# Automation Using Selenium(Advanced Level)

## Day 1:

### Page object Model in Selenium:

Page object model is not a new concept this concept is used by the programmer to make code robust and make code clean. We will also use Page Object model in Selenium to make our framework efficient.

It is a design pattern which will help to maintain the code and code duplication. You can store all locators and respective methods in separate class and call them from the test which you have to use. So the benefit of this if there is any change in the page then you don’t have to modify the test just modify the respective page. It will behave has an object repository where all locators are saved.

There are two choices if you want to implement the POM.

1. POM with page factory
2. POM without page factory

#### POM without page factory

Program 1:Login Page

Approach: Create a separate login page that will store the three locators username, password and login button and create a methods to access them.

Then create a test class so you can use the login class and call all the methods accordingly.

Algorithm:

1. Import the packages
2. Create a package
3. Create Login Class
4. Declare the three locator – username , password and login button
5. Create a new package inside same project
6. Import the login class package
7. Create the test class
8. Call the login method from 1st package

* he length of your resume looks appropriate given your career level and objective
* Headings throughout your resume seem inconsistent at some places
* Font type and size are not consistent across the resume. We recommend you use one of the below fonts (recommended font size indicated in brackets for each font): Verdana (9-11), Calibri (10-12), Cambria (10-12), Arial (9-11) and Lucida Sans Unicode (9-11); Time New Roman (9-11) and use Black Font Color across the document.( Non headings)
* Type of bullet points seem consistent throughout your resume
* You have used bolding most appropriately in your resume
* Line spacing throughout your resume is not consistent
* Your resume has some underlined content. It is suggested that you remove the underline and bold the same
* Paragraphs & bullet points are not aligned properly in your resume
* Some of the paragraphs in your resume are lengthy. They should not exceed 20 words
* Length of bullet points throughout your resume seems lengthy and can be reduced
* Work experience is correctly listed in reverse chronological order in your resume

SCORE : **58/70**Content

* Your resume does not contain summary section or does not have enough content in summary section
* Your resume seems to have limited number of sections. It is suggested that you add a few more to enhance the readability of your resume
* Targeted functional area or industry is appropriately mentioned in the objective, summary or personal details section
* Your resume carries skillset and skills are appropriately substantiated in work experience highlighting your areas of specialization
* You have not quantified your achievements in your resume. It is suggested that you do so to add more credibility to your achievements
* Your resume does not have any spelling errors.
* Your resume carries certain repetitive information. It is suggested that you avoid repetition for better impact
* Your resume is packed with action phrases and Functional Area specific keywords providing better ranking in recruiter searches
* Your resume has valid Email ID for contact
* Your resume has valid mobile number for contact
* Your resume has most appropriate words and does not contain any banned words.
* Your resume has most appropriate words and does not contain any commonly mis-spelt words.
* Your resume does not have any error related to use of commas, parenthesis, correct case, word repetition, extra space, correct use of vowels etc.
* Your resume has sufficient soft skills effectively portraying your personality.
* Your resume carries certain repetitive information. It is suggested that you avoid repetition for better impa

### Accessing elements from dropdown

To select the elements from the drop down first do the following 2 things.

1. Import the package org.openqa.selenium.support.ui.select
2. Instantiate the drop down box as a select object

##### Selecting items in a multiple Select Items

We can select multiple option by selecting the item using the select object one by one.

Common Methods in Select class

1. Select By Visible Text & Deselect By Visible text
2. Select By Value & Deselect by Value
3. Select By Index & Deselect by Index
4. Is Mutliple()
5. DeselectALL()

### Accessing Links and tables in Selenium

Link can be accessed using the full text match by “by.LinkText” or by the partial matching

If there are two links with same name then selenium will get the first link

If there are two links contains the same matching words when using the partial link text then selenium will get the first link

Both By.LinkText and By.PartialLinkText are case-sensitive

If you want to check all links in web page is working or not then follow the below procedure

* Get all the links and store in a List of Web Element
* Create a String array with the size of list of web Elements
* Run For Each loop and store the link text in each String array
* Run for each loop with String array and click on the link
* Before all these create a string var storing the common under construction message for broken links
* In the for each loop with string array compare the title with constant
* If both equals then print not working
* If not then print working

##### Link outside and inside a block

Web driver access both inside the block link and outside the block link

##### Images Link

Images link cannot be accessed through LinkText or PartialLinkText because the images link doesn’t have any LinkText . To access images link we have to use the Css or Xpath functionality you can use the css because of its functionality

If the link is inside the frame then first switch to the frame and then search for the link

Clicking link by href value also can be use first store the link in a webelement and then get the attribute of the webelement and match with expecting href and if both matches then click the element

Other way to click on a hyperlink is by xpath or css

By.xpath(“//\*a[@href=”------”]”)

By.cssSelector(“a[@href=’------- ’]”)

##### Handling Tables using Selenium

To access the HTML table data we have to use XPATH

//table/tbody/tr[1 or 2 or…depending on row]/td[1 or 2 or 3..]

##### Accessing Nested Tables

Nested tables are tables located within another table

Use the same principle of xpath to get he nested table data

//table/tbody/tr[1 or 2…]/td[1 or 2….]/table/tbody/tr[1…]/td[1..]

##### Using Attribute as predicate

When the table is buried deep into the html page and it difficult to get the which number of row and column the value is stored i.e; predicate

Then the attribute of the table can be used to get the value stored there

Remember when using the attribute value stored within the “” use the escape character \ slash to prevent the string to prematurely end within xpath

\”240”\

//table[@width=\”240”\]/tbody/tr[1]/td[1]

### Keyboard and Mouse Event in Selenium

Keyboard Events

KeyUp(keys Modifer Key)

KeysUp(WebElement Element , Keys Modifier Key)

KeyDown(Keys Modifier Key)

KeyDown(WebElement Element, Keys Modifier Key)

Release()

SendKeys(CharSequence KeysToSend)

SendKeys(WebElement element, CharSequence KeysToSend)

Mouse Events

ClickAndHold()

doubleClick()

contextClick()

dragAndDrop()

dragAndDropBy()

MoveToElement()

MoveToOffset()

### Download and upload using selenium

Uploading a file – You can upload a file using the send keys method on the file select input field passing the path of the file which needs to be uploaded

Downloading- Selenium cannot handle the window dialog box for download rather there are below options to download

* 1. Window automation - AutoIT
  2. Change the browser default behaviour-
  3. Direct Download- wget

Wget: Wget is a small command line program to automate download , we can access the wget from selenium webdriver

Step 1: Import java.io.IOException

Step 2: using the GetCssAttribute get the href attribute of the and store in the String

Step 3: Set up the syntax for Wget command

String wget\_command= “cmd/c wget –P D: --no-check-certificate” +sourcelocation(location of href attribute)

Step 4: Initiate the run

Process exec= Runtime.getRuntime().exec(wget\_command)

Int exitVal= exec.waitFor()

### TestNG

Features of TestNG

* Support for annotations
* Support for parameterization
* Provide data driven testing by using data providers
* Advance execution methodology that does not requires test suite be created
* Option to execute particular test by using textng.xml
* Prioritization
* Report mechanism
* Support integration with other tools like ANT, Maven Eclipse
* Execution pattern can be set
* Concurrent execution of the test suite
* Test Case dependencies can be set

### Assertion in Selenium

AssertEqual(“String actual”,”String Expected”,”Message if fails”)

AssertEquals(“Boolean actual, Boolean Expected”)

AssertEquals(“java.utill.collection actual, java.utill.collection expected , “messgage”)

AssertTrue(Condition)

AssertTrue(Condition,message)

AssertFalse(Condition)

AssertFalse(Condition, Message)

TestNG.xml

Testing.xml file is created to handle multiple test cases, In this file we can add test case dependency , include or exclude any test,class,method or package, set priority , provide data

### Excel file operations in Selenium

###### Reading the data from Excel

1. Create a Read method of type void with three parameters(Filepath,FileName , SheetName)
2. Create a file object to open the file
3. Create FileInputStream object to start reading the file
4. Using the FileInputStream object create a Workbook object
5. Using the Workbook object and SheetName from parameters create Sheet object
6. Now get the total numbers of rows in the sheet
7. Run the loop for till the last row using the number of rows count
8. Create a row object and get the first row
9. Using the first row create a loop till last cell value
10. Using the row get the Cell value and Print or store the value
11. Create a Main method and create an object of the read class
12. Declare the FilePath
13. Call the Read class with Filepath , Filename and Sting name

Writing in excel

### Taking Screenshot :

GetScreenShotAs method of interface TakeScreenshot is used to take the screenshot.

Public <X> X getScreenshotAS(OutputType<X> target)

OutputType- another interface

File filename= ((TakeScreenshot)driver).getScreenShotAs(outputType.file);

### Handing windows:

First get the window handle then use the command to switch to that window handle

Driver.switchTo.window(WindowHandle)

To get all the window handles use .getWindowhandles

### Handling Frames

We can switch to frame by using the Frame Id, Name or index

##### Select Frame by Its index

Driver.SwitchTo.Frame(int index)

Throws: NoSuchFrameException – if the frame is not found

##### Select Frame by Its name

Driver.SwitchTo.Frame(String framename)

Throws: NoSuchFrameException – if the frame is not found

##### Select Frame by Its name

Driver.SwitchTo.Frame(String framename)

Throws: NoSuchFrameException – if the frame is not found

##### String Programs:

1. Write a java program to find the duplicate words and their number of occurrences in a string?
2. Write a java program to count the number of words in a string?
3. Write a java program to count the total number of occurrences of a given character in a string without using any loop?
4. Write a java program to reverse a string?
5. Write a java program to count the number of occurrences of each character in a string?
6. Write a java program to remove all white spaces from a string?
7. Write a java program to find duplicate characters in a string?
8. Write a java program to check whether one string is a rotation of another?
9. Write a java program to check whether two strings are anagram or not?
10. Write a java program to reverse a given string with preserving the position of spaces?
11. Write a java program to reverse each word of a given string?
12. How do you convert string to integer and integer to string in java?
13. Write a java program to find the percentage of uppercase letters, lowercase letters, digits and special characters in a given string?
14. Write a java program to prove that strings are immutable in java?
15. How do you find longest substring without repeating characters in the given string?
16. How do you swap two string variables without using third or temp variable in java?
17. Write a java program to find all permutations of a string?
18. How do you find first repeated and non-repeated character in the given string in java?
19. Write a java program to append a given string to a text file?
20. How do you find the number of characters, words and lines in the given text file in java?
21. **How do you find the most repeated word in a text file in java?**
22. program to print first non repeated character from String?
23. How to check if a String contains only digits?
24. How to find duplicate characters in a String?
25. How to count number of vowels and consonants in a String?
26. How to replace each given character to other e.g. blank with %20?
27. How to find all permutations of String?
28. How to check if a String is valid shuffle of two String?  
      
    Read more: <http://javarevisited.blogspot.com/2015/01/top-20-string-coding-interview-question-programming-interview.html#ixzz4dNaA7WQZ>
29. Write a program to check if a String contains another String e.g. indexOf()?
30. Write a program to remove a given characters from String?
31. How to sort String on their length in Java?
32. Program to reverse each word of a given string
33. Count numbers of vowels and constants in a string
34. Program to append a given string to the text file
35. Sort String
36. Find most repeated word in a text file in java
37. Program to find number of words, number of characters, number of lines in a text file

# Q/A

##### Tell me about yourself

Should address:

1. What are your primary selling point for this job
2. Why are you interested in this position

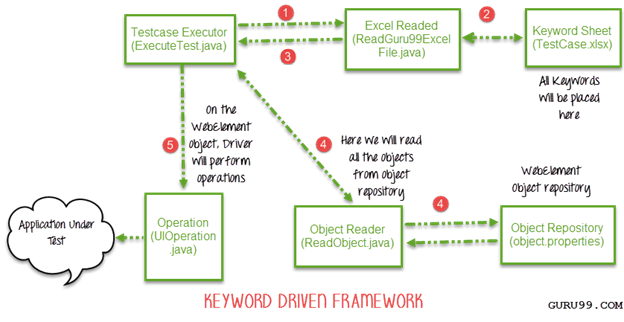
Answer:

1. Who you are: I am a proficient software testing professional with over 3 and half years of experience in software testing industry managing and conducting all aspects of software testing life cycle for a telecom and internet provider company.
2. Expertise Highlights: I have spent last 3 years developing my skills as a software tester in both manual and automation methods for CGI where I took complete responsibilities of delivering a successful test projects. I love finding defects and deriving different scenarios.
3. Why you are here: Although I am ok with my current role , I feel I am ready for more challenging assignment and I see this position is for Automation tester and Automation really interest me and I think I’ll do great job if given opportunity

##### Tell me about your work experience?

Tell me about roles and responsibilities?

Hybrid framework



1. Create a driver script Class : This is a Test Case executer
2. Driver Script will call the Excel Reader class to read the test case , Keywords to be executed
3. Create an Excel reader classes which will read the Test case name , Keyword , test Data an other required things
   1. Create 3 classes , Set Excel File , Get Cell Data , Set Cell Data
   2. Create 2 helper method to help Excel Reader classes GetRowContains and GetRowUsed
4. Excel Reader class should also contain a class to write the results insert row , columns when necessary
5. Based on the test case, test data, keyword return by the Excel Reader classes Driver script will call the UI operation / Action classes to perform the operation on the web page
6. Create Actions/Operations classes
7. Operation classes will use the object repository or the POM classes to perform the operations
8. Create an object repository or Page object classes for every page
9. Also create the object repository reader classes which will read the object from the repository and return it to Driver Script which will pass it to operations classes to perform operations
10. Also create the utilities classes such class to declare all the constants
11. Listner classes to print the Result , Log and status
12. Create log class which will print logs
13. Create class for reporting

# Java

02 May 2017

* Prepare Common Interview Questions
* Read Selenium thoroughly and Interview Question

03 May 2017

* Prepare Common Interview Questions
* Read Java and Interview Questions

04 May 2017

* Prepare Common Interview Questions
* Read SQL thoroughly and Interview Questions

05 May 2017

* Go for interview

06 May 2017

* Revise what happened in the interview and prepare again

3rd May 2017

Selenium 2 Test Automation Framework [By Grazitti Interactive](http://www.grazitti.com/)

Version 3.0

# ==================== FRAMEWORK OVERVIEW

## Selenium 2 (Webdriver) Framework

Setting up Selenium Automation for the first time or running it for separate web applications, you will require some bit of study to get started. You can read up a lot of documentation, experiment with scripts, and repeat the process every time your testing object changes. Or you can simply use our Selenium 2 (Webdriver) Framework. This [Selenium 2 (Webdriver) Framework](http://www.grazitti.com/resources/tools/selenium-2-framework) is a set of guidelines, including coding standards, test-data handling, object repository treatment etc., which when followed will not only save precious time but will also provide additional benefits including increased code re-usage, higher portability, and reduced script maintenance cost.

## Our Approach

For [our framework](http://www.grazitti.com/resources/tools/selenium-2-framework) we have used ‘Data-Driven Framework’ with Selenium 2 (Webdriver) and Java Programming language. A data-driven framework is where test input and output values are read from data files (ODBC sources, CVS files, Excel files and DAO objects) and are loaded into variables in captured or manually coded scripts. In this framework, variables are used for both input values and output verification values. Navigation through the program, reading of the data files, and logging of test status and information are all coded in the test script.

## Framework Features

* Well defined architectural design
* Less time to test large data
* Script execution in multiple environments
* Easier, faster and efficient analysis of result logs
* Communication of results
* Easy debugging and script maintenance
* Robust and stable due to error and exception handling
* 100% reliability of utility scripts, online execution, report packs

## Architecture Explanation

Architecture forms the foundation of any software application. It should be robust enough to handle the desired functions efficiently and effectively. In this approach, the goal is to develop an application-independent reusable Data-Driven Framework that can be used directly across any application without spending any extra time on it. In order to make all the components of the system work in sync, it is important to define the components and its functionalities, as well as the binding relationship between them.

This Package Includes:

* Config - Keeps all the configuration files such as property files
* InputTestData - Has files containing input data for application
* OutputData - Contains downloaded docs/images, fetched data in excel
* TestReports - Contains ANT generated reports
* Util - Utility package contains all generic functions & business functions such as email configuration setting and all other utilities
* TestLogs - Contains log file corresponding to tests
* DAO - Classes for accessing persistent storage, such as to a database
* Pages - Page classes for particular pages

## Understanding of Page Object Model

Page Object Pattern

* A simple design pattern that models each page (or page chunk) as a class object that other classes / objects can interact with
* A very common pattern for implementing an automation framework using Selenium 2 (Webdriver)
* Classes representing pages / chunks should provide the services that a user would execute on the page

## Retrieving Data From DataBase

RegisterTest extends DBCon class having function loadDataFromExcel() retrieving data from excel sheet which further extends GUI\_automation\_base class setting up the browser to be used as mentioned in gui\_automation.properties in config folder.

## Logs:

Initialize the logger for getting logs with: static Logger log = Logger.getLogger(RegisterTest.class.getName());

## Setup Function:

In setup() function call the GUI\_automation\_base setup() function by getting the browser from gui\_automation.properties in config folder, configure the log4j.properties file, load the data from excel sheet and initialize the RegisterPage instance.

## Rules for Automation Framework

* Methods performing an action that results into other page should return the page object to user. Assert and wait to load the page completely with all elements of the page. Example: Clicking a button on page A redirects the user to page B, then returns the object of page A
* Naming of variables and methods: a. Standard JAVA naming conventions should be utilized b. If a method performs an action that clicks on any element on the page, the name of the method should start with click c. If a method retrieves any value or entity, or a list of values or entities, the method name should start with get d. If a method assigns a value to a web element, the method name should start with set e. If a method interacts with radio or check-boxes it should start with enable or disable f. Instance variables should be named using an underscore as a leading character. Example: \_register, \_pageName
* Encapsulation should be used as follow: a. All methods intended for the implementation of tests for the automation of the web project, should be public b. All methods not intended for the implementation of tests should be private c. All variables, with the exception of enums, should be private
* Common methods for similar web elements: a. Lists (HTML tables): getAll\* or getAllByName, findByName, click\*\_ByName b. List-boxes: getAllOptions or getAllOptions\_ByText, click\*Option
* All assertions and verifications should be within the tests using JUnit framework asserts

## Some amazing and distinct features

Selenium 2 (Webdriver) Framework Version 3 has all new features, which will make testing a whole lot easier. Key new features include:

1. Parameterization of Different Input Types: In the earlier versions, we did parameterization only for textboxes. But in version 3.0 we have also implemented it for other input types like radio button, drop downs, check boxes, auto-suggest search drop down, and file upload.
2. Auto-suggest Search Dropdown: The common Webdriver code for selecting drop down is:

Select gender = new Select(driver.findElement(By.id(“gender”))); gender.selectByVisibleText(“Female”);

This does not work for auto-suggest search dropdowns Therefore, we have used following code in our framework for these types of dropdowns:

driver.findElement(locator).sendKeys(value); driver.findElement(locator).sendKeys(Keys.TAB);

1. Image Comparison: This feature helps you to take screenshots of images and compare them with sample images. In our code, we have added 3 sample images in the ‘InputTestData’ folder of the Project and compared them with the screenshots that are stored in the ‘OutputData’ folder.
2. Drag and Drop Functionality: Drag and drop is a very common feature. It is when you “grab” an object and drag it to a different location. We have also also added this functionality using the Actions class:

WebElement element = driver.findElement(By.name(“source”)); WebElement target = driver.findElement(By.name(“target”)); (new Actions(driver)).dragAndDrop(element, target).perform();

1. Sorting Values: Data sort is a common functionality used in different websites especially Business to Consumer sites. You can use framework’s sorting functionality to sort values and check if the sorted order is correct or not. In the test script – SortingTest.java, we have checked sorting of table rows for three columns.
2. Auto-Focusing On New Tab: Some links on the websites open in a new tab but when we click on them via automation script, a new window opens instead of tab; and the automation script breaks. We have now added code in our framework for managing this new window using webdriver’s “switchTo” method.
3. Check and Delete Cookies: Webdriver has amazing feature of cookie handling that we have used in our framework to check and delete cookies for a specific domain.

## Conclusion

A robust framework helps keep pace with agile software delivery and maximize on the benefit of Selenium automation testing. It drives productivity and facilitates code reuse ultimately enhancing the quality of resulting software.

##### Maven-

* POM.xml
* Maven Clean- Run As – Maven Build- Goal- Clean – Run

When it runs the target folder will be deleted , all the jars file after compilation will be saved in target folder

Compile

* Maven Built… - Run AS- Maven Built..- Goal – Compile

When it runs it will create target folder – it will create class file by Maven from the source file

* Maven Built… for Test- Run As – Maven Built…- Goal – test-compile it will create test class folder inside target folder

Run

* Maven Built…. – Run As- mavenbuilt…- Goal- test

In the console window you will see the result whether the test has passed or not , it will create a sorfire-report folder inside the target folder which will have two file among which is a text file where we can see the result of test run

###### Create Jar

* Create Jar- Maven Built…- Goal- install – Run – It will create a jar file inside the target folder , it will same name as class and the version

#### Maven Build Life Cycle

###### Commands

* Validate
* Compile
* Test
* Package
* Integration-test
* Verify
* Install
* Deploy

#### Transitive Dependencies in Maven

Any dependent jar will also be downloaded along with the main dependency jar

#### Exclude Dependencies

To exclude the dependencies one has to know the Group ID and Artifact ID of the jar

Add element <exclusions> and inside exculsion put the Group ID and Artifact ID of the Jar

Scope: Scope can be added to the Dependency to limit the access to the file For example if Scope is Test then it will access only Test file and not Main Source file

Other Scope can be: compile(default), test , runtime, provided

Ant

Jenkin

**Steps:**

* Create three different project

## 4th May 2017

Resume work:

Tim Jobs(Senior Software Engineer and Architect at eBay)

Expert in Designing and implementing scalable architectures; 6+ years of experience in multiplatform environment and mutli- tiered web application design and development at Yahoo, Amazon and Ebay

Tim Jobs(Software Developer with 6+ years of experience at Yahoo, Amazon and Ebay)

Expert in Java and scalable architectures; dabbles in Hadoop and python; Algorithm and Machine Learning enthusiast. When I am not working I’m tackling competitive programming problems, contributing to open source project and playing squash.

## 10th May 2017

Array List

* Add function
* Get Function
* Set function
* To loop through the Array list we can use For loop, Enhanced For Loop and Iterator
* Contains method
* ContainsALL method
* Clear method
* IsEmpty

Linked List

* Same as array list
* AddLast method
* AddFirst
* Set Method
* Remove Function
* Get Method
* GetLast Method
* Contains
* Index OF method
* ISEmpty method
* Size method
* Peek() method
* Poll() Method
* PollLast() Method

Cover latter

**Tim Jobs**

Senior Software Engineer

Phone: 112234455

Website:

Email: [tim.jobs@gmail.com](mailto:tim.jobs@gmail.com)

March 21,2016

XXXX, YYYYY

Vice President , Engineering

$Company

$Company\_Address

Re: Lead Software Engineer opening (Ref ID: CS-45623

Detailed Resume Quality Report

SCORE : **12/30**Format

* The length of your resume looks appropriate given your career level and objective
* Headings throughout your resume seem inconsistent at some places
* Font type and size are not consistent across the resume. We recommend you use one of the below fonts (recommended font size indicated in brackets for each font): Verdana (9-11), Calibri (10-12), Cambria (10-12), Arial (9-11) and Lucida Sans Unicode (9-11); Time New Roman (9-11) and use Black Font Color across the document.( Non headings)
* Type of bullet points seem consistent throughout your resume
* You have used bolding most appropriately in your resume
* Line spacing throughout your resume is not consistent
* Your resume has some underlined content. It is suggested that you remove the underline and bold the same
* Paragraphs & bullet points are not aligned properly in your resume
* Some of the paragraphs in your resume are lengthy. They should not exceed 20 words
* Length of bullet points throughout your resume seems lengthy and can be reduced
* Work experience is correctly listed in reverse chronological order in your resume

SCORE : **47/70**Content

* Your resume does not contain summary section or does not have enough content in summary section
* Your resume seems to have limited number of sections. It is suggested that you add a few more to enhance the readability of your resume
* Targeted functional area or industry is appropriately mentioned in the objective, summary or personal details section
* Your resume does not carry your skillset. It is suggested that you include it in your resume and substantiate your skills in work experience to highlight your areas of specialization
* You have not quantified your achievements in your resume. It is suggested that you do so to add more credibility to your achievements
* Your resume does not have any spelling errors.
* Your resume carries certain repetitive information. It is suggested that you avoid repetition for better impact
* Your resume is packed with action phrases and Functional Area specific keywords providing better ranking in recruiter searches
* Your resume has valid Email ID for contact
* Your resume has valid mobile number for contact
* Your resume has most appropriate words and does not contain any banned words.
* Your resume has most appropriate words and does not contain any commonly mis-spelt words.
* Your resume has few errors related to use of commas, parenthesis, correct case, word repetition, extra space, correct use of vowels etc.
* Your resume has sufficient soft skills effectively portraying your personality.
* Your resume carries certain repetitive information. It is suggested that you avoid repetition for better impact.

## 18th May 2017

Problem faced during Selenium automation project

* As soon as I navigated to the url windows authentication pop up window appears first I thought it to handle it through Auto IT but it didn’t work as the driver.get command never completed , to resolve this I had to use multi-threading concept of java and ran both auto it and get command at the same time

## 19th May 2017

Coding

* Regular expression in java to check string contains digit or not

Pattern pattern= Pattern.compile(“.\*[^0-9].\*”);

Pattern pattern = Pattern.compile(“.\*\\D.\*”);

In character class /d will represent a digit and /D will represent Non Digit characters

* Given a String find the longest substring without repeating characters
* **1-** [Write code to filter duplicate elements from an array and print as a list?](http://www.techbeamers.com/java-coding-questions-software-testers/#q1)  
  **2-** [Write code to sort the list of strings using Java collection?](http://www.techbeamers.com/java-coding-questions-software-testers/#q2)  
  **3-** [Write a function to reverse a number in Java?](http://www.techbeamers.com/java-coding-questions-software-testers/#q3)  
  **4-** [Write a method to check prime no. in Java?](http://www.techbeamers.com/java-coding-questions-software-testers/#q4)  
  **5-** [Write a Java program to find out the first two max values from an array?](http://www.techbeamers.com/java-coding-questions-software-testers/#q5)  
  **6-** [Write a Java program to find the longest substring from a given string which doesn’t contain any duplicate characters?](http://www.techbeamers.com/java-coding-questions-software-testers/#q6)  
  **7-** [Write Java code to get rid of multiple spaces from a string?](http://www.techbeamers.com/java-coding-questions-software-testers/#q7)  
  **8-** [Write Java code to identify a number as Palindrome?](http://www.techbeamers.com/java-coding-questions-software-testers/#q8)  
  **9-** [Write Java code to swap two numbers without using a temporary variable?](http://www.techbeamers.com/java-coding-questions-software-testers/#q9)  
  **10-** [Write a Java program to demonstrate string reverse with and without StringBuffer class?](http://www.techbeamers.com/java-coding-questions-software-testers/#q10)

Testing techniques:

1. Equivalence Partitioning.
2. Use Case Testing.
3. Data Flow Analysis.
4. Exploratory Testing.
5. Decision Testing.
6. Inspections.

## 24th May 2017

Amazon framework.

1. First build a

**Tools:**

\* \*\*IntelliJ IDE\*\* - Best Java IDE in the Whole Wide World!!!

\* \*\*Eclipse Java EE\*\*

\* \*\*Java 7+\*\*

\* \*\*WebDriverJs\*\* - JavaScript

\* \*\*MySQL Connector\*\*

\* \*\*WebDriver 2+\*\*

\* \*\*Selenium Grid\*\* - Not implemented yet!!

\* \*\*SauceLabs\*\* - Not Implemented yet!!

\* \*\*Object Repository\*\* - Not Implemented yet!!

\* \*\*Selenium Server\*\*

\* \*\*Apacha POI\*\*

\* \*\*Apache Ant\*\*

\* \*\*Apache Maven\*\*

\* \*\*JUnit 4.10\*\*

\* \*\*TestNG\*\*

\* \*\*Jenkins\*\* (Hudson) - Not implemented yet!!

\* \*\*XML\*\*

--

The framework consists of the following components.

* com.totsy.config - config.properties - config.or
* com.totsy.log - Application.log
* com.totsy.test - Constants.java - DriverScript.java - Keywords.java
* com.totsy.utils - ReportUtil.java - SendMail.java - Zip.java
* com.totsy.xls - C Suite.xls - Check Items.xls - Links Verification.xls - Login Functionality.xls - My Account.xls - Sales page.xls - Search Suite.xls - Suite.xls
* test(Sample scripts) - Arraylist.java

**Amazon framework.**

* com.amazon.config- config properties – config or
* com.amazon.log- application log
* com.amazon.pages- will contain the page objects of each pages
* com.amazon.test- Constants.java, DriverScript.java, keywords.java
* com.amazon.utils- ReportUtil.java, SendMail.java, Zip.java
* com.amazon.xls- C Suite xls, checkItem.xls, Links Verification.xls, Login Functionali.xls, My Account.xls, Sales page.xls, Search Suite.Xls , suite.xls
* Test (Sameple scripts)- ArrayList.java

com.amazon.config

**config.properties: This classs is created to declare all the configuration data like database details, test url , credential, browser confugiration.**

**com.amazon.pages**

**BaseClass.java: This class contains the webdriver object and result variable object which will be inherited by every other class who will use webdriver object**

# GitHub:

* 1. Error : Git Push Non rejected non-fast-forward
* Configure Fetch in Remote -> Merge Master with Origin/master
* In eclipse, open the view 'Git Repositories'.
* Ensure you see your local repository and can see the remote repository as a subfolder. In my version, it's called Remotes, and then I can see the remote project within that.
* Look for the green arrow pointing to the left, this is the 'fetch' arrow. Right click and select 'Configure Fetch'.
* You should see the URI, ensure that it points to the remote repository.
* Look in the ref mappings section of the pop-up. Mine was empty. This will indicate which remote references you want to fetch. Click 'Add'.
* Type in the branch name you need to fetch from the remote repository. Mine was 'master' (btw, a dropdown here would be great!!, for now, you have to type it). Continue through the pop-up, eventually clicking 'Finish'.
* Click 'Save and Fetch'. This will fetch that remote reference.
* Look in the 'Branches' folder of your local repository. You should now see that remote branch in the remote folder. Again, I see 'master'.
* Right-Click on the local branch in the 'Local' folder of 'Branches', which is named 'master'. Select 'Merge', and then select the remote branch, which is named 'origin/master'.
* Process through the merge.
* Commit any changes to your local repository.
* Push your changes to the remote repository.
* Go have a tasty beverage, congratulating yourself. Take the rest of the day off.

Steps to set up git in eclipse

1. Create a repository in github.com
2. Right click on project and -> Team -> Share Project
3. Create a local repository and give it a name -> Finish
4. Adjust setting to display Github toolbar = Window-> Show view-> Other-> Git -> Git repository & Git staging
5. Adjust setting to display Team History -> Window-> Show view-> Other-> Team-> History
6. Customize perspective for git-> Window-> perspective-> customize perspective-> Tool Bar visibility-> Git -> Action Set Availability-> Git
7. Create a local master branch -> Right click on project-> Team-> Commit -> Provide commit comment -> Select file and then -> Commit
8. Create Remote -> Origin-> Type Url-> Add spec -> source and destination
9. Fetch the repository from Github.com-> Right click on GreenArrow pull-> configure fetch-> type url->add spec -> source and destination -> finish and fetch
10. Merge Local master branch with Remote -> Right click on Branch named master -> merge-> select remote branch -> click merge
11. Now push the code -> right click on push from remote origin/master -> push

#### 28th June

## How to get the row count of excel sheet

* Create a method and pass the sheetname as parameter
* Get the index by using the method getSheetindex on workbook object
* Using the sheet object get the sheet using the index
* Get the row count by using the lastrownum+1

#### Jenkin

1. Get MSI file
2. Install
3. Copy the path of program file jenkin
4. Java –jar Jenkins.war –httpPort-9999
5. Get the password from command prompt
6. http:/localhost:9999

###### Day 2

http://localhost:8080/jenkins/restart

under build

javac -cp src.\DemoJenkin\src\demo1\JenkinDemo1.java

C:\Users\shashi.prasad\.jenkins\workspace\DemoJob

maven

build dependency

right click

run as- maven build

goal- clean test

clean -Dtest=AppTest test

go to directory leave there is class AppTest , test it

* Jenkin job creation – linking with git then build(compile and run)
* Maven Project- Maven build with test- upload to Git-create job in jenkin – link with git – run the jenkin job build
* git config --global user.name "Shashi Prasad"
* git config --global user.email "shashi.prsd@yahoo.com"
* **Create a new repository**
* git clone git@gitlab.com:ShultzShining/ManualCommit.git
* cd ManualCommit
* touch README.md
* git add README.md
* git commit -m "add README"
* git push -u origin master
* **Existing folder**
* cd existing\_folder
* git init
* git remote add origin git@gitlab.com:ShultzShining/ManualCommit.git
* git add .
* git commit -m "Initial commit"
* git push -u origin master
* **Existing Git repository**
* cd existing\_repo
* git remote add origin git@gitlab.com:ShultzShining/ManualCommit.git
* git push -u origin --all
* git push -u origin --tags
* -cp src directory packagename.classname

### Day 3

##### Web project - Deployment

* Install Tomact
* Configure the server in eclipse
* Create a dynamic web project select the server
* Upload it to git
* Create jenkin job add git
* Build goal- clean install
* Post build- deploy war or ear to container

##### Send Mail:

* Configure system- setup smtp server
* Build setting – Set recipient , mail will be sent for failed build

<https://github.com/eswaribala>

##### Pipeline:

|  |
| --- |
| node |
|  | { |
|  | // This shows a simple build wrapper example, using the Timestamper plugin. |
|  | node { |
|  | // Adds timestamps to the output logged by steps inside the wrapper. |
|  | timestamps { |
|  | // Just some echoes to show the timestamps. |
|  | stage "First echo" |
|  | echo "Hey, look, I'm echoing with a timestamp!" |
|  |  |
|  | // A sleep to make sure we actually get a real difference! |
|  | stage "Sleeping" |
|  | sleep 30 |
|  |  |
|  | // And a final echo to show the time when we wrap up. |
|  | stage "Second echo" |
|  | echo "Wonder what time it is now?" |
|  | } |
|  | } |
|  |  |
|  |  |
|  | } |

Groovy script

* ShultzShining
* becky23

|  |
| --- |
| Import groovy.sql.sql  def url = 'jdbc:mysql://localhost:3306/nessu' |
|  | def user = 'root' |
|  | def password = 'vignesh' |
|  | def driver = 'com.mysql.jdbc.Driver' |
|  | def sql = Sql.newInstance(url, user, password, driver) |
|  |  |
|  | // use 'sql' instance ... |
|  | sql.query('SELECT \* FROM employee') { resultSet -> |
|  | while (resultSet.next()) { |
|  |  |
|  | print(resultSet.getString(1)) |
|  | println(resultSet.getString(2)) |
|  | } |
|  | } |
|  | sql.connection.autoCommit=false |
|  | def sqlstr = """INSERT INTO Project(project\_id, project\_name) VALUES (1280,'GeoSpace')""" |
|  | try |
|  | { |
|  | sql.execute(sqlstr); |
|  | sql.commit() |
|  | println("Successfully committed") |
|  | } |
|  | catch(Exception ex) |
|  | { |
|  | sql.rollback() |
|  | println("Transaction rollback") |
|  | } |
|  | sql.close() |
|  | } |

##### Security

Configure global security

Jenkins own user data base , allow user to sing up

**Cameron Philipp-Edmonds:**Tell us a little about yourself.

**Jonathan Cooper:** I’m Jon Cooper, principal quality engineer for the ExactTarget Marketing Cloud. I have worked in software testing for nearly ten years, starting out at a small software company in Indianapolis, testing telecom billing software. There, I manually tested the telecom billing software using session-based exploratory testing techniques. For the next six years, I worked with a consulting company that had clients in both Indianapolis and Cincinnati. From there, I joined the ExactTarget Marketing Cloud team and recently celebrated my two-year anniversary.

**Cameron:**Can you tell us about your role at ExactTarget Marketing Cloud?

**Jon:**As part of the platform team, my primary focus is testing our core platform APIs and our mobile products. Most of my testing revolves around REST APIs. Additionally, I train team members on why and how to test APIs. For our mobile products, I work with SMS and push message emulators that simulate messages moving back and forth from SMS and push aggregators.

**Cameron:**In your words, what is API testing?

**Jon:**API stands for application programming interface, and defines how software components interact with each other. In other words, an API is a way for a developer to expose functionality for use internally or externally. Just as traditional testers test how an operator would use the front end of their product, an API tester can test how internal and external users use the APIs. The concepts are very similar to non-API testing; the only difference is that in most cases you are manipulating the API outside of a user interface (UI) using different tools.

**Cameron:**Do you think API testing is more a white box testing task or a black box testing task?

**Jon:**The vast majority of the testing I do can be done without having any expressed knowledge of what the code behind the API is doing. The same things that a tester would do in the UI, you can do with the parameters of an API. Now, it can be very useful to have knowledge of the code and data structures behind the scenes because it opens up lots of different testing ideas and allows you to shift your focus to areas where you can see that there might be vulnerabilities that could be exposed with certain testing techniques. But at most, my testing could be considered “Gray Box”, almost never White Box.

**Cameron:**You specialize in API testing with an emphasis on both SOAP and REST. In layman's terms, can you tell us the difference between the two and why someone would choose one over the other?

**Jon:**Simple object access protocol (SOAP) and representational state transfer (REST) are two formats for implementing web services. Soap APIs have customer actions defined by developers, where REST APIs use only HTTP actions and define the state of an object using payload.

Typically users of REST routes are those who depend on their services being very quick to run and quick to build, including a significant amount of web apps. Newer companies seem to prefer REST due to its ease of implementation, and versatility.

**Cameron:**You also do mobile testing. Can you tell us what kind of testing for mobile you work on?

**Jon:**I assist with the testing of our two mobile products, MobileConnect, our SMS product, and MobilePush, our push product. MobileConnect allows our customers to interact with their mobile subscribers via customized SMS campaigns. For instance, at the movies when you see a sign in the lobby that says “Text POPCORN to 123456 to get concessions discounts,” MobileConnect is a tool that would allow brands to set up and monitor that marketing campaign.

The testing I do is around our mobile APIs and also around the mobile product itself, including the sending and receiving of SMS messages. We also test to ensure these interactions work across all of the aggregators we partner with both here in the US and internationally. Finally, we test to ensure message sizes and encoding work internationally.

Our push product enables brands to interact with their mobile subscribers via mobile apps on their phones and tablets. MobilePush enables a brand to “push” a message to a subscriber’s phone from an installed app. For example, if on your phone you get a notification saying that you have a bill ready to be paid from your banking app, it may have been triggered using one of our APIs. We also handle geo-targeted notifications. For example, if you are walking past a restaurant, and happen to have a dining coupons app, our APIs might trigger a message from that app giving you a coupon code for 10% off an appetizer.

**Cameron:**What is the difference between SMS and push notifications, and how does testing of those mobile traits differ?

**Jon:**Both SMS and mobile app messaging are critical communications for almost any organization— [**mobile stats and consumer behavior shows it**](http://www.exacttarget.com/blog/7-mobile-usage-stats-you-didnt-know-yesterday/). SMS is a text message and push notifications are messages delivered in-app or sent to the home scree of customer’s mobile devices. Each requires unique testing with variables on country, mobile app, size and encoding.

**Cameron:**You describe yourself as an adherent of the context-driven methodology of software testing. What about the method do you find so appealing?

**Jon:**As a consultant for several years, I’ve worked with interesting products in unique places. There was not a single instance where I would use the same testing procedure verbatim. The whole idea behind the context-driven methodology is that there is no single testing best practice that fits all situations, and that is up to the entire team to determine what the best way to test. This is what I have found to be true during my testing career.

It really is all about people taking a purposeful look at what they are trying to build and working together to define the “best practices” in their own unique context. However, that does not mean that for every testing project everything must be built from scratch. Tools, ideas, and processes can and should be reused where prudent.

**Cameron:** You've been a testing professional for roughly a decade, and testing seems to have changed a lot in the last ten years. How do you think the world of testing, especially relating to mobile, will change in the next ten years?

**Jon:**In the last ten years, I have seen a considerable shift to more and more automation, simply because the products we test and environments we operate in have become so complex that manual regression testing is impractical. In the next ten years we are likely going to continue to see more movement toward automation, especially in the mobile space. This does not however mean that there will be fewer testers.

In the last few years, I have seen more companies discover that automated tools are only a part (a SMALL part) of good software testing, and that on site testing teams actually save money and provide better ROI.

All in all, I believe we will see the number of testers grow to support the ever-increasing amounts of software that we interact with every day. Average testers ten years from now will be more knowledgeable about the technologies that drive the products they work with, and they will be working much more closely with development teams to build better and more stable products quickly.

**Cameron:**Knowing what you know now, what advice would you give to yourself when you were just starting out?

**Jon:** I would get more involved with the development and test communities locally and online. There are lots of great people and organizations out there to interact with professionally that can assist with professional growth and provide an environment for mentorship. Early on in my testing career I got to know some of the people in the AST, and the advice and techniques I learned from them are invaluable to my work even today. The software testing community is a friendly, professional, and resourceful community, so get involved!

## 16th August

Roles and responsibilities as a manual Tester:

Roles and responsibilities as an Automation Tester:

## 28th November

## Jmeter

#### SMTP Transfer

1. Download java mail jar
2. Place in lib folder
3. Add details to smtp transfer
4. Get server details

Software installed- Badboy, Filezilla, Mysql

#### Workbench

#### Ui Tools in Jemter

<https://openweathermap.org/api>

# 7Th Decemeber

Selenium Group Link

<https://groups.google.com/forum/?fromgroups#!forum/webdriver>

System.setproperty("webdriver.gecko.driver","C:/samplepath")

Webdriver driver = new firefoxdriver();

driver.get("http://example.com/reg.html");

//positive tests

driver.findElement(By.id("id-username")).sendKeys("Cd123456");

driver.findElement(By.id("id-password1")).sendKeys("Cd123456");

driver.findElement(By.id("id-password2")).click();

Assert.assertTrue(driver.getPageSource().containsIgnoreCase("successful"));

driver.findElement(By.id("id-username")).sendKeys("Cd123456");

driver.findElement(By.id("id-password1")).sendKeys("Cd123456");

driver.findElement(By.id("id-password2")).click();

Assert.assertTrue(driver.getPageSource().containsIgnoreCase("successful"));

//negative tests

driver.findElement(By.id("id-username")).sendKeys("C23456");

driver.findElement(By.id("id-password1")).sendKeys("Cd123456");

driver.findElement(By.id("id-password2")).click();

Assert.assertTrue(driver.getPageSource().containsIgnoreCase("username too short"));

driver.findElement(By.id("id-username")).sendKeys("C1223456");

driver.findElement(By.id("id-password1")).sendKeys("3456");

driver.findElement(By.id("id-password2")).click();

Assert.assertTrue(driver.getPageSource().containsIgnoreCase("too short"));

driver.findElement(By.id("id-username")).sendKeys("11223456");

driver.findElement(By.id("id-password1")).sendKeys("34563456");

driver.findElement(By.id("id-password2")).click();

Assert.assertTrue(driver.getPageSource().containsIgnoreCase("username not correct"));

# *15th December*

Website to refer to for interview preparation

* LeetCode
* HackerRank
* GeeksForGeeks
* Programming- InterviewBit
* CareerCup
* Glassdoor
* Coursera
* Edx
* Udemey
* Mock Interview Sites
* AmbitionBox

# *20th December*

## Mandatory Skills:

### JAVA:

* OOPs Concepts, Collections
* Design Patterns - PageFactory, Singleton, Abstract factory
* Maven in automation solution (Like: Setup, configure, Dependencies, Plugins, parameters, transitive dependencies and integration with Jenkins etc.)

### Automation:

* Automation of APIs (REST, XML & SOAP)
* Automation using Selenium Java BDD implementation using Cucumber or JBehave (Specify JBehave or Cucumber)
* Experience in using Selenium Grid
* Automation Testing across Platforms: Cross browser testing coverage
* Testing management across environments
* Develop test tools, test beds, data mock ups, and other test infrastructure to continuously measure quality through Continuous testing and integration (CI)

### Testing:

* Design End to End Test Strategy Requirement elaborations, Feature and requirements quality Testing and Design End to End Test Scenario
* Defect Management and RCA

### CI/CD:

* CI Integration experience using Jenkins/Bamboo
* Code Review and Analysis Experience in using GIT/SVN (Specify the tool)
* Develop test tools, test beds, data mock ups, and other test infrastructure to continuously measure quality through Continuous testing and integration (CI)

# Selenium WebDriver With Java - Novice To Ninja + Interview - Udemey

* Selenium Webdriver 3.x
* Java Concepts in details
* TestNG Framework
* Advanced Reporting
* Logging infrastructure with Log4j
* Page Object Model
* Page Factory Framework
* Data Driven Framework
* Executing tests on a remote machine using Selenium Grid 2.0
* Build Management with Maven
* Continuous Integration with Jenkins
* Database Testing
* Performance Testing
* Behavior Driven Testing Using Cucumber and Gherkin language

Curriculum For This Course

Expand All

271 Lectures

40:37:15

–

## Selenium Introduction

37:15

[Course Outcome - \*\*\* MUST WATCH \*\*\*](javascript:void(0))

[Preview](javascript:void(0)) 05:03

What made this a Top Course and Top Responder - \*\*\* MUST WATCH \*\*\*

08:07

[Instructor Introduction](javascript:void(0))

[Preview](javascript:void(0)) 01:27

[How to reach me anytime and ask questions? \*\*\* MUST WATCH \*\*\*](javascript:void(0))

[Preview](javascript:void(0)) 05:14

[Why Selenium?](javascript:void(0))

[Preview](javascript:void(0)) 09:24

[Selenium WebDriver Architecture - How WebDriver Works?](javascript:void(0))

[Preview](javascript:void(0)) 08:00

How to fix blurry videos

1 page

–

Setup and installation of required tools and plugins

28:50

## Overview and Install Java

05:09

Java Environment Setup - Mac

06:36

Java Environment Setup - Windows

04:05

Install Eclipse - Windows

05:16

Eclipse Installation - Mac

04:35

Install Maven Plugin

03:09

–

Java Concepts - Data Types

01:29:31

First Program and Some Tips

14:48

Variables And Data Types

18:19

Default Variable Values

08:57

Reference Data Type - Strings Example

11:32

String Methods - Part 1

11:19

String Methods - Part 2

08:53

Strings Quiz \*\*\* Test Your Knowledge \*\*\*

6 questions

String Vs StringBuffer Vs StringBuilder \*\*\* Interview Question \*\*\*

1 page

Arrays

15:43

–

Classes And Methods -> Object Oriented Concepts

01:16:59

Methods Introduction

09:20

Methods - Practical Example

10:52

Understanding Return Type

10:19

Class Introduction

11:00

Getters-Setters And This Keyword - Part 1

11:02

Getters-Setters And This Keyword - Part 2

09:28

Getters-Setters And This Keyword - Part 3

03:52

Constructors

11:06

–

Conditional Statements and Loops

44:12

Conditional Statement

10:07

Switch Statement

09:34

While Loop

14:43

For Loops

09:48

–

Static Keyword

24:31

Static Keyword - Part 1

14:39

Static Keyword - Part 2

09:52

–

Practice Exercise With Solution + \*\*\* Interview Question \*\*\*

22:57

Practice Exercise With Solution + \*\*\* Interview Question \*\*\*

11:08

Practice Exercise + Interview Question -> Reverse Characters Of A String

02:31

Practice Exercise + Interview Solution -> Reverse Characters Of A String

09:18

–

Java Concepts - Object Oriented Programming Concepts

02:08:22

[Inheritance](javascript:void(0))

[Preview](javascript:void(0)) 15:23

[Access Modifiers and Packages](javascript:void(0))

[Preview](javascript:void(0)) 17:24

Abstract Class - Abstraction Part 1

15:51

Interface - Abstraction Part 2

12:57

Interface vs Abstract Class \*\*\* Interview Question \*\*\*

1 page

Method Overloading

14:24

More On Method Overloading

04:16

Method Overriding

12:34

Overloading vs Overriding \*\*\* Interview Question \*\*\*

1 page

Exceptions - Checked

17:21

Exceptions - RunTime

10:02

Reading Properties File

08:10

–

How To Inspect Elements On Different Browsers - Add-Ons

01:01:54

Introduction

04:03

How To Inspect Elements Using Firefox DevTools

12:41

Firefox Add-On - Try XPath

06:58

How To Inspect Elements Using Chrome DevTools

10:07

Chrome Extension - Part 1

07:15

Chrome Extension - Part 2

04:43

Tricks To Generate XPath

10:10

FirePath Fans -> If you want to use FirePath

05:57

–

Selenium WebDriver - Setup

23:01

Setup Selenium 2.x Environment -> If you are using Selenium WebDriver 2 Version

08:45

Selenium WebDriver Working Versions

00:11

Setup Selenium 3.x Environment

05:59

Selenium 3.x Update

08:06

-

Selenium WebDriver -> Running Tests On Various Browsers

55:50

[Running Tests On Firefox with Selenium 2.x](javascript:void(0))

[Preview](javascript:void(0)) 04:46

Running Tests On Firefox With Selenium 3.x

07:50

Desired Capabilities Introduction

07:21

[Running Tests On Google Chrome](javascript:void(0))

[Preview](javascript:void(0)) 09:36

Running Tests On Internet Explorer

10:25

Resolving IE Related Issues

04:46

Requirements To Run Tests On Safari

00:39

Running Tests On Safari

07:21

\*\*\* Interview Questions \*\*\*

03:06

-

Multiple Ways To Locate Elements

35:50

[Find Elements Using ID And XPATH](javascript:void(0))

[Preview](javascript:void(0)) 08:24

[Find Elements Using Name](javascript:void(0))

[Preview](javascript:void(0)) 04:42

[Find Elements Using LinkText And PartialLinkText](javascript:void(0))

[Preview](javascript:void(0)) 09:33

[Find Elements Using ClassName](javascript:void(0))

[Preview](javascript:void(0)) 06:39

Find Elements Using TagName And \*\*\* Interview Question \*\*\*

06:32

-

CSS Selectors - Advanced Locators

40:49

Using Ids With CSS Selectors To Find Elements

09:55

Using Multiple CSS Classes To Find Elements

11:27

Using Wildcards With CSS Selectors

11:28

Finding Child Nodes Using CSS Selectors

07:09

CSS Cheat Sheet

00:50

-

Xpath - Advanced Locators

01:00:22

Difference Between Absolute And Relative Xpath

12:27

How to Build An Effective Xpath

05:31

Using Text To Build An Effective Xpath

07:45

Build Xpath Using Contains Keyword

06:24

Build Xpath Using Starts-With Keyword

08:01

How To Find Parent and Sibling Nodes

07:38

Exercise With Solution + \*\*\* Interview Question \*\*\*

11:40

Xpath Cheat Sheet

00:56

-

Selenium IDE / WebDriver -> Basics

31:02

Selenium IDE - Important Update

00:15

[Install Selenium IDE And First Script](javascript:void(0))

[Preview](javascript:void(0)) 09:01

Generating WebDriver Code Using Selenium IDE

15:09

Assert vs Verify - Selenium IDE Commands

06:37

-

Selenium WebDriver -> Working With Web Elements

01:59:57

JUnit Introduction

09:08

[How To Click And Type On A Web Element](javascript:void(0))

[Preview](javascript:void(0)) 14:27

How To Navigate Between Web Pages

16:30

How To Find The State Of A Web Element (Disabled And Enabled Elements)

10:22

Radio Buttons And CheckBoxes

14:05

Working With List Of Elements

12:09

Understanding Dropdown Elements

05:09

Working With A Dropdown Element \*\*\* Practical Example \*\*\*

09:37

Understanding Multiple Select Elements

03:41

Working With Multiple Select \*\*\* Practical Example \*\*\*

08:44

How To Work With Hidden Elements

05:42

Working With Hidden Elements \*\*\* Practical Example \*\*\*

10:23

-

Selenium WebDriver -> Useful Methods And Properties

46:51

How To Get The Text On Element

06:27

How To Get Value Of Element Attribute

06:10

Generic Method To Find Elements

12:33

Generic Method To Find Element List

06:26

How To Check If Element Is Present

09:06

Generic Methods -> Code Refactoring

06:09

-

Selenium WebDriver -> Wait Types - Handling Synchronization Issues

50:56

Implicit Wait Vs Explicit Wait

10:59

Implicit Wait \*\*\* Practical Example \*\*\*

06:30

Explicit Wait \*\*\* Practical Example \*\*\*

08:39

Generic Method To Work With Explicit Wait - Part 1

11:04

Generic Method To Work With Explicit Wait - Part 2

07:49

\*\*\* Interview Questions \*\*\*

05:55

-

Selenium WebDriver -> Advanced

01:04:34

Calendar Selection - working with date pickers

08:47

Calendar Selection \*\*\* Practical Example \*\*\*

04:08

Calendar Selection \*\*\* Real Time Example \*\*\*

06:48

Autocomplete

09:35

Executing JavaScript Commands

08:40

How To Find Size Of The Window

05:31

How To Scroll Element Into View

08:51

Capturing screenshots of a web page

12:14

-

Selenium WebDriver -> Switch Window And IFrames

48:28

How To Switch Window Focus

11:04

Switch To Window \*\*\* Practical Example \*\*\*

08:28

How To Work With IFrames

11:00

Switch To IFrame \*\*\* Practical Example \*\*\*

08:04

Handling JavaScript Popup

09:52

Switch Focus Quiz + \*\*\* Interview Questions \*\*\*

6 questions

-

Selenium WebDriver -> Working With Actions Class

23:55

Mouse Hover Actions

09:34

[How To Drag And Drop Element On A Web Page](javascript:void(0))

[Preview](javascript:void(0)) 08:01

Working With Sliders Actions

06:20

Actions Class Quiz + \*\*\* Interview Questions \*\*\*

3 questions

-

Selenium WebDriver - Key Press Events

30:43

Introduction

04:42

How To Press Keyboard Keys

10:13

How To Send A Key Combination

08:44

KeyPress Events Using Actions Class

07:04

-

Automation Framework

01:11:25

[Automation Framework Introduction](javascript:void(0))

[Preview](javascript:void(0)) 14:38

Page Object Model

15:10

Creating Object Repository and Page Class

10:21

Page Factory

16:18

Find all links on a web page - BONUS

14:58

-

Logging Infrastructure - Using Log4j2

55:12

Log4j2 Introduction

10:56

Using Default Configuration

09:56

Using Configuration File - Console Logging

10:27

Using Configuration File - File Logging

10:52

Working With Custom Loggers

06:42

Test Case Example Using Log4j2

06:19

-

TestNG Setup

21:20

TestNG Introduction

05:51

TestNG Setup

09:00

How To Add TestNG JavaDoc

06:29

-

TestNG -> Annotations And Asserts

01:10:07

First TestNG Class - Test Annotation

09:54

TestNG Asserts

14:10

TestNG Soft Asserts

08:17

Method And Class Annotations

08:41

Running A Test Suite

12:03

BeforeSuite And AfterSuite Annotations

08:01

BeforeTest And AfterTest Annotations

09:01

-

TestNG -> Advanced Features

42:38

Prioritizing Test Methods In A Desired Sequence

05:11

How To Group Test Methods - Part 1

09:02

How To Group Test Methods - Part 2

04:52

Understanding Dependent Test Methods

10:49

How To Disable And Timeout Test Methods

05:55

How To Preserve Order Of Execution

06:49

-

TestNG -> Parameters And Parallel

21:54

How To Provide Parameters To Test Methods

12:27

How To Run Tests In Parallel

09:27

-

TestNG -> Parameters And Parallel - Practice Exercise

14:23

TestNG Pracitce Exercise Question

03:04

Practice Exercise Solution - Selenium WebDriver Practical Example

11:19

-

TestNG -> DataProviders

13:24

DataProviders - Part 1

08:35

DataProviders - Part 2

04:49

-

TestNG -> ITestResult

08:35

ITestResult Interface

08:35

-

TestNG -> Listeners

43:40

IInvokedMethodListener

14:10

ITestListener

16:20

ISuiteListener

06:33

TestNG Listeners - Code Refactoring

06:37

-

TestNG -> Reporter Logs and HTML Reports

12:35

TestNG Reporters And Logs

12:35

-

Extent Reports -> Advanced Reporting

59:42

Advanced Reporting Introduction And Features

09:56

Extent Reports JARs Download Location

00:09

Advanced Reporting - Practical Example

12:17

Attaching Screenshots To Advanced Reports

09:05

Multiple Test Cases \*\*\* Practical Example \*\*\*

09:23

Reporting With Page Object Model

11:18

Exercise With Solution \*\*\* Homework \*\*\*

07:34

-

Selenium WebDriver -> Data Driven Testing

49:55

Data Driven Testing Setup

07:21

Reading Data From Excel File

10:31

Reading Multiple Data Sets From Excel File

08:40

Multiple Data Sets Test Case \*\*\* Practical Example \*\*\*

14:38

Writing Data To Excel File

08:45

-

Working With Firefox Profiles

23:52

Firefox Profiles Introduction

05:47

Using Firefox Profile In Web Automation Code

07:24

Chrome Options - Bring Up Chrome With Extensions

10:41

-

Selenium WebDriver -> File Upload And Windows Authentication

49:07

File Upload Windows - Robot

10:47

File Upload Mac - Robot

10:36

AutoIT Installation And Finder Tool

06:32

File Upload Windows - AutoIT

11:33

Handling Windows Authentication Using AutoIT

09:39

-

WebDriver Event Listener

15:54

WebDriver Event Listener - Part 1

08:29

WebDriver Event Listener - Part 2

07:25

-

Cross-Browser Testing Using Selenium Grid 2.0

01:25:29

[Selenium Grid Introduction](javascript:void(0))

[Preview](javascript:void(0)) 07:52

Hub And Nodes Configuration - Part 1

14:54

Hub And Nodes Configuration - Part 2

11:47

Grid Configuration Using JSON File - Part 1

12:45

Grid Configuration Using JSON File - Part 2

05:37

Running Test Case On Selenium Grid Environment

14:46

Real Time Practical Example Of Selenium Grid

17:48

-

Build Management With Maven

01:54:07

[Maven Features and Advantages](javascript:void(0))

[Preview](javascript:void(0)) 08:42

[Setup and Installation (MAC and Windows)](javascript:void(0))

[Preview](javascript:void(0)) 14:57

Creating And Importing Maven Project

12:33

Maven Local And Central Repositories

09:44

POM Explanation - Core Of Maven

19:10

Build Lifecycle And Maven Commands

16:38

Maven Naming Convention

00:22

Maven Project Migration

09:32

Maven TestNG Integration And Maven Profiles

16:08

Where To Find TestNG Reports In Maven?

06:21

-

Continuous Integration With Jenkins

01:10:47

Jenkins Features And Advantages

08:50

Jenkins Setup And Installation

05:49

Jenkins Configuration

08:04

Securing Jenkins

06:27

Jenkins Plugins Management

04:41

Jenkins Git Integration

03:10

Building A Maven Project Using Jenkins

06:58

Building A Freestyle Project Using Jenkins

06:30

Building A Remote Project Using Git And Jenkins

07:47

Jenkins Scheduling Builds On A Git Repository

12:31

-

Database Testing

29:20

Testing MySql And Oracle

14:34

Testing MongoDB

14:46

-

Performance Testing

24:16

Performance Testing Going The System Way

14:46

Performance Testing By Making A StopWatch

09:30

-

Behavior Driven Development Using Cucumber

01:29:15

Features And Advantages

09:10

An Example Of Cucumber Template

07:18

Cucumber Eclipse Plugin Installation

02:51

Cucumber JAR Files - Important Update

00:14

Downloading Cucumber Jars

08:00

Setting Up Cucumber Project

04:44

Selenium Login Test

07:22

Understanding The Feature File

08:58

Gherkin Keywords Explanation

12:03

Running The Tests Using Feature File

10:23

Understanding Cucumber Options

06:24

Converting Selenium Test To Cucumber With Step Definition

11:48

-

Sauce Labs Integration

18:07

Sauce Labs Introduction

06:32

How To Get Access Key

04:31

Sauce Labs Practical Example

07:04

-

Java Advanced - Collections Framework

01:02:26

Array List

12:38

Linked List

07:41

Array List vs Linked List \*\*\* Interview Question \*\*\*

10:29

Sets

11:28

Maps

09:29

Different Kind Of Maps \*\*\* Interview Question \*\*\*

10:41

-

Conclusion

02:26

BONUS: Other Cool Stuff

02:20

Java Code Files

00:03

Selenium WebDriver Automation Code Files

00:03

-

Logging Infrastructure - Using Log4j - Obsolete (Replaced by new lectures)

01:00:07

Log4j Introduction

15:34

Log4j Console Appenders

09:58

Configuring Properties File

15:48

Test Case Using Logging Infrastructure

11:21

Appenders Initialization Issues Of Log4j

07:26

-

TestNG Framework - Obsolete (Replaced by new lectures)

02:36:46

Downloading TestNG JAR Files

00:23

TestNG Introduction

14:24

Running Test Suite

12:58

Grouping Tests

10:07

Dependent Tests

09:39

Prioritizing Tests

08:37

Reporters And Asserts

18:55

Parameters And DataProviders

18:41

[Multiple Browsers And Parallel Tests](javascript:void(0))

[Preview](javascript:void(0)) 12:10

Listeners

19:53

MultiThreading And Parallel Tests

14:45

Taking Screenshot On Test Failure - Part 1

10:23

Taking Screenshot On Test Failure - Part 2

05:51

# Selenium Webdriver Complete Course - Build A Framework- Udemey C# Nicolai

Curriculum For This Course

Expand All

541 Lectures

31:02:21

–

Getting Started

43:39

[Course Overview](javascript:void(0))

[Preview](javascript:void(0)) 02:26

[What software is required for this course?](javascript:void(0))

[Preview](javascript:void(0)) 01:49

All course resources

00:06

[Syllabus for C# section](javascript:void(0))

[Preview](javascript:void(0)) 03:41

[Syllabus for selenium section](javascript:void(0))

[Preview](javascript:void(0)) 02:52

[HELP I have an error!](javascript:void(0))

[Preview](javascript:void(0)) 07:58

[What information should be supplied in questions](javascript:void(0))

[Preview](javascript:void(0)) 01:33

[Sample question](javascript:void(0))

[Preview](javascript:void(0)) 06:05

[Quizzes](javascript:void(0))

[Preview](javascript:void(0)) 01:59

[Frequently Asked Questions](javascript:void(0))

[Preview](javascript:void(0)) 00:49

Your Extra Bonus

00:16

[Visual Studio options for Mac](javascript:void(0))

[Preview](javascript:void(0)) 10:05

[How to use Parallels for Mac](javascript:void(0))

[Preview](javascript:void(0)) 02:39

Clarification

01:11

What do you struggle with in test automation?

00:07

[Quiz](javascript:void(0))

[Preview](javascript:void(0)) 00:01

–

Introduction To C#

41:14

Instructor Intro

01:00

Concepts Covered

02:44

[How to download Visual Studio Community](javascript:void(0))

[Preview](javascript:void(0)) 04:23

[Getting to know VS](javascript:void(0))

[Preview](javascript:void(0)) 03:37

Creating a Console Project

04:28

Understanding solutions

05:02

Creating a Class Library

04:23

Creating a Unit Test Project

02:32

C# Coding Standards

03:48

Coding Convention Example pt1

04:24

Coding Convention Example pt2

04:50

[Visual Studio and Coding Conventions Quiz](javascript:void(0))

[Preview](javascript:void(0)) 00:03

–

Variables and Strings

30:22

Code Solutions for Variables and Strings

00:03

[Intro to Variables](javascript:void(0))

[Preview](javascript:void(0)) 04:16

[Declaring Variables of Simple Types](javascript:void(0))

[Preview](javascript:void(0)) 04:50

Formatting text using concatenation

05:02

Formatting text for Output

04:29

Using Numeric Format Strings

02:07

Using StringFormat Method

03:08

[Verbatim Strings and Escape Sequences](javascript:void(0))

[Preview](javascript:void(0)) 03:13

Lesson 3 Exercise

03:11

Variables and strings assessment

00:03

–

Different types of VS Projects

22:10

[Code solutions for Different types of VS Projects](javascript:void(0))

[Preview](javascript:void(0)) 00:01

Running a Console App

03:03

Creating a Unit Test

04:42

Using Assertions in Unit Tests

03:35

Basic Debug Commands

04:12

The Watch Window

02:25

Other Debugging Functions

04:10

Quiz

00:01

–

Math,equality operators, If statement

21:18

[Code Solutions for Math,equality operators, If statement](javascript:void(0))

[Preview](javascript:void(0)) 00:03

Math with Integers

04:14

Different Floating Point Data Types

03:02

Math with Decimals

04:25

Equality and Relational Operators

03:01

If Statements

03:35

If Else Statements

02:57

[Quiz](javascript:void(0))

[Preview](javascript:void(0)) 00:01

–

C# Classes and Methods

48:54

[Code Solutions for Classes](javascript:void(0))

[Preview](javascript:void(0)) 00:03

[Classes and Objects](javascript:void(0))

[Preview](javascript:void(0)) 03:18

[Writing a Class](javascript:void(0))

[Preview](javascript:void(0)) 03:55

[Get and Set Accessors](javascript:void(0))

[Preview](javascript:void(0)) 03:48

Auto-implemented properties

02:25

Constructors

01:45

Instatiating a class object

04:35

Methods in Classes

03:48

Value vs Reference Types

02:57

Scope of Declarations

04:25

[Class demo pt1](javascript:void(0))

[Preview](javascript:void(0)) 03:21

[Class demo pt 2](javascript:void(0))

[Preview](javascript:void(0)) 02:59

[Class demo pt3](javascript:void(0))

[Preview](javascript:void(0)) 03:43

[Class demo pt4](javascript:void(0))

[Preview](javascript:void(0)) 04:09

Correction

00:03

[Class demo pt5](javascript:void(0))

[Preview](javascript:void(0)) 03:39

Quiz

00:01

–

Loops and Switch

33:37

[Code Solutions for Loops and Switch](javascript:void(0))

[Preview](javascript:void(0)) 00:03

While Repetition

02:14

Counter Controlled Repetition

02:24

Sentinel controlled repetition

03:04

Nested Control Statements

03:30

Compound Assignment Operators

01:27

Increment-Decrement Operators

02:35

For Repetition

04:13

DoWhile Loop

02:39

Switch Statements

04:01

Break and Continue

03:14

Logical Operators

04:13

–

Methods

17:50

[Code Solutions for Methods](javascript:void(0))

[Preview](javascript:void(0)) 00:03

The Method Call Stack

02:47

Static Methods

04:11

Why Static Members

05:00

Method Overloading

02:44

Optional Parameters

03:05

–

Arrays and Lists

19:21

[Code Solutions for arrays and lists](javascript:void(0))

[Preview](javascript:void(0)) 00:03

Declaring and Creating Arrays

04:04

Foreach Repetition

03:28

Passing Arrays to Methods

02:57

Passing array element to method

02:29

Arrays vs Collections

02:53

List Collection

03:27

–

Introduction to Object Oriented Programming

01:39:16

[Code Solutions for OOP Concepts](javascript:void(0))

[Preview](javascript:void(0)) 00:01

[Intro to Polymorphism](javascript:void(0))

[Preview](javascript:void(0)) 02:20

Composition

05:05

[Programming with Inheritance](javascript:void(0))

[Preview](javascript:void(0)) 03:45

Coding Inheritance pt1

03:06

Coding Inheritance pt2

02:58

Coding Inheritance pt3

03:37

Coding Inheritance pt4

03:43

Coding Inheritance pt 5

04:04

Protected Keyword

02:27

Abstract Classes

03:28

[Abstract Methods](javascript:void(0))

[Preview](javascript:void(0)) 03:58

Virtual Keyword

03:18

Sealed classes and methods

02:52

Intro to Interfaces

03:34

Creating an Interface

04:00

Implementing Interfaces

03:36

Using polymorphism with interfaces

03:14

Testing Interfaces

04:22

[Interface Code Comparison pt 1](javascript:void(0))

[Preview](javascript:void(0)) 04:53

[Interface Code Comparison pt 2](javascript:void(0))

[Preview](javascript:void(0)) 04:59

Common Interfaces in FCL pt1

03:31

Common Interfaces in FCL pt2

03:54

Carbon Footprint Exercise pt1

05:01

Carbon Footprint Exercise pt2

04:55

Carbon Footprint Exercise pt3

04:36

Carbon Footprint Exercise pt4

03:58

OOP Exam

00:01

-

Exception Handling

13:57

Code Solutions for Exception Handling

00:03

Exception Handling

02:44

Try Catch Block

04:01

Catch Block

01:55

Finally Block

02:55

Throw Keyword

02:19

-

Understanding The Automation Framework

01:10:25

Important notes about the future code

00:46

What is automated testing

03:57

Framework classes overview

04:34

AutomationTestingPracticePage Class

04:42

Pages Class

04:34

Pages Class 2

04:29

Pages Class 3

04:49

Browser class fix

00:24

Browser Class

04:36

Browser Class 2

04:47

TestBase Class

04:28

TestBase Class 2

04:43

AutomationTestingPageOpens Class

04:49

AutomationTestingPageOpens Class 2

04:38

AutomationTestingPageOpens Class 3

04:37

[Running the test code](javascript:void(0))

[Preview](javascript:void(0)) 04:38

[Running the test code 2](javascript:void(0))

[Preview](javascript:void(0)) 04:54

-

What is automation testing?

01:23:48

[introduction to automation testing](javascript:void(0))

[Preview](javascript:void(0)) 03:03

[next topics](javascript:void(0))

[Preview](javascript:void(0)) 01:26

[What is scrum](javascript:void(0))

[Preview](javascript:void(0)) 05:02

Why learn scrum

04:40

Why learn scrum 2

02:14

What is the Product Backlog in Scrum

03:48

How to provide valid estimates in Scrum

03:21

What is a Sprint Planning in Scrum

05:00

What is a Sprint Planning in Scrum 2

03:30

What is a Sprint Review and Sprint Retrospective

05:32

Summarizing everything that we learned about Scrum

05:36

Introduction to the Sprint of an Automation Tester

03:27

Starting the sprint as an automation tester

04:49

What is a regression suite

05:26

Finishing the first sprint

02:33

Finishing the second sprint as an automation engineer

05:13

What is the purpose of testing

03:46

Designing our first test case

05:48

Designing more test cases

04:42

Understanding why manual testing can be a burden

04:48

Was this section useful?

00:04

-

Benefits of Automation Testing

21:34

Automation Testing saves time

05:16

Automated tests are faster

01:48

Automated tests are repeteable

03:24

Automated tests are reusable

02:50

Automated tests reduce costs

01:19

Automated tests are powerful and versatile

04:06

Summary of this section

02:51

-

What is Selenium Webdriver?

13:16

Introduction to selenium webdriver

00:51

What are the benefits of using Selenium Webdriver

04:53

What are the negatives of using Selenium Webdriver

03:03

How does Selenium help us get the perfect framework

02:33

Conclusions regrading this section

01:56

-

Installing the right tools

36:49

What is the software that you will need to work with Selenium WebDriver

03:55

ChromeDriver Version!!!

00:23

Resource pages

01:17

How to download and install Selenium WebDriver

05:04

Critical info about FirefoxDriver

00:28

How to download ChromeDriver

03:01

How to get ChromeDriver path dynamically

06:12

How to download Github projects and open them

04:16

503:Service Temporarily Unavailable on QTPTutorial.net

00:15

[Installing the right tools](javascript:void(0))

[Preview](javascript:void(0)) 03:06

[making sure that Firefox won't auto update](javascript:void(0))

[Preview](javascript:void(0)) 01:08

downloading and opening Selenium IDE

02:11

downloading and opening Firebug Plugin

02:11

downloading and opening firepath Plugin

03:19

Quiz

00:01

-

Advanced guide to locating web elements with Webdriver

02:29:31

Introduction to Element Identification

02:53

Prerequisites and Resources

02:11

Resources for Element Identification

00:10

Critical Updates To Course

03:45

Updates to the Framework Project

02:35

Element Interactions Set Up Quiz

00:01

[introduction to html tags](javascript:void(0))

[Preview](javascript:void(0)) 04:42

[introduction to html attributes](javascript:void(0))

[Preview](javascript:void(0)) 04:28

[using selenium webdriver for element identification](javascript:void(0))

[Preview](javascript:void(0)) 04:23

[selenium ide for object identification](javascript:void(0))

[Preview](javascript:void(0)) 04:20

[using an id to identify elements](javascript:void(0))

[Preview](javascript:void(0)) 02:58

[using an id to identify elements with selenium webdriver](javascript:void(0))

[Preview](javascript:void(0)) 03:35

[understanding the webdriver commands](javascript:void(0))

[Preview](javascript:void(0)) 03:13

[how to use className to identify an element](javascript:void(0))

[Preview](javascript:void(0)) 03:45

[how to use Name property to identify an element](javascript:void(0))

[Preview](javascript:void(0)) 03:59

[how to use link text to identify an element in selenium](javascript:void(0))

[Preview](javascript:void(0)) 04:06

[how to use partial link text to identify an element in selenium](javascript:void(0))

[Preview](javascript:void(0)) 03:48

how to use css to identify an element in selenium

05:17

how to use absolute xpath to identify an element in selenium

05:55

how to use relative xpath to identify an element in selenium

02:43

how to use pick any tag from html body

05:27

how to use attributes in xpath

04:06

how to use special operators in xpath

03:55

9-19-how to identify a button using xpath

03:16

how to figure out if an element contains some text using xpath

02:39

how to identify link and icon using xpath

02:50

how to identify fields using xpath

02:05

how to identify radio buttons using xpath

03:47

how to identify radio buttons using index

03:18

how to use lists using xpath

04:02

working with tabs and toggles

04:16

introduction to html tables

04:34

how to get an html table in a variable

04:29

how to use c# to find a cell value

03:19

how to use c# to find all cells in a row

03:47

how to use c# to find a value in a cell

05:00

how to use c# to find a value in a cell 2

04:48

how to use c# to find a value in a cell 3

02:16

how to find a table with no id

03:28

dom in selenium webdriver

01:36

How to evaluate elements at run time

07:37

[Object Identification Quiz](javascript:void(0))

[Preview](javascript:void(0)) 00:08

-

Navigation, Manipulation, Interrogation with Selenium Webdriver

57:36

[Navigation with Selenium WebDriver](javascript:void(0))

[Preview](javascript:void(0)) 02:02

[Navigation quiz](javascript:void(0))

[Preview](javascript:void(0)) 02:49

[Navigation quiz answer](javascript:void(0))

[Preview](javascript:void(0)) 03:49

Element manipulation with WebDriver

04:39

How to fill out a form

05:49

Quiz on element manipulation

01:48

Answer to manipulation quiz

09:42

CurrentWindowHandle, WindowHandes, PageSource, Title

05:37

WebDriver URL

00:56

Interrogating IWebElements

03:53

Learning all IWebElement properties

05:56

Element interrogation quiz

03:10

Element interrogation quiz answer

04:06

Conclusions

03:20

-

Mouse and keyboard actions

08:12

Introduction

02:09

What topics will be covered

01:17

Section Resources

00:09

What are user interactions in Selenium Webdriver

02:11

What are the different kind of user interactions

02:26

-

Mouse and Keyboard Actions - Drag and Drops

32:00

[How to create an Actions class in Webdriver](javascript:void(0))

[Preview](javascript:void(0)) 03:20

[Analyzing our application under test](javascript:void(0))

[Preview](javascript:void(0)) 03:42

Writing the drag and drop test

04:40

Finishing the first drag and drop test

03:15

Executing the first drag and drop test

03:32

Starting to write the 2nd drag and drop example

03:47

Writing a 2nd drag and drop example

02:57

Running a 2nd drag and drop example

02:03

Drag and drop quiz question

01:06

Drag and drop quiz answer

03:38

-

Mouse and Keyboard Actions - Developer Tools, Resize, Draw

29:24

[How to do a resize action with selenium webdriver](javascript:void(0))

[Preview](javascript:void(0)) 04:35

[How to open developer tools with selenium webdriver](javascript:void(0))

[Preview](javascript:void(0)) 05:27

[Running a test that opens a network tab in firefox](javascript:void(0))

[Preview](javascript:void(0)) 01:59

Drag and drop with html 5 quiz question

00:57

Drag and drop with html 5 answer part 1

04:31

Drag and drop with html 5 answer part 2

02:47

How to draw on a canvas with selenium quiz

00:32

How to draw on a canvas with selenium quiz answer

04:06

Conclusions

03:37

Exam on Mouse and Keyboard Actions

00:01

Quick overview of the topics covered in the Implic and Explicit Waits section

00:52

-

Implicit and Explicit Waits - Introduction

27:00

Introduction and pre-reqs

02:27

Important updates to Implicit Waits

09:45

Notes for Implicit and Explicit Waits

00:21

Why Is Synchronization Important?

03:49

Quiz-Why does our test fail when it runs

03:12

Answer-Why does our test fail when it runs

02:46

What kinds of problems can arise as a result of improper sync

04:40

-

Implicit and Explicit Waits - Implicit Waits

21:41

[What are implicit waits in Selenium WebDriver](javascript:void(0))

[Preview](javascript:void(0)) 08:24

[Implicit Waits Quiz](javascript:void(0))

[Preview](javascript:void(0)) 01:32

[Implicit Waits Quiz Answer](javascript:void(0))

[Preview](javascript:void(0)) 02:34

disadvantage of an implicit wait 1

05:53

disadvantage of an implicit wait 2

03:18

-

Implicit and Explicit Waits - Explicit Waits

14:34

Introduction to Explicit Waits

04:12

Introduction WebDriverWait Class

05:36

Explicit waits quiz answer

04:46

-

ExpectedConditions, DefaultWait, and Best Practices

25:30

Understanding the ExpectedConditions class

03:20

Webdriver timeout exception and quiz

03:26

Explicit waits quiz answer

02:52

Dont mix implicit and explicit waits

01:49

DefaultWait class and quiz

06:04

DefaultWait class and quiz answer

03:00

Conclusions

04:58

Exam for Implicit and Explicit Waits Section

00:01

-

Why record and replay automation doesn't work

19:08

Overview of the rest of the course

02:03

Introduction to the next section and tools used

01:23

Recording automation test using selenium builder

03:17

Changing the remove webdriver to firefox

03:36

Why did the test fail a 2nd time?

01:45

Finally getting the recorded test to run

01:53

What are the problems with recorded functional tests

01:52

Conclusions regarding recorded tests

01:52

What can we do to fix problems with recorded functional tests?

01:26

Quiz

00:01

-

Why keyword driven test automation doesn't work

16:04

Introduction to keyword driven testing

04:24

What can change about KDF

08:13

Advantages and Disadvantages of KDF tests

02:14

Transitioning into Page Object Pattern

01:13

-

Page Objects

10:48

Introduction to the Page Object Pattern

01:55

Advantages and Disadvantages of the Page Object Pattern

04:23

First look at a functional test using the Page Object Model

03:28

Page objects look similar

01:02

-

TDD for automation frameworks

01:07:56

[Automation quiz](javascript:void(0))

[Preview](javascript:void(0)) 02:25

[02-Why did you struggle?](javascript:void(0))

[Preview](javascript:void(0)) 03:17

[Quiz](javascript:void(0))

[Preview](javascript:void(0)) 00:01

[Examples of bad automation tests](javascript:void(0))

[Preview](javascript:void(0)) 06:02

[Introduction to TDD](javascript:void(0))

[Preview](javascript:void(0)) 04:54

[Check execution environment](javascript:void(0))

[Preview](javascript:void(0)) 04:33

[Red and green phase](javascript:void(0))

[Preview](javascript:void(0)) 03:49

[Opening complicated page](javascript:void(0))

[Preview](javascript:void(0)) 07:50

[Finish green phase](javascript:void(0))

[Preview](javascript:void(0)) 02:42

Refactor

01:50

Red

01:43

Locating amazon search bar

03:18

Green

04:56

Green 2

01:45

Another refactor

02:55

Quiz on red and green

01:59

Quiz answer

07:47

Conclusions for TDD

03:51

Conclusions for the last several sections

02:19

-

Starting Framework Development

01:09:57

Getting started with Selenium framework

02:10

Writing the test skeleton

06:11

Fixing compiler errors in test

06:51

Quiz-How to setup ChromeDriver

00:59

Answer to how to setup ChromeDriver

05:22

Getting ChromeDriver to run

04:21

Making GoTo and IsVisible work

05:11

FillOutFormAndSubmit method

07:08

Making 1st test pass

05:56

Refactor

05:44

Quiz on best practices

00:55

Answer on best practices

02:49

Creating a BasePage class

04:53

Making functional test fail

05:38

Summary

05:49

-

Sprint2

58:50

Introduction to sprint 2

05:06

Adding Lastname field

04:45

Fixing a possible problem

04:18

Refactor phase

02:22

Adding a TestCleanup

04:51

Quiz answer on duplication

04:09

Answer on duplication 2

03:38

Answer on duplication 3

03:31

Red phase

06:33

Red phase 2

05:06

Red phase 3

04:48

Importance of Refactoring

05:17

Refactoring 2

04:26

-

Sprints 3+4

01:01:37

Sprint 3 Quiz

01:27

Adding a Gender

03:24

Update FillOutFormAndSubmit

04:54

Refactoring

03:06

Red and Green phase

03:57

Quiz on Other gender

02:03

Quiz answer

02:24

Refactoring

08:10

Refactoring 2

05:01

Refactoring 3

01:37

Sprint 4 Requirements

01:51

FillOutEmergencyContactForm method

07:11

Refactoring all tests 1

04:19

Refactoring 2

02:46

Red and green phases of sprint 4

02:38

Conclusions

03:40

Conclusions 2

03:08

Exam on Test Creating and Refactoring

00:01

-

Test organization and BaseTest class

01:02:07

How to organize your test cases and classes

04:03

Test organization in code

02:10

Quiz on test case 1

01:36

Skeleton of test case

03:06

Implementing the test

09:09

WebDriverFactory

10:42

Making test pass

03:37

Making test fail

04:02

Explaining WebDriverFactory

03:42

Quiz 2

02:14

Skeleton of test

01:44

Implementing the test

06:53

Running Test 2

02:29

A new problem

01:26

Creating BaseTest

05:14

-

Keeping classes small and solution organized

28:25

Problems with large page objects

05:07

How to keep page objects small

04:00

Quiz on slider

01:34

Quiz answer on slider

07:45

How to organize the solution

05:51

How to encapsulate page locators

04:08

-

Logging in test automation

59:34

Course syllabus and prerequisites

03:55

Why is logging important

02:49

Introduction to NLog

07:07

How to install NLog

04:24

Starting to log

04:23

Why Info level is important

02:53

Understanding NLog

05:46

Layout in NLog

02:21

fileName and keepFileOpen in NLog

02:00

More information about Info logging

05:02

Creating the first log records

05:32

NLog Exam

05:35

NLog exam answer

05:19

NLog bug logger

02:28

-

How to create HTML Reports

47:46

What to focus on

01:33

Why do reporting in automation

01:45

Overview of Extent Reports

02:21

Overview of how to use Extent Reports

04:07

Solution layout and new classes

02:54

The Reporter class

04:09

The BaseTest class

04:07

The ScreenshotTaker class

02:32

Viewing test results

03:32

How to use the Reporter class

05:45

Quiz on Reporter.cs

02:03

Quiz answer for Reporter.cs usage

07:33

Conclusions about Logging and Reporting

05:25

-

Conclusions and final exam

37:03

Final exam

04:42

HTML Report and steps

04:12

TCID4 Answer

04:29

TCID5 Answer

04:33

TCID6 Answer part 1

06:44

TCID6 Answer part 2

07:03

TCID7 Answer

05:20

-

Developing an automation testing framework

55:21

[Introduction](javascript:void(0))

[Preview](javascript:void(0)) 02:48

Getting weird Selenium Webdriver Errors?

00:04

[Prerequisites before proceeding](javascript:void(0))

[Preview](javascript:void(0)) 02:41

[Positives and negatives of the POM Framework](javascript:void(0))

[Preview](javascript:void(0)) 03:45

[Downloading the POM Framework](javascript:void(0))

[Preview](javascript:void(0)) 03:35

[Downloading Resharper](javascript:void(0))

[Preview](javascript:void(0)) 02:35

Opening visual studio and starting a new project

02:56

[How to install Selenium WebDriver Nuget packages](javascript:void(0))

[Preview](javascript:void(0)) 02:34

Add a reference to Selenium Webdriver for your QtpTests project

00:27

How to create a unit test project and add a reference

02:14

How to rename a test

02:41

Important Updates to QTP Tutorial

01:22

proper naming conventions of tests

01:11

How to code the first test

02:06

10-12 Understanding the first test

02:36

understanding the first test 2

02:37

understanding the first test 3

03:26

coding the second test

03:25

identifying elements of the second test

03:45

Fail, refactor, pass

03:15

fixing the tests

02:47

how to install chrome driver

02:30

-

Refactoring

25:08

[Understanding a problem with these tests](javascript:void(0))

[Preview](javascript:void(0)) 04:49

[Refactoring our test based on the DRY principle](javascript:void(0))

[Preview](javascript:void(0)) 04:21

[Refactoring our test based on the DRY principle 2](javascript:void(0))

[Preview](javascript:void(0)) 01:46

[Comparing the 2 tests before and after refactoring](javascript:void(0))

[Preview](javascript:void(0)) 02:24

Considering other problems with the current test

03:15

Doing a second refactoring of the test

03:29

Running the test and analyzing it one more time

01:44

Automation testing Exercise

03:17

What are your thoughts so far?

00:03

-

Creating the first automation test

01:23:02

Make sure that you have the latest source code!

00:07

[Introduction to the Page Object Model](javascript:void(0))

[Preview](javascript:void(0)) 03:12

[Introduction to the Page Object Model 2](javascript:void(0))

[Preview](javascript:void(0)) 02:21

[Benefits of the Page Object Model](javascript:void(0))

[Preview](javascript:void(0)) 03:20

[Benefits of the Page Object Model 2](javascript:void(0))

[Preview](javascript:void(0)) 03:18

[Words of wisdom from Albert Einstein](javascript:void(0))

[Preview](javascript:void(0)) 04:00

How to start removing problems and creating a TestBase class

02:15

Browser class fix

00:25

Fix to the TestBase class

00:30

How to code an Initialize method for the TestBase class

02:19

How to create a static driver and the Initialize method

03:59

Moving the Browser class to its own file and cleaning up

02:21

Cleaning up the new test to inherit from TestBase

04:19

Comparing the new test with the older test

02:45

How to create a test cleanup method

02:32

Hermetic Test Design Pattern

04:46

More design patterns

02:12

Designing the test

03:07

How to create a Pages class

03:37

Coding the Pages class

02:32

Coding the Pages class 2

03:11

Coding the LoginPage

02:27

Adding a goto method to the Browser class

02:34

Writing the login method of the Login page

03:55

Running the test and checking the results

04:03

Creating a membership page object

02:14

Implementing the IsAt method

03:07

Rerunning the test

03:22

Seeing the test results of the test

01:12

Making sure that the test fails successfully and closing comments

03:00

-

Analyzing the automation test

02:04:48

[comparing the 2 tests](javascript:void(0))

[Preview](javascript:void(0)) 05:09

[analyzing the test](javascript:void(0))

[Preview](javascript:void(0)) 04:07

[analyzing the test 2](javascript:void(0))

[Preview](javascript:void(0)) 03:27

[analyzing the test 3](javascript:void(0))

[Preview](javascript:void(0)) 04:27

[Writing the second test](javascript:void(0))

[Preview](javascript:void(0)) 04:18

analyzing the second test

04:32

writing the 3rd test

04:05

writing the 4th test

03:39

coding out the page objects

03:48

finishing the 4th test coding

02:15

[running the 4th test](javascript:void(0))

[Preview](javascript:void(0)) 04:16

[what is an implicit wait and why is it bad](javascript:void(0))

[Preview](javascript:void(0)) 04:25

[what is an explicit wait](javascript:void(0))

[Preview](javascript:void(0)) 04:19

designing an explicit wait

03:35

designing an explicit wait 2

04:03

designing an explicit wait 3

04:18

designing an explicit wait 4

02:13

designing an explicit wait 5

03:47

designing an explicit wait 6

04:01

refactoring the test

03:24

writing a unit test

04:28

writing a unit test 2

04:02

writing a unit test 3

04:00

refactoring our unit test

04:45

refactoring the Goto method

04:05

adding 2 more unit tests to the suite

04:04

final run of the unit tests

05:07

how to switch tabs using webdriver

04:47

how to switch tabs using webdriver 2

03:30

how to switch tabs using webdriver 3

04:47

final test run of the last test

03:05

-

Final refactoring

52:26

introduction to this section

08:27

refactoring the MyMembership page

04:09

how to run all the unit tests together

03:40

final validation of a test

02:32

conclusions 1

04:53

conclusions 2

04:44

conclusions 3

04:22

conclusions 4

04:37

conclusions 5

04:55

conclusions 6

05:21

JSON Currency Project pt 1

# 5th Jan :

### Selenium Basics

s

1. Udemey : lecture 34:

CSS

Tagname[attribute=’value’]

Regular expression:

Tagname[attribute\*=’value’]

Xpath

//tagname[@attribute=’value’]

Regular expression

//tagname[contains(@attribute,’value’)]

Xpath using parent child relation

To check xpath in chrome- in console $x(“xpath”)

Parentxpath/div/div[2]/div/input - - each tag if more than one child use indexing

#### Lecture 38: Cropath

Add plugin to chrome to validate the xpath and css

#### Lecture 39: Interview questions

1. Relative vs absolute
2. Moving between siblings- ParentXpath/FollowingXapth::sibling tagname
3. Traversing from child to parent – childXpath/parent::parent node or xpath

.//\*[@id=’we’]/parent::lu

#### Lecture 40:

1. How to identify element with text based

//\*[text()=’Selenium’]

package Tests;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

public class testit {

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

// TODO Auto-generated method stub

System.setProperty("webdriver.chrome.driver","C:\\work\\chromedriver.exe");

WebDriver driver =new ChromeDriver();

driver.get("http://www.qaclickacademy.com/interview.php");

driver.findElement(By.xpath("//li[text()=' Selenium ']")).click();

driver.findElement(By.xpath("//ul[@class='responsive-tabs\_\_list']/li[1]/following-sibling::li[2]")).click();

System.out.println(driver.findElement(By.xpath(".//\*[@id='tablist1-tab2']/parent::ul")).getAttribute("role"));

//

}

}

#### Lecture 42: CSS Selector locator techniques

[Attribute=’value’]

Tagname[attribute=’value’]

Tagname[attribute\*=’value’]

Classname = tagname.classname

Id= tagname#id

#### Lecture 43: tips and tricks to identify elements

#### Lecture 44: handling dropdown

* Static Drop down: Select tag with options comes into static dropdown category
* Select select = new Select(driver.findElement(By.id(“er”)));
* Select.selectbyValue(“2”);
* Select.selectbyIndex(6);
* Select.selectbyvisibletext(“Text1”);

#### Lecture 45: Handling dynamic dropdown

* When the same code is used for the two different dropdown then to identify the second dropdown wrap the xpath and mention the index as 2 or so…
* Xpath(“.//[@id=’DEL’])[2]

#### Lecture 46: Dealing with Checkboxes

* Click to check using simple xpath or id
* Validate checkbox is selected or not – using IsSelectod Method
* Element is not clickable at point (1,2) , other element would receive the click – when the drop down or some other element overlaps the element you want to operate

#### Lecture 47: handling Radio Button

* Similar to checkbox using type , value we can write xpath to identify the radio button
* Without using value option to select the first option from radio button find a common class name or attribute and use index to select the first element- xpath(“//input[@name=’group1’][1]
* To get the numbers of radio button for a specific goup use find elements.size- findElements.By.xpath(“//[@name=’group1’]).size

#### Lecture 52: Handling java alerts using webdriver API

* Driver.switchTo().Alert().accept()
* If cancel button is present on Alert- dismiss()
* If want to send text in alert- .switchTo().Alert().sendKeys()

#### Lecture 54: Web Elements validation

* How to check if UI element got hidden or is displayed – webelement.isDisplayed()

#### Lecture 55: Web Element validation – 2

* Isdiplayed will return only false when element is present but in hidden mode- should be present in code
* To validate any particular element is present in web page use webelement.size() method if size is 0 then there is no such element

#### Lectuer 56: Web Element validation -3 IsEnabled()

* Will return true if element is enabled otherwise false
* getText() method- get text from website use to validate text Sometimes element are not clickable on the site at that time we can use java script executer to click on the element- WebElement iafforcec\_checkbox=driver.findElement(By.xpath("//\*[@id='ctl00\_mainContent\_chk\_IndArm']")); ((JavascriptExecutor)driver).executeScript("arguments[0].click();",iafforcec\_checkbox );

#### Lecture 57: Web element validation – 4 Calender

* Practice

#### Lecture 58: End to End Practice

* Assertion- Assert class
* Assert.AssertMethod – Assert.assertTrue- if the value is false then the assertion gets failed and the script execution stops with an error
* Assertion- Practice

#### Lecture 60: Interview questions

* DropDown
* Alert
* Web Driver Methods
* If all radio button have same name and attribute the take the radio buttons in list using findelements then traverse through the list and select based on index

### Section 7: Advanced topics

#### Lecture 62: Synchronization in Selenium

* Implicit wait- define waiting time globally , if the element is found before the given time it stop waiting and comes out of the waiting and executes next step. If dosent find the element after waiting till that time it throws timeout exception
* Fluent Wait
* Thread.sleep

#### Lecture 64: Implicit Wait

* driver.manage().timeouts().implicitlywait(5,Timeunit.seconds)
* Position of declaring the implicitly wait is not important if defined at the beginning then also it will wait for the given time whenever element is not found
* Time unit defined in the implicit wait will work for all the element
* If you declare multiple implicit wait then the latest declaration will take over

#### Lectuer 65: Explicit wait

* WebDriverWait wait = new WebDriverWait(driver,5)
* wait.until(ExpectedCondition.VisibilityOfElementLocator(locator))
* Go through all Expected Condition method

#### Lectuer 66: Handling ajax and mouse interaction

* Action Class:

1. How to mouse hover on object with selenium

* Create object of Actions class- Actions action= new Actions(driver)
* **Action.moveToElement(locator).build().perform()**
* Build() is used so that we can use composite actions

1. Performing mouse and keyboard interactions with selenium

* To type capital letters in textbox , locate the text box -> click on it -> send keys
* Action.moveToElement(findElementByXpath(locator)).click().KeysDown(keys.SHIFT).sendKeys(“hello”).build().perform()
* Select text(double click) – doubleclick()
* Using KeyDown we can only use ALT, CTRL,SHIFT
* Any handling of Keyboard and mouse should be done by Action or Actions class as these are parts of Advanced user interactions API

1. Context click on selenium -> moveToElement(loc).contextClick()
2. Double click on element-> moveToElement(loc).doubleClick()
3. Drag and dropping the element

* Frames:

1. What are frames?
2. How to identify Frames in application
3. How to handle frames
4. Best Practices when working with frames application

#### Lecture 69: Handling multiple windows

* By default driver will stick to the window which it is initiated initially

#### Lecture 70: Window handle Concept real time Example

* getWindowhandles() – return set of string , it returns the windows id

Set<String> Ids= Driver.getWindowhandles();

* Set- FIFO – 1st – parent window id – 2nd – 2nd child window – 3rd- 3rd child window and so on…

Parent window

1st child

2nd Child

* To switch to child window use driver.switchTo().window(arg0)
* In place of argument we should pass the ids of the window
* To get the id of the child window iterate through the Set<Strings> using Iterator , Iterator return the Set of Strings so use Iterator<Strings> it=Ids.Iterator();

**Iterator<Strings> it = Ids.Iterator()**

* **Now the iterator at place above parent**
* **It.next() transfer the controls to the next place that is parent window**
* *String parentID= it.next();*

*String childID= it.next();*

#### Lecture 71: Live example on working with child windows

#### Lecture 72: How to handle frames

* Jqueryui.com/droppable/
* Selenium cannot identify frame directly , we need to tell selenium that this is frame
* How to identify frame – click on the element and look at the left side of web developer inspector if it is written as iframe then it is a frame , generally it will written as top window
* We can switch to a frame using frame id, frame index or web element
* Alert popup to enter user name and password

WebDriverWait wait = new WebDriverWait(driver, 10);

Alert alert = wait.until(ExpectedConditions.alertIsPresent());

alert.authenticateUsing(new UserAndPassword("username", "password"));

#### Lecture 73: Handling frame real time example

* To know how many frames are present – driver.findelements(by.tagname(“iframe”).size()
* To come to default content- driver.switchTo().defaultContent()
* Action.dragAndDrop(source locator, target locator)
* [info@qaclickacademy.com](mailto:info@qaclickacademy.com)

#### Lecture 74: interview Questions

### Section 10: Test Cases

#### Lecture 75: Test Cases Practice exercise

#### Lecture 76: Limiting webdriver scope

* How to find all the links- driver.findelements(by.tagname(“a”).size();
* Count the links in footer section- Webelement Footer = driver.findelelements(By.xpath(“.//\*[@id=’glbfooter’]).
* Footer.findelements(by.tagname(“a”).size();

#### Lecture 77: Counts of links in the page, section

* Driver.findelements(locator).get(index).gettext();

#### Lecture 78: Practice Exercise

#### Lecture 79: Dynamic Data in website

#### Lecture 80: Dynamic link handling

#### Lecture 81: validations and checkpoints

#### Lecture 83: Handling calendar

* First get the xpath of date field
* To select a date say – 23 August 2018
* Use css to select the date
* In most cases all day in a month will have same class name so get all the elements with the class name and store in a list and later retrieve them – List<WebElement> date = driver.findelements(By.ClassName(“.day”)
* Run the loop over list and get the text of each item using gettext
* If text is equal to given date text then click on that element
* If(text.equalsignorecase(“23”)
* Use findelements

#### Lecture 84: handle calendar- month & Years

* Run while loop till you get month , using contains , and if the month doesn’t matches with the given month then click next

While(!driver.(findelementBy.cssSelector(“[class=’datepicker-days][class=’datepicker-switc h’]”)).getText().contains(“April”){

Driver.findelementBy.cssSelector(“(“[class=’datepicker-days] th[class=’next’])).click()

}

#### Lecture 85: Solved example of Calender- makemytrip.com

* Debug- put a toggle point where ever you want the execution to stop , f6 or step over to go to the next line
* To remove the break point click disable breakpoint and click resume

while(!driver.findElement(By.cssSelector("[class='datepicker-days'] [class='datepicker-switch']")).getText().contains("May"))

{

driver.findElement(By.cssSelector("[class='datepicker-days'] th[class='next']")).click();

}

List<WebElement> dates= driver.findElements(By.className("day"));

//Grab common attribute//Put into list and iterate

int count=driver.findElements(By.className("day")).size();

for(int i=0;i<count;i++)

{

String text=driver.findElements(By.className("day")).get(i).getText();

if(text.equalsIgnoreCase("21"))

{

driver.findElements(By.className("day")).get(i).click();

break;

}

}

}

}

#### Lecture 88: problem- dream11.com

* Captcha, I am not robot- to click on I am not robot switch to frame
* Fir get all the frames in the site and get the size-> run loop till frame count and witch to the frame
* Then inside for loop check whether the I am not robot checkbox is present inside the frame

#### Lecture 89: validate element present on thepage

* isDisplayed- checks visible or invisible
* get the size using find elements

#### Lecture 91`: Generic function to identify frames

* parent child frame can be switched directly
* independent frame = switchTo().defaultContent()
* to pass web element to method we can use By by and pass the web element as method para meter

public void findFrameNumber(Webdriver driver, By by{

driver.findelements(by).size()

}

#### Lecture 92: Demonstrating multiple frames

* How to find one frame is another frame- HTML code

#### Lecture 93: how to handle auto suggest- dream11.com

* After typing the character in the text box put a explicit wait till the auto suggest window appears
* Wait.ExpectedCondition(Visibility of element located(findElement(By.xpath(“whole box of suggestion box”)
* Then click on the element from the auto suggestion

#### Lecture 94:Exception technique

* When element may or may not be present
* Try Catch
* Inside catch block take the screenshot

// TODO Auto-generated method stub

WebDriver driver=new FirefoxDriver();

WebDriverWait wd=new WebDriverWait(driver,7);

driver.get("<https://fantasycricket.dream11.com/IN/>");

driver.manage().window().maximize();

int m=gotoframe(driver,By.xpath(".//\*[@id='recaptcha-anchor']"));

driver.switchTo().frame(m);

driver.findElement(By.xpath(".//\*[@id='recaptcha-anchor']/div[5]")).click();

driver.switchTo().defaultContent();

wd.until(ExpectedConditions.frameToBeAvailableAndSwitchToIt(By.id("I1\_1441700500937")));

int j=gotoframe(driver,By.xpath(".//\*[@id='recaptcha-verify-button']"));

if(j!=-1)

{

driver.switchTo().frame(j);

//WebDriverWait wd=new WebDriverWait(driver,5);

//wd.until(ExpectedConditions.(By.xpath(".//\*[@id='recaptcha-verify-button']")));

driver.findElement(By.xpath(".//\*[@id='recaptcha-verify-button']")).click();

}

else{

System.out.println("ops");

}

}

}

// Code

package basics;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebDriverException;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.interactions.Actions;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.WebDriverWait;

public class Day1 {

public static int gotoframe(WebDriver driver1,By by)

{

driver1.switchTo().defaultContent();

int i;

int num = -1;

int a=driver1.findElements(By.tagName("iframe")).size();

for(i=0;i<a;i++)

{

driver1.switchTo().defaultContent();

driver1.switchTo().frame(i);

int b=driver1.findElements(by).size();

if(b>0)

{

num=i;

break;

}

}

driver1.switchTo().defaultContent();

return num;

}}

### Section 12- cricbuzz.com

#### Lecture 96: How to handle table grid in webpage

* Find a common class name but traverse through parent table name relevant to your table
* Find a parent table and declare in a web element
* Then find elements by (webelement)table.findelements

Table.findelements.by.cssSelector(“cb-col items”).size()

To traverse to third child

In css= space tagname:nth-child(number)

Space div:nth-child[3]

Int rowCount= table.findelements(By.cssSelector(“div[class=’cb-col cb-col-100 cb-ltst-wgt-hdr’]”))

Int count= table.findElements(By.cssSelector(“cb-col cb-col-100 cb-scrbd-itms div:nth-child(3) “)).size()

For(int i=0;i<count;i++)

{

table.findElements(By.cssSelector(“cb-col cb-col-100 cb-scrbd-itms div:nth-child(3) “)).get(i).getText()’

}

#### Lecture 98: practice on table

* To get the extra run from the table since all the class name is same use text and following sibling

Driver.findelementBy.Xpath(“.//div[@text()=’Extras’]/Following-sibling::div

#### Lecture 99: Practice on table – sum all runs and match with total

public class Tableexercises {

public static void main(String[] args) {  
// TODO Auto-generated method stub  
int sum=0;  
System.setProperty("webdriver.chrome.driver", "C://work//chromedriver.exe");  
WebDriver driver = new ChromeDriver();  
driver.get("http://www.cricbuzz.com/live-cricket-scorecard/18970/pak-vs-sl-2nd-t20i-pakistan-v-sri-lanka-in-uae-2017");  
  
WebElement table=driver.findElement(By.cssSelector("div[class='cb-col cb-col-100 cb-ltst-wgt-hdr']"));  
int rowcount= table.findElements(By.cssSelector("cb-col cb-col-100 cb-scrd-itms")).size();  
int count=table.findElements(By.cssSelector("div[class='cb-col cb-col-100 cb-scrd-itms'] div:nth-child(3)")).size();  
  
for(int i=0;i<count-2;i++)  
{  
String value=table.findElements(By.cssSelector("div[class='cb-col cb-col-100 cb-scrd-itms'] div:nth-child(3)")).get(i).getText();  
int valueinteger=  Integer.parseInt(value);  
sum=sum+valueinteger;//103  
}  
//System.out.println(sum);  
  
String Extras=driver.findElement(By.xpath("//div[text()='Extras']/following-sibling::div")).getText();  
int extrasValue=Integer.parseInt(Extras);  
int TotalSumValue=sum+extrasValue;  
System.out.println(TotalSumValue);  
  
  
String ActualTotal=driver.findElement(By.xpath("//div[text()='Total']/following-sibling::div")).getText();  
int ActualTotalVAlue=Integer.parseInt(ActualTotal);  
if(ActualTotalVAlue==TotalSumValue)  
{  
System.out.println("Count Matches");  
}  
else  
{  
System.out.println("count fails");  
}  
}

}

#### Lecture 101: Handling HTTPS Certification

* ***Ssl certifications***- For Chrome – there is a class we can use for this which is chromeOption
* ***Insecure Certification***

ChromeOption c= new ChromeOption();

c.merge()

* Desired capabilities is a class which can be used to customized the browser
* First we need to create desired capabilities class object

DesiredCapabilities ch = desiredcpabilities.chrome();

Ch.acceptInsecureCert()

Or

Ch.setCapability(capabilityType.Accept\_INSECURE\_CERTS, true)

Ch.setCapability((capabilityType.ACCEPT\_SSL\_CERTS, true)

* Then merge the capabilities with the chrome option which belongs to local browser , Desired capabilitites is general profile

ChromeOptions c = new ChromeOptions()

c.merge(ch);

* Then pass the ChromeOptions object to the ChromeDriver object

WebDriver driver= new chromeDriver();

* Desired Capability is used to create a general profile and chrome options is used to customize the local browser

#### Lecture 103: Maximizing windows and deleting cookies

* Maximize= driver.window.manage().maximize()
* To delete cookies- driver.manage().cookies()
* To delete a particular cookie- driver.manage.deleteCookieNamed(“asdf”)
* How to bring login page in the middle of the test-

Driver.manage.deleteCookienamed(“sessionKey”)

#### Lecture 104: Taking screen shot

* To take the screenshot we have to convert the driver object t to the screenshot object to change the behavior
* Cast driver object to the take screenshot object

File file= ((takeScreenShot)driver).getScreenshotAS(OutputType.file)

* After taking the screenshot we are telling the object to show the screensot as a file

FileUtils.copyfile(src, new file(C://”newFile.jpg”)

* Fileutils- package name – org.apache.common.io.FileUtils
* Common Error- Access is denied if you are writing to the C drive- change the drive

#### Lecture 105: Killing the process and cookies using selenium

* Kill process/window- WindowsUtil class

WindowsUtils.killByName(“excel.exe”)

#### Lecture 106: customize way of identifying the locators

* Click esc to open console window
* To check in console- $x()

.//tagName[@Attribute=’value’]

* If the id is same then to get the 2nd element

Xpath = (“(.//a[@id=’same’)[3]”)

#### Lecture 107: Customize xpath

* Starts-with(@attribute,’value’)

.//a[starts-with(@id,’new’)]

* Ends-with

### Section 16: TestNG

#### Lecture 128: Why testing

* Either we can add tesng jar or add eclipse plug in

#### Lecture 129: running TestNg

* Test ng acts as java compiler and runs the program

Testng

* Use of @test

#### Lecture 130: importance of xml file in TestNg configuration

* To create testng xml – right click and convert to testng
* Test Suite-> Test Folder (module Name)-> Test Cases
* We can change the test name
* Class name= package.classname
* testNG.xml

#### Lecture 131: Prioritizing the test cases with testNG

* we can create category in test sui by creating different test sets and run according to the requirement

#### Lecture 132: Include and exclude mechanism in TestNG

* to exclude a particular tag modify testng xml – inside the test set create a method tag inside class and then use <exclude> tag to exclude that test
* include- whatever define in include will only execute

#### Lecture 133: Executing the test cases at package level with regex

* if you want to exclude any test with a particular test case name
* then we can use – name.\* in exclude tag
* package level- instead of classes tag after test use packages tag and then package

#### Lecture 134: Test NG Annotations

* before test- before executing every test method from testng xml test module
* after test- same scope limited to test from testNG

#### Lecture 135: TestNG annotation

* Before Suite-
* After Suite-
* Beforemethod- before each test method inside the class

#### Lecture 136: TestNG Groups

* Before Class
* AfterClass
* Order of test execution based on alphabetical order
* Groups- should be placed after test tag

Test->groups->run->include->name of group

#### Lecture 137: Annotations helper attributes

* **Depends** on method declared in @test tag

@test(groups={“name”})

* Depends on method scope is within class
* **Enable**- to skip the test

@test(groups={“name”}, enable=false)

* **Timeout -**  set the timeout for a particular test

@test(timeout=4000)

#### Lecture 138: parameterization in TestNG

* Parameter>- if written just after suite tag then it will be applicable to all the test methods

#### Lecture 139: multiple parameter, data providers

@parameters({“url”})

@test

<parameters name=”url” value=”web”>

* Create a multi dimensional array – row is how many times the test should run , col- how many types data

@DataProvider

Public Object[] [] getData(){

Object[][] data = new Object[3][2];

Data[0][0]= “userNm”;

Data[0][1]= “pwd”;

Return data;

}

#### Lecture 140 : Data providers

* To call the data provider

@test(dataProvider=”getData”)

Public void test(String un, String pd){

}

#### Lecture 141: Listener Interface