What is the main difference between xpath and css selector?

Xpath can traverse the elements backward also (along with forward)like we can select the parent of the current element but using css Selector only forward traversing is done.

If there 4 frames in the application . How will you get the 4rth frame ID.

4th frame can be navigated using driver.switchTo().frame(3); //here 3 is index [if the frames are independent]

If the frames are nested then: driver.switchTo().frame(0); driver.switchTo().frame(0); driver.switchTo().frame(0);driver.switchTo().frame(0);

If there is a ok button. How many ways you can click.

1 Using Return Key

//signOnImage.sendKeys(Keys.RETURN);

2 Using JavascriptExecutor

2.1

JavascriptExecutor js = (JavascriptExecutor)driver;

js.executeScript("arguments[0].click();", signOnImage);

2.2

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("document.getElementsByName('login')[0].click()");

3 Using Actions class

3.1

Actions actions = new Actions(driver);

actions.click(signOnImage).perform();3.2

Actions actions = new Actions(driver);

actions.moveToElement(signOnImage).click().perform();3.3

Actions actions = new Actions(driver);

actions.clickAndHold(signOnImage).release().perform();3.4

Actions actions = new Actions(driver);

actions.sendKeys(signOnImage, Keys.RETURN).perform();

4. What is the difference between try-catch block and throws clause.

The try block will execute a sensitive code which can throw exceptions

The catch block will be used whenever an exception (of the type caught) is thrown in the try block

The finally block is called in every case after the try/catch blocks. Even if the exception isn't caught or if your previous blocks break the execution flow.

The throw keyword will allow you to throw an exception (which will break the execution flow and can be caught in a catch block).

The throws keyword in the method prototype is used to specify that your method might throw exceptions of the specified type. It's useful when you have checked exception (exception that you have to handle) that you don't want to catch in your current method.

**throws will pass the error to his caller.**

In the following example, the error occurs in the throwsMethod() but it is handled in the catchMethod

public class CatchThrow {

private static void throwsMethod() throws NumberFormatException {

String intNumber = "5A";

Integer.parseInt(intNumber);

}

private static void catchMethod() {

try {

throwsMethod();

} catch (NumberFormatException e) {

System.out.println("Convertion Error");

}

}

public static void main(String[] args) {

// TODO Auto-generated method stub

catchMethod();

}

}  
5. If there is a abstract class A with methods M1,M2, M3. And M1 and M2 are abstract methods and M3 is not abstract. Class B extends A. For M1 and M2 we get the implementation in next year. How will you use M3 in B class.

Public abstract Class A {

Public void M1();

Public void M2();

Public static void M3(){

System.out.println(“I am concrete”);

}

}

Public abstract class B extends A {

//This class should not implement the methods M1 and M2, then class B should also be abstract

// To access M3 in class B ,it should be made static..so that u can access with classname itself

A.M3();

B.M3();

}

6. If there is a class A extends Class B. If we create a reference of A and object of B , we can call only the method from Class A. If same methods are present in class A and class B.

For the same scenario we can call the common methods present in both the classes and methods present in A why…????

**public** **class** B {

**public** **void** me1() {

System.***out***.println("meow");

}

}

**public** **class** A **extends** B{

**public** **void** me1() {

**super**.me1();

System.***out***.println("cat meow");

}

**public** **static** **void** main(String[]args) {

B bobj = **new** A();

bobj.me1();

}

}

**Console Output:**

meow

cat meow

7. A a1 = new B(); and B b1 = new B();

How the method call will happen at run time and compile time in both the cases..

A a1 = new B(); - method call will happen at run time

B b1 = new B(); - method call will happen at compile time

8. What is spin and re spin in agile and which phase the demo will show to the customer.

Respin is to fix bug encountered during testing . After respin it is necessary to perform retesting to ensure the new product is compliant.The process of getting the multiple build in a single test cycle is known as re-spin.