**Q #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 us create test scripts these scripts can be executed repeatedly and doesn’t require any manual intervention.

**Q #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

**Q #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

**Q #4) What is Selenium? What are the different Selenium components?**

Selenium is one of the most recent 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)**](http://www.softwaretestinghelp.com/selenium-ide-download-and-installation-selenium-tutorial-2/) – Selenium IDE is a record and playback tool. It is distributed as a Firefox Plug-in. Using Selenium IDE we can able to view the source code of the tests in other formats like java, C#, Perl etc.
* **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. RC controls operations on browser, architecture of selenium RC is in client-Server passion

here we need to start the server which launches the browser and per forms the operations on it.

* [**Selenium Web Driver**](http://www.softwaretestinghelp.com/selenium-webdriver-selenium-tutorial-8/) – Web Driver is a different tool altogether that has various advantages over Selenium RC. Web Driver directly communicates with the web browser and uses its native compatibility to automate.
* [**Selenium Grid**](http://www.softwaretestinghelp.com/selenium-grid-selenium-tutorial-29/) – Selenium Grid is used to distribute your test execution on multiple platforms and environments concurrently.

**Q #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

**Q #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.

**Q #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 plugin, 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 which is a GUI less or headless browser. |
| 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 before executing the test scripts | | Selenium RC requires server to be started before executing the test scripts | WebDriver doesn't require any server to be started before executing the test scripts |
| Architecture | Selenium IDE is a Javascript 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. |
| WAP (iPhone/Android) Testing | Selenium IDE doesn't support testing of iPhone/Andriod applications | | Selenium RC doesn't support testing of iPhone/Andriod 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 communicates directly | | Selenium RC is slower than WebDriver as it doesn't communicates directly with the browser; rather it sends selenese commands over to Selenium Core which in turn communicates with the browser. | WebDriver communicates directly with the web browsers. Thus making it much faster. |

**Q #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.

**Q #9) What is Selenese?**

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

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

Locator can be termed as an address that uniquely identifies a web element within the webpage. Thus, to identify web elements accurately and precisely we have [different types of locators in Selenium](http://www.softwaretestinghelp.com/using-selenium-xpath-and-other-locators-selenium-tutorial-5/):

* ID
* ClassName
* Name
* TagName
* LinkText
* PartialLinkText
* Xpath
* CSS Selector
* DOM

**Q #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.

*assertTrue(“The title of the window is incorrect.”,driver.getTitle().equals(“Title of the page”));*

Assert.assertEquals(21, multiply(10, 5));

Verify **(Verify is implemented by soft assert):**

At times, we might require the test method to continue execution even after the failure of the assertion statements.

InTestNG, **Verify** isimplementedusing **SoftAssert** class.   
In case of SoftAssert, all the statements in the test method are executed (including multiple assertions). Once, all the statements are executed, the test result is collated basedontheassertionresultsand test is marked as passed or fail.   
Example-

**@Test**

**public** **void** **softAssertionTest**(){

//Creating softAssert object

SoftAssert softAssert = **new** SoftAssert();

//Assertion failing

softAssert.fail("Failing first assertion");

System.out.println("Failing 1");

//Assertion failing

softAssert.fail("Failing second assertion");

System.out.println("Failing 2");

//Collates the assertion results and marks test as pass or fail

softAssert.assertAll();

}

Output-

Failing **1**

Failing **2**

**FAILED:** softAssertionTest

java.lang.AssertionError: The following asserts **failed:**

Failing first assertion,

Failing second assertion

**Q #34) How to assert title of the web page?**

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

**Q #12) What is an Xpath?**

[Xpath](http://www.softwaretestinghelp.com/using-selenium-xpath-and-other-locators-selenium-tutorial-5/) 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, Xpath can be used to locate HTML elements.

[Xpath](http://www.softwaretestinghelp.com/using-selenium-xpath-and-other-locators-selenium-tutorial-5/) is used to locate a web element based on its XML path, the fundamental behind locating elements using Xpath 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. Generally we use Xpath when we don't have properties like ID and Value etc.

**Q #13) What is the difference between “/” and “//” in Xpath?**

**Single Slash “/” –**Single slash is used to create Xpath with absolute path i.e. the xpath would be created by selecting the start of the document node/start node to the element you require.

Example:

/html/body/div[@id='page']

**Double Slash “//” -** Double slash is used to create Xpath with relative path i.e. the xpath would be created to start selection from anywhere within the document.

Syntax : //tagname[@property='value']

Example: //Button[@name='btng']

**Q #51) How to get parent node for a child node “title 50”?**

**Example:** How could I get its parent? Result should be the store node

<?xml version="1.0" encoding="utf-8"?>

<d:data xmlns:d="defiant-namespace" d:mi="23">

<store d:mi="22">

<book price="12.99" d:price="Number" d:mi="4">

<title d:constr="String" d:mi="1">Sword of Honour</title>

<category d:constr="String" d:mi="2">fiction</category>

<author d:constr="String" d:mi="3">Evelyn Waugh</author>

</book>

<book price="8.99" d:price="Number" d:mi="9">

<title d:constr="String" d:mi="5">Moby Dick</title>

<category d:constr="String" d:mi="6">fiction</category>

<author d:constr="String" d:mi="7">Herman Melville</author>

<isbn d:constr="String" d:mi="8">0-553-21311-3</isbn>

</book>

<book price="8.95" d:price="Number" d:mi="13">

<title d:constr="String" d:mi="10">50</title>

<category d:constr="String" d:mi="11">reference</category>

<author d:constr="String" d:mi="12">Nigel Rees</author>

</book>

<book price="22.99" d:price="Number" d:mi="18">

<title d:constr="String" d:mi="14">The Lord of the Rings</title>

<category d:constr="String" d:mi="15">fiction</category>

<author d:constr="String" d:mi="16">J. R. R. Tolkien</author>

<isbn d:constr="String" d:mi="17">0-395-19395-8</isbn>

</book>

<bicycle price="19.95" d:price="Number" d:mi="21">

<brand d:constr="String" d:mi="19">Cannondale</brand>

<color d:constr="String" d:mi="20">red</color>

</bicycle>

</store>

</d:data>

**Answer:**

Syntax:

//\*[title="50"]/parent::store

OR else we can use one of these

//\*[title="50"]/parent::\*

//\*[title="50"]/..

**Q #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.

OR

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. This is nothing but RC

**Q #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.

How it works?

Selenium Grid can be used to run multiple instances of Selenium RC on various operating system and browser configurations.

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

**Q #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.

**Q #17) which is the latest Selenium tool?**

WebDriver

**Q #18) How do I launch the browser using Web Driver?**

The following syntax can be used to launch Browser:

For firefox

WebDriver driver = new FirefoxDriver();

For Chrome browser

File file = new File("D:\\selnium webdriver\\driver\\chromedriver.exe");

System.setProperty("webdriver.chrome.driver", file.getAbsolutePath() );

WebDriver driver = new ChromeDriver();

For Internet explorer

File file = new File("D:\\selnium webdriver\\driver\\IEDriverServer.exe");

System.setProperty("webdriver.ie.driver", file.getAbsolutePath());

WebDriver driver = new InternetExplorerDriver();

**Q #19) What are the different types of Drivers available in Web Driver?**

The different drivers available in WebDriver are:

* FirefoxDriver
* InternetExplorerDriver
* ChromeDriver
* SafariDriver
* OperaDriver
* AndroidDriver
* IPhoneDriver
* HtmlUnitDriver

**Q #20) What are the different types of waits available in Web Driver?**

There are two [types of waits available in Web Driver](http://www.softwaretestinghelp.com/selenium-webdriver-waits-selenium-tutorial-15/):

1. Implicit Wait
2. Explicit Wait

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

Implicit wait will not work on all the commands/statements in the application. It will work only for "FindElement" and "FindElements" statements.

Syntax:   
 driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

OR

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:** 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.

**FluentWait Command:**

Each **FluentWait** instance defines the maximum amount of time to wait for a condition, as well as the frequency with which to check the condition. Furthermore, the user may configure the wait to ignore specific types of exceptions whilst waiting, such as **NoSuchElementExceptions** when searching for an element on the page.

Syntax:

|  |
| --- |
| // Waiting 30 seconds for an element to be present on the page, checking for its presence once every 5 seconds.       Wait wait = new FluentWait(driver)        .withTimeout(30, SECONDS)        .pollingEvery(5, SECONDS)        .ignoring(NoSuchElementException.class);      WebElement foo = wait.until(new Function() {        public WebElement apply(WebDriver driver) {        return driver.findElement(By.id("foo"));        }       }); |
|  |

**Explicit wait: ExpectedConditions Command**: Purpose: Models a condition that might reasonably be expected to eventually evaluate to something that is neither null nor false.

Java

|  |  |
| --- | --- |
|  | WebDriverWait wait = new WebDriverWait(driver, 10);  WebElement element = wait.until(ExpectedConditions.elementToBeClickable(By.id(>someid>)));  More methods:  ExpectedConditions.presenceOfElementLocated(container)  ExpectedConditions.visibilityOfElementLocated(locator) |

**PageLoadTimeout** Command:

**Purpose**: Sets the amount of time to wait for a page load to complete before throwing an error. If the timeout is negative, page loads can be indefinite.

driver.manage().timeouts().pageLoadTimeout(100, SECONDS);

**Sleep Command:**

Purpose: This is rarely used, as it always force the browser to wait for a specific time. **Thread.Sleep** is never a good idea and that’s why Selenium provides wait primitives. If you use them you can specify much higher timeout value which makes tests more reliable without slowing them down as the condition can be evaluated as often as it’s required.

Ex: Thread.sleep(1000);

**Q #21)** **How to type in a textbox using Selenium?**

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

**Syntax:**  
*WebElement username = drv.findElement(By.id(“Email”));*  
*// entering username*  
*username.sendKeys(“sth”);*

**Q #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. isDisplayed()
2. isSelected()
3. isEnabled()

**Syntax:**

**isDisplayed():**  
***boolean****buttonPresence = driver.findElement(By.id(“gbqfba”)).isDisplayed();*

**isSelected():**  
***boolean****buttonSelected = driver.findElement(By.id(“gbqfba”)).isSelected();*

**isEnabled():**  
***boolean****searchIconEnabled = driver.findElement(By.id(“gbqfb”)).isEnabled();*

**Q #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”)).getText();*

**Q #24) How to select value in a dropdown?**

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

**Syntax:**

**selectByValue:**  
*Select selectByValue =****new*** *Select(driver.findElement(By.id(“SelectID\_One”)));*  
*selectByValue.selectByValue(“greenvalue”);*

**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);*

**Q #25) What are the different types of navigation commands?**

Following are the [navigation commands](http://www.softwaretestinghelp.com/selenium-webdriver-waits-selenium-tutorial-15/):  
**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”);*

**Q #26) How to click on a hyper link using linkText?**

*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.

**Q #27)** **How to**[**handle frame in Web Driver**](http://www.softwaretestinghelp.com/selenium-tutorial-18/)**?**

An inline frame acronym as iframe 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 iframe by id**  
*driver.switchTo().frame(“ID of the frame“);*

**Locating iframe using tagName**  
*driver.switchTo().frame(driver.findElements(By.tagName(“iframe”).get(0));*

**Locating iframe using index**

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

------------

**frame(Name of Frame)**  
*driver.switchTo().frame(“name of the frame”);*

**frame(WebElement element)**  
**Select Parent Window**  
*driver.switchTo().defaultContent();*

**Q #28) When do we use findElement() and findElements()?**

**findElement():**findElement() 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:**

*WebElement element =driver.findElement(By.xpath(“//div[@id=’example’]//ul//li”));*  
**findElements():**findElements() 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 WebElements.

**Syntax:**  
*List <WebElement> elementList =driver.findElements(By.xpath(“//div[@id=’example’]//ul//li”));*

**Q #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 WebElement List.

**Sample Code**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | // Storing the list  List <WebElement> elementList = driver.findElements(By.xpath("//div[@id='example']//ul//li"));  // Fetching the size of the list  int listSize = elementList.size();  for (int i=0; i<listSize; i++)  {  // Clicking on each service provider link  serviceProviderLinks.get(i).click();  // Navigating back to the previous page that stores link to service providers  driver.navigate().back();  } |

**Q #30) What is the difference between driver.close() and driver.quit command?**

**close()**: WebDriver’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.

**Q #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.

**Q #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](http://www.softwaretestinghelp.com/handle-alerts-popups-selenium-webdriver-selenium-tutorial-16/). There are the four methods that we would be using along with the Alert interface.

* void dismiss() – The dismiss() 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 getText() – The getText() method returns the text displayed on the alert box.
* void sendKeys(String stringToSend) – The sendKeys() method enters the specified string pattern into the alert box.

**Syntax:**  
*// accepting javascript alert*  
*Alert alert = driver.switchTo().alert();*  
*alert.accept();*

**Q #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 AutoIT, Robot class etc.

**Q #35) How to mouse over 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:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | // Instantiating Action Interface  Actions actions=new Actions(driver);  // howering on the dropdown  actions.moveToElement(driver.findElement(By.id("id of the dropdown"))).perform();  // Clicking on one of the items in the list options  WebElement subLinkOption=driver.findElement(By.id("id of the sub link"));  subLinkOption.click(); |

**Q #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”);*

**Q #37) How to capture screenshot in WebDriver?**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31 | import org.junit.After;  import org.junit.Before;  import org.junit.Test;  import java.io.File;  import java.io.IOException;  import org.apache.commons.io.FileUtils;  import org.openqa.selenium.OutputType;  import org.openqa.selenium.TakesScreenshot;  import org.openqa.selenium.WebDriver;  import org.openqa.selenium.firefox.FirefoxDriver;    public class CaptureScreenshot {         WebDriver driver;         @Before         public void setUp() throws Exception {              driver = new FirefoxDriver();              driver.get("[https://google.com](https://google.com/)");       }       @After       public void tearDown() throws Exception {              driver.quit();       }         @Test       public void test() throws IOException {              // Code to capture the screenshot  File scrFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);              // Code to copy the screenshot in the desired location  FileUtils.copyFile(scrFile, new File("C:\\CaptureScreenshot\\google.jpg"));       }  } |

**Q #38) What is Junit?**

[Junit](http://www.softwaretestinghelp.com/selenium-junit-framework-selenium-tutorial-11/) is a unit testing framework introduced by Apache. Junit is based on Java.

**Q #39) What are Junit annotations?**

Following are the Junit 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.
* **@BeforeClass:**Method annotated as @BeforeClass lets the system know that this method shall be executed once before any of the test method.
* **@AfterClass:**Method annotated as @AfterClass 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.

**Q #40)** **What is TestNG and how is it better than Junit?**

[TestNG](http://www.softwaretestinghelp.com/testng-framework-selenium-tutorial-12/) is an advance framework designed in a way to leverage the benefits by both the developers and testers. With the commencement of the frameworks, JUnit 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, JUnit 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 WebDriver’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 t hem are:

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

**Q #41)** **How to set test case priority in Test NG?**

**Setting Priority in TestNG**

**Code Snippet**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | package TestNG;    public class SettingPriority {        @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

**Q #42) What is a framework?**

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

**Q #43)** **What are the advantages of Automation framework?**

**Advantage of**[**Test Automation framework**](http://www.softwaretestinghelp.com/test-automation-frameworks-selenium-tutorial-20/)

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

**Q #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.

**Q #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](http://www.softwaretestinghelp.com/selenium-framework-design-selenium-tutorial-21/).

**Q #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 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 |

**Q #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 |

**Q #48) Can WebDriver test Mobile applications?**

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

**Q #49) Can captcha be automated?**

No, captcha and bar code reader cannot be automated.

**Q #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.

That’s all for now.

Hope in this article you will find answers to most frequently asked Selenium and WebDriver Interview questions. The answers provided here are also helpful for understanding the Selenium basics and advanced WebDriver topics.

**Q #52) How would you make sure that a page is loaded using Selenium and Webdriver?**

The best approach is by selecting an element from the page & stand by till it becomes clickable.

**selenium.waitForPageToLoad("5000");**

**// Or**

**while (!(selenium.isElementPresent("any page element ")==true)) {**

**selenium.setSpeed("5");**

**Thread.sleep(5);**

**}**

**Interview Questions:**

1. **Common Exceptions in selenium:**

### Stale Element Reference Exception – Selenium

A stale element reference exception is thrown in one of two cases, the first being more common than the second:

* The element has been deleted entirely.
* The element is no longer attached to the DOM.

1. **NoSuchElementException** : WebDriver is unable to identify the elements during run time, i.e. FindBy method can’t find the element.
2. **ElementNotVisibleException**: Although an element is present in the DOM, it is not visible (cannot be interacted with). E.g. Hidden Elements – defined in HTML using type=”hidden”.
3. **ElementNotSelectableException**: Although an element is present in the DOM, it may be disabled (cannot be clicked/selected).
4. **TimeoutException**: The command did not complete in enough time. E.g. the element didn’t display in the specified time. **Encountered when working with waits.**

# Java Exception propagation

In the below example exception occurs in m() method where it is not handled,so it is propagated to previous n() method where it is not handled, again it is propagated to p() method where exception is handled.

Exception can be handled in any method in call stack either in main() method,p() method,n() method or m() method.

1. class TestExceptionPropagation1{
2. void m(){
3. int data=50/0;
4. }
5. void n(){
6. m();
7. }
8. void p(){
9. try{
10. n();
11. }catch(Exception e){System.out.println("exception handled");}
12. }
13. public static void main(String args[]){
14. TestExceptionPropagation1 obj=new TestExceptionPropagation1();
15. obj.p();
16. System.out.println("normal flow...");
17. }

}

### Q #53) Ajax Call Handling – You can get more information from <https://www.guru99.com/handling-ajax-call-selenium-webdriver.html>

* AJAX allows the Web page to retrieve small amounts of data from the server without reloading the entire page.
* To test Ajax application, different wait methods should be applied
  + ThreadSleep
  + Implicit Wait
  + Explicit Wait
  + WebdriverWait
  + Fluent Wait
* Creating automated test request may be difficult for testing tools as such AJAX application often use different encoding or serialization technique to submit POST data.

**Assertions:**

**Example:**

1. **assertEquals:**

**public** **void** **testCaseVerifyHomePage**() {

driver= **new** FirefoxDriver();

driver.navigate().to("[http://google.com"](http://google.com/));

Assert.assertEquals("Google", driver.getTitle()); }

1. **assertTrue:**

**public** **void** **testCaseVerifyHomePage**() {

driver= **new** FirefoxDriver();

driver.navigate().to("[http://google.com"](http://google.com/));

Assert.assertEquals("Gooooogle", driver.getTitle());

}

1. **assertFalse**

public void testCaseVerifyHomePage() {

driver= new FirefoxDriver();

driver.navigate().to("[http://google.com"](http://google.com/));

Assert.assertEquals("Gooogle", driver.getTitle(), "Strings are not matching");

//Write a code to login and write a method called isUserLoggedInSuccessfully and isUserLoggedOut which returns boolean. Assert.assertTrue(isUserLoggedInSuccessfully(), "User failed to login");

Assert.assertFalse(isUserLoggedOut())

}

}

**Handling Dynamic Elements:**

**For more** [**http://www.optimusinfo.com/dynamic-ids-headache-automation-great-challenge/**](http://www.optimusinfo.com/dynamic-ids-headache-automation-great-challenge/)

### Stable preceding text:

For web elements such as text fields and text areas we can identify them by using the stable preceding text as in the sample below.

web\_element\_name=//label[text()=’preceding\_web\_element:’]/following::input

Example:

Identify this field with preceding Text City.

City

Location 

Example: Xpath = //label[text()=’City’]//following::input

**I . Get Default Value from text field:**

Sometimes in a text field without entering any input we see default value is populated how to get that value

Get default value from below text field

City

Hyderabad

Answer:

The getText function will usually return text content inside the HTML tag.

Eg.

<div>Hello World</div>

Here,

driver.findElement(By,tagName("div")).getText();

Will return the text "Hello World"

Usually if you are working with Textbox, the getText function will return null. The Textbox values are stored in "value" attribute. So, in that case you will have to use **getAttribute("value")**;

Here is the example of typical text box.

<input value="submit" />

Using,

driver.findElement(By.tagName("input")).getAttribute("value");

will return the text "submit"

**Scrum:** Scrum is an agile way to manage a project, usually software development. Agile software development with Scrum is often perceived as a methodology; but rather than viewing Scrum as methodology, think of it as a framework for managing a process.

## Agile: Agile methodology in project management generally used especially for software development that is characterized by the division of tasks into short phases of work and frequent reassessment and adaptation of plans.

## What is Scrum?

In the agile Scrum world, instead of providing complete, detailed descriptions of how everything is to be done on a project,

much of it is left up to the Scrum software development team. This is because the team will know best how to solve the problem they are presented.

Within agile development, Scrum teams are supported by two specific roles. The first is a Scrum Master, who can be thought of as a

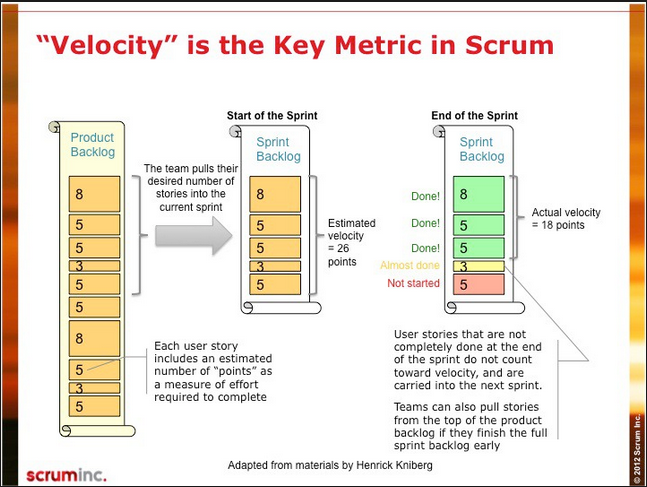
coach for the team, helping team members use the Scrum process to perform at the highest level.

The product owner (PO) is the other role, and in Scrum software development, represents the business, customers or users, and guides the team toward building the right product.

**Sprint:** A **sprint** (or iteration) is the basic unit of development in **Scrum**. The **sprint** is a time-boxed effort; that is, it is restricted to a specific duration.

The duration is fixed in advance for each **sprint** and is normally between one week and one month, with two weeks being the most common.

**Velocity: Velocity** is a measure of the amount of work a Team can tackle during a single Sprint and is the key metric in **Scrum**. **Velocity** is calculated at the end of the Sprint by totaling the Points for all fully completed User Stories.



**Sprint Backlog:**

The **sprint backlog** is a list of tasks identified by the Scrum team to be completed during the Scrum **sprint**.

During the **sprint** planning meeting, the team selects some number of product **backlog** items,

usually in the form of user stories, and identifies the tasks necessary to complete each user story.

**Question 1: SQL Query to find second highest salary of Employee**

select MAX(Salary) from Employee WHERE Salary NOT IN (select MAX(Salary) from Employee );

**SELECT** TOP 1 salary **FROM** (**SELECT** TOP 2 salary **FROM** employees **ORDER** **BY** salary **DESC**) **AS** emp **ORDER** **BY** salary **ASC**

Question 2: To find third highest salary. Using this we can find nth highest salary

**SELECT** TOP 1 salary **FROM** ( **SELECT** TOP 3 salary **FROM** employees **ORDER** **BY** salary **DESC**) **AS** emp **ORDER** **BY** salary **ASC**

**Question 2: SQL Query to find Max Salary from each department.**

SELECT DeptID, MAX(Salary) FROM Employee  GROUP BY DeptID.

**Question 5: Write a SQL Query to print the name of distinct employee whose DOB is between 01/01/1960 to 31/12/1975.**

Ans:

SELECT DISTINCT EmpName FROM Employees WHERE DOB  BETWEEN ‘01/01/1960’ AND ‘31/12/1975’;

**Question 6: Write an SQL Query find number of employees according to gender  whose DOB is between 01/01/1960 to 31/12/1975.**

Answer : SELECT COUNT(\*), sex from Employees  WHERE  DOB BETWEEN ‘01/01/1960 ' AND ‘31/12/1975’ GROUP BY sex;

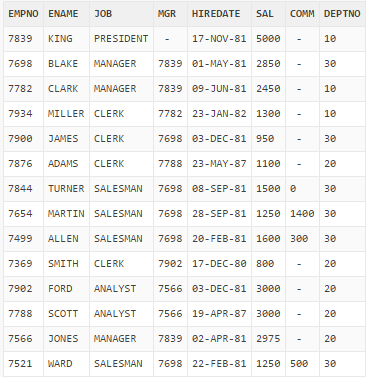
**Question 9: find all Employee records containing the word "Joe", regardless of whether it was stored as JOE, Joe, or joe.**

Answer : SELECT  \* from Employees  WHERE  upper(EmpName) like upper('joe%');

**Question 10: Write a SQL Query to find year from date.**

Answer:  SELECT YEAR(GETDATE()) as "Year";

Question 11:

DEPT EMP

We want to see only those Departments where Average salary is greater than 80. Here the condition is associated with a non-static aggregated information which is “average of salary”. We will need to use HAVING clause here:

select dept.DNAME Department, avg(emp.sal) Salary from dept,emp where dept.DEPTNO=emp.DEPTNO

group by dept.DNAME having avg(emp.sal)>20

**Question 12: Inner Join and outer join**

Inner Join: Result set generated by the combination of rows in two tables containing only those records which are present in both the tables based on a joining condition

OR

Inner join returns rows when there is at least one match in both tables

(If none of the records matches between two tables, inner join will return NULL set)

select dept.dname Department, emp.ename Employee from dept,emp

where dept.DEPTNO=emp.DEPTNO

Outer Join: Outer Join will return matching rows from both the tables as well as unmatched rows from one or both the tables.

SELECT dept.name DEPARTMENT, emp.name EMPLOYEE

FROM DEPT dept LEFT OUTER JOIN EMPLOYEE emp

ON dept.id = emp.dept\_id

**Question 13**: List the number of customers in each country. Only include countries with more than 10 customers

**SELECT COUNT(Id), Country FROM Customer**

**GROUP BY Country**

**HAVING COUNT(Id) > 10**

**Question 14:** List the number of customers in each country, except the USA, sorted high to low.   
Only include countries with 9 or more customers.

**SELECT COUNT(Id), Country**

**FROM Customer**

**WHERE Country <> 'USA'**

**GROUP BY Country**

**HAVING COUNT(Id) >= 9**

**ORDER BY COUNT(Id) DESC**

**Question 15**: If you want to delete all the rows if the selected columns repeated more than 1 time then use below query.

Query to delete 3 duplicated rows (in our example table) or repeated more than 1 time using "GROUP BY" clause.

delete from EmpDup where EmpID in(select EmpID from EmpDup group by EmpId having

count(\*) >1)

**Question 16:** If you want to delete duplicates of a  particular ID(here empid) then use"Delete Top( )" clause as shown below

delete top(2) From EmpDup where empid=2

OR

delete top(select count(\*)-1 From EmpDup x where x.empid=2) From EmpDup where empid=2

TestNg : [TestNG](http://testng.org/) is a Java testing framework, inspired by JUnit and NUnit. It overcomes the limitations and drawbacks of JUnit. TestNG introduces a whole new set of features making it a powerful and user friendly tool

Advantages of TestNG:

Simplified Annotations: Annotations have been simplified, making it easier for testers to understand them.

* **Results in HTML |**It generates reports in HTML.
* **Multiple test-cases at once |**Using TestNG Suite (testing.xml), which is essentially an xml file, testers can execute a number of test cases at a time. In this xml file, users need to mention the number of classes and test-cases they wish to execute.
* **Runs failed test-cases |**TestNG can also be used to run failed test-cases. This is one of most important advantages of TestNG over JUnit.
* **Allows grouping |**TestNG allows supports grouping. Using this feature, testers can group test-cases without too much effort, which was not possible with JUnit.
* **Allows running on multiple browsers |**TestNG allows testers to execute and run one script in multiple browsers.
* **Parametric testing |**Most of the time, testers have to execute a large number of varied tests, mainly due to the nature of their business logic. However, this lengthy process can be eliminated as parametric testing allows you to run the same test a number of times by simply changing the values. It allows you to pass parameters to your test methods in two ways: parameter and @dataprovider.
* **Bypassing or ignoring test-cases |**This feature is useful if you do not want execute a particular testcase(s). In such instances, TestNG, with the help of annotation @Test(enabled = false), allows you to disable or bypass the particular test-case(s).
* **Reporter class (it generates logs |**In TestNG, with the help of the reporter class, users can log messages for the test. Let’s say you are running a test-case, and you want to log related information. This could be just surface or in-depth information, depending on what you want.
* **Expected exceptions |**It allows you to trace the exception handling of the code. While writing a code, there may be situations where testers want to verify if an exception is being presented when executed. This method will give you the details of the exceptions that are expected to be presented by that particular method. Use this method with @Test annotation.
* **Dependent On Method |**TestNG supports the ‘dependence’ method. We can have dependence as an attriubute of a method, for example, if one method is dependent on another. This is not available in JUnit.
* **TestCase priority |**We can execute the test cases in a particular order. To accomplish this, we define the order in @Test Annotation.

TestNG Test Suite: <http://toolsqa.com/selenium-webdriver/testng-testsuite/>

testng.xml syntax:

<suite name=”Test - Suite”>

<test name=”Tools QA”>

<classes>

<class name=”automationFramework.TestNG”>

</classes>

<test>

</suite>

### ****TestNG Annotations:****

**Following is the list of the most useful and favorable annotations in TestNG:**

| **Annotation** | **Description** |
| --- | --- |
| @Test | The annotation notifies the system that the method annotated as @Test is a test method |
| @BeforeSuite | The annotation notifies the system that the method annotated as @BeforeSuite must be executed before executing the tests in the entire suite |
| @AfterSuite | The annotation notifies the system that the method annotated as @AfterSuite must be executed after executing the tests in the entire suite |
| @BeforeTest | The annotation notifies the system that the method annotated as @BeforeTest must be executed before executing any test method within the same test class |
| @AfterTest | The annotation notifies the system that the method annotated as @AfterTest must be executed after executing any test method within the same test class |
| @BeforeClass | The annotation notifies the system that the method annotated as @BeforeClass must be executed before executing the first test method within the same test class |
| @AfterClass | The annotation notifies the system that the method annotated as @AfterClass must be executed after executing the last test method within the same test class |
| @BeforeMethod | The annotation notifies the system that the method annotated as @BeforeMethod must be executed before executing any and every test method within the same test class |
| @AfterMethod | The annotation notifies the system that the method annotated as @AfterMethod must be executed after executing any and every test method within the same test class |
| @BeforeGroups | The annotation notifies the system that the method annotated as @BeforeGroups is a configuration method that enlists a group and that must be executed before executing the first test method of the group |
| @AfterGroups | The annotation notifies the system that the method annotated as @AfterGroups is a configuration method that enlists a group and that must be executed after executing the last test method of the group |

MBS Product Knowledge:

**Mortgage-backed securities (MBS)** are securities that represent an interest in a pool of [mortgage](http://www.investinganswers.com/node/1608)[loans](http://www.investinganswers.com/node/5825).

## HOW IT WORKS (EXAMPLE):

To understand how MBS work, it's important to understand how they're created. Let's assume you want to buy a house, so you get a [mortgage](http://www.investinganswers.com/node/1608) from XYZ Bank. XYZ Bank transfers [money](http://www.investinganswers.com/node/5074) into your account, and you agree to repay the money according to a set schedule. XYZ Bank may then choose to hold the mortgage in its portfolio (i.e., simply collect the interest and [principal](http://www.investinganswers.com/node/928) payments over the next several years) or sell it.

[If you're ready to buy a home, use our [Mortgage Calculator](http://www.investinganswers.com/calc/mortgage-calculator-what-will-my-monthly-principal-interest-payment-be-2084) to see what your [monthly principal and interest payment](http://www.investinganswers.com/calc/mortgage-calculator-what-will-my-monthly-principal-interest-payment-be-2084) [will](http://www.investinganswers.com/node/4974) be.]

If XYZ Bank sells the mortgage, it gets [cash](http://www.investinganswers.com/node/5011) to make other [loans](http://www.investinganswers.com/node/5825). So let's assume that XYZ Bank sells your mortgage to ABC Company, which could be a governmental, quasi-governmental, or private entity. ABC Company groups your mortgage with similar mortgages it has already purchased (referred to as pooling the mortgages). The mortgages in the pool have common characteristics (i.e., similar interest rates, [maturities](http://www.investinganswers.com/node/4911), etc.).

ABC Company then sells securities that represent an interest in the pool of mortgages, of which your mortgage is a small part (called securitizing the pool). It sells these MBS to investors in the [open](http://www.investinganswers.com/node/5848)[market](http://www.investinganswers.com/node/3609). With the [funds](http://www.investinganswers.com/node/5054) from the [sale](http://www.investinganswers.com/node/5682) of the MBS, ABC Company can purchase more mortgages and create more MBS.  
   
When you make your monthly mortgage payment to XYZ Bank, they keep a fee or spread and send the rest of the payment to ABC Company. ABC Company in turn takes a fee and passes what's left of your principal and interest payment along to the investors who hold the MBS.  
   
There are two kinds of MBS. The first, called pass-throughs or participation certificates (PCs), represent a direct claim on the pool of mortgage loans. The second kind, called collateralized mortgage[*obligation*](http://www.investinganswers.com/node/3919)mortgage obligations (CMOs) or [*real estate*](http://www.investinganswers.com/node/2146)mortgage[*investment*](http://www.investinganswers.com/node/4904)conduits (REMICs) are more complicated. These securities essentially take the interest and principal payments from several MBS and create additional securities with varying maturities and coupons.  
   
For investors, an MBS is much like a [bond](http://www.investinganswers.com/node/1287). Most [offer](http://www.investinganswers.com/node/3909) semi-annual or monthly [income](http://www.investinganswers.com/node/5798), and this payment frequency enhances the [compounding](http://www.investinganswers.com/node/341) effects of reinvestment. Prepayment risk is a large concern for MBS investors. When people move, for example, they sell their houses, payoff their mortgages with the proceeds, and buy new houses with new mortgages. When interest rates fall, many homeowners refinance their mortgages, meaning they obtain new, lower-rate mortgages and pay off their higher-rate mortgages with the proceeds. Like [bonds](http://www.investinganswers.com/node/1287), changes in interest rates affect MBS prices, but the change is exacerbated by the fact that MBS investors are more likely to get their principal back early. They might have to reinvest that principal at rates below what their MBS were yielding.  
   
The Role of the Government in MBS  
Although several private institutions (brokerage firms, financial institutions, and even construction companies) create and sell MBS, the Federal National Mortgage Association ([FNMA](http://www.investinganswers.com/node/2140), or "[Fannie Mae](http://www.investinganswers.com/node/2352)") and the Federal Home Loan Mortgage [Corporation](http://www.investinganswers.com/node/2862) (FHLMC, or "[Freddie Mac](http://www.investinganswers.com/node/999)") purchase a very large portion of mortgages. Freddie Mac and Fannie Mae (both government- sponsored entities) [guarantee](http://www.investinganswers.com/node/993) the timely payment of interest and principal on the MBS they [issue](http://www.investinganswers.com/node/5068) -- that is, if the borrowers do not make their mortgage payments on time, Freddie Mac and Fannie Mae will still make their payments to their MBS investors. It is important to [note](http://www.investinganswers.com/node/5082) that the U.S. government does not guarantee Freddie Mac or Fannie Mae. That is, if these entities cannot fulfill their obligations to their MBS investors, the federal government has no responsibility to rescue them. However, both entities have lines of [credit](http://www.investinganswers.com/node/5244) with the government, and investors generally believe that the government would not actually let them [default](http://www.investinganswers.com/node/5956) on any of their securities.  
   
The Government National Mortgage Association (GNMA, or "[Ginnie Mae](http://www.investinganswers.com/node/2390)"), on the other hand, is a governmental entity that does not purchase mortgages but does guarantee (with the full faith and credit of the U.S. government) the mortgage-backed securities of certain MBS issuers. GNMA MBS have the lowest risk of the three, because they carry an explicit guarantee from the federal government.

Maven: ( <http://www.mkyong.com/maven/how-to-create-a-java-project-with-maven/> )

**1) Explain what is Maven? How does it work?**

**Maven is a project management tool**. It provides the developer a complete build lifecycle framework. On executing Maven commands, it will look for POM file in Maven; it will run the command on the resources described in the POM.

**2) List out what are the aspects does Maven Manages?**

Maven handles following activities of a developer

• Build  
• Documentation  
• Reporting  
• Dependencies  
• SCMs  
• Releases  
• Distribution  
• Mailing list

**3) Mention the three build lifecycle of Maven?**

• Clean: Cleans up artifacts that are created by prior builds  
• Default (build): Used to create the application  
• Site: For the project generates site documentation

**4) Explain what is POM?**

In Maven, a standard pom.xml is generated. This POM file is like the Ant build.xml file, it describes the entire project information, everything from directory structure, project plugins, project dependencies, how to build this project and etc, read this [official POM guide](http://maven.apache.org/guides/introduction/introduction-to-the-pom.html).

OR

In Maven, POM (Project Object Model) is the fundamental unit of work. It is an XML file which holds the information about the project and configuration details used to build a project by Maven.

**5) Explain what is Maven artifact?**

Usually an artifact is a JAR file which gets arrayed to a Maven repository. One or more artifacts a maven build produces such as compiled JAR and a sources JAR.

Each artifact includes a group ID, an artifact ID and a version string.

**6) Explain what is Maven Repository? What are their types?**

A Maven repository is a location where all the project jars, library jars, plug-ins or any other particular project related artifacts are stored and can be easily used by Maven.

Their types are local, central and remote

Differences:

Your "local" repository is just like a cache of all the maven artifacts that have ever been retrieved from remote repositories. It is where maven artifacts are referenced from when building/running applications via maven tools. You have no choice to use/not use this as all maven tools will first check the local repository and then if not present retrieve an artifact from remote repositories.  
  
"central" and "remote" repositories are essentially the same, they are just repositories that are located remote from yourself (and therefore remote/different from your local cache). These are the true source location for the various artifacts. The difference between "central" and just any other "remote" repository, is just that the central repository is the one that is "built-in" to the maven tools, ie. you don't need to explicitly configure this one and so you can still resolve a large number of artifacts with no extra config. Other "remote" repositories are required to store and distribute artifacts that people want to distribute but don't necessarily want in the offical "central" repository, ie. if you build a JAR library and you want to distribute to a few of your friends, you probably wouldn't want that in the main, offical central repository so you just setup your own remote repository.

**7) Why Maven Plugins are used?**

Maven plugins are used to  
• Create a jar file  
• Create war file  
• Compile code files  
• Unit testing of code  
• Documenting projects  
• Reporting

**8) List out the dependency scope in Maven?**

The various dependency scope used in Maven are:

• Compile: It is the default scope, and it indicates what dependency is available in the classpath of the project  
• Provided: It indicates that the dependency is provided by JDK or web server or container at runtime  
• Runtime: This tells that the dependency is not needed for compilation but is required during execution  
• Test: It says dependency is available only for the test compilation and execution phases  
• System: It indicates you have to provide the system path  
• Import: This indicates that the identified or specified POM should be replaced with the dependencies in that POM’s section

**9) Mention how profiles are specified in Maven?**

Profiles are specified in Maven by using a subset of the elements existing in the POM itself.

**10) Explain how you can exclude dependency?**

By using the exclusion element, dependency can be excluded

**11) Mention the difference between Apache Ant and Maven?**

Apache Ant Maven  
• Ant is a toolbox – Maven is a framework  
• Ant does not have formal conventions like project directory structure – Maven has conventions  
• Ant is procedural; you have to tell to compile, copy and compress – Maven is declarative ( information on what to make & how to build)  
• Ant does not have lifecycle; you have to add sequence of tasks manually – Maven has a lifecycle  
• Ant scripts are not reusable – Maven plugins are reusable

**12) In Maven what are the two setting files called and what are their location?**

In Maven, the setting files are called settings.xml, and the two setting files are located at

• Maven installation directory: $M2\_Home/conf/settings.xml  
• User’s home directory: ${ user.home }/ .m2 / settings.xml

# Q) Where is Maven local repository?

The maven local repository is a local folder that is used to store all your project’s dependencies (plugin jars and other files which are downloaded by Maven). In simple, when you build a Maven project, all dependency files will be stored in your Maven local repository.

By default, Maven local repository is default to .m2 folder :

1. Unix/Mac OS X – ~/.m2
2. Windows – C:\Documents and Settings\{your-username}\.m2

**13) List out what are the build phases in Maven?**

Build phases in Maven are

• Validate  
• Compile  
• Test  
• Package  
• Install  
• Deploy

**14) List out the build, source and test source directory for POM in Maven?**

• Build = Target  
• Source = src/main/java  
• Test = src/main/test

**15) Where do you find the class files when you compile a Maven project?**

You will find the class files ${basedir}/target/classes/.

**16) Explain what would the “jar: jar” goal do?**

jar: jar will not recompile sources; it will imply just create a JAR from the target/classes directory considering that everything else has been done

**17) List out what are the Maven’s order of inheritance?**

The maven’s order of inheritance is

• Parent Pom  
• Project Pom  
• Settings  
• CLI parameters

**18) For POM what are the minimum required elements?**

The minimum required elements for POM are project root, modelVersion, groupID, artifactID and version

**19) Explain how you can produce execution debug output or error messages?**

To produce execution debug output you could call Maven with X parameter or e parameter

**20) Explain how to run test classes in Maven?**

To run test classes in Maven, you need surefire plugin, check and configure your settings in setting.xml and pom.xml for a property named “test.”