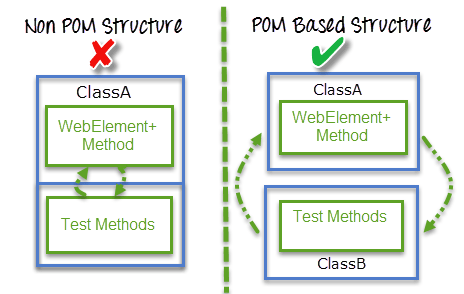
POM:

* Page Object Model is a design pattern to create Object Repository for web UI elements. Under this model, for each web page in the application, there should be corresponding page class. This Page class will find the WebElements of that web page and also contains Page methods which perform operations on those WebElements.
* It helps make the code more readable, maintainable, and reusable



Advantages of POM:

* Page Object Pattern says operations and flows in the UI should be separated from verification. This concept makes our code cleaner and easy to understand.
* The Second benefit is the object repository is independent of test cases, so we can use the same object repository for a different purpose with different tools. For example, we can integrate POM with TestNG/JUnit for functionalTesting and at the same time with JBehave/Cucumber for acceptance testing.
* Code becomes less and optimized because of the reusable page methods in the POM classes.
* Methods get more realistic names which can be easily mapped with the operation happening in UI. i.e. if after clicking on the button we land on the home page, the method name will be like 'gotoHomePage()'.

What is Page Factory?

* Page Factory is an inbuilt Page Object Model concept for Selenium WebDriver but it is very optimized.
* Here as well, we follow the concept of separation of Page Object Repository and Test Methods. Additionally, with the help of PageFactory class, we use annotations @FindBy to find WebElement. We use initElements method to initialize web elements
* @FindBy can accept tagName, partialLinkText, name, linkText, id, css, className, xpath as attributes