* SearchContext is the super most interface in selenium, which is extended by another interface called WebDriver.
* All the abstract methods of SearchContext and WebDriver interfaces are implemented in RemoteWebDriver class.
* All the browser related classes such as FirefoxDriver, ChromeDriver etc., extends the RemoteWebdriver class.
* WebDriver defines common methods which all browser classes (such as Firefox, Chrome etc.,) use. All these class methods are derived from WebDriver interface.
* All the [abstract](https://www.softwaretestingmaterial.com/abstraction-in-java/) methods of both the [interfaces](https://www.softwaretestingmaterial.com/interface-in-java/) are implemented in RemoteWebDriver class which is extended by browser classes such as Firefox Driver, Chrome Driver etc.

**ABSTRACTION**

* In Page Object Model design pattern, we write locators (such as id, name, xpath etc.,) in a Page Class. We utilize these locators in tests but we can’t see these locators in the tests. Literally we hide the locators from the tests.
* Abstraction is the methodology of hiding the implementation of internal details and showing the functionality to the users.

### ****INTERFACE****

* Basic statement we all know in Selenium is WebDriver driver = new FirefoxDriver();
* WebDriver itself is an Interface. So based on the above statement WebDriver driver = new FirefoxDriver(); we are initializing Firefox browser using Selenium WebDriver. It means we are creating a reference variable (driver) of the interface (WebDriver) and creating an Object. Here WebDriver is an Interface as mentioned earlier and FirefoxDriver is a class.
* An interface in Java looks similar to a class but both the interface and class are two different concepts. An interface can have methods and variables just like the class but the methods declared in interface are by default abstract. We can achieve 100% abstraction and multiple inheritance in Java with Interface.

### ****INHERITANCE****

* We create a Base Class in the Framework to initialize WebDriver interface, WebDriver waits, Property files, Excels, etc., in the Base Class.
* We extend the Base Class in other classes such as Tests and Utility Class. Extending one class into other class is known as Inheritance.

### ****POLYMORPHISM****

* Combination of overloading and overriding is known as Polymorphism. We will see both overloading and overriding below.
* Polymorphism allows us to perform a task in multiple ways.

### ****METHOD OVERLOADING****

* We use implicit wait in Selenium. Implicit wait is an example of overloading. In Implicit wait we use different time stamps such as SECONDS, MINUTES, HOURS etc.,
* A class having multiple methods with same name but different parameters is called Method Overloading

### ****METHOD OVERRIDING****

* We use a method which was already implemented in another class by changing its parameters. To understand this you need to understand Overriding in Java.
* Declaring a method in child class which is already present in the parent class is called Method Overriding. Examples are get and navigate methods of different drivers in Selenium .

### ****ENCAPSULATION****

* All the classes in a framework are an example of Encapsulation. In POM classes, we declare the data members using @FindBy and initialization of data members will be done using Constructor to utilize those in methods.
* Encapsulation is a mechanism of binding code and data together in a single unit.

### ****WEB ELEMENT:****

* Web element is an interface used to identify the elements in a web page.

### ****WEBDRIVER:****

* WebDriver is an interface used to launch different browsers such as Firefox, Chrome, Internet Explorer, Safari etc.,

### ****FIND BY:****

* FindBy is an annotation used in Page Object Model design pattern to identify the elements.

### ****What is Page Object Model Design Patten (POM):****

Page Object Model is a Design Pattern which has become popular in Selenium Test Automation. It is widely used design pattern in Selenium for enhancing test maintenance and reducing code duplication. Page object model (POM) can be used in any [kind of framework](https://www.softwaretestingmaterial.com/types-test-automation-frameworks/) such as modular, [data-driven](https://www.softwaretestingmaterial.com/data-driven-framework-selenium-webdriver/), keyword driven, hybrid framework etc.

### ****What is Page Factory?****

We have seen that ‘Page Object Model’ is a way of representing an application in a test framework. For every ‘page’ in the application, we create a Page Object to reference the ‘page’ whereas a ‘Page Factory’ is one way of implementing the ‘Page Object Model’.

### ****What is the difference between Page Object Model (POM) and Page Factory?****

Page Object is a class that represents a web page and holds the functionality and members.  
Page Factory is a way to initialize the web elements you want to interact with within the page object when you create an instance of it.