SELENIUM CUCUMBER Interview Preparations Questions

SELENIUM

1. **What is Selenium and what is composed of?**

- Selenium is a suite of tools for automated web testing. It is composed of;
  - Selenium IDE(Integrated Development Environment); a Firefox plugin that works for recording and playing back.
  - Selenium RC(Remote Control) (1.0) ; is a test tool and is used to work on JS to automate the web application. (2004)
  - WebDriver (2.0); is a web automation framework and allows you to execute your tests in different browsers. (2011)
  - Selenium Grid; allows tests to run in parallel across multiple machines.

2. **What are the advantages of Selenium?**
- Selenium is open source and free to use without any licensing cost
- It supports multiple languages like Java, Ruby, Python, C#...
- It supports multi-browser testing
- It has a good amount of resources and helping community
- It supports many operating systems like Windows, Mac, Linux ...
- Interact with the web application

3. **What are the disadvantages of Selenium?**
- Selenium supports only web-based applications, does not support windows-based application
- No built-in reporting tool, it needs third party tools for report generation activity
- Cannot work with graphics, captchas, barcodes, shapes
- It does not support file upload facility.
- Hard to master, requires developer level knowledge
- Hard to write good locators
- Hard to synchronize

4. **What are the limitations of Selenium?**
- We cannot test desktop application
- We cannot test web services
- Ewe have to use external libraries and tools for performing tasks like testing framework (TestNG, JUnit), reading from external files (Apache POI for excel)
- Automating Captcha is not possible using Selenium
- It does not support file upload facility.

5. **What types of testing you automate with Selenium?**
- functional tests (positive/negative, UI)
- smoke tests
- regression tests
- integration tests
- end to end testing
- data driven

6. **What we don't do with selenium?**
- Performance, load, stress testing, manual ad hoc testing, (These tests are done by experts trained in these tools)
- Pure database testing (if we only test the DB itself),
- Unit tests..., look and feel based testing (color, shapes, etc.),
- static testing

7. **What is in the Selenium tool set?**
   - Selenium IDE □ implemented as a Chrome and Firefox extension, and allows you to record, edit, and debug tests.
   - Selenium RC □ to write automated web application UI tests in any programming language
   - Selenium WebDriver□ execute your tests against different browsers
   - Selenium GRID □ run your tests on different machines against different browsers in parallel.

8. **What version of Selenium do you use right now?**
   - JDK (JAVA) - 1.8 □ I like it because of □ Lambda exp. and, Try catch error handling you may add multiple catches.
   - IntelliJ - 2018.03.04
   - Selenium - 3.141.59
   - TestNG - 6.14.3
   - Cucumber – 4.2.6
   - Maven - 3.6.0
   - GIT - 2.17.2

9. **Implicit Wait vs Explicit Wait?**
   - **Implicit wait** is a wait which waits for a specified time while locating an element before throwing "NoSuchElementException". As by default selenium tries to find elements immediately without any wait. So, it is good to use implicit wait. This wait applied to all elements of the current driver instance.
   - **Explicit wait** is a wait which is applied to a particular webelement until the ExpectedCondition specified is met.
   - Implicit wait is simply; if condition is met before the timeout, it will continue to next step, if condition is not met within timeout throw "No Such Element" exception.
   - Explicit wait sometimes we need to wait for a certain event/condition such as element is visible, clickable, enabled....

```
driver.manage().timeouts().implicitlyWait(5,TimeUnit.SECONDS);
webDriverWait wait = new WebDriverWait (driver, 5);
wait.until (ExpectedConditions.visibilityOf(element);
```

10. **What is fluentWait?**
    - Let's say you have an element which sometime appears in just 1 second and some time it takes minutes to appear. In that case it is better to use fluent wait, as this will try to find element again and again until it find it or until the final timer runs out. Example is AJAX or JQuery
    - Subtype of explicit wait but you can override the conditions

```
Wait<WebDriver>wait=new
FluentWait<Webdriver>(driver).withTimout(5,timeUnit.seconds).pollingEvery(100,timeunit.
milliseconds).ignoring(NoSuchElementException.class);
```

11. **What are various ways of locating an element in Selenium?**
    - Selenium Locators □ Id & name
    - In selenium locator is a means of finding an element in the html :
    - Id, name, className, xpath, css, linkText, partialLinkText, tagName

12. **Why I cannot find element?**
    - Locator changed
    - There is an iframe
    - Waiting time:: page is loading slowly or Element is dynamic:: locator
    - Page is not fully loaded/opened
    - Page changes and that element does not exist anymore

13. **How to highlight an element?**
    - Selenium WebDriver doesn't have highlight action.
    - But we can use JavaScript to do it

```
JavascriptExecutor js = ((JavascriptExecutor) driver);
String bgcolor = element.getCssValue("backgroundColor");
for(inti=0;i< 10;i++){
   changeColor("rgb(0,200,0)",
   element,driver);//1
   changeColor(bgcolor,
   element,driver);//2
}
```

14. **What is Xpath?**
    - Xpath is used to find the location of any element on a webpage using html structure.
    - We could navigate through elements and attributes in an XML document to locate web Elements such as textbox.
    - button. checkbox, Image ext... in web Page

15. **Absolute (/) and Relative (//) Xpath?**
    - Syntax □ //tagname[@attribute=`value`]
    - Absolute xpath starts with single slash (/), starting from root element and all the way to the element.
    - Relative xpath starts with double slash (//), starting selection matching anywhere in the document.

16. **How do you handle dynamic elements?**
    - Find the static part of the id and write a locator(xpath or css) □ And then use Startswith, contains, EndsWith
    - contains( ) □ //*[contains(@name=`btn`)]
    - startwith( ) □ //label[startwith(@id, `message`)]
    - text( ) □ //td[text() = `usedId`]
    - or & and □ //input[@type = `submit` AND @name = `login`]

17. **How to test dynamic web page?**
    - There is no one size fits all solution to this problem. We have to understand the application very well
        - Use explicit waits where necessary.
        - Use custom xpaths and css locators
            - Xpath: contains, starts with, ends with, contains text.
            - By finding the element in relation to another stable element using parent, child, sibling relationships

18. **How to test dynamic table?**
    - Use custom xpaths and css locators
        - Xpath: contains, starts with, ends with, contains text.
        - By finding the element in relation to another stable element using parent, child, sibling relationships
    - I have utility methods that work with table. I have method that takes a table webelement and returns all the column names. I have a method that takes a table, number and returns all the data in that row.

19. **How can we move to parent element using xpath?**
    - Using (..) expression in xpath, we can move to parent element

20. **Difference between close() and quit() command?**
    - driver.close()□ used to close the current browser
    - driver.quit()□ used to close all the browser instances

21. **How can we move to nth child element using xpath?**
    - **There are two ways:**
        - using square brackets with index position
          For ex: div[2] will find the second div element
        - using position ( ) method
          For ex: div[position()=2] will find the second div element

22. **Difference between xpath and css selector?**
    - with xpath, we can search elements backward or forward... while css works only in forward direction
    - Xpath can work with text, css cannot work
    - Xpath has more combination and can search by index
      css cannot search by index, but css is working faster than xpath.

23. **What is framework?**
    - In test automation, framework is the blueprint of test automation.
    - It includes your folder structures, where to save you function library, test results, test data, resources.
    - It is essential because when you are working on a automation project everyone will have a guideline to follow and our script will be easier to maintain.

24. **Talking about HTML reporting during the interview?**
    - I use multiple methods of reporting in my framework, driver script writes pass/fail to the test cases excel sheet,
    - Reporter utility object writes to UFT report, also I have developed a custom HTML reporting engine.
    - It sends HTML code to the Notepad and creates a nice HTML report document that nontechnical people can easily understand and use.

25. **How to maximize a web page?**

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

26. **In some cases, maximize() will not work > so what will be the way around?**
    - Actions or change version.

```
ChromeOptions options = new ChromeOptions();
options.addArguments("startmaximized");
```

27. **What is the key class in Selenium?**
    - Gives us option for pressing keys from keyboard
    - Key.ENTER
    - MUST BE PASSED TO SendKeys() method
    - Ex; .sendkeys(“charger” + keys.ENTER)

28. **What if there is a dynamic popup that comes up randomly**
    - Use try/catch with alert

29. **What is Thread.sleep()?**
    - Slows down selenium to catch up
    - Throws exception so must handle it or throw it

30. **What is Selenium Framework?**
    - It is a code structure that helps to make code maintenance easy, code readability and code reuse.
    - There are mainly 3 type of frameworks created by Selenium WebDriver to automate test cases:

    Data Driven Framework
    - It is one of the most popular automation frameworks in the market
    - All of our test data is generated from some external files;
        - excel
        - or scenario outline in feature file
        - or TestNG Data Provider
    - Selenium WebDriver is a great tool to automate web-based applications. But it does not support read and write operations on excel files. Therefore, we use third party APIs like **Apache POI**

    Keyword Driven Framework
    - Keyword driven testing is a scripting technique that uses data files to contain the keywords related to the application being tested.
    - Keywords are written in some external files like excel file and Java code will call this file and execute test cases.

    HybridDriven Framework
    - A combination of the DDF and KDF is commonly said to be HDF.
    - Both the test data and test action are kept in external files.

31. **How did you use overloaded Methods in Selenium?**
    - When asserting if two values are equal, I use Assert.assertEquals(actual, Expected)from TestNG
    - You can put in the parameters String, Objects, int, boolean values

32. **Why we get NoSuchElementException?**
    - Check if locator is correct
    - Check if timing is correct
    - Check if element is hidden inside an iframe

33. **How you handle js alerts?**
    - If the alert on the browser comes from JavaScript, we use the Alert class.

```
Alert alert = driver.switchTo.alert();
alert.accept();
alert.dismiss();
alert.sendKeys();
alert.getText()
```

34. **How to handle multiple frames?**
    - If there are 4 frames, you have to go through each from consecutively to reach certain frame. Can't jump to the 3rd frame from 1st frame.

35. **What is the difference between driver.get() and driver.navigateto() ?**
    - driver.get() To open an URL and it will wait till the whole page gets loaded
    - driver.navigateto() To navigate to an URL and it will not wait till the whole page get loaded

36. **How to handle frames in Selenium?**
    - Frames used to embed a html page into another
    - Steps
        - Locate the iframe
        - Switch to another iframe with driver.switchTo().frame();
          .frame() □ takes string, Integer, webElement, name or id directly as parameter

```
driver.switchTo().frame(webElement);
driver.switchTo().frame();
```

    - Now you are in the 2nd frame, if you want to find an element outside of the 2nd frame (that you're currently on) throws NosuchElementException
    - If you need to switch back to previous frame
        - driver.switchTo().parentFrame() □ Goes one level up
        - driver.switchTo().defaultcontent() □ Goes to the very top
    - Can switch using count
        - driver.switchTo(0) □ Counts anything that is not the default frame

    These methods might give you different results based on what browser you are using

37. **How you handle browser pop ups?**
    - **void dismiss()** □ clicks on the "Cancel" button as soon as the pop-up window appears.
    - **void accept()** □ clicks on the "Ok" button as soon as the pop-up window appears.
    - **String getText()** □ returns the text displayed on the alert box.
    - **void sendKeys(String stringToSend)** □ enters the specified string pattern into the alert box.

38. **How you handle windows/ OS pop ups?**
    - Selenium doesn't support windows-based apps, it is an automation testing tool that supports only web application testing.
    - We could handle windows-based popups in Selenium using some third-party tools such as AutoIT, Robot class
    - **driver.getWindowHandle();** This will handle the current window that uniquely identifies it within this driver instance.
    - **driver.getWindowHandles();** To handle all opened windows

39. **How to handle Headless browser**
    - Headless browser: browser that does not open, it runs as a background service / program.
    - Example is htmlunitdriver from selenium
        - WebDriver = new htmlunitdriver()
        - Not very stable
    - Phantonjsbrowser
        - More stable
        - browser = new phantomjsbrowser()

40. **findElement vs findElements?**
    - FindElement > this method returns first WebElement !
        - gives Exception if the element not found
    - FindElements > returns List <WebElement>;
        - does not give Exception if the element not found as a result list has null values

41. **How to handle multiple windows/tabs?**
    - Selenium stays on one window
    - If you open a window and then 5 tabs popped open, selenium is focused on the first window
    - If you are on a new window and you tell selenium to print an element on the default window, it will still work even that user's focus is on the new window
    - Must switch to new window
        - Use windowHandle()
        - Driver.getWindowHandle()
            - Everytime Selenium opens a browser, it's going to give an index ID for the page called Handles
            - Returns the handle/id of current page (as a string)
        - driver.switchTo().window(string handle)
        - driver.getWindowHandles() for multiple windows
            - Returns a Set of window handles
        - Switch using titles

```
for(string handle: driver.getWindowHandles()){
   driver.switchTo().Window(handle)
   if(driver.getTitle().equals(targetTitle){
   break;
   }
}
```

42. **How to find all links in the page?**

```
List<WebElement> list = driver.findElements(By.tagName("a"));
```

43. **Difference between isDisplayed(), isEnabled(). And is Selected() method in selenium WebDriver?**
    - isDispalyed() □ verify the presence of a web element within the web page. If found □ true, If not found □ false
    - isDisplayed() □ check for the presence of all kinds of web elements available
    - isEnabled() □ verify if the web element is enabled or disabled within the web page.
    - isEnabled() □ is primarily used with buttons
    - isSelected() □ verifies if the web element is selected or not
    - isSelected() □ used with radio buttons, dropdowns and checkboxes.

44. **How to Drag And Drop ?**

```
Actions action = new Actions(driver);
action.clickAndHold(driver.findElement(By.id("item")))
.moveToElement(driver.findElement(By.id("destination")))
.release().build()
.perform();
```

45. **For Scroll down:**

```
WebDriver driver = new ChromeDriver(); JavascriptExecutor jse = (JavascriptExecutor)driver;
jse.executeScript("window.scrollBy(0,250)", "");
```

- OR, we can do as follows:

```
jse.executeScript("scroll(0, 250);");
```

46. **For Scroll up:**

```
jse.executeScript("window.scrollBy(0,-250)", ""); OR, jse.executeScript("scroll(0,-250);");
```

47. **How to handle cookies?**

- In Selenium Webdriver, we can query and interact with cookies with below built-in method:

```
driver.manage().getCookies();   // Return The List of all Cookies
driver.manage().getCookieNamed(arg0); //Return specific cookie according to name
driver.manage().addCookie(arg0); //Create and add the cookie
driver.manage().deleteCookie(arg0); // Delete specific cookie
driver.manage().deleteCookieNamed(arg0); // Delete specific cookie according Name
driver.manage().deleteAllCookies(); // Delete all cookies
```

48. **Why do we need to handle cookies?**

- Each cookie is associated with a name, value, domain, path, expiry, and the status of whether it is secure or not. In order to validate a client, a server parses all of these values in a cookie.
- When Testing a web application using selenium web driver, we may need to create, update or delete a cookie.
- For example, when automating Online Shopping Application, we many need to automate test scenarios like place order, View Cart, Payment Information, order confirmation, etc.
- If cookies are not stored, we will need to perform login action every time before we execute above listed test scenarios. This will increase your coding effort and execution time.
- The solution is to store cookies in a File. Later, retrieve the values of cookie from this file and add to it your current browser session. As a result, you can skip the login steps in every Test Case because your driver session has this information in it.
- The application server now treats your browser session as authenticated and directly takes you to your requested URL.

49. **Store cookies cookie information**

```
public class cookieRead extends BasePage{
    public static void main(String[] args){
        driver.get("http://demo.guru99.com/test/cookie/selenium_aut.php");

        // Input Email id and Password If you are already Register
        driver.findElement(By.name("username")).sendKeys("abc123");
        driver.findElement(By.name("password")).sendKeys("123xyz");
        driver.findElement(By.name("submit")).click();

        File file = new File("Cookies.data");    // create file to store cookies

        try {
            file.delete(); // Delete old file if exists
            file.createNewFile();
            FileWriter fileWrite = new FileWriter(file);
            BufferedWriter Bwrite = new BufferedWriter(fileWrite);

            // loop for getting the cookie information
            for(Cookie ck : driver.manage().getCookies()){
               Bwrite.write((ck.getName() +";"+ ck.getValue() +";"+ ck.getDomain() +";"+ ck.getPath()
                                        + ";" + ck.getExpiry() +";"+ ck.isSecure()));

               Bwrite.newLine();
            }

            Bwrite.flush(); Bwrite.close(); fileWrite.close();
        } catch(Exception ex) {
            ex.printStackTrace();
        }
    }
}
```

50. **Use stored cookies to login information**

```
public static void main(String[] args){
    try{
        File file = new File("Cookies.data");
        FileReader fileReader = new FileReader(file);
        BufferedReader Buffreader = new BufferedReader(fileReader);

        String strline;
        while((strline=Buffreader.readLine())!=null){
            StringTokenizer token = new StringTokenizer(strline,";");
            while(token.hasMoreTokens()){
                 String name = token.nextToken();
                 String value = token.nextToken();
                 String domain = token.nextToken();
                 String path = token.nextToken();
                 Date expiry = null;

                 String val;
                 if(!(val=token.nextToken()).equals("null")) {
                     expiry = new Date(val);
                 }
                 Boolean isSecure = new Boolean(token.nextToken()).booleanValue();
                 Cookie ck = new Cookie(name,value,domain,path,expiry,isSecure);
                 System.out.println(ck);
                 driver.manage().addCookie(ck); // This will add the stored cookie to current session
            }
        }
    }catch(Exception ex){
        ex.printStackTrace();
    }
    driver.get("http://demo.guru99.com/test/cookie/selenium_aut.php");

}
```

OUTPUT: You are taken directly to the login success screen without entering the input user id and password NOTE: Use hard refresh in case you see the login page after executing the above script.

51. **Do you use JavaScriptExecutor?**

- This helps me write my own JavaScript. JS has way more control than selenium.
- we can send JS commands to the browser with using this class JavaScriptExecutor jsExecutor=(JavaScriptExecutor)driver;
  - executeScript(); performs the command
  - Inside the parameter is where you put JS code
- jsExecutor.executeScript("alert('WARNING: This is a useless message');" □ This code will bring up a JS popup
- You can also put 2 parameters is .executeScript("js code",element);
  - Used for scrolling (selenium is not good with scrolling, you can say a challenge is when I was working on terms and condition page, where you have to read the page before clicking on continue.
  - When I tried using selenium and actions class it didn't work, so i used javaexecutor) and clicking an element;

52. **How to check if element is present/visible/enable/ and to check text present?**

- To check Element Present:

```
if(driver.findElements(By.xpath("value")).size() != 0){
   System.out.println("Element is Present");
}else{
   System.out.println("Element is Absent");}
```

- or

```
if(driver.findElement(By.xpath("value"))!= null){
   System.out.println("Element is Present");
}else{
   System.out.println("Element is Absent"); }
```

- To check Visible:

```
if(driver.findElement(By.cssSelector("a > font")).isDisplayed()){
   System.out.println("Element is Visible");
}else{
   System.out.println("Element is InVisible"); }
```

- To check Enable:

```
if(driver.findElement(By.cssSelector("a > font")).isEnabled()){
   System.out.println("Element is Enable");
}else{
   System.out.println("Element is Disabled"); }
```

- To check text present

```
if(driver.getPageSource().contains("Text to check")){
   System.out.println("Text is present");
}else{
   System.out.println("Text is absent"); }
```

53. **How check the multiple selected values in dropdown?**

- Select carsList = new Select(el)
- carList.getSelectedOptions(): //returns the the selected options a list (List<webelement>)
- for each : carList.getSelectedOptions()

54. **How to use actions class?**

- Actions class lets us do advanced mouse and keyboard operations:
- Control the mouse
- Class that provides methods for advanced user interactions
  - Hovering
  - Scroll
  - Double click
  - Drag and drop
  - Right click
  - mix/match operators

- Actions action=new Actions(driver)
- Action methods
  - click()
  - perform()
  - dragAndDrop(source,target).perform()
  - hold()
  - keydown()build()
  - moveToElement(element)
  - sendKeys() different from the one we usually use

        - Let's you do the sendkeys operation on different elements
        - Regular sendkeys that comes from webelement will throw an exception on something that is not input text.
        - The long way is;
          actions.moveToElement(source).clickAndHold().moveToElement(target).release().perform();
    - Actions won't work unless perform() is used
    - If you are chaining methods, you must use build() before perform()

55. **What is the syntax for double click action ?**
    - To perform any actions against web element using actions class, we need to locate the element first:

```
WebElement el = driver.findELement
Actions actions = new Actions (driver).perform actions.doubleClick(el).perform()
actions.moveTo(el).perform actions.doubleClick.perform
actions.moveTo(el).doubleClick().build.perform()
```

56. **File download and upload**
    - **Download**
        - Selenium itself cannot verify file downloads, can click on download link but can't go outside the browser and open the downloaded file
        - Other tools need to be used for that Robot and AutoIT
    - **Upload**
        - Selenium handles the upload, but does it differently compared to actual user
        - Steps
            - Find the element that triggers the upload window
            - Find the path of the file you want to upload
                - Store into a String
                    - Ex: String □ file="C:\\Users\\Andy\\Desktop\\folder1\\file.key";
                    - Then driver.findElement(upload button).sendkeys(file);

57. **How check the selected value in dropdown?**

```
Select carsList = new Select(el)
carList.getFirstSelectedOption()
assertequals("some text",carList.getFirstSelectedOption().getText() )
```

58. **How to work with dropdown without the select tag?**
    - If the dropdown list has no select tag, we cannot use the select class
    - Treat the dropdown list and its options as separate elements, locate every element separately
    - To select an option:
        - 1. Find and click on the list
        - 2. Find and click on the option

59. **What if there's no select tag?**
    - You have to select the label for the dropdown separately as a webelement.
    - Then manually use click method

60. **Sometimes sendKeys does not work**
    - Robot or AutoIT
    - library==jar file==dependency

61. What is the syntax for switching frame ?

- Frame is a html document inside another html document.
- Web driver handles one page/html document at a time. To control another frame, we always need to switch
- Driver.switchTo.frame(webelement) □ find the iframe and pass as a param
- Driver.switchTo.frame(string) □ find the id or name of the iframe and pass as a param
- Driver.switchTo.frame(int) □ find the index and pass as a param

62. What is the syntax for switching windows ?

- To handle separate tabs/windows we have to switch to that tab
- Web driver handles one page/html document at a time.
- To control another tab, we always need to switch
- To be able switch we need to get the window handle first using

```
getWindowHandles() method driver.switchTo.window(String) // → window handle
//for each loop : driverGetWIndowHandles:
Driver.switchTo.window("handle")
Ifdriver.getTitle==expectedtitle;
Break;
```

63. What is the syntax for uploading a file?

```
Public void fileUpload(Stirng path){
   WebELement upload = driver.findELement; Upload.sendKeys(path)
}
```

- We need to locate the upload button in html.
- The element will have tag input.
- Then we do sendKeys by passing the path to file which we want to upload

64. Sometimes sendKeys/path does not work

- Building a dynamic path for a file inside our project Path to the project location:

```
String projectDir= System.getProperty("user.dir") // project directory
String file= "src/test/resources/test_data/myfile.txt";
Element.sendKeys(projectDir+file);
```

65. How to input text in the text box without calling the sendKeys()?

```
   //Use                                                                    javascriptExecutor
JavascriptExecutor JS = (JavascriptExecutor)webdriver;
   //To                                    enter                                    username
JS.executeScript("document.getElementById('User').value= 'www.google.com'");
   //To enter password
JS.executeScript("document.getElementById('pass').value=' tester'");
```

66. How to press ENTER key on text box in Selenium WebDriver?

- To press Enter key using Selenium WebDriver,
- We need to use Selenium Enum keys with its constant Enter
- Driver.findElement(By.xpath("xpath")).sendKeys(Keys.ENTER);

67. Have you done any cross-browser testing? cross browser testing

- Always mention that you have a control file for keywords like browser type, main url, username, password, environment.

68. **How you resolve certification issue?**
    - CHROME, IE □ DesiredCapabilities capability = DesiredCapabilities.chrome();
    - on Jenkins we need to insert □ .relaxedHTTPSValidation

```
Response response=RestAssured.given()
                    .contentType(ContentType.JSON)
                            .relaxedHTTPSValidation()
                            .get("https://api.got.show/api/continents");
System.out.println(response.asString());
```

69. **How would you verify the position of the Web Element on the page?**
    - element.getLocation();
    - WebElement class has a get Location method with returns the top left corner of the element

70. **Page Factory class?**
    - Page Factory class comes with Selenium.
    - And it is used whenever we create page object classes.
    - Its purpose is to initialize webElements that were defined in the class.

71. **Explain me your test execution flow with cucumber.**
    - Runner > Feature file > Scenario > Steps > Step def > Selenium code using POM

72. **What tools are you using to test UX and Restful webServices?**
    - UX □ User Experience. First ensure UX is acceptable manually.
    - After that since it is UI testing, I use Selenium WebDriver to automate it.
    - RESTFul API Automation > RestAssured Library, PostMan for manual tests

73. **How To resize browser Window Using Selenium WebDriver?**
    - To resize the browser window to particular dimensions, we use 'Dimension' class to resize the browser window.
    - //Create object of Dimensions class
      Dimension d = new Dimension(480,620);
    - //Resize the current window to the give n
      dimension
      driver.manage().window().setSize(d);

74. **What exceptions do you know in Selenium?**
    - I often have **NoSuchElementException**
    - **StaleElementException**
        - The element has been deleted entirely.
        - The element is no longer attached to the DOM.
        - How we handle StaleElementException;
            - Element is not attached to DOM □ 'try catch block' within 'for loop' Or
            - 1. Refresh the page and try again for the same element.
            - 2. Wait for the element till it gets available
    - **TimeOutException**

75. **ASSERT(hard assert) VS VERIFY(soft assert)**
    - Hard assert throws an AssertException immediately when an assert statement fails, and test suite continues with next @Test. If Assert steps fails, execution of test stops at that point! and will go to next test if present!

- (Example: just simple Assert.assertTrue(boolean);)
- Soft assert collects errors during @Test Soft Assert does not throw an exception when an assert fails and would continue with the next step after the assert statement. If Verify steps fails, it will report a fail but will continue execution!
  - Example: SoftAssert soft=new SoftAssert(); //for soft create object
  - soft.assertTrue(boolean);
  - soft.assertAll(); //put at the end it will report what is failing!

76. What the verification point available in Selenium ?

- In selenium IDE, We use Selenium Verify and Assert Commands as Verification points
- In Selenium WebDriver, There is no built-in features for verification points, it totally depends on our coding style. Some of the Verification points are
  - to check for page title
  - to check for certain text
  - to check for certain element(text box, button, drop down, etc.)

77. Verify text exists?

- VerifyTextPresent returns TRUE if the specified text string was FOUND somewhere in the page; FALSE if otherwise.
- VerifyTextNotPresent returns TRUE if the specified text string was NOT FOUND anywhere in the page; FALSE if it was found.

78. How do you find a text in a webpage?

- //tagname[contains(text(),'text')] contains certain test
- //tagname[.='text'] contains exact text sometimes doesn't work Selenium

79. How to get all the preceding siblings of Apple?

- Xpath: "//ul/li[contains(text(),'Apple Mobiles')]/precedingsibling::li"
- This will give "Samsung Mobiles"

80. How to get all the following siblings of Apple?

- Xpath: "//ul/li[contains(text(),'Apple Mobiles')]/followingsibling::li"
- This will give all the preceding siblings (Nokia Mobiles, HTC Mobiles, Sony Mobiles, Micromax mobiles)

81. How to handle Web Tables/grid?

- Table tag used for table data is arranged in a grid format
  - th tag for column name Example –

```
<tr>
   <th>FirstName</th> column names on the very top row
   <th>Lastname</th>
   <th>Age</th>
</tr>
```

  - </tr> tr tag used to indicate a row, applies to whole column td tag to indicate a column in a row Example

```
<tr>
   <td>Danny</td> actual_data_on_the_very_first_row
   <td>Smith</td>
   <td>29</td>
</tr>
```

- Some tables have tbody Used to indicate the data of the table, usually does not include column names (th)

82. How to use Excel?

```
FileInputStream ExcelFile = new FileInputStream(path);
excelWBook = new XSSFWorkbook(ExcelFile);
excelWSheet =  excelWBook.getSheet(sheetName);
cell = excelWSheet.getRow(rowNum).getCell(colNum);
```

83. How do you like Selenium version 3? Is Selenium 3 drastically different from Selenium 2?

- Selenium 3 has bug fixes from selenium 2 also it is more mobile automation focused.
- We aim for Selenium 3 to be “a tool for user-focused automation of mobile and web apps”.
- Here is the summary of the change.
    - For WebDriver users, it’s more of bug fixes and drop-in replacement for 2.
    - Selenium Grid bug fixes are done as well.
    - Selenium project will not actively support only the WebDriver API.
    - By a quirk of timing, Mozilla have made changes to Firefox that mean that from Firefox 48 you must use their geckodriver to use that browser, regardless of whether you're using Selenium 2 or
    - As we know Selenium 3.0 is the latest version of Selenium Jar

MAVEN

1. **What is Maven?**

- A build tool and command prompt tool that called POM xml file that calls my runner class and manages my dependencies.
- Maven is a build automation tool or a project management tool. With Maven we can import all libraries and can also create project structures. In Maven we have many inbuilt templates. These templates are called archetypes. A Maven is basically a tool used to compile our applications.
- Command Prompt mvn archetype; generate
  - Creates project
- Choose a # press enter
- Choose a # press enter
- GroupId; com.nameOfProject (usually a reversed domain name, like com.example.foo)
- ArtifactID; testmavenproject
  - Version enter
  - Package enter
  - Y; enter

2. **Why Maven? How it helps you developing your project effectively?**
   - It helps to develop and managing project structure or applications like deployment, clean, packaging, jar and many more features for the Java-based project.
   - In another word, it is a Java tool. If you want to create a sample project or skeleton project you can use Maven. It is an automated build tool. The Maven focused on simplicity that it generates intelligent starters and assumes intelligence defaults. It also covers build-oriented phases in Application Lifecycle Management i.e. testing, deployment, builds management, and release versioning.
   - **It helps** to setup project very quickly and it avoids complicated build files like build.xml. Maven required files like POM.xml; it serves the purpose for Maven only. POM.xml is a collection of dependencies of your Java Project which one can specify to Maven and then Maven will download all of them from the internet and then store it to some repository i.e. local repository, central repository, and remote repository.

3. **What is Maven Artifact?**
   - An artifact is a file, usually a JAR, that gets deployed to a Maven repository.
   - A Maven build produces one or more artifacts, such as a compiled JAR and a "sources" JAR.
   - Each artifact has a group ID (usually a reversed domain name, like com.example.foo), an artifact ID (just a name), and a version string. The three together uniquely identify the artifact. Example:

```
<groupId>org.seleniumhq.selenium</groupId>
<artifactId>seleniumjava</artifactId>
<version>3.11.0</version>
```

   - A project's dependencies are specified as artifacts.

4. **Explain me the maven lifecycle?**
   - Commands can only run in the same directory where the specific **pom xml** file is located
   - 3 built in build lifecycles
     - Default Handles your project deployment
     - Clean Handles project cleaning
     - Site Handles creation of project's site documentation

5. **A build lifecycle is made up <u>of phases</u>**
   - <u>Validate</u> □ Validate the project is correct and all necessary information is available
   - <u>Compile</u> □ Run the source code of the project (checking if there is any error or not, if not→ build success)
     □ Target folder is created, and Reports will be stored here
   - <u>Test</u>
     - Test the compiled source code using a suitable unit testing framework.
     - Should not require the code to be packaged or deployed
     - Mvn D(VariableName) = testname □ Run specific tests based on the parameter
   - <u>Package→</u> Take the compiled code and package it in a distributed format, like JAR
   - <u>Verify</u> →Runs any checks on results of integration tests to ensure quality criteria are met
   - <u>Install</u> → Install the package into local repo, for use as dependency
   - <u>Deploy</u> → Done in the build environment, copies the final package to the remote repository for sharing with other devs and projects

6. **How do you convert maven project to eclipse project?**
   - Mvn eclipse

7. **How java projects are made?**
   1. Create folders/packages
   2. Add libraries/dependencies
   3. Create class files
   4. Compile
   5. Run tests
   6. Deploy

8. **Where do you find your dependencies/libraries?**
   - Mvnrepository.com
   - Update project if maven not working
     - When you have dependencies inside your pom file and you use update, maven will pull the JAR files from internet and add it to your project

9. **What is .m2 folder?**
   - Where your jar files/repositories are saved in your computer

10. **What is POM xml file?**
    - A file that manages the whole project
    - When you run a maven command, everything should be done through the pom.xml

11. **Versions of tools?**
    - RestAssured 3.3.0 release date: 2019-01-11

12. **Log4j?**
    - Used by any application
    - Example: LOG4J2 □ From Apache
    - Records activity
    - Dev will look at the logs, look at the time, go to the IP address and see what going on if there was a bug
    - Loggers are very important part of applications and it keeps each step/event happened with timestamp
    - Normally logs are written programmatically into .log file
    - There are ready tools/libraries to add to any framework or application.
    - In java, the most famous logging library/framework is LOG4J from apache

13. **Purpose of logs?**
    - Help us debug the issues that you may have with application.
    - Sometimes when a bug is found in application, developers firstly check the logs. In order to see which steps the user was taking and application did not behave as expected.
    - Logs MAY help you find the source of the problem (in application perspective, not testing)

14. **What is the role of logs in Test automation?**
    - ●We are looking at console or html report to see the status of our test runs. If anything fails, we find from there.
    - ● If we implement logging into our framework, it will be another way of looking at automation execution steps and will help us find the problem whenever our test fails

TESTNG & JUNIT

1. **What is TestNG?**

- You have 500 test cases→We create a Java Package and 500 Class for each test cases
  Client asked you run only 40 of them for smoke test → We handle it in Jasmine with it blocks and reporting mechanism.
- TestNG is a testing framework
- Centralized controller: manage run different test cases then create reports, logs
- Batch execution: 100 test cases and run them one by one
- Optional execution: we can skip some test cases

2. **What is assertions in TestNG?**

- We run the test and title test case failed. It will not affect the other test cases, so we don't want our script to stop.
  - Critical □ stop/failure Assert
    - It takes one boolean argument and String message. It Asserts that a condition is true. If it isn't, an AssertionError, with the given message, is thrown.
  - Non critical □ failure/continue SoftAssert
    - Soft Assert does not throw an exception when an assert fails and would continue with the next step after the assert statement.

3. **Difference between JUnit and TestNG**

- Annotations; **JUnit:** @Test, @BeforeClass, @AfterClass, @Before, @After, @Ignore
  TestNG: @Test, @BeforeTest, @BeforeClass, @BeforeSuite, @BeforeMethod, @AfterTest, @AfterClass, @AfterSuite, @AfterMethod
- Both are testing framework to help us running automation scripts.
- TestNG provide html report
- TestNG has @Dataprovider annotation same as Cucumber Scenario Outline for Data Driven Testing.
- In TestNG, we can do parallel testing, but JUnit doesn't support to parallel test, so we use sauceLab for it.
- TestNG support group test but JUnit doesn't support
- TestNG and JUnit both of them have parameterize testing but TestNG parameterized test configuration is very easy to configure. There are two ways to achieve parameterization in TestNG;
  - @Parameters and TestNG xml file
  - @DataPrivider

FEATURE	JUNIT	TESTNG
Purpose	General unit testing	Focus on Integration testing for Enterprise projects
IDE support	yes	Yes
Maven support	yes	Yes
setup/teardown for test	@Before / @After	@BeforeMethod / @AfterMethod
setup/teardown for class	@Before / @After	@BeforeClass / @AfterClass
setup/teardown for suite	no	@BeforeSuite / @AfterSuite
setup/teardown for test groups	no	@BeforeGroups/ @AfterGroups
setup/teardown for test	in annotations	In annotations and/or XML file
Parameterised tests	Yes, but in a limited way	Yes
Test groups	Yes with Categories (new feature)	Yes
Test for Exceptions	Yes	Yes
Timeouts in tests	Yes	Yes
Test order	Non-Deterministic or alphabetical	Can be defined in detail with dependencies
Dynamic test input	No	Yes with DataProviders
Can run tests of the other library	No	Yes, TestNG can run JUnit tests
Assumptions before running a test	Yes	No
Dependency injection for tests	No	Yes, with Google Guice
Ignore/disable test	Yes	Yes
Parallel testing	No	Yes
Test listeners	No	Yes
Test reporters	No	Yes

4. **Cross Browser and Parallel Test**
   - In my current project, we use sauceLab for cross browser testing. But my previous project I used testng.xml file.
   - Basically, inside the suite there are 3 keys (name, thread count, parallel) and I created 2 different tests, one of them is for
   - Chrome and the other one is for Firefox.
   - There is also parameter annotation and include name and value; name is browser and value is Chrome.

```
<?xml version="1.0" encoding="UTF 8"?>
<!DOCTYPE suite SYSTEM ...>
<suite ...>
   <test name="ChromeTest" ... >
            <parameter name="browser" value="chrome"/>
                     <classes>
                              <class name="testsuite..."/>
                     </classes>
   </test> <!   First Test   >
   <test name="FireFox" ... >
            <parameter name="browser" value="FireFox"/>
                     <classes>
                              <class name="testsuite…."/>
                     </classes>
   </test> <!   Second Test   >
</suite> <!   Suite   >
```

CUCUMBER & GHERKIN

1. **Tell me more about Cucumber, how did you guys decide to start using Cucumber ?**
   - In the past few years, more and more IT teams follow Agile methodology in their development process to adapt to the rapid changes of the market. This is also a challenge for the test team in managing test cases and test scripts which can be changed when the requirements are updated monthly. Finding a suitable testing method from the beginning is one of the keys to the success of an Agile software project.
   - Many Agile teams have successfully applied Behavior Driven Development (or BDD) approach in testing process using the Cucumber tool. So, what is Cucumber? And why is it one of the good approaches in Agile projects, used together with BDD?
   - Cucumber is a tool for running automated acceptance tests written in a behavior driven development style. One of its wonderful main features is the ability to execute plain text functional description (written in language named Gherkin) as automated tests. Here is an example:

```
Feature: Update password
Scenario: Admin user can update user password
Given I am in the HR system with an Admin account
When I update password of another "user"
Then I receive a message for updating password successfully
And user's password is updated to the new password
```

   - This great feature has played a primary role in supporting the BDD approach with the following **advantages**:
     - Writing BDD tests in Ubiquitous language, a language structured around the domain model and used by all team members including developers, testers, BAs, etc.
     - Building bridges between the technical and nontechnical members of a software team
     - Allows interaction directly with the developers' code, but written in a language that business stakeholders can understand
     - Last but not least, Cucumber is an Automated Acceptance Test Tool which running tests written in a behavior driven development (BDD) style.
   - **Cucumber Tool helps to improve communication between technical and non-technical members in a project.**

2. **Tell me what are the most important things in Cucumber, what makes it unique ?**
   - Features file, Step Defs, Runner Classes, Hook Class, Tags

3. **How to see your reports in cucumber?**
   - My framework generates cucumber reports folder in the target folder which contains the reports.
   - When we run the tests on Jenkins, Jenkins saves the report of every run.
   - Home page of the Jenkins job always points to the last run reports.
   - All the reports for previous runs can be found under the build number.
   - Go to target folder
   - Open with system explorer
   - Go to target>cucumber report>index shows the tests you ran

4. **What is Gherkin?**
   - Language used by feature files
   - Feature, Scenario, Given, Then, When, And, But, BackGround, Scenario Outline

5. **What are the components of Cucumber BDD framework?**
    1. **Feature files**
        - Consists of scenarios that test a certain feature or functionality
        - Feature is main story while scenarios are the test cases to the story(feature)
    2. **Cukes Runner**
        - A class that strictly runs the tests, generates codes for step definition
        - @smoketest
        - Cukesrunner → IN CUCKESRUNNER I HAVE A FEATURE LOCATION THAT SHOWS WHERE MY FEATURE ARE LOCATED
    3. **Step definition**
        - A class that made of steps that starts with Gherkin language
        - Make sure the step definition is in the same package as cukes Runner, or child package (not parent or sibling)
- FOR NON-TECH PPL TO UNDERSTAND
- DEPENDECY BDD IS A DEPEN
- MVN REPOSITORY IN THE POM.XML FILE
- CUCUMBER BDD FROM CUCUMBER.IO
- Combine techs of TDD
- Behavior driven
- Express the flow customer behavior → Don't focus on the elements

6. **What does @CucumberOptions do?**
    - Tag used to customize the running of the cucumber tests
    - Inside @CucumberOptions you can add:
        - dryRun
        - Plugin
            - "**Pretty**"
                - Adds more info in the console □ Gives you tag, scenario, method info.
                - "html:target/cucumber report" □ Generates html report located in target/cucumber report folder "json:target/cucumber.json
            - **Tags**
                - Tags must be located in feature path
                - Can add multiple tags...tags= "@Dog, @Cat"
            - **Features** location of where feature files are
            - **Glue** where to look for step definition steps. hook class is part of glue too.

7. **How to run Cucumber with JUnit?**
    - Add cucumber JUnit dependency
    - Adding @RunWith (Cucumber.class) on top of cukesRunner class

8. **How to run Cucumber with TestNG?**
    - Add cucumber testNG dependency
    - Make CukesRunner extend to AbstractTestNG CucumberTests

9. **What happens we you run your runner class with no tags?**
    - All the feature files will run from top to bottom but only the feature files that are located in the @CucumberOptions "features="

10. **What are Hooks in cucumber?**
    - Cucumber hook allows us to better manage the code workflow and helps us to reduce the code redundancy. We can say that it is an unseen step, which allows us to perform our scenarios or tests.
    - Class that uses
        - @Before → runs before each cucumber scenario
        - @After →runs after each scenario (It will always run no matter if scenario passes or fails)
    - Class must be in same package as stepDefinition
    - I implemented screenshots inside hook class
    - Hook Class will not run if dryRun=true
    - I use Scenario as a parameter in my before/after method

11. **How do you take screenshots in cucumber?**
    - In my Aftermethod I use a code:
    - I use TakeScreenShot interface
    - You can store screenshot as a byte or file
        - @After

```
public void tearDown(Scenario scenario) {
   if(scenario.isFailed()) {
   //taking a screenshot
 final byte[] screenshot = ((TakesScreenshot)
                              Driver.getDriver()).getScreenshotAs(OutputType.BYTES);
 //adding the screenshot to the report
 scenario.embed(screenshot, "image/png"); }
```

12. **How to run a Cucumber with DDT?**
    - I use Cucumber tables:
      | Home | Emails | Documents | Projects |
    - You get the method with (DataTable arg1)
    - In the parameter DataTable you can change it to
      List<YourType>, List<List<E>>, List<Map<K,V>>, and Map<K,V>
    - Prints in order for list
    - No order for map

13. **What is Background?**
    - Cucumber has their own before method
    - The one in hooks is for java
    - A step that runs BEFORE a scenario inside the feature file
    - Can only put on top, before all scenarios
    - Cannot put pipelines in backgrounds (Only in scenario outline)

14. **What is Scenario Outline? vs Scenario?**
    - Scenario in cucumber runs once.
    - Used for data driven testing
    - Have the same cucumber steps but we provide data after the scenario as a table using keyword examples

15. **How do I limit the types of variables I can pass?**
    - In the gherkin parenthesis you can add (Collaboration | Sales | Marking, etc.)
    - Ex: @When("^I hover over the (Collaboration | Sales | Marketing | Activities | All) menu$")

```
public void i_hover_over_the_Collaboration_menu(String menu) {
switch(menu) {
case "Sales":
   BrowserUtils.hover(dashboard.sales); break;
case "Marketing":
   BrowserUtils.hover(dashboard.marketing);  break;
case "Collaboration":
   BrowserUtils.hover(dashboard.collaboration);  break;
case "Activities":
   BrowserUtils.hover(dashboard.activities);  break;
case  "All":
   BrowserUtils.hover(dashboard.all); break;};
```

16. **What if you have a scenario that has two parameters (limiting parameter, table parameter)?**
    - Example :
        - Scenario: Verify Collaboration menu options
        - Given I logged into suiteCRM
        - When I hover over the Collaboration menu
        - Then the following menu options should be visible for Collaboration:
          | Home | Emails | Documents | Projects |
        - In this scenario i have a table, I want to limit collaboration to just collaboration and the other menus categories
    - Solution:
        - @Then("^following menu options should be visible for
          (Collaboration | Sales | Marketing | Activities | All):$")
        - public void following_menu_options_should_be_visisble_for_Collaboration(String menu, List<String> options) {
        - String menu represents the 5 menu options (Collaboration | Sales | Marketing | Activities |
          All) List<String>options represents the tables; | Home | Emails | Documents | Projects |

17. **How do I use cucumber scenario for DDT?**
    - In my current project I use Scenario Outline with Examples
    - In my scenario feature file, whenever I'm using a variable as a data driven, I use "<variable>"
    - Then in Examples:
      | variable | column name
      | data1. | row1
      | data 2 | row 2
      | data3 | row3

20. **Data driven**
    - Test data is separated from code and stored into external sources: Cucumber Examples table, Excel files, CSV files, Database.
    - If the amount of data is not that huge, then I use Cucumber Scenario outline with Examples table.
    - And other times I maintain test data in Excel files, and I use Apache POI library to read and write data
    - If data comes from a database, or I need to do database validation, I use SQL queries along with JDBC library in java.

18. **How to use Maps in cucumber?**
    - Using a nonScenario Outline
    - Scenario: Create contact using a map
        - Given I logged into suiteCRM
        - When I create a new contact:
          | first_name | John |
          | last_name | Smith |
          | cell_phone | 801 888 8889 |
        - Then I should see contact information for "John Smith"
        - Left side is key, and right is value 2 columns only
    - Using a Scenario Outline
        - Scenario Outline: Create contact using a map
        - Given I logged into suiteCRM
        - When I create a new contact:
          | first_name | <first_name> |
          | last_name | <lname> |
          | cell_phone | <cell_phone> |
          | office_phone |<office_phone> |
        - Then I should see contact information for "<first_name> <lname>"
        - Examples: **| first_name | lname | cell_phone | office_phone |**
          | Michael | Jackson | 1234567890 | 2345678891 |
          | Bonnie | Garcia | 4569871234 | 4567890987 |
    - In step def I write;

```
@When("^I create a new contact:$")
 public void i_create_a_new_contact(Map<String,String>contact) {
 // open the create contact dialog
```

21. **How to use POJO in cucumber?**
    - Create **contactBean** class
        - Add all variables
        - Add the getter/setters
    - Create bean feature file
    - Create a table with first row containing the variables in the contactBean class
        - Add values under the table
        - Implement method with parameter (List<ContactBean>contacts)
    - Scenario: Create contact
        - Given I logged into suiteCRM
        - When I save a new contact:
          | **firstName** | **lastName** | **officePhone** | **cellphone** | **email**|
          | Steve | Gates | 3456758888 | 1234329999 | SteveGates123@gmail.com |
        - Then I should see contact information for "Steve Gates"

22. **How to run a group of test case using TestNG?**

```
@Test (groups={“smokeTest”,”FunctionalTest”})
public void loginTest(){
            System.out.println(“Logged in successfully”);
}
```

23. Data Driven Testing

- ***WHEN:*** Whenever a functionality or a module in an app requires testing with multiple sets of data(Parametrization), Multiple inputs then we need to perform data driven testing and automation.
- These scenarios are one of the things That must be automated.
- ***HOW:*** Test data is separated from code and stored into external sources: Cucumber Examples table, Excel files, CSV files, Database.
- ***BENEFIT:*** More organized, Data centralized, Collaboration on test data - it can come from BA, MTs etc

24. How can we create data driven framework using TestNG?

- By using @DataProvider annotation, we can create a Data Driven Framework

```
@DataProvider(name=”getData”) Public Object[][] getData(){ Object [][] data = new Object[2][2];
Data[0][0] = “firstUid”; Data[0][1] = “FirstPWD”;
Data[1][0] = “SecondUid”;
Data[1][1] = “SecondPWD”; Return data; }
```

25. How to create Group of Groups in TestNG?

- These groups are called metagroups.
- Example: you might want to define a group all that includes smokeTest and FunctionalTest. Let’s modify our testing.xmlfile:

```
<groups>
   <define name=”all”>
            <include name =”smoke Test”/>
            <include name = “functionalTest”/>
   </define>
   <run>
            <include name = “all”/>
   </run>
</groups>
```

26. How to run test cases in parallel using TestNG?

- We can use “parallel” attribute in testng.xml to accomplish parallel test execution in TestNG
- The parallel attribute of suite tag can accept four values:
    - Classes → All the test cases inside a java class will run parallel
    - Methods → All the methods with @Test annotation will execute parallel
    - Instances → Test cases in same instance will execute parallel but two method of two different instances will run in different thread. <suite name=”softwaretestingmaterial” parallel=”methods”>

27. How to ignore a test case in testNG?

- To ignore the test case, we use the parameter enabled = false to the
- @Test annotation @Test(enabled=false)

28. How to exclude a particular test method from a test case execution?

- By adding the exclude tag in the testing.xml

```
<classes>
   <class name="TestCaseName">
      <methods>
         <exclude name="TestMethodNameToExclude"/>
      </methods>
   </class>
</classes>
```

29. **How to exclude a particular test group from a test case execution?**
    - By adding the exclude tag in the testing.xml

```
<groups>
  <run>
     <exclude name="TestGroupNameToExclude"/>
  </run>
</groups>
```

30. **What are the different way to produce reports for TestNG results?**
    - TestNG offers two ways to produce a report
        - Listeners implement the interface **org.testng.testListener** and are notified in real time of when a test starts, passes, fails, etc…
        - Reporters implement the interface **org.testng.reporter** and are notified when all the suites have been run by TestNG.
    - The IReporter instance receives a list of objects that describe the entire test run

31. **What is the use of @Listener annotation in TestNG?**
    - configure reports and logging.
    - widely used listeners : ITestListener interface.
    - It has methods like onTestStart, onTestSucess. onTestFailure, on TestSkipped...
    - we should implement this interface creating a listener class of our own,
    - Next, we should add the listeners annotation (@Listeners) in the class

32. **What Is a Regular Expression, Regexp, or Regex?**
    - A regular expression is a special text string for describing a search pattern.
    - You can think of regular expressions as wildcards on steroids.
    - You are probably familiar with wildcard notations such as *.txt to find all text files in a file manager.
    - Regex equivalent is.*\.txt.

33. **How to write regular expression in testing.xml file to search @Test methods containing ”smoke” keyword?**
    - Regular expression to find @Test method containing keyword “smoke” is a mentioned below

```
<methods>
   <include name=".*smoke.*"/>
</methods>
```

34. **What is the time unit we specify in test suites and test cases ?**
    - We specify the time unit in test suites and test cases is in milliseconds.

35. **What is the use of @Test(invocationCount= someInteger)?**

```
@Test(invocationCount=10)
Public void testcase(){}
```

    - //the invocation count attribute tells how many times TestNG should run a test method

36. **What is the use of @Test(threadPoolSize=someInteger)?**
    - The threadPoolSize attribute tells to from a thread pool to run the test method through multiple threads
    - Note: this attribute is ignored if invocation count IS NOT SPECIFIED

37. **What does the test timeout mean in testing?**
    - The maximum number of milliseconds a test case should take

```
@Test1(threadPoolSize=3,invocationCount=10,timeOut=10000)
public void test() {}
```

    - : // in this example: the function test1 will be invoked ten times from three different threads, Additionally, a time-out often seconds guarantees that none of the threads will block on this thread forever.

38. **What are @Factory and @DataProvider annotation?**
    - @Factory → executes all the test methods present inside a test class using a separate instance of the class with different set of data
    - @DataProvider → a test method that uses dataProvider will be executed the specific methods multiple number of times based on the data provided by the dataProvider.

39. **annotations - priority**
    - Doesn't matter what number you start Ex: @Test(priority=0)
    - DependsOnMethods = “test method name” You Can add multiple test names
    - If the first one fails, the 2nd test won't run at all
    - If the first method failed, your report will show that the 2nd test will be skipped

40. **parallel execution in testNG**
    - In xml file write.
        - parallel=”tests”thread-count=”4”
    - Thread-count is how many browsers you want to open same time
    - In xml file you can add .* to run everything
        - Ex:<package name=”.*”></package>
    - TestNG has its own reports -When you run xml, it gives you the report in test-output folder
    - Contains the test report in html

41. **Framework Tools : Cucumber BDD framework**
    - Junit, Cucumber Java, Maven
    - Selenium, HTML reporting with screenshots Log4j,
    - JDBC, Rest Assured, Apache POI, Git, Jenkins

42. **Framework Tools: TestNG + Selenium**
    - Java, Maven, TestNG,
    - Selenium, Extend Reports with screenshots Log4J,
    - JDBC, Rest Assured, Apache POI, Git, Jenkins

43. **How does your framework generate reports?**
    - Our Cucumber BDD framework generates HTML reports.
    - The report shows the pass/fail coverage for feature files, tags, steps
    - The report contains all the steps for each test The report has screenshots for failures

44. **How to run tests selectively cucumber?**
    - tags keyword the cukesrunner
    - feature keyword the cukesrunner
    - tags and features can also be passed using the command line
    - mvn test -Dcucumber.options="--tag @smoke”

45. **What do you use for logging?**
    - I use Log4J for logging. I always log important steps in the test execution. That helps me to debug when there is a failure.
    - Log4J is not a replacement for HTML reports.

```
<dependency>
   <groupId>org.apache.logging.log4j</groupId>
   <artifactId>log4j- core</artifactId>
   <version>2.11.0</version>
</dependency>
```

46. **How does the FEATURE FILE WORK?**
    - **Feature** →description of what is being tested @tags. Sample feature file;
        - Feature: login functionality → Background:
        - Given I am on the login page → Scenario: 1, Scenario: 2
        - The background runs before both of the scenarios
    - **Scenario** → description of the scenario being test
        - Given I am on the login page
        - And I enter username and password
        - When I click on the submit button
        - Then I should be able to see the profile picture
        - But the submit button should not be displayed
    - **Given** → a precondition
    - **When** →condition that triggers the expected result Then —> expected condition

47. **What is test base Class ? and How do you implement in your framework ?**
    - Test Base class is class where I have most used methods in my tests.
    - My test classes extend the Test Base class and thus have access to those methods. This helps me us make my code reusable
    - Before/after test methods wait/synchronization utility methods.
        - SwitchToWindow(title)
        - WebDriver driver;

48. **How to rerun the failed tests again in TestNG?**
    - In my TestNG framework, failed tests are reported in the testng_failed_.xml file in the target folder.
    - We can add this file in the pom file so that maven will try to run the failed tests every time.
    - If will only run when there are failures in the test.

49. **How to rerun the failed tests again in Cucumber?**
    - we use the re-run option in the CukesRunner.
    - Add the rerun to cukes runner.
    - This option will create a file with a list of failed tests
    - Create a second runner class which points to file with a list of failed tests
    - Add the second runner in the pom file

50. **How to rerun the failed tests again in Jenkins?**
    - In Jenkins there are plugin that re run the failed tests Unit cases.
    - So you can configure your Maven build execution on Jenkins using the option:
        - Dsurefire.rerunFailingTestsCount=2