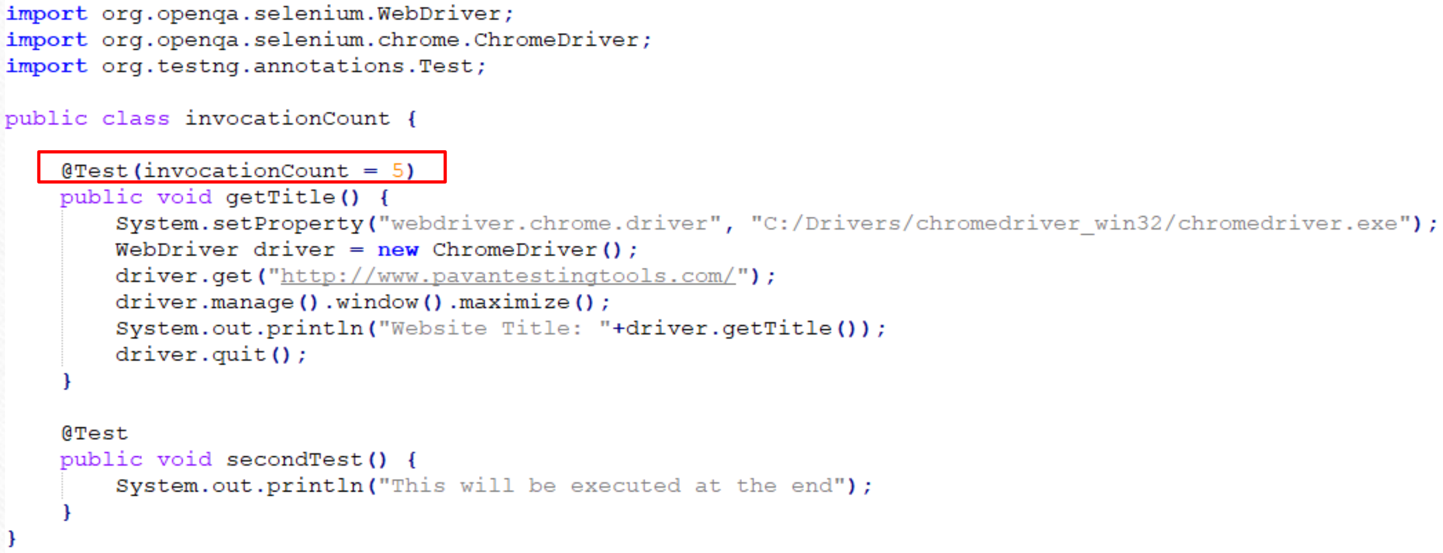
SELENIUM FAQ’S

Part-5

Ques.101. How can we run a Test method multiple times in a loop(without using any data provider)?

- Ans. Using invocationCount parameter and setting its value to an integer value, makes the test method to run n number of times in a loop.



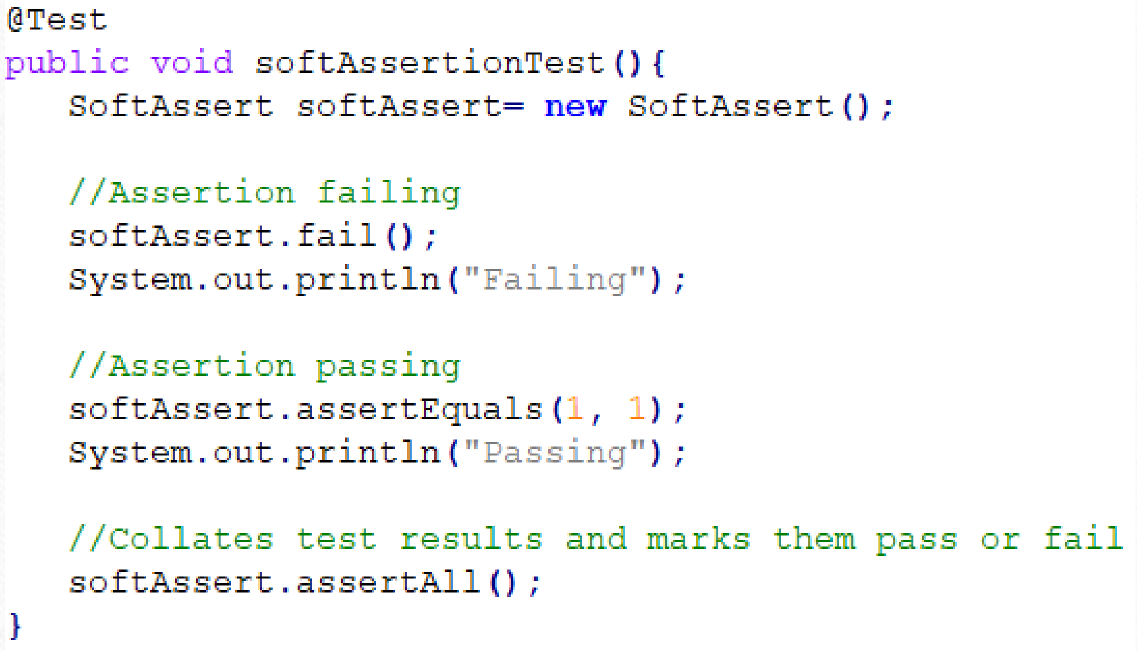
```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;

public class invocationCount {

    @Test(invocationCount = 5)
    public void getTitle() {
        System.setProperty("webdriver.chrome.driver", "C:/Drivers/chromedriver_win32/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("http://www.pavantestingtools.com/");
        driver.manage().window().maximize();
        System.out.println("Website Title: "+driver.getTitle());
        driver.quit();
    }

    @Test
    public void secondTest() {
        System.out.println("This will be executed at the end");
    }
}
```

Ques.102. What is the default priority of test cases in TestNG?

- Ans. The default priority of test when not specified is integer value 0. So, if we have one test case with priority 1 and one without any priority then the test without any priority value will get executed first (as default value will be 0 and tests with lower priority are executed first).

Ques.103. What is the difference between soft assertion and hard assertion in TestNG?

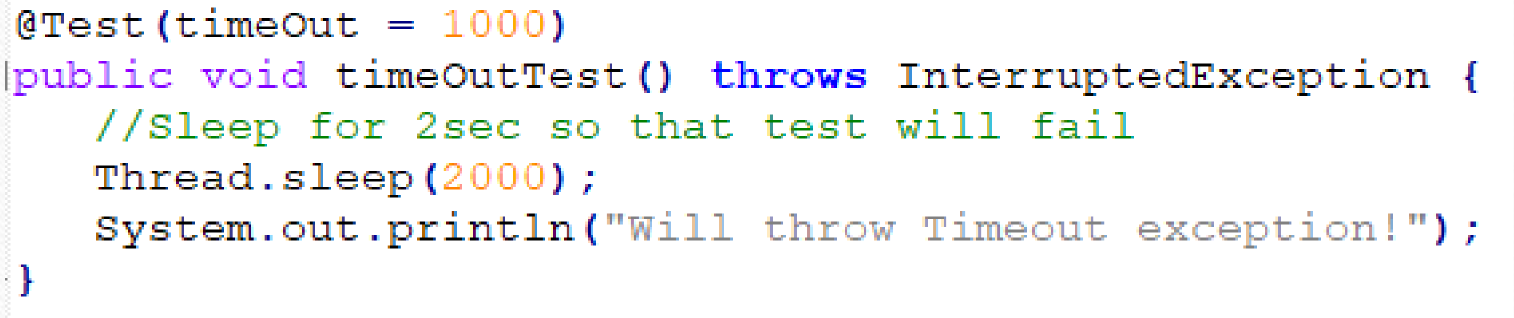
- Ans. Soft assertions (SoftAssert) allows us to have multiple assertions within a test method, even when an assertion fails the test method continues with the remaining test execution.
- The result of all the assertions can be collated at the end using softAssert.assertAll() method.
- Here, even though the first assertion fails still the test will continue with execution and print the message below the second assertion.
- Hard assertions on the other hand are the usual assertions prodived by TestNG. In case of hard assertion in case of any failure, the test execution stops, preventing execution of any further steps within the test method.



```
@Test
public void softAssertionTest(){
   SoftAssert softAssert= new SoftAssert();

   //Assertion failing
   softAssert.fail();
   System.out.println("Failing");

   //Assertion passing
   softAssert.assertEquals(1, 1);
   System.out.println("Passing");

   //Collates test results and marks them pass or fail
   softAssert.assertAll();
}
```

Ques.104. How to fail a testNG test if it doesn't get executed within a specified time?

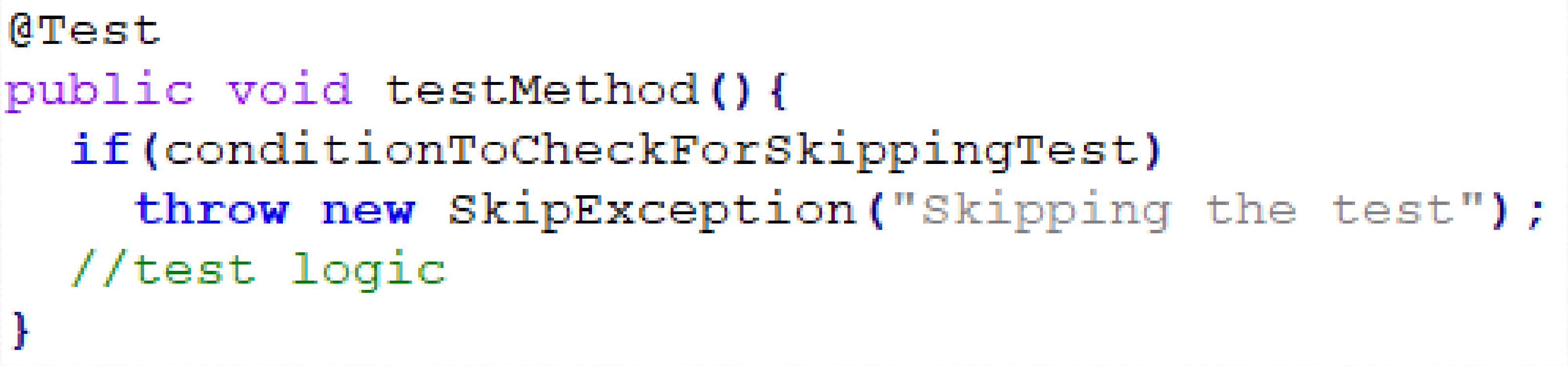
- Ans. We can use **timeOut** attribute of @Test annotation.
- The value assigned to this timeOut attribute will act as an upperbound, if test doesn't get executed within this time frame then it will fail with timeOut exception.



```
@Test(timeOut = 1000)
public void timeOutTest() throws InterruptedException {
    //Sleep for 2sec so that test will fail
    Thread.sleep(2000);
    System.out.println("Will throw Timeout exception!");
}
```

Ques.105. How can we skip a test case conditionally?

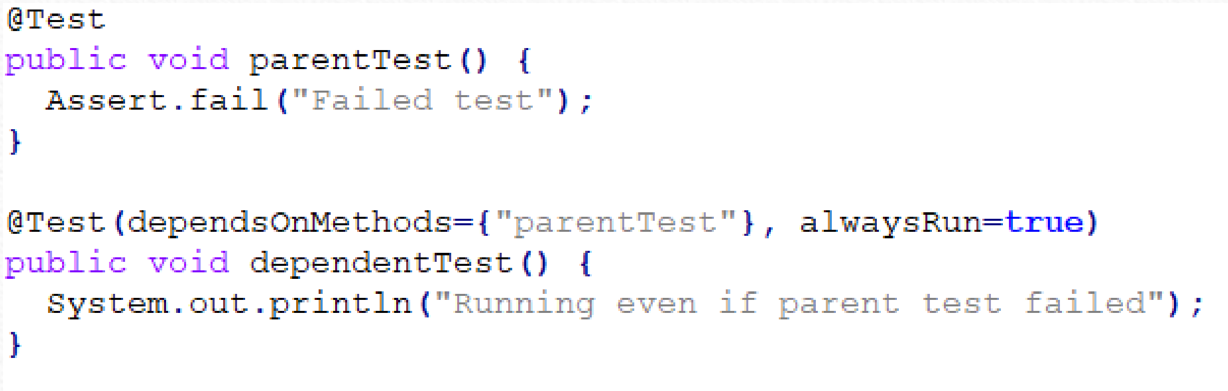
- Ans. Using SkipException, we can conditionally skip a test case. On throwing the skipException, the test method marked as skipped in the test execution report and any statement after throwing the exception will not get executed.



```
@Test
public void testMethod(){
  if(conditionToCheckForSkippingTest)
    throw new SkipException("Skipping the test");
  //test logic
}
```

Ques.106. How can we make sure a test method runs even if the test methods or groups on which it depends fail or get skipped?

- Ans. Using "alwaysRun" attribute of @Test annotation, we can make sure the test method will run even if the test methods or groups on which it depends fail or get skipped.
- Here, even though the parentTest failed, the dependentTest will not get skipped instead it will executed because of "alwaysRun=true". In case, we remove the "alwaysRun=true" attribute from @Test then the report will show one failure and one skipped test, without trying to run the dependentTest method.



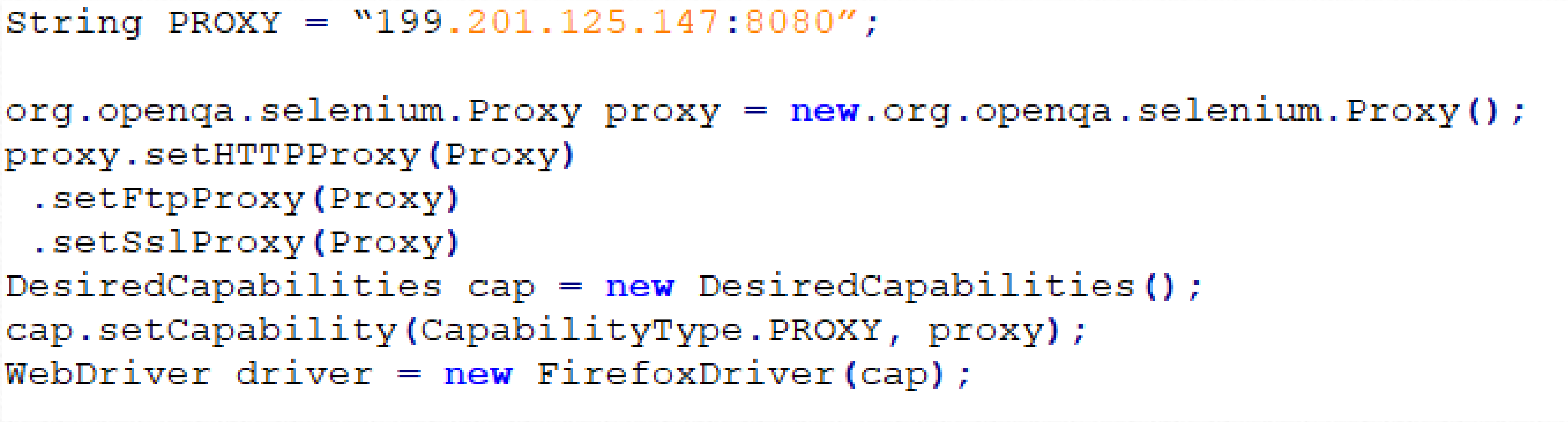
```
@Test
public void parentTest() {
  Assert.fail("Failed test");
}

@Test(dependsOnMethods={"parentTest"}, alwaysRun=true)
public void dependentTest() {
  System.out.println("Running even if parent test failed");
}
```

Ques.107. Why and how will you use an Excel Sheet in your project?

- The reason we use Excel sheets is because it can be used as data source for tests. An excel sheet can also be used to store the data set while performing DataDriven Testing.

Ques.108. How can you redirect browsing from a browser through some proxy?

- Selenium provides a PROXY class to redirect browsing from a proxy. Look at the example below:



```
String PROXY = “199.201.125.147:8080”;

org.openqa.selenium.Proxy proxy = new.org.openqa.selenium.Proxy();
proxy.setHTTPProxy(Proxy)
 .setFtpProxy(Proxy)
 .setSslProxy(Proxy)
DesiredCapabilities cap = new DesiredCapabilities();
cap.setCapability(CapabilityType.PROXY, proxy);
WebDriver driver = new FirefoxDriver(cap);
```

Ques.109. How to scroll down a page using JavaScript in Selenium?

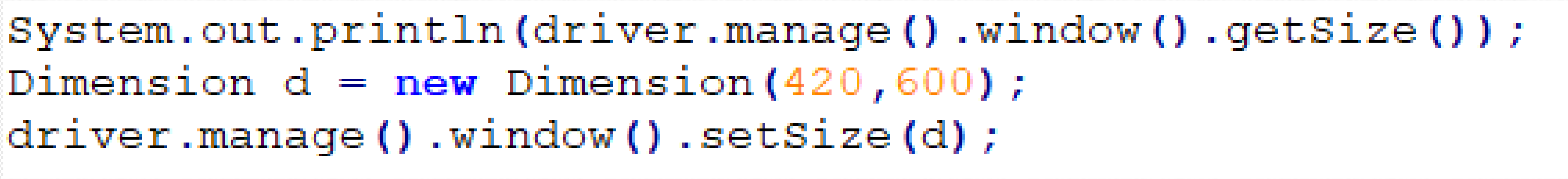
- We can scroll down a page by using window.scrollBy() function.
- Example:
- ((JavascriptExecutor) driver).executeScript("window.scrollBy(0,500)")

Ques.110. How to scroll down to a particular element?

- To scroll down to a particular element on a web page, we can use the function **scrollIntoView()**.
- Example:
- ((JavascriptExecutor) driver).executeScript("arguments[0].scrollIntoView();", element);

Ques.111. How to set the size of browser window using Selenium?

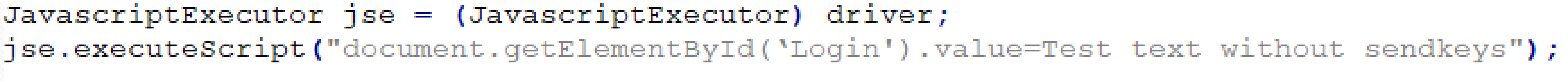
- To maximize the size of browser window, you can use the following piece of code:
  driver.manage().window().maximize(); – To maximize the window
- To resize the current window to a particular dimension, you can use the **setSize()** method.



```
System.out.println(driver.manage().window().getSize());
Dimension d = new Dimension(420,600);
driver.manage().window().setSize(d);
```

Ques.112. Can we enter text without using sendKeys()?

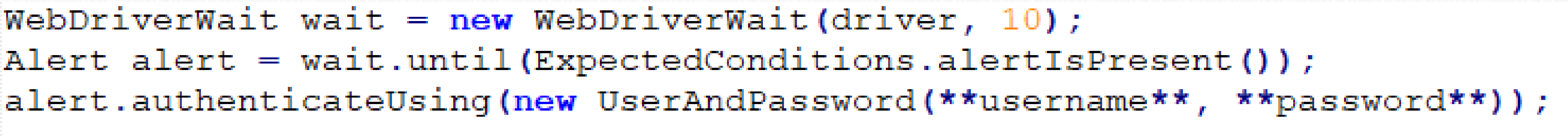
- Yes. We can enter/ send text without using **sendKeys()** method. We can do it using JavaScriptExecutor.



```
JavascriptExecutor jse = (JavascriptExecutor) driver;
jse.executeScript("document.getElementById(‘Login').value=Test text without sendkeys");
```

Ques.113. Explain how you will login into any site if it is showing any authentication popup for username and password?

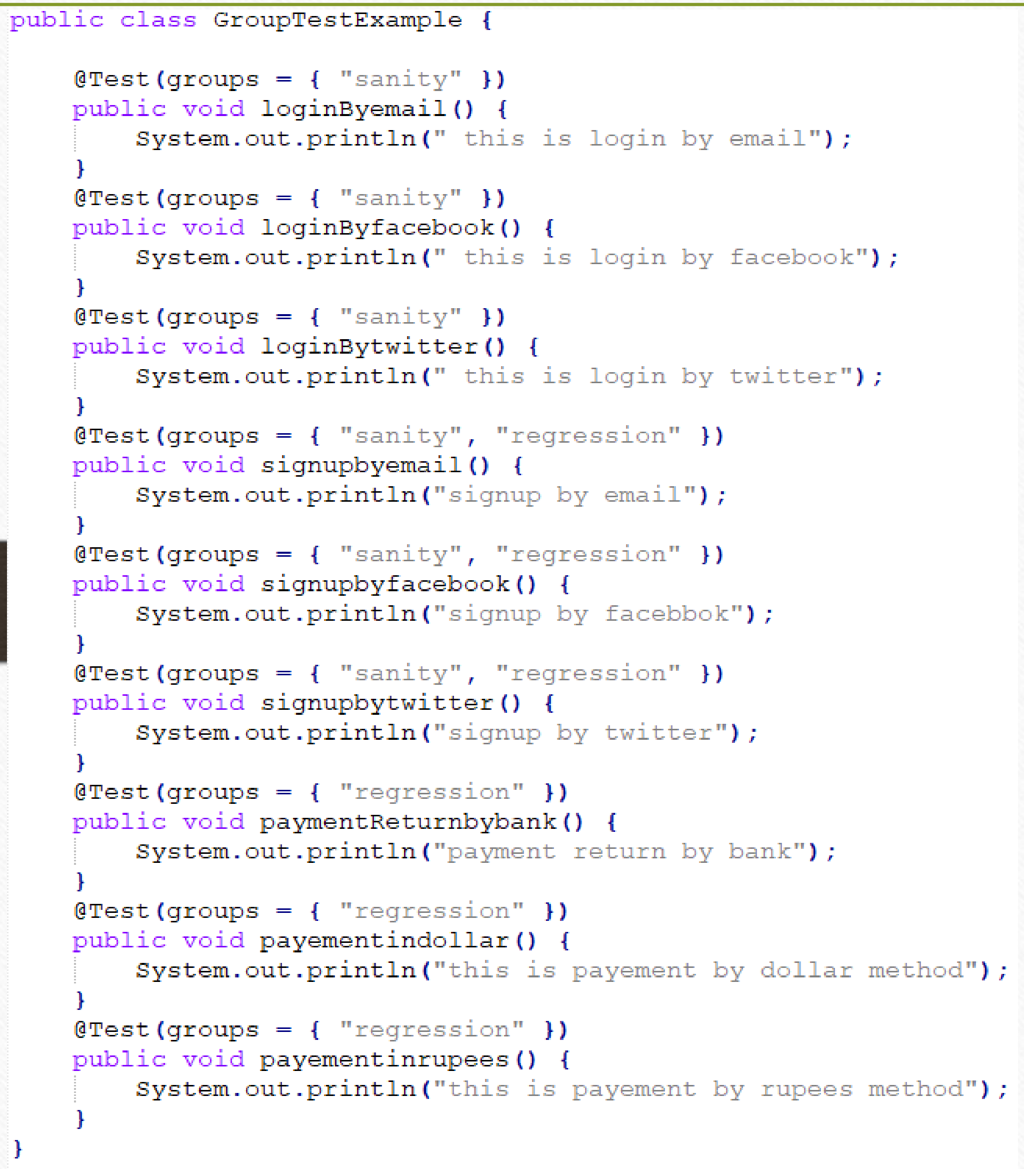
- Since there will be popup for logging in, we need to use the explicit command and verify if the alert is actually present. Only if the alert is present, we need to pass the username and password credentials.
- The sample code:



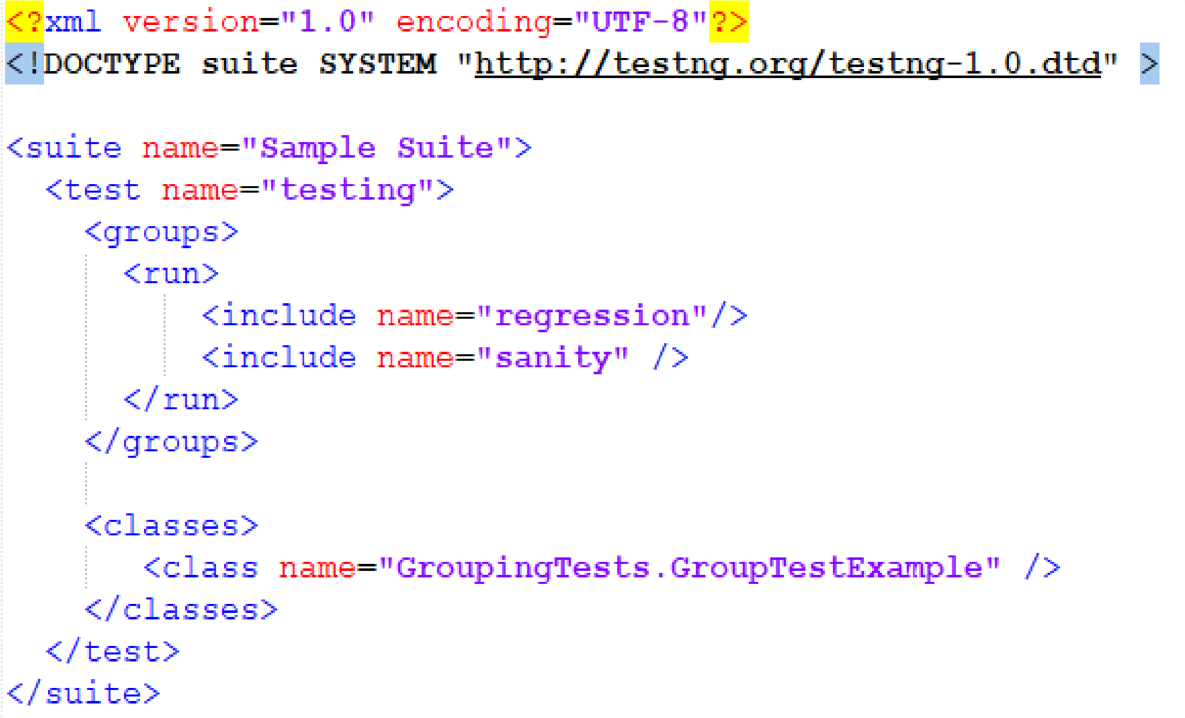
```
WebDriverWait wait = new WebDriverWait(driver, 10);
Alert alert = wait.until(ExpectedConditions.alertIsPresent());
alert.authenticateUsing(new UserAndPassword(**username**, **password**));
```

Ques.114. Explain what is Group Test in TestNG?

- In TestNG, methods can be categorized into groups. When a particular group is being executed, all the methods in that group will be executed. We can execute a group by parameterizing it's name in group attribute of **@Test** annotation. Example: @Test(groups={"xxx"})



```java
public class GroupTestExample {

    @Test(groups = { "sanity" })
    public void loginByemail() {
        System.out.println(" this is login by email");
    }
    @Test(groups = { "sanity" })
    public void loginByfacebook() {
        System.out.println(" this is login by facebook");
    }
    @Test(groups = { "sanity" })
    public void loginBytwitter() {
        System.out.println(" this is login by twitter");
    }
    @Test(groups = { "sanity", "regression" })
    public void signupbyemail() {
        System.out.println("signup by email");
    }
    @Test(groups = { "sanity", "regression" })
    public void signupbyfacebook() {
        System.out.println("signup by facebbok");
    }
    @Test(groups = { "sanity", "regression" })
    public void signupbytwitter() {
        System.out.println("signup by twitter");
    }
    @Test(groups = { "regression" })
    public void paymentReturnbybank() {
        System.out.println("payment return by bank");
    }
    @Test(groups = { "regression" })
    public void payementindollar() {
        System.out.println("this is payement by dollar method");
    }
    @Test(groups = { "regression" })
    public void payementinrupees() {
        System.out.println("this is payement by rupees method");
    }
}
```



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

<suite name="Sample Suite">
  <test name="testing">
    <groups>
      <run>
        <include name="regression"/>
        <include name="sanity" />
      </run>
    </groups>

    <classes>
      <class name="GroupingTests.GroupTestExample" />
    </classes>
  </test>
</suite>
```

Ques.115. How To Run Failed Test Cases Using TestNG In Selenium WebDriver

- By using “testng-failed.xml”

Ques.116. What is Stale Element Exception? How to handle it?

- Stale means old, decayed, no longer fresh.
- Stale Element means an old element or no longer available element.
- Assume there is an element that is found on a web page referenced as a WebElement in WebDriver. If the DOM changes then the WebElement goes stale. If we try to interact with an element which is staled then the **StaleElementReferenceException** is thrown.
- When this happens you will need to refresh your reference, or find the element again.

Ques.117. What are different XPath functions that you have used in your Project?

- Contains()
- Using OR & AND
- Start-with() function
- Text()

XPath axes methods

Following
Ancestor
Child
Preceding
Following
Parent
Self
Descendant

Ques.118. What will happen in background when execute new FirefoxDriver() ?

- Firefox binary will be triggered and Fiefox browser will open with default options.
- FirefoxDriver object is created

Ques.119. What is the below statement means and Why?

WebDriver driver = new FirefoxDriver();

- WebDriver is an interface which contain several abstract methods such as get(...), findElamentBy(...) etc.
- We simply create reference of web Driver and we can assign objects (Firefox driver, ChromeDriver, IEDriver, Andriod driver etc) to it.

Ques.120. How do you handle inner Frames and Adjacent Frames?

- SwitchTo frame1, SwitchTo frame2 (inner frame) work on the element and switchto default content
- Use SwitchTo frame to move the control inside frame.

Ques.121. How to click on an element which is not visible using selenium WebDriver?

- We can use JavascriptExecutor to click.

```
WebElement element = driver.findElement(By.id("gbqfd"));
JavascriptExecutor executor = (JavascriptExecutor)driver;

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

Ques.122. Difference between verify and assert?

- **Assert:** Assert command checks if the given condition is true or false. If the condition is true, the program control will execute the next phase of testing, and if the condition is false, execution will stop and nothing will be executed.
- **Verify:** Verify command also checks if the given condition is true or false. It doesn't halts program execution i.e. any failure during verification would not stop the execution and all the test phases would be executed.

Ques.123. What is the use of @FindBy annotation?

- @FindBy is used to identify element in the Page Factory approach.

Ques.124. Do you use Thread.sleep?

- Rarely

Ques.125. What are different pop-ups that you have handle in your projects?

- JavaScript Pop
  - Alert alert = driver.switchTo().alert();
- Browser Pop Ups
  - Browser Profiles, Robot Class, AutoIT, Sikuli
- Native OS Pop Ups
  - Browser Profiles, Robot Class, AutoIT, Sikuli