mp.getKey());
System.out.println(mp.getValue());
}
}}
Markers
Properties
Servers
Data Source Explorer
Snippets
Console
<terminated> hashMapexample [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (12-Apr-2016 6:34:30 pm)
Exception in thread "main" morning
null
0=hello
1
Gudbye
3=evening


1. Synchronization or Thread Safe : This is the most important difference between two . HashMap is non synchronized and not thread safe.On the other hand, HashTable is thread safe and synchronized.
When to use HashMap ? answer is if your application do not require any multi-threading task, in other words hashmap is better for non-threading applications. HashTable should be used in multithreading applications.
2. Null keys and null values : Hashmap allows one null key and any number of null values, while Hashtable do not allow null keys and null values in the HashTable object.
3. Iterating the values: Hashmap object values are iterated by using iterator .HashTable is the only class other than vector which uses enumerator to iterate the values of HashTable object.

----------CTS----------

https://www.geeksforgeeks.org/difference-between-throw-and-throws-in-java/

THROW	THROWS
throw keyword is used to throw an exception explicitly.	throws keyword is used to declare one or more exceptions, separated by commas.
Only single exception is thrown by using throw.	Multiple exceptions can be thrown by using throws.
throw keyword is used within the method.	throws keyword is used with the method signature.
Syntax wise throw keyword is followed by the instance variable.	Syntax wise throws keyword is followed by exception class names.
Checked exception cannot be propagated using throw only.Unchecked exception can be propagated using throw.	For the propagation checked exception must use throws keyword followed by specific exception class name.

https://sqa.stackexchange.com/questions/18658/how-to-get-entered-text-from-textbox-when-value-is-empty-in-selenium-using-java

Or

https://stackoverflow.com/questions/38091241/how-to-get-entered-text-from-a-textbox-having-no-value-attribute-in-selenium

https://www.baeldung.com/cucumber-scenario-outline

or

https://medium.com/@priyank.it/cucumber-difference-between-examples-table-data-table-21501f2becbd

A **merge conflict happens** when two branches both modify the same region of a file and are subsequently **merged**. **Git** can't know which of the changes to keep, and thus needs human intervention to resolve the **conflict**. In this case, your steps 2 and 3 create two branches that have **conflicting** changes.

----sorting using SQL query----
SELECT * FROM table_name ORDER BY column_name ASC|DESC

https://www.geeksforgeeks.org/sql-query-to-find-second-largest-salary/

Git pull is basically a **git fetch** followed by a **git** merge. **Git fetch** only downloads data from a remote repository, but it does not integrate any **of** these new data into your working files. **Git pull** updates the current HEAD branch with the latest **changes** from the remote server

How to handle dynamic web table

https://www.guru99.com/handling-dynamic-selenium-webdriver.html

Assert

https://www.softwaretestinghelp.com/assertions-in-selenium/#targetText=assert%20equals%20verifies%20that%20two,arrays%20are%20not%20considered%20equal.&targetText=message%20%E2%80%93%20Message%20to%20be%20displayed%20in%20case%20of%20an%20assertion%20error.,-expected%20%E2%80%93%20Array%20of

To Print *
ex.1

```
public class GeeksForGeeks
{
    // Function to demonstrate printing pattern
    public static void main(String args[]) {
   int i, j;
   int n=5;
          // outer loop to handle number of rows
          //  n in this case
          for(i=0; i<n; i++)        {
          //  inner loop to handle number of columns
          //  values changing acc. to outer loop
```

```
        for(j=0; j<=i; j++)        {
                // printing stars
                System.out.print("*");
        }
        // ending line after each row
        System.out.println();
        }
   }
  }
```

Output

```
*
**
***
****
*****
```

Ex.2

```
public class GeeksForGeeks
{
    // Function to demonstrate printing pattern
    public static void main(String args[])
    {
        int n=5;
        // outer loop to handle number of rows
        //  n in this case
        for (int i=0; i<n; i++)
        {
            // inner loop to handle number spaces
            // values changing acc. to requirement
            for (int j=n-i; j>1; j--)
            {
                // printing spaces
                System.out.print(" ");
            }
            //  inner loop to handle number of columns
            //  values changing acc. to outer loop
            for (int j=0; j<=i; j++ )
            {
                // printing stars
                System.out.print("* ");
            }
            // ending line after each row
```

```
        System.out.println();
        }
   }
  }
```
Output : pyramid

ex.3

```
public class GeeksForGeeks
{
    // Function to demonstrate printing pattern
    public static void main(String args[]) {
  {
        int i, j,num;
        int n=5;
        // outer loop to handle number of rows
        //  n in this case
        for(i=0; i<n; i++)
        {
        // initialising starting number
        num=1;
        //  inner loop to handle number of columns
        //  values changing acc. to outer loop
        for(j=0; j<=i; j++)
        {
                // printing num with a space
                System.out.print(num+ " ");
                //incrementing value of num
                num++;
        }
        // ending line after each row
        System.out.println();
        }
        }
   }
  }
```

Output

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

ex.4

```
public class GeeksForGeeks
{
```

```
  // Function to demonstrate printing pattern
  public static void main(String args[]) {
     // initialising starting number
     int i, j, num=1;
     int n=5;
     // outer loop to handle number of rows
     // n in this case
     for(i=0; i<n; i++)      {
        // without re assigning num
        // num = 1;
        for(j=0; j<=i; j++)
        {
           // printing num with a space
           System.out.print(num+ " ");

           // incrementing num at each column
           num = num + 1;
        }
        // ending line after each row
        System.out.println();
     }
  }
 }
```

Output:

1
2 3
4 5 6
7 8 9 10
11 12 13 14 15

Ragesh Question's

Realtime interview questions:

- What kind of test cases will u automate

https://www.quora.com/How-do-you-choose-which-test-cases-to-automate

---

- How do we merge code when conflict occurs in Git
- Why can't use code directly in Jenkins without using maven and Github
- dB connectivity code for DB2...
- How to do sorting using SQL query

- Diff between is displayed & is enabled

https://www.softwaretestinghelp.com/webdriver-commands-selenium-tutorial-14/
https://reviewdb.io/posts/1510069286016/difference-between-isdisplayed-and-isenabled-methods-in-selenium-webdriver

---

- What is page object model, what is the reason to prefer page object model

Answer reference : https://www.quora.com/What-is-a-page-object-pattern-in-Selenium
https://www.guru99.com/page-object-model-pom-page-factory-in-selenium-ultimate-guide.html

---

- Difference between pull and fetch command used in git.
- How do I convince customer on the selection of Automation tool
- How Jenkins is used in ur office
- Some questions on github, SQL query to fetch employee with highest salary

---

- Explain different xpaths and types

https://www.guru99.com/xpath-selenium.html

- Read and write the data from excel sheet

- How will you do Right click on the mouse

```
WebElement searchbox = driver.findElement(By.id("search"));
searchbox.sendKeys("test");
Actions act = new Actions(driver);
act.contextClick(searchbox).perform();
```

- How will you verify the highlight element

```
JavascriptExecutor js = (JavascriptExecutor) driver;
            js.executeScript("arguments[0].setAttribute('style', 'background: yellow; border: 2px solid red;');", element);
```

https://stackoverflow.com/questions/29475414/how-to-check-if-a-text-is-highlighted-on-the-page-using-selenium

- first time button is enabled after some time button should be disabled,how will you verify it

if(!searchbox.isEnabled());

- There are 1000 test cases in a test suite which has defect leakage. What strategy you will come up for the solution
- css selectors types
- how to handle windows pop up ... Code using robot class.

https://www.guru99.com/using-robot-api-selenium.html

- Selenium code to highlight text, and get colour and background colour.
- How to underline an element
- what does @Test will do?

https://www.guru99.com/all-about-testng-and-selenium.html

- How to handle a dynamic web table

https://www.guru99.com/handling-dynamic-selenium-webdriver.html

- frames handling code and inline frame handling ?
- On clicking date field two month Calendar will be displayed (July and August) at the same time and date 4 will be enabled at this time and how will you handle this situation
- How to handle alert with Yes and No option
- a web element was removed in the previous release and now you need to verify whether the element is present or not..how will u do this ?
- how you will write assert statement for comparing the difference between actual and expected login name
- In web page there is an element you don't know whether it appears or not. now how you'll find that element also you have to catch that exception. explain ?

- Difference between scenario and scenario outline in cucumber
- Difference between throw and throws
- Syntax for fluent wait
- Code for encapsulation
- Diff between implicit and explicit wait
- How to get the value present in text field
- Program to write action class
- Program to write armstrong number1

- arrays sorting, second largest number in an array
- difference between throw and throws, hashmap and hashtable

SET 2 (5 years experience questions)

Question asked in interview in different organisations for 5 years of exp in in last one month.
1) Explain your current automation framework.
2) current roles and responsibilities.
3) TestNg sequence.

---

4) why no main method is required in tetsng execution.
Answer: **TestNG** uses annotations like @Test to take care of the executions while running the **TestNG** class.

---

5) how to connect to database.
https://www.guru99.com/database-testing-using-selenium-step-by-step-guide.html

---

6) simple SQL queries on group by and conditions.
The GROUP BY statement is often used with aggregate functions (COUNT, MAX, MIN, SUM, AVG) to group the result-set by one or more columns.

The following SQL statement lists the number of customers in each country:

```
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country;
```

The following SQL statement lists the number of customers in each country, sorted high to low:

```
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country
ORDER BY COUNT(CustomerID) DESC;
```

```
DDL = CARD(create,alter,rename and drop)
DML = CRUD(create(insert into),retrieve(select),update,delete)
Golden Rule = Select From Where Groupby Having Orderby
```

7) SQL query to find second highest salary without using sub query.

```
SELECT Salary FROM table ORDER BY Salary DESC LIMIT 1,1
```

8) SQL query to sort the column in ascending.

```
SELECT * FROM Student ORDER BY ROLL_NO DESC;
```

```
SELECT * FROM Student ORDER BY Age ASC , ROLL_NO DESC;
```

9) SQL query to find duplicate using self join.
https://www.xaprb.com/blog/2006/10/09/how-to-find-duplicate-rows-with-sql/

10) what is the difference between http and https.
HTTP is unsecured while **HTTPS** is secured. **HTTP** sends data over port 80 while **HTTPS** uses port 443. **HTTP** operates at application layer, while **HTTPS** operates at transport layer. No SSL certificates are required for **HTTP**, with **HTTPS** it is required that you have an SSL certificate and it is signed by a CA.

11) java program on find duplicate character.
Given above
12) java program on bubble sort.
https://www.javatpoint.com/bubble-sort-in-java

13) excel utilty to read excel using hashmap of hashmap.
https://www.inviul.com/store-data-excel-sheet-hashmap/

14) what is dependency injection in cucumber and testng Both and how we can achieve.

15)java program to remove white space and replace it with comma.

```
class BlankSpace {
	public static void main(String[] args)
	{
		String str = "     Geeks for Geeks        ";
		// Call the replaceAll() method
		str = str.replaceAll("\\s", "");
		System.out.println(str);
	}
}
```

Output:

```
GeeksforGeeks
```

```
replaceAll("\\s", ""); where \\s is a single space in unicode
```

// Java program to demonstrate working
// of java string trim() method
---It returns the omitted string with no leading and trailing spaces.---

```
class Gfg {
        // driver code
        public static void main(String args[])
        {
        // trims the trailing and leading spaces
        String s = " geeks for geeks has all java functions to read ";
        System.out.println(s.trim());
        // trims the leading spaces
        s = " Chetna loves reading books";
        System.out.println(s.trim());
                }
}
```

Output:
geeks for geeks has all java functions to read
Chetna loves reading books

16) java program for taking input in arraylist and returning in array.

```
 class GFG
 {
    public static void main (String[] args)
    {
       List<Integer> al = new ArrayList<Integer>();
       al.add(10);
       al.add(20);
       al.add(30);
       al.add(40);

       Object[] objects = al.toArray();

       // Printing array of objects
       for (Object obj : objects)
          System.out.print(obj + " ");
    }
 }
```

Output: 10 20 30 40

17) what is thrust testing.

18) write creative test case for lift.

https://artoftesting.com/lift

19) write cucumber feature file using datatable.
http://www.automationtestinghub.com/cucumber-data-table/

20) what are latest enhancement in selenium 4
https://dzone.com/articles/selenium-4-is-releasing-soon-what-every-qa-must-kn

https://blog.testproject.io/2019/05/01/open-source-selenium-4-test-automation/

21) how to pass parameters in Jenkins.
https://wiki.jenkins.io/display/JENKINS/Parameterized+Build

22) what is java genrics and where you have used in our current automation framework.
https://howtodoinjava.com/java/generics/complete-java-generics-tutorial/#:~:targetText=%E2%80%9CJava%20Generics%20are%20a%20language,replace%20the%20formal%20type%20parameters.

23) wap to take pass a get request using rest assured and from response verify if data present in database using SQL.

https://semaphoreci.com/community/tutorials/testing-rest-endpoints-using-rest-assured

24) difference between hashmap and linkedhashmap
HashMap and LinkedHashMap are two of the most common used Map implementation in Java. Main **difference between HashMap and LinkedHashMap** is that **LinkedHashMap** maintains insertion order of keys, order in which keys are inserted in to **LinkedHashMap**. On the other hand **HashMap** doesn't maintain any order or keys or values.

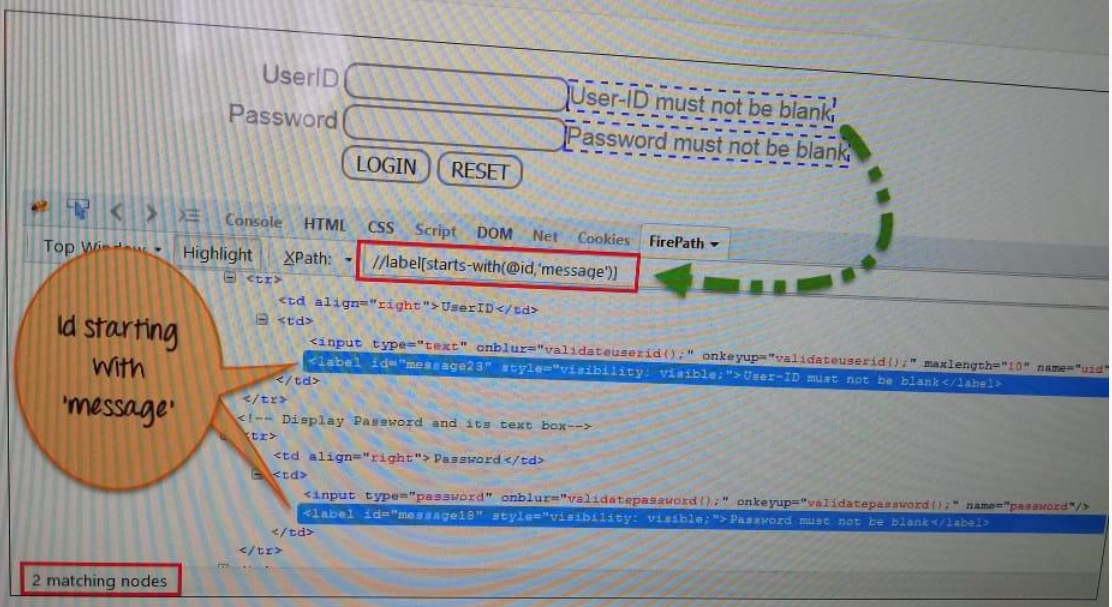
25) different error code.
https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

40) what is effective pom in maven.
41) how many different build phase avaialt in maven.
42) how to pass testng in maven file.
43) how to create profile in maven and execute same .
44) how the optimze ur current framework.
45) what is expectedconditions in explicit wait is it a class or interface.

46) how to take ss of failed tc.
47) how to rerun failed tc for 3 times.
50) have you use JavaScript executor in ur framework is yes where.
51) which framework is easy to handle data driven, keyword driven or hybrid.
52) different design pattern we can use with selenium.
53) can we integrate Sprint boot with selenium.
54) what is jira.how ur project maintains testcase and defect
55) what are deliverables you deliver after project is done.
56) how you see automation growth in next fiver years.

In below example, XPath finds those element whose 'ID' starting with 'message'.

```
Xpath=//label[starts-with(@id,'message')]
```



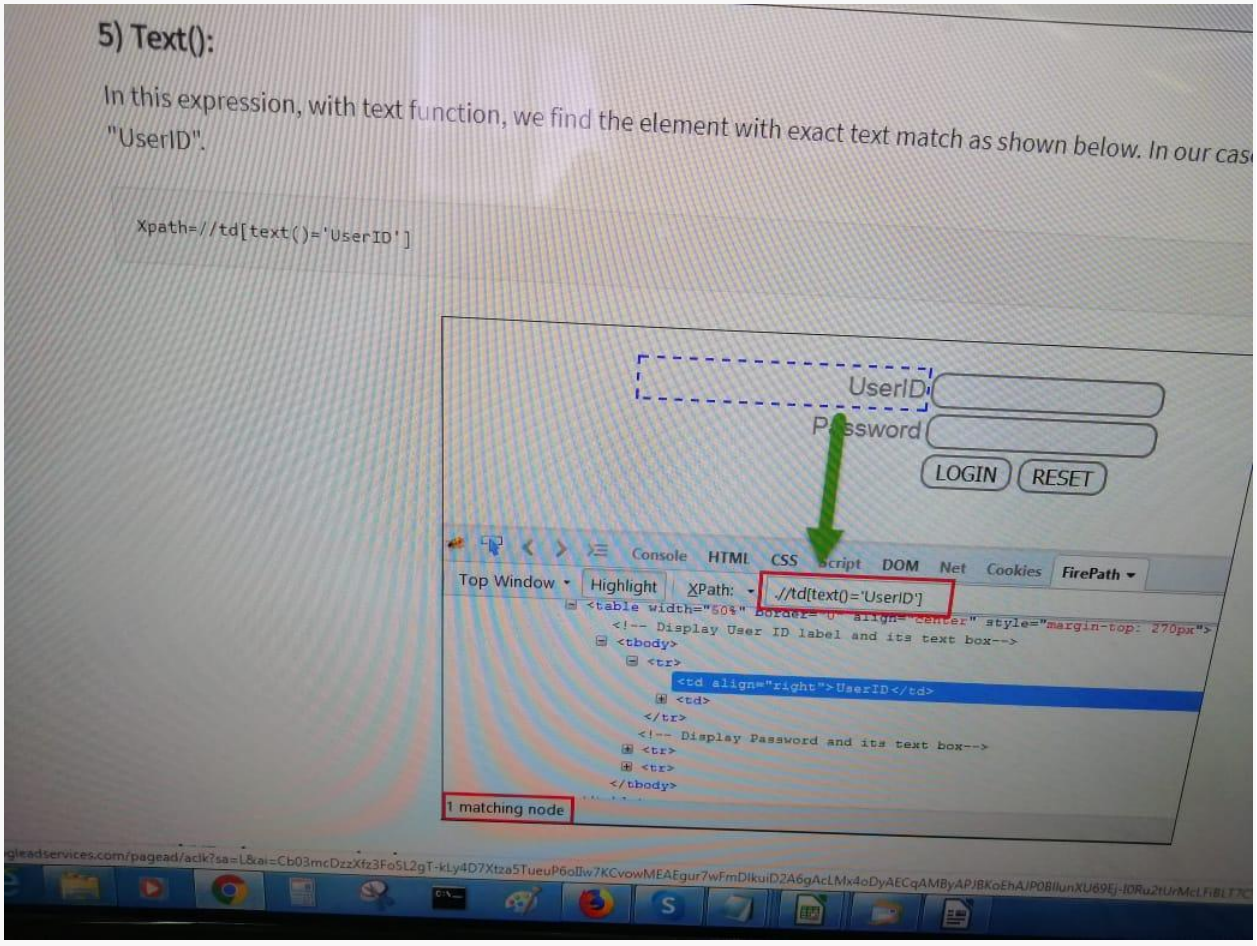
5) Text():

In this expression, with text function, we find the element with exact text match as shown below. In our case, we find the element with text "UserID".

5) Text():

In this expression, with text function, we find the element with exact text match as shown below. In our cas
"UserID".

```
Xpath=//td[text()='UserID']
```

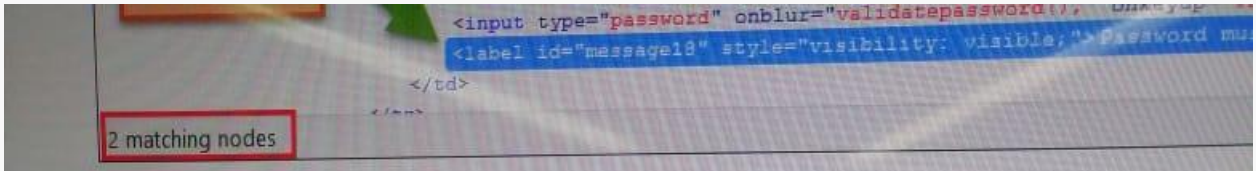


xpath,css selector and etc.....

Syntax as below:Class which is having element locators collection

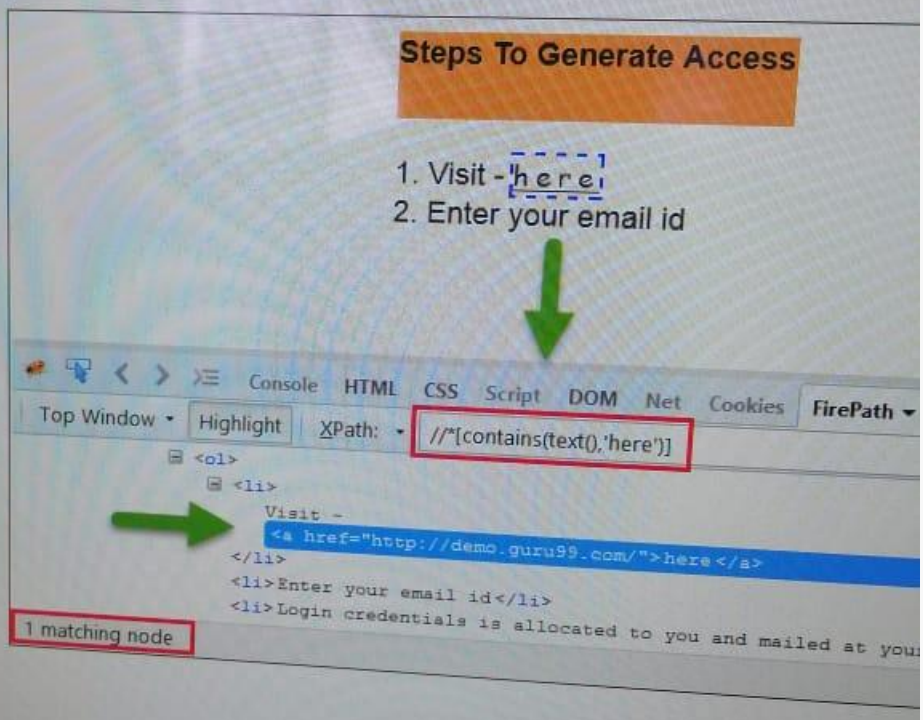
```
public class name {
public WebDriver driver;
    By
menu=By.xpath("//*[@id='side-menu']/li[5]/a/span");
    By
fullbreadscrum=By.xpath("//*[@id='content']/div/div[1]/div
/div/ul");
                                    ......
                                    ......
                                    ......
            public source(WebDriver driver) {
             this.driver=driver;
    }
    public WebElement getmenu(){
    return driver.findElement(menu);
}
    public WebElement getfullbreadscrum(){
   return driver.findElement(fullbreadscrum);
    }
                                    ......
                                    ......
                                    ......
                                    ......
                                    ......
}
```

5.We should return the inspect element ids fro



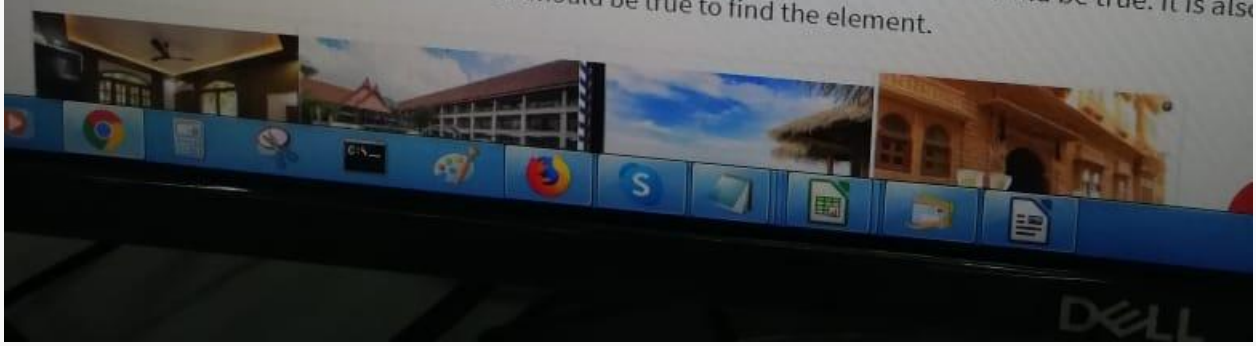
the below expression, we have taken the "text" of the link as an attribute and 'here' as a partial v
ill find the link ('here') as it displays the text 'here'.

```
Xpath=//*[contains(text(),'here')]
Xpath=//*[contains(@href,'guru99.com')]
```



3) Using OR & AND:

In OR expression, two conditions are used, whether 1st condition OR 2nd condition should be true. It is als
true or maybe both. Means any one condition should be true to find the element.



INTERVIEW QUESTIONS (PART 2)

Project specific Questions

1. What was the duration of your project?
2. Explain about your project.
3. How many testers were their on this project.
4. According to you which was the complex part of the project from testing point of view.
5. How did you do the testing of your project?
6. How many test cases have you designed? How many you wrote in a day?
7. How many bugs did you find?
8. Tell me any high Severity bugs that you found.
9. What happens when the client changes requirements?
10. Which tool you used for defect reporting?
11. What is the database used for your project.
12. In which technology this application is developed?

Company specific Questions

1. Where are you working currently?
2. How that Seed has sent you here for an interview when you are working in the same company.
3. Where it is located?
4. How many testers and developers are their?
5. Are testers and developers sitting at same location?
6. What is the hierarchy in testing team?
7. Whom do you report to?
8. Is this a dummy project?
9. How do you come to know about your tasks?
10. Are you a permanent employee?
11. How long you are working in Seed?
12. Can you name some clients of your company?

HR Questions

1. Introduce yourself.
2. What are your strengths?
3. What are your weaknesses?
4. What you do not like in this world?
5. How do you commute?
6. Are you a team player?

7. How long will you take to join?
8. Are you ready to work on contract position?
9. Where would you like to see yourself 5 years down the line.
10. Are you a permanent employee?

1. Define what is Software Testing?
2. What is difference between Severity and Priority?
3. What is difference between Regression and Retesting?
4. What is the Role of the Tester?
5. How will you convince a developer who is not ready to resolve the bug you have raised?
6. What is difference between Debugging and Unit testing?
7. Have you done White Box Testing? Do you know what type of bugs you can trace through this type of testing?
8. What is the important column in Test case?
9. What is the use of any Test Case Management tool?
10. Why Testing is necessary?
11. What is the Bug Life Cycle?
12. How do you decide which test cases to consider for Regression Testing?
13. Is testing QA or QC?
14. Differentiate between Integration Testing and System Testing.
15. Have you participated in User Acceptance Testing (UAT)?
16. Have you seen a Test Plan? Who prepares it ?
17. What are the various contents of a Test Plan?
18. What methods of Black Box Testing you have used?
19. Do you know Exploratory Testing?
20. What is Prototyping? What are the various types?
21. Define phases of SDLC.
22. Which SDLC model your company follows?
23. When do we write Stubs?
24. What is the use of Driver script?
25. What kinds of reviews are conducted during SDLC phase?
26. What is the role of Metrics in Testing?
27. What are key challenges of Software Testing field?
28. Explain your views about Quality.
29. What is the advantage of using any bug tracking tool.
30. Explain about website Usability testing.
31. What are defect attributes?
32. What is pesticide paradox?
33. Explain V model.
34. What is Load and Stress testing, explain with example
35. Differentiate between Verification and Validation?

36. Explain Boundary Value Analysis with an example.
37. What is the difference between Web application and Client Server Application?
38. What is Localization testing?
39. What is difference between Black box testing and White box testing?
40. What is the difference between Authentication and Authorization give an example.
41. Other than functionality what else you should test in Web Application?
42. How you know that testing is enough?
43. How will you test the reports?
44. What is compatibility testing?
45. If we test the application on IE 8 then is it necessary to test it on IE 7?
46. Difference between Smoke & Sanity testing.
47. What is the difference between Requirement specification & Functional Specification?
48. On what basis you will decide whether all the requirements are covered in the test cases?
49. What is Database? Which database you are aware of?
50. What is Database Testing?
51. What is Cookies? Any idea on Cookies Testing?
52. Explain Spiral Model?
53. What are Stored Procedures?
54. How many types of Joins are their?
55. What are DML and DDL commands give example.
56. What is Normalization?
57. What is Primary key and Foreign key?
58. What is RDBMS?
59. What is difference between delete & truncate?
60. What design documents you receive from PM?
61. What are documents are created in Design phase?
62. What challenges you have faced in testing so far?
63. Have you seen our company web site? What are the usability defects you have found?
64. Differentiate between Load testing and Stress testing.
65. Which documents does the test team submit during testing period?
66. How will you test an application without and requirement documents?
67. What is windows registry?
68. What is Object, multiple Inheritance and Class in OOPS?
69. What is Multitasking, Multithreading and Multi processing?
70. What is Stack, and Queue in DS?
71. Explain STLC.
72. What is Functional Testing?
73. What activities are done in Inspection.
74. What is Deferred Defect?
75. In Defect Life cycle, you find that the defect is not fixed but the developer says that it is fixed will you every time tell the Project lead about that?

76. What is Equivalence Partitioning?
77. How many regression cycles you have participated in?
78. Realizing you won't be able to test everything - how do you decide what to test first?
79. What is SQA?
80. What are the contents of defect report?
81. Explain RTM template.
82. What is Big Bang type of testing?
83. What is Traceability Matrix & Coverage Matrix?
84. What is a fault, failure & defect?
85. Which is the efficient method to write (a) test scenario (b) Many test cases?
86. What is UAT and Alpha and Beta Testing?
87. Explain doubly and singly linked list?
88. Explain OOPS features?
89. What are the key challenges in testing?
90. What is Hot fix?
91. Where Exit and Entry Criteria are written in Test Plan?
92. What are different scenarios for verifying Email address?
93. What is the query for update table?
94. If there is mistake in company Logo, what is Priority and Severity Justify your answer?
95. Can Stub/Driver be use as an independent module?
96. Who prepares Test Plan in your organization?
97. Where feature to be tested and features not to be tested are written in test plan?
98. What is Test Strategy and Test Approach.
99. What is the data flow in your project?
100. Where Regression Testing comes in Defect Life Cycle
101. What is deference between Integration and System Testing
102. Examples of Non-Functional Testing?
103. What is difference between QC and QA?
104. If 4 modules are there like A, B, C and D. All are dependent on each other. If changes is done to Module C. Will you perform Regression Testing? If yes how many test case you run again?
105. How much test cases you wrote in a single day?
106. Are you submitting the defects you found each day or submit them when you finish a complete module?
107. What is Data Migration Testing? What are the different approaches followed in Data Migration?
108. What are broken links?
109. What is Closer Report?
110. What is System Testing?
111. What type of testing should we performed while testing a website
112. What is Security Testing? Different aspect we should cover while doing security Testing
113. Arrange the following testing types in sequence in which we test any application
Functional Smoke Integration System Sanity Retesting Regression

114. If same software convert English language PDF file to Spanish language word file And some lines are showing the converted data like @#$#$%#$%$#%$% (special Character). Then is it a functional issue or usability issue.
115. If any bug is reproducible only on Testing server and not on Development server, how you make sure that the developer fixes that bug .
116. If any bug is reproducible at client side (in production environment) not in the test environment what are the different possibilities that this scenario occurs.
117. What is security testing?
118. What is Installation Testing?
119. What is Application Server? Give example.
120. Have you come across any severe bugs in your application and did you contribute so that it gets resolved.
121. Have you ever prepared Test Plan?
122. How you define Severity to a Bug?
123 You are given little understanding of application, how much time will you take to understand it?
124. What do you think why software has bugs?
125. Explain error guessing with an example.
126. In bug life cycle, who opens the bug?
127. What is Compatibility, Multiplatform and Configuration testing?
128. What are Session and Cookies and what is difference between them?
129. What is the process followed in your company for testing?
130. What is Mutation testing?
131. Difference between Ad-hoc & Exploratory testing?
132. What is Test Suit?
133. How you are tracking defects?
134. What Qualities does QA Engineer Require?
135. What are CMM Levels?
136. Does Code review come under white box testing?
137. You have been given a form, how will you test it?
138. What is Static and what is Dynamic Web Page?
139. Difference between HTTP and HTTPS? Explain how the data is secured in HTTPS?
140. Have you heard of Agile Methodology? Explain?
141. Have you used any Configuration Management tool?
142. Do we always need to do regression after retesting?
143. What is Web server? Which web servers you know?
144. What is Inter-system testing? Have you done that?
145. What is a build? What is the release?
146. What is production server?
147. What is Recovery Testing?
148. Any disadvantages of V model?
149. What are the properties of a good Requirements Document?

150. What is Accessibility Testing?
151. Write test cases on :
Chess board, white board, pen, water bottle, elevator, telephone instrument, mobile SMS service, ATM, Ceiling fan

2017 Selenium Automation Testing Interview questions asked in MNC's
Aasked Automation Interview Questions in MNC's
Company (1)
1) Tell me about yourself?
2) Tell me Your Roles and Responsibility?
3) Have you worked on designing?
4) Flow of your framework, Model project?
5) Define Encapsulation?
6) Difference between method overloading and method overriding?
7) Interface and abstract?
8) Define Interface?
9) Static keyword in java?
10) explain “this” keyword?
11) Multiple and Multilevel Inheritance.
12) Different types of Exceptions.
13) Define Checked and Unchecked exception.
14) Difference between List and Set.
15) What is ArrayList?
16) Different types of “wait” in selenium, Describe each with syntax.
17) How to handle web based popups?
18) What are different types of alert method?
19) How to switch from one window to another?
20) What is table handle and window handle?
21) How to read & write data from excel sheet?
22) How to take screenshot in selenium?
23) How to highlight a link in selenium?
24) What is additional setting and proxy setting in selenium?
25) What is Firefox Profile?
26) What is Jenkins?
27) How to upload data in Jenkins?
28) Jar files for logging
29) Automation of testing
30) Parameterized in TestNG
31) Execute test case parallel in TestNG?
32) Can you give some Priority to two test cases?
33) Parallel parameter= true? Explain.

34) Have you worked on SQL query? What you have done?
35) What is Primary Key please explain.
36) If there are same multiple entry in the table, how to find Unique element in the table?
37) How to get 3rd Highest Salary from employee table?
38) How to find any particular date of joining of employee from Employees table, date is in number format then how to change it to character format?

Aasked Automation Interview Questions in MNC's Company (2)

1) Tell me about yourself?
2) What is payroll? Explain about your project with your rolls and responsibility?
3) What is Integration testing you performed in your project please explain with scenarios.
4) Types of Integration testing.
5) What is hybrid Integration testing?
6) Explain Top-down and Bottom-up Integration testing.
7) What is depth and breadth means in Integration testing?
8) In Integration testing which testing performed first?
9) Tell me about System Integration testing?
10) Have you written any Test Plan?
11) Parameter used in creating test plan?
12) Why Regression testing is required?
13) If fault found who will be originally documented by?
14) Have you logged any defect?
15) What is V-Model?
16) What is end to end testing?
17) What is use case?
18) Difference between use case and test case?
19) What is boundary values analysis?
20) What is traceability matric?
21) Have you sending any test report?
22) What parameter you used to send test report?
23) What is spike testing?
24) What is soap testing?
25) Why Bug occur?
26) What is difference between manual and automation testing?
27) What is better manual or automation testing?
28) Manual testing framework?
29) What is execution testing and non-execution testing?
30) Any query?

Aasked Automation Interview Questions in MNC's

Company (3)

1) How many Test Cases you have automated till yet?
2) How many Test Cases you have executed in regression in current project?
3) Total test cases count in regression suite?
4) How much time it will take to prepared & execute these 80 test cases of regression test suite
5) Did you face any difficulty at the time to preparing Test Suite?
6) Did all the test case passed when you execute regression suite?
7) Did you got functional or non-functional defect?
8) Difference between private and public access modifiers?
9) Can private member of class be access outside the class?
10) Difference between private and protected?
11) How many web pages are opened in system can we count in selenium?

```
ArrayList<String> tabs = new ArrayList<String>(driver.getWindowHandles());
System.out.println("No. of tabs: " + tabs.size());
```

12) Difference between find element and find elements?
13) What is getWindowHandles();
14) What is web based popup and window based popups?
15) What is generic function in selenium?
16) How many generic function you have written in your current project? Please specify.
17) Did you know how to connect to excel sheet through java?
18) Do you know what is jenkins?
19) Did you performed continue integration in your current project?
20) Which framework you have used in your project?
21) Have you used TestNG in your project?

Aasked Automation Interview Questions in MNC's

Company (4)

1) Tell me about yourself?
2) How to read from property file in selenium and its method?

```
File src = new File(path);
FileInputStream fs = new FileInputStream(src);
Properities pro = new Properities();
pro.load(fis);
pro.getproperty(keyvalue);
```

3) How to read excel file?
4) How to get Row count and Column count in excel file?
5) How to read cell data in excel?
6) Please explain Xpath.

7) How to perform double click on any component?

```
Actions actions = new Actions(driver);
WebElement elementLocator = driver.findElement(By.id("ID"));
actions.doubleClick(elementLocator).perform();
```

For right click

```
actions.contextClick(elementLocator).perform();
```

8) Did you perform any Data Base validation using java selenium code?

9) Difference between Abstract and Interface.

10) Can Interface be extended to another Interface?

12) Have you done any Parallel testing execution in selenium TestNG?

13) What kind of Report Generation you have done in TestNG? Is it XML, HTML excel report?