1) What is Automation Testing?

Automation testing or Test Automation is a process of automating the manual process to test the application/system under test. Automation testing involves use to a separate testing tool which lets you create test scripts which can be executed repeatedly and doesn't require any manual intervention.

2) What are the benefits of Automation Testing?

Benefits of Automation testing are:

1. Supports execution of repeated test cases
2. Aids in testing a large test matrix
3. Enables parallel execution
4. Encourages unattended execution
5. Improves accuracy thereby reducing human generated errors
6. Saves time and money

3) Why should Selenium be selected as a test tool?

Selenium

1. is free and open source
2. have a large user base and helping communities
3. Have cross Browser compatibility (Firefox, chrome, Internet Explorer, Safari etc.)
4. Have great platform compatibility (Windows, Mac OS, Linux etc.)
5. Supports multiple programming languages (Java, C#, Ruby, Python, Pearl etc.)
6. has fresh and regular repository developments
7. supports distributed testing

4) What is Selenium? What are the different Selenium components?

Selenium is one of the most popular automated testing suites. Selenium is designed in a way to support and encourage automation testing of functional aspects of web based applications and a wide range of browsers and platforms. Due to its existence in the open source community, it has become one of the most accepted tools amongst the testing professionals.

Selenium is not just a single tool or a utility, rather a package of several testing tools and for the same reason it is referred to as a Suite. Each of these tools is designed to cater different testing and test environment requirements.

The suite package constitutes of the following sets of tools:

- **Selenium Integrated Development Environment (IDE)** – Selenium IDE is a record and playback tool. It is distributed as a Firefox Plug-in.
- **Selenium Remote Control (RC)** – Selenium RC is a server that allows user to create test scripts in a desired programming language. It also allows executing test scripts within the large spectrum of browsers.
- **Selenium WebDriver** – WebDriver is a different tool altogether that has various advantages over Selenium RC. WebDriver directly communicates with the web browser and uses its native compatibility to automate.
- **Selenium Grid** – Selenium Grid is used to distribute your test execution on multiple platforms and environments concurrently.

5) What are the testing types that can be supported by Selenium?

Selenium supports the following types of testing:

1. Functional Testing
2. Regression Testing

6) What are the limitations of Selenium?

Following are the limitations of Selenium:

- Selenium supports testing of only web based applications
- Mobile applications cannot be tested using Selenium
- Captcha and Bar code readers cannot be tested using Selenium
- Reports can only be generated using third party tools like TestNG or Junit.

- As Selenium is a free tool, thus there is no ready vendor support though the user can find numerous helping communities.
- User is expected to possess prior programming language knowledge.

7) What is the difference between Selenium IDE, Selenium RC and WebDriver?

Feature	Selenium IDE	Selenium RC	WebDriver
Browser Compatibility	Selenium IDE comes as a Firefox plug-in, thus it supports only Firefox	Selenium RC supports a varied range of versions of Mozilla Firefox, Google Chrome, Internet Explorer and Opera	WebDriver supports a varied range of versions of Mozilla Firefox, Google Chrome, Internet Explorer and Opera. Also supports HtmlUnitDriver.
Record and Playback	Selenium IDE supports record and playback feature	Selenium RC doesn't supports record and playback feature	WebDriver doesn't support record and playback feature
Server Requirement	Selenium IDE doesn't require any server to be started	Selenium RC requires server to be started before executing the	WebDriver doesn't require any server to be started before

Feature	Selenium IDE	Selenium RC	WebDriver
	before executing the test scripts	test scripts	executing the test scripts
Architecture	Selenium IDE is a Java script based framework	Selenium RC is a JavaScript based Framework	WebDriver uses the browser's native compatibility to automation
Object Oriented	Selenium IDE is not an object oriented tool	Selenium RC is semi object oriented tool	WebDriver is a purely object oriented tool
Dynamic Finders (for locating web elements on a webpage)	Selenium IDE doesn't support dynamic finders	Selenium RC doesn't support dynamic finders	WebDriver supports dynamic finders
Handling Alerts, Navigations, Dropdowns	Selenium IDE doesn't explicitly provides aids to handle alerts, navigations, dropdowns	Selenium RC doesn't explicitly provides aids to handle alerts, navigations, dropdowns	WebDriver offers a wide range of utilities and classes that helps in handling alerts, navigations, and dropdowns efficiently and effectively.

Feature	Selenium IDE	Selenium RC	WebDriver
WAP (I Phone/Android) Testing	Selenium IDE doesn't support testing of I Phone/ Andriod applications	Selenium RC doesn't support testing of iPhone/Android applications	WebDriver is designed in a way to efficiently support testing of iPhone/Android applications. The tool comes with a large range of drivers for WAP based testing. For example, AndroidDriver, iPhoneDriver
Listener Support	Selenium IDE doesn't support listeners	Selenium RC doesn't support listeners	WebDriver supports the implementation of Listeners
Speed	Selenium IDE is fast as it is plugged in with the web -browser that launches the test. Thus, the IDE and browser communicat	Selenium RC is slower than WebDriver as it doesn't communicates directly with the browser; rather it sends selenese commands over to	WebDriver communicates directly with the web browsers. Thus making it much faster.

Feature	Selenium IDE	Selenium RC	WebDriver
	es directly	Selenium Core which in turn communicates with the browser.	

8) When should I use Selenium IDE?

Selenium IDE is the simplest and easiest of all the tools within the Selenium Package. Its record and playback feature makes it exceptionally easy to learn with minimal acquaintances to any programming language. Selenium IDE is an ideal tool for a naïve user.

9) What is Selenese?

Selenese is the language which is used to write test scripts in Selenium IDE.

10) What are the different types of locators in Selenium?

Locator can be termed as an address that identifies a web element uniquely within the webpage. Thus, to identify web elements accurately and precisely we have different types of locators in Selenium:

- ID
- Class Name
- Name
- Tag Name
- Link Text
- Partial Link Text
- X path
- CSS Selector
- DOM

11) What is difference between assert and verify commands?

Assert: Assert command checks whether the given condition is true or false. Let's say we assert whether the given element is present on the web page or not. If the condition is true then the program control will execute the next test step but if the condition is false, the execution would stop and no further test would be executed.

Verify: Verify command also checks whether the given condition is true or false. Irrespective of the condition being true or false, the program execution doesn't halts i.e. any failure during verification would not stop the execution and all the test steps would be executed.

12) What is an X path?

X path is used to locate a web element based on its XML path. XML stands for Extensible Markup Language and is used to store, organize and transport arbitrary data. It stores data in a key-value pair which is very much similar to HTML tags. Both being markup languages and since they fall under the same umbrella, X path can be used to locate HTML elements.

The fundamental behind locating elements using X path is the traversing between various elements across the entire page and thus enabling a user to find an element with the reference of another element.

13) What is the difference between "/" and "//" in X path?

Single Slash "/" – Single slash is used to create X path with absolute path i.e. the x path would be created to start selection from the document node/start node.

Double Slash "//" – Double slash is used to create X path with relative path i.e. the x path would be created to start selection from anywhere within the document.

14) What is same origin policy and how it can be handled?

The problem of same origin policy disallows to access the DOM of a document from an origin that is different from the origin we are trying to access the document.

Origin is a sequential combination of scheme, host and port of the URL. For example, for a URL http:// http://www.softwaretestinghelp.com/resources/, the origin is a combination of http, softwaretestinghelp.com, 80 correspondingly.

Thus the Selenium Core (JavaScript Program) cannot access the elements from an origin that is different from where it was launched. For Example, if I have launched the JavaScript Program from “http://www.softwaretestinghelp.com”, then I would be able to access the pages within the same domain such as “http://www.softwaretestinghelp.com/resources” or “http://www.softwaretestinghelp.com/istqb-free-updates/”. The other domains like google.com, seleniumhq.org would no more be accessible.

So, In order to handle same origin policy, Selenium Remote Control was introduced.

15) When should I use Selenium Grid?

Selenium Grid can be used to execute same or different test scripts on multiple platforms and browsers concurrently so as to achieve distributed test execution, testing under different environments and saving execution time remarkably.

16) What do we mean by Selenium 1 and Selenium 2?

Selenium RC and WebDriver, in a combination are popularly known as Selenium 2. Selenium RC alone is also referred as Selenium 1.

17) Which is the latest Selenium tool?

WebDriver

18) How do I launch the browser using WebDriver?

The following syntax can be used to launch Browser:

```
WebDriver driver = new Firefox Driver();
WebDriver driver = new Chrome Driver();
```

*WebDriver driver = **new** InternetExplorerDriver();*

19) What are the different types of Drivers available in WebDriver?

The different drivers available in WebDriver are:

- Firefox Driver
- InternetExplorerDriver
- Chrome Driver
- Safari Driver
- Opera Driver
- Android Driver
- Iphone Driver
- HtmlUnitDriver

20) What are the different types of waits available in WebDriver?

There are two types of waits available in WebDriver:

1. Implicit Wait
2. Explicit Wait

Implicit Wait: Implicit waits are used to provide a default waiting time (say 30 seconds) between each consecutive test step/command across the entire test script. Thus, subsequent test step would only execute when the 30 seconds have elapsed after executing the previous test step/command.

Explicit Wait: Explicit waits are used to halt the execution till the time a particular condition is met or the maximum time has elapsed. Unlike Implicit waits, explicit waits are applied for a particular instance only.

21) How to type in a textbox using Selenium?

User can use send Keys (“String to be entered”) to enter the string in the textbox.

Syntax:
Web Element username = drv.findElement(By.id(“Email”));

// entering username
username.sendKeys(“sty”);

22) How can you find if an element in displayed on the screen?

WebDriver facilitates the user with the following methods to check the visibility of the web elements. These web elements can be buttons, drop boxes, checkboxes, radio buttons, labels etc.

1. is Displayed()
2. is Selected()
3. is Enabled()

Syntax:

Is Displayed ():
***Boolean** buttonPresence = driver.findElement (By.id (“gbqfba”)).is Displayed();*

Is Selected ():
***Boolean** buttonSelected = driver.findElement(By.id(“gbqfba”)).is Displayed();*

Is Enabled ():
***Boolean** searchIconEnabled = driver.findElement(By.id(“gbqfb”)).is Enabled();*

23) How can we get a text of a web element?

Get command is used to retrieve the inner text of the specified web element. The command doesn’t require any parameter but returns a string value. It is also one of the extensively used commands for verification of messages, labels, errors etc displayed on the web pages.

Syntax:
String Text = driver.findElement(By.id(“Text”)).get Text();

24) How to select value in a dropdown?

Value in the drop down can be selected using Web Driver’s Select class.

Syntax:

SelectByValue:
*Select selectByValue = **new** Select(driver.findElement(By.id(“SelectID_One”)));*
selectByValue.selectByValue(“Greenvale”);

SelectByVisibleText:
*Select selectByVisibleText = **new** Select*
(driver.findElement(By.id(“SelectID_Two”)));
selectByVisibleText. SelectByVisibleText (“Lime”);

SelectByIndex:
*Select selectByIndex = **new** Select(driver.findElement(By.id(“SelectID_Three”)));*
selectByIndex.selectByIndex(2);

25) What are the different types of navigation commands?

Following are the navigation commands:
navigate().back() – The above command requires no parameters and takes back the user to the previous webpage in the web browser’s history.

Sample code:
driver. Navigate ().back ();

Navigate ().forward () – This command lets the user to navigate to the next web page with reference to the browser’s history.

Sample code:
driver. Navigate ().forward ();

Navigate ().refresh () – This command lets the user to refresh the current web page there by reloading all the web elements.

Sample code:
driver. Navigate ().refresh ();

Navigate ().to () – This command lets the user to launch a new web browser window and navigate to the specified URL.

Sample code:
driver. Navigate ().to ("https://google.com");

26) How to click on a hyper link using link Text?

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

The command finds the element using link text and then click on that element and thus the user would be re-directed to the corresponding page.

The above mentioned link can also be accessed by using the following command.

driver.findElement (By.partialLinkText ("Goo")).click ();

The above command find the element based on the substring of the link provided in the parenthesis and thus partialLinkText () finds the web element with the specified substring and then clicks on it.

27) How to handle frame in WebDriver?

An inline frame acronym as I frame is used to insert another document with in the current HTML document or simply a web page into a web page by enabling nesting.

Select I frame by id
driver.switchTo ().frame ("ID of the frame");

Locating I frame using tag Name
driver.switchTo ().frame (driver.findElement (By.tagName ("frame").get (0)) ;

Locating iframe using index

Frame (index)
driver.switchTo().frame(0);

Frame (Name of Frame)
driver.switchTo().frame("name of the frame");

Frame (Web Element element)
Select Parent Window
driver.switchTo().default Content();

28) When do we use find Element() and find Elements()?

Find Element (): find Element () is used to find the first element in the current web page matching to the specified locator value. Take a note that only first matching element would be fetched.

Syntax:

Web Element element = driver.findElements (By.xpath ("//div [@id='example']//ul//li"));
find Elements(): find Elements() is used to find all the elements in the current web page matching to the specified locator value. Take a note that all the matching elements would be fetched and stored in the list of Web Elements.

Syntax:
List <Web Element> element List =
driver.findElements(By.xpath("//div[@id='example']//ul//li"));

29) How to find more than one web element in the list?

At times, we may come across elements of same type like multiple hyperlinks, images etc arranged in an ordered or unordered list. Thus, it makes absolute sense to deal with such elements by a single piece of code and this can be done using Web Element List.

<u>Syntax:</u>

List<Web element> options=driver.findelements (by.tagname ("a"));

String text=options.gettext ();

30) What is the difference between driver. Close () and driver. Quit() command?

Close(): Web Driver’s close() method closes the web browser window that the user is currently working on or we can also say the window that is being currently accessed by the WebDriver. The command neither requires any parameter nor does is return any value.

Quit(): Unlike close() method, quit() method closes down all the windows that the program has opened. Same as close() method, the command neither requires any parameter nor does is return any value.

31) Can Selenium handle windows based pop up?

Selenium is an automation testing tool which supports only web application testing. Therefore, windows pop up cannot be handled using Selenium.

32) How can we handle web based pop up?

WebDriver offers the users with a very efficient way to handle these pop ups using Alert interface. There are the four methods that we would be using along with the Alert interface.

- Void dismiss() – The accept() method clicks on the “Cancel” button as soon as the pop up window appears.
- Void accept() – The accept() method clicks on the “Ok” button as soon as the pop up window appears.
- String get Text() – The get Text() method returns the text displayed on the alert box.
- Void send Keys(String stringToSend) – The send Keys() method enters the specified string pattern into the alert box.

Syntax:

```
// accepting java script alert
          Alert alert = driver.switchTo().alert();
```

```
alert. Accept();
```

33) How can we handle windows based pop up?

Selenium is an automation testing tool which supports only web application testing, that means, it doesn’t support testing of windows based applications. However Selenium alone can’t help the situation but along with some third party intervention, this problem can be overcome. There are several third party tools available for handling window based pop ups along with the selenium like Auto IT, Robot class etc.

34) How to assert title of the web page?

```
//verify the title of the web page
assert True(“The title of the window is incorrect.”,driver.getTitle().equals(“Title of
the page”));
```

35) How to mouse hover on a web element using WebDriver?

WebDriver offers a wide range of interaction utilities that the user can exploit to automate mouse and keyboard events. Action Interface is one such utility which simulates the single user interactions.

Thus, in the following scenario, we have used Action Interface to mouse hover on a drop down which then opens a list of options.

Sample Code:

```
   // Instantiating Action Interface
   Actions actions=new
 Actions(driver);
   // hovering on the dropdown
 actions.moveToElement(driver.findElement(By.id("id of the
 dropdown"))).perform();
  // Clicking on one of the items in the list options
  Web Element subLinkOption=driver.findElement(By.id("id of the sub
 link"));
 subLinkOption.click(
 );
```

36) How to retrieve css properties of an element?

The values of the css properties can be retrieved using a get() method:

Syntax:

```
driver.findElement(By.id(“id“)).getCssValue(“name of css attribute”);
driver.findElement(By.id(“id“)).getCssValue(“font-size”);
```

37) How to capture screenshot in WebDriver?

```
//Time Stamp using java

DateFormat df=new SimpleDateFormat("yyyy_MMM_dd hh_mm_ss");

Date d=new Date();

String time=df.format(d);



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

FileUtils.copyFile(src,
newFile("E:\\selenium_Workspace_Week20\\screens\\facebook"+time+".p
ng"));
```

```
} catch (Exception e)

{

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

}
```

38) What is Junit?

Junit is a unit testing framework introduced by Apache. J unit is based on Java.

39) What are J unit annotations?

Following are the J unit Annotations:

- **@Test:** Annotation lets the system know that the method annotated as @Test is a test method. There can be multiple test methods in a single test script.
- **@Before:** Method annotated as @Before lets the system know that this method shall be executed every time before each of the test method.
- **@After:** Method annotated as @After lets the system know that this method shall be executed every time after each of the test method.
- **@Before Class:** Method annotated as @Before Class lets the system know that this method shall be executed once before any of the test method.
- **@After Class:** Method annotated as @After Class lets the system know that this method shall be executed once after any of the test method.
- **@Ignore:** Method annotated as @Ignore lets the system know that this method shall not be executed.

40) What is TestNG and how is it better than J unit?

TestNG is an advance framework designed in a way to leverage the benefits by both the developers and testers. With the commencement of the frameworks, J Unit gained an enormous popularity across the Java applications, Java developers and Java testers with remarkably increasing the code quality. Despite being easy to use and straightforward, J Unit has its own limitations which give rise to the need of bringing TestNG into the picture. TestNG is an open source framework which is distributed under the Apache software License and is readily available for download.

TestNG with WebDriver provides an efficient and effective test result format that can in turn be shared with the stake holders to have a glimpse on the product's/application's health thereby eliminating the drawback of Web Driver's incapability to generate test reports. TestNG has an inbuilt exception handling mechanism which lets the program to run without terminating unexpectedly.

There are various advantages that make TestNG superior to JUnit. Some of them are:

- Added advance and easy annotations
- Execution patterns can set
- Concurrent execution of test scripts
- Test case dependencies can be set

41) How to set test case priority in TestNG?

Setting Priority in TestNG

Code Snippet

```
 @Test(priority=0)
      public void method1()
 {
      }
     @Test(priority=1)
     public void method2()
{
      }

     @Test(priority=2)
```

```
    public void method3()
{
    }
}
```

Test Execution Sequence:

1. Method1
2. Method2
3. Method3

42) What is a framework?

Framework is a constructive blend of various guidelines, coding standards, concepts, processes, practices, project hierarchies, modularity, reporting mechanism, test data injections etc. to pillar automation testing.

43) What are the advantages of Automation framework?

Advantage of Test Automation framework

- Reusability of code
- Maximum coverage
- Recovery scenario
- Low cost maintenance
- Minimal manual intervention
- Easy Reporting

44) What are the different types of frameworks?

Below are the different types of frameworks:

1. **Module Based Testing Framework:** The framework divides the entire “Application under Test” into number of logical and isolated modules. For each module, we create a separate and independent test script. Thus, when these test scripts taken together builds a larger test script representing more than one module.

2. **Library Architecture Testing Framework:** The basic fundamental behind the framework is to determine the common steps and group them into functions under a library and call those functions in the test scripts whenever required.
3. Data Driven Testing Framework: Data Driven Testing Framework helps the user segregate the test script logic and the test data from each other. It lets the user store the test data into an external database. The data is conventionally stored in “Key-Value” pairs. Thus, the key can be used to access and populate the data within the test scripts.
4. **Keyword Driven Testing Framework:** The Keyword driven testing framework is an extension to Data driven Testing Framework in a sense that it not only segregates the test data from the scripts, it also keeps the certain set of code belonging to the test script into an external data file.
5. **Hybrid Testing Framework:** Hybrid Testing Framework is a combination of more than one above mentioned frameworks. The best thing about such a setup is that it leverages the benefits of all kinds of associated frameworks.
6. **Behavior Driven Development Framework:** Behavior Driven Development framework allows automation of functional validations in easily readable and understandable format to Business Analysts, Developers, Testers, etc.

45) How can I read test data from excels?

Test data can efficiently be read from excel using JXL or POI API. See detailed tutorial here.

46) What is the difference between POI and jxl jar?

	JXL jar	**POI jar**
1	JXL supports “.xls” format i.e. binary based format. JXL doesn’t support Excel 2007 and “.xlsx” format i.e. XML based format	POI jar supports all of these formats
2	JXL API was last updated in the year 2009	POI is regularly updated and released
3	The JXL documentation is not as comprehensive as that of POI	POI has a well prepared and highly comprehensive

	JXL jar	POI jar
		documentation
4	JXL API doesn’t support rich text formatting	POI API supports rich text formatting
5	JXL API is faster than POI API	POI API is slower than JXL API

47) What is the difference between Selenium and QTP?

Feature	Selenium	Quick Test Professional (QTP)
Browser Compatibility	Selenium supports almost all the popular browsers like Firefox, Chrome, Safari, Internet Explorer, Opera etc	QTP supports Internet Explorer, Firefox and Chrome. QTP only supports Windows Operating System
Distribution	Selenium is distributed as an open source tool and is freely available	QTP is distributed as a licensed tool and is commercialized
Application under Test	Selenium supports testing of only web based applications	QTP supports testing of both the web based application and windows based application
Object Repository	Object Repository needs to be created as a separate entity	QTP automatically creates and maintains Object Repository
Language Support	Selenium supports multiple programming languages like Java, C#, Ruby, Python, Perl etc	QTP supports only VB Script
Vendor Support	As Selenium is a free tool, user would not get the vendor’s support in troubleshooting issues	Users can easily get the vendor’s support in case of any issue

48) Can WebDriver test Mobile applications?

WebDriver cannot test Mobile applications. WebDriver is a web based testing tool, therefore applications on the mobile browsers can be tested.

49) Can Captcha be automated?

No, Captcha and bar code reader cannot be automated.

50) What is Object Repository? How can we create Object Repository in Selenium?

Object Repository is a term used to refer to the collection of web elements belonging to Application Under Test (AUT) along with their locator values. Thus, whenever the element is required within the script, the locator value can be populated from the Object Repository. Object Repository is used to store locators in a centralized location instead of hard coding them within the scripts.

In Selenium, objects can be stored in an excel sheet which can be populated inside the script whenever required.

51) 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) :** It is a tool for recording and playing back. It is a Firefox plug-in.
- **WebDriver and RC:** It provides the APIs for a variety of languages like Java, .NET, PHP, etc. With most of the browsers WebDriver and RC works.
- **Grid:** With the help of Grid you can distribute tests on multiple machines so that test can be run parallel which helps in cutting down the time required for running in browser test suites.

52) What is Selenium 2.0?

Web testing tools Selenium RC and WebDriver are consolidated in single tool in Selenium 2.0

53) How will you find an element using Selenium?

In Selenium every object or control in a web page is referred as an elements, there are different ways to find an element in a web page they are

- ID
- Name
- Tag
- Attribute
- CSS
- Link text
- Partial Link Text
- Xpath etc

54) List out the test types that are supported by Selenium?

For web based application testing selenium can be used

The test types can be supported are

a) Functional

b) Regression

For post release validation with continuous integration automation tool could be used

a) Jenkins

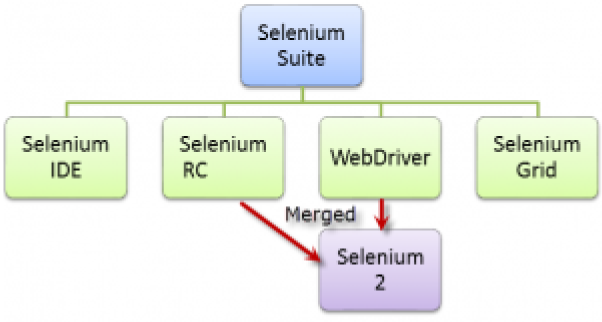
b) Hudson

c) Quick Build

d) CruiseCont

55) Explain what assertion in Selenium is and what are the types of assertion?

Assertion is used as a verification point. It verifies that the state of the application conforms to what is expected. The types of assertion are “assert”, “Verify” and “waif For”.



56) Mention what is the use of X-path?

X-Path is used to find the Web Element in web pages. It is also useful in identifying the dynamic elements.

57) Explain the difference between single and double slash in X-path?

Single slash ‘/ ’

- Single slash (/) start selection from the document node
- It allows you to create ‘absolute’ path expressions

Double Slash ‘// ’

- Double slash (//) start selection matching anywhere in the document
- It enables to create ‘relative’ path expressions

58) List out the technical challenges with Selenium?

Technical challenges with Selenium are

- Selenium supports only web based applications
- It does not support the Bitmap comparison
- For any reporting related capabilities have to depend on third party tools

- No vendor support for tool compared to commercial tools like HP UFT
- As there is no object repository concept in Selenium, maintainability of objects becomes difficult

59) What is the difference between type keys and type commands ?

Type Keys () will trigger JavaScript event in most of the cases whereas .type () won't. Type key populates the value attribute using JavaScript whereas .type keys() emulates like actual user typing.

60) What is the difference between verify and assert commands?

Assert: Assert allows checking whether an element is on the page or not. The test will stop on the step failed, if the asserted element is not available. In other words, the test will terminated at the point where check fails.

Verify: Verify command will check whether the element is on the page, if it is not then the test will carry on executing. In verification, all the commands are going to run guaranteed even if any of test fails.

61) What is J Unit Annotations and what are different types of annotations which are useful?

In JAVA a special form of syntactic meta-data can be added to Java source code, this is known as Annotations. Variables, parameters, packages, methods and classes are annotated some of the J Unit annotations which can be useful are

- Test
- Before
- After
- Ignore
- Before Class
- After Class

62) While using click command can you use screen coordinate?

To click on specific part of element, you would need to use click AT command. Click At command accepts element locator and x, y co-ordinates as arguments-

Click At (locator, cord String)

63) What are the advantages of Selenium?

- It supports C#, PHP, Java, Perl, Python
- It supports different OS like Windows, Linux and Mac OS
- It has got powerful methods to locate elements (Xpath, DOM , CSS)
- It has highly developer community supported by Google

64) Why testers should opt for Selenium and not QTP?

Selenium is more popular than QTP as

- Selenium is an open source whereas QTP is a commercial tool
- Selenium is used specially for testing web based applications while QTP can be used for testing client server application also
- Selenium supports Firefox, IE, Opera, and Safari on operating systems like Windows, Mac, Linux etc. however QTP is limited to Internet Explorer on Windows.
- Selenium supports many programming languages like Ruby, Perl, Python whereas QTP supports only VB script

65) What are the four parameter you have to pass in Selenium?

Four parameters that you have to pass in Selenium are

- Host
- Port Number
- Browser
- URL

66) What is the difference between set Speed () and sleep () methods?

Both will delay the speed of execution.

Thread. Sleep (): It will stop the current (java) thread for the specified period of time. Its done only once

- It takes a single argument in integer format

Ex: thread. Sleep (2000) - It will wait for 2 seconds

- It waits only once at the command given at sleep

Set Speed (): For specific amount of time it will stop the execution for every selenium command.

- It takes a single argument in integer format

Ex: selenium.setSpeed("2000")- It will wait for 2 seconds

- Runs each command after set Speed delay by the number of milliseconds mentioned in set Speed

This command is useful for demonstration purpose or if you are using a slow web application

67) What is same origin policy? How you can avoid same origin policy?

The **"Same Origin Policy"** is introduced for security reason, and it ensures that content of your site will never be accessible by a script from another site. As per the policy, any code loaded within the browser can only operate within that website's domain.

To avoid "Same Origin Policy" proxy injection method is used, in proxy injection mode the Selenium Server acts as a client configured **HTTP proxy** , which sits between the browser and application under test and then masks the AUT under a fictional URL

68) What are heightened privileges browsers?

The purpose of heightened privileges is similar to Proxy Injection, allows websites to do something that are not commonly permitted. The key difference is that the browsers are launched in a special mode called heightened privileges. By using these browser modes, Selenium core can open the AUT directly and also read/write its content without passing the whole AUT through the Selenium RC server.

69) How you can use “submit” a form using Selenium ?

You can use “submit” method on element to submit form-

Element. Submit ();

Alternatively you can use click method on the element which does form submission

70) What are the features of TestNG and list some of the functionality in TestNG which makes it more effective?

TestNG is a testing framework based on JUnit and NUnit to simplify a broad range of testing needs, from unit testing to integration testing. And the functionality which makes it efficient testing framework is

- Support for annotations
- Support for data-driven testing
- Flexible test configuration
- Ability to re-execute failed test cases

71) Mention what is the difference between implicit wait and explicit wait?

Implicit Wait: Sets a timeout for all successive Web Element searches. For the specified amount of time it will try looking for element again and again before throwing a **NoSuchElementException**. It waits for elements to show up.

Explicit Wait: It is a one-timer, used for a particular search.

72) Which attribute you should consider throughout the script in frame for “if no frame Id as well as no frame name”?

You can use…..driver.findElements (By.xpath (“//iframe”))….

This will return list of frames.

You will need to switch to each and every frame and search for locator which we want.

Then break the loop

73) Explain what is the difference between find elements () and find element () ?

Find element ():

It finds the first element within the current page using the given “locating mechanism”. It returns a single Web Element

Find Elements (): Using the given “locating mechanism” find all the elements within the current page. It returns a list of web elements.

74) Explain what are the JUnit annotations linked with Selenium?

The JUnit annotations linked with Selenium are

- @Before public void method() – It will perform the method () before each test, this method can prepare the test
- @Test public void method() – Annotations @Test identifies that this method is a test method environment
- @After public void method()- To execute a method before this annotation is used, test method must start with test
- @Before

75) Explain what is Data driven framework and Keyword driven?

Data driven framework: In this framework, the test data is separated and kept outside the Test Scripts, while test case logic resides in Test Scripts. Test data is read from the external files (Excel Files) and are loaded into the variables inside the Test Script. Variables are used for both for input values and for verification values.

Keyword driven framework: The keyword driven frameworks requires the development of data tables and keywords, independent of the test automation. In a keyword driven test, the functionality of the application under test is documented in a table as well as step by step instructions for each test.

76) Explain how you can login into any site if it’s showing any authentication popup for password and username?

Pass the username and password with URL

- Syntax-http://username:password@url
- ex- http://creyate:tom@www.gmail.com

77) Explain how to assert text of webpage using selenium 2.0 ?

```
Web Element el = driver.findElement (By.id (“Element”))

//get test from element and stored in text variable

String text = el.getText();

//assert text from expected

Assert.assertEquals (“Element Text”, text);
```

78) Explain what is the difference between Borland Silk and Selenium?

Silk Test Tool	Selenium Test Tool
• Borland Silk test is not a free testing tool	• Selenium is completely free test automation tool

• Silk test supports only Internet Explorer and Firefox	• Selenium supports many browsers like Internet Explorer, Firefox, Safari, Opera and so on
• Silk test uses test scripting language	• Selenium suite has the flexibility to use many languages like Java, Ruby,Perl and so on
• Silk test can be used for client server applications	• Selenium can be used for only web application

79) What is Object Repository?

An object repository is an essential entity in any UI automations which allows a tester to store all objects that will be used in the scripts in one or more centralized locations rather than scattered all over the test scripts.

80) Explain how Selenium Grid works?

Selenium Grid sent the tests to the hub. These tests are redirected to Selenium WebDriver, which launch the browser and run the test. With entire test suite, it allows for running tests in parallel.

81) Can we use Selenium grid for performance testing?

Yes. But not as effectively as a dedicated performance testing tool like Load runner.

82) What is Selenium 2.0?

Web driver is open source automation tool for web application. WebDriver is designed to provide a simpler, more concise programming interface in addition to addressing some limitations in the Selenium-RC API.

83) What is cost of web driver is this commercial or open source?
Selenium is open source and free of cost.

84) How you specify browser configurations with Selenium 2.0?
Following driver classes are used for browser configuration

- Android Driver,
- Chrome Driver,
- Event Firing WebDriver,
- Firefox Driver,
- HtmlUnitDriver,
- InternetExplorerDriver,
- Iphone Driver,
- Iphone Simulator Driver,
- Remote WebDriver

85) How is Selenium 2.0 configuration different than Selenium 1.0?
In case of Selenium 1.0 you need Selenium jar file pertaining to one library for example in case of java you need java client driver and also Selenium server jar file. While with Selenium 2.0 you need language binding (i.e. java, C# etc) and Selenium server jar if you are using Remote Control or Remote WebDriver.

86) Can you show me one code example of setting Selenium 2.0?
Below is example

```
WebDriver driver = new Firefox Driver ();
Driver. Get ("https://www.google.co.in/");
driver.findElement(By.cssSelector("#gb_2 > span.gbts")).click();
driver.findElement(By.cssSelector("#gb_1 > span.gbts")).click();
driver.findElement(By.cssSelector("#gb_8 > span.gbts")).click();
driver.findElement(By.cssSelector("#gb_1 > span.gbts")).click();
```

87) Which web driver implementation is fastest?
HtmlUnitDriver. Simple reason is HtmlUnitDriver does not execute tests on browser but plain http request – response which is far quick than launching a browser and executing tests. But then you may like to execute tests on a real

browser than something running behind the scenes

88) what all different element locators are available with Selenium 2.0?

Selenium 2.0 uses same set of locators which are used by Selenium 1.0 – id, name, css, Xpath but how Selenium 2.0 accesses them is different. In case of Selenium 1.0 you don't have to specify a different method for each locator while in case of Selenium 2.0 there is a different method available to use a different element locator. Selenium 2.0 uses following method to access elements with id, name, css and Xpath locator –

```
driver.findElement(By.id("HTMLid"));
driver.findElement(By.name("HTMLname"));
driver.findElement(By.cssSelector("cssLocator"));
driver.findElement (By. class Name ("CalssName”));
driver.findElement (By. link Text ("LinkeText”));
driver.findElement (By. partialLinkText ("Partial Link”));
driver.findElement (By. tag Name ("Tag Name”));
driver.findElement(By.xpath("XPathLocator));
```

89) How do I submit a form using Selenium?

```
Web Element el  =  driver.findElement(By.id("Element"));
el.submit();
```

90) How to capture screen shot in WebDriver?

```
File file= ((Takes Screenshot) driver).getScreenshotAs (OutputType.FILE);
FileUtils.copyFile(file, new File("c:\\name.png"));
```

91) How do I clear content of a text box in Selenium 2.0?

```
Web Element el = driver.findElement (By.id ("Element"));
el.clear();
```

92) How to execute java scripts function?

```
JavaScript Executor   jess = (JavaScript Executor) driver;
String title = (String) js.executeScript("pass your java scripts");
```

93) How to automate radio button in Selenium 2.0?

```
Web Element el = driver.findElement (By.id ("Radio button id"));

//to perform check operation
el.click ();

//verify to radio button is check it return true if selected else false
el.isSelected ()
```

94) How to count total number of rows of a table using Selenium 2.0?

```
List <Web Element> rows = driver.findElements (By.className ("//table
[@id='tabloid']/tr"));
int total Row = rows. size ();
```

95) How to capture page title using Selenium 2.0?

```
String title = driver.getTitle ()
```

96) How to store page source using Selenium 2.0?

```
String page source = driver.getPageSource ()
```

97) How to store current url using selenium 2.0?

```
String currentURL  = driver.getCurrentUrl()
```

98) How to assert text assert text of webpage using selenium 2.0?

```
Web Element el  =  driver.findElement(By.id("Element"));
```

```
//get test from element and stored in text variable
String text = el.getText();

//assert text from expected
Assert.assertEquals("Element Text", text);
```

99) How to get element attribute using Selenium 2.0?

```
Web Element el  =  driver.findElement(By.id("Element"));
//get test from element and stored in text variable
String attribute Value = el. get Attribute ("Attribute Name") ;
```

100) How to double click on element using selenium 2.0?

```
Web Element el = driver.findElement (By.id ("Element"));

Actions builder = new Actions(driver);
builder.doubleClick(el).build().perform();
```

101) How to perform drag and drop in selenium 2.0?

```
Web Element source = driver.findElement (By.id ("Source Element"));
Web Element destination  =  driver.findElement(By.id("Target ElementID"));

Actions builder = new Actions(driver);
builder.dragAndDrop(source, destination ).perform();
```

102) How to maximize window using selenium 2.0?

```
Driver. Manage ().window ().maximize ();
```

103) How to verify pdf content using selenium 2.0?

```
//get current URL pdf file URL
 URL url = new URL (driver.getCurrentUrl ());
 //create buffer reader object
 BufferedInputStream fileToParse = new BufferedInputStream (url.openStream
());
 PDFParser pdfParser = new PDFParser (fileToParse);
 PdfParser. Parse ();

 //save pdf text into strong variable
 String pdftxt = new PDFTextStripper ().get Text (pdfParser.getPDDocument ());

 //close PDFParser object
 pdfParser.getPDDocument ().close ();
```

104) How to capture video of running scripts in selenium 2.0?

```
    Public static void main (String [] args) throws Exception
{

        VideoReord videoReord = new VideoReord ();

        videoReord.startRecording ();

        WebDriver driver = new Firefox Driver();

        Driver. Get ("http://www.google.com");

        Web Element element = driver.findElement (By.name
("q"));

        element.sendKeys ("testing");

        Element. Submit ();

        System.out.println ("Page title is: " +driver.getTitle
());

        Driver. Quit ();
    videoReord.stopRecording ();

  }

  Public void start Recording () throws Exception

   {

     Graphics Configuration go = Graphics Environment

      .getLocalGraphicsEnvironment ()

      .getDefaultScreenDevice ()


      .getDefaultConfiguration ();

    this.screenRecorder = new Screen Recorder (go,

      New Format (MediaTypeKey, MediaType.FILE, MimeTypeKey,
MIME_AVI),

      new Format(MediaTypeKey, MediaType.VIDEO, Encoding Key,
ENCODING_AVI_TECHSMITH_SCREEN_CAPTURE,

        CompressorNameKey,
ENCODING_AVI_TECHSMITH_SCREEN_CAPTURE,

        Depth Key, 24, FrameRateKey, Rational.valueOf (15),

        Quality Key, 1.0f,

        KeyFrameIntervalKey, 15 * 60),

      new Format(MediaTypeKey, MediaType.VIDEO, Encoding Key, "black",

        FrameRateKey, Rational.valueOf (30)),

      null);

    This.screenRecorder. Start ();
```

```
       }
 Public void stop Recording () throws Exception
     {
       This.screenRecorder. Stop ();
     }
```

105) How to verify response 200 code using selenium 2.0?

```
        String URL = "http://www.google.com/";

 Web Client webClient = new WebClient ();
 HtmlPage htmlPage = webClient.getPage (URL);

 //verify response
 Assert.assertEquals (200, htmlPage.getWebResponse ().getStatusCode ());
 Assert.assertEquals ("OK", htmlPage.getWebResponse ().getStatusMessage ());
```

106) How to verify image using selenium 2.0?

```
        BufferedImage bufileInput = ImageIO.read (file Input);

      Data Buffer dafileInput = bufileInput.getData ().getDataBuffer ();
      Int sizefileInput = dafileInput.getSize ();
      BufferedImage bufileOutPut = ImageIO.read (file Output);
      Data Buffer dafileOutPut = bufileOutPut.getData ().getDataBuffer ();
      int sizefileOutPut = dafileOutPut.getSize();
```

107) Can we automate HTML 5 video using selenium 2.0?

1. Play video

```
        document.getElementById ("Video ID").play();
```

2. Pause video

```
        document.getElementById ("Video ID").pause() ;
```

3. To check video is paused or not use below code

```
document.getElementById ("Video ID").Paused;
```

4. To increase volume

```
document.getElementById ("Video ID").volume=0.5;
```

5. To reload video

```
document.getElementById ("Video ID").load ();
```

6. To replay from stating or forward and back playing video. You can set current playing time.

```
document.getElementById ("Video ID").current Time=0
document.getElementById ("Video ID").current Time=10
document.getElementById ("Video ID").current Time=20
```

7. To check video is muted.

```
document.getElementById ("Video ID").muted
```