**AUTOMATION FRAMEWORK**

**What is Framework in Automation Testing?**

A **Test Automation Framework** is a set of guidelines like coding standards, test-data handling, object repository treatment etc... which when followed during automation scripting produces beneficial outcomes like increased code re-usage, higher portability, reduced script maintenance cost etc. These are just guidelines and not rules; they are not mandatory and you can still script without following the guidelines. But you will miss out on the advantages of having a Framework.

**Features & Benefits of Automation Framework?**

* It reduces the cost as well and lowers risks also.
* It improves test efficiency.
* It lowers the cost of maintenance.
* It maximizes the test coverage and functionality of the application.
* It also defines the reusability of code.
* It helps in avoiding the duplication of test cases as well that is used to automate across the platforms.
* The test framework is used to automate the testing of an independent application.
* It helps the team to organize the test suites and helps in improving the test efficiency.
* It saves a lot of time and effort.
* It is the fastest way to test things quickly.
* It is easy to maintain and scaling is also easy.
* The multiple data sets can be executed easily.
* It helps in avoiding the hard code data.
* Test scripts can be built independently of the platform and applications under test.

**Different Type of Automation Framework**

**A picture containing diagram

Description automatically generated**

**PAGE OBJECT MODEL**

**What is Page Object Model?**

**Page Object Model (POM)** is a design pattern, popularly used in test automation that creates Object Repository for web UI elements. The advantage of the model is that it reduces code duplication and improves test maintenance.

Under this model, for each web page in the application, there should be a corresponding Page Class. This Page class will identify the Web Elements of that web page and also contains Page methods which perform operations on those WebElements. Name of these methods should be given as per the task they are performing.

A picture containing graphical user interface

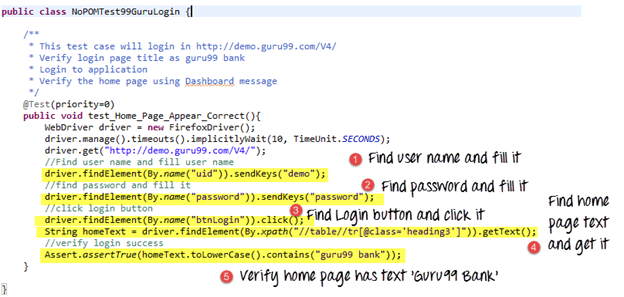
Description automatically generated

**Why Page Object Model?**

Starting an UI Automation in Selenium WebDriver is NOT a tough task. You just need to find elements, perform operations on it.

What is Linux Linux Beginner Tutorial

Consider this simple script to login into a website

[](https://www.guru99.com/images/AdvanceSelenium/071514_0722_PageObjectM1.png)

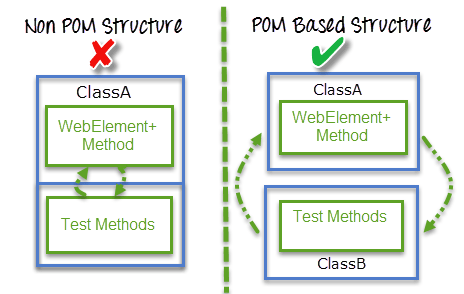
As you can observe, all we are doing is finding elements and filling values for those elements.

This is a small script. Script maintenance looks easy. But with time test suite will grow. As you add more and more lines to your code, things become tough.

The chief problem with script maintenance is that if 10 different scripts are using the same page element, with any change in that element, you need to change all 10 scripts. This is time consuming and error prone.

A better approach to script maintenance is to create a separate class file which would find web elements, fill them or verify them. This class can be reused in all the scripts using that element. In future, if there is a change in the web element, we need to make the change in just 1 class file and not 10 different scripts.

This approach is called Page Object Model in Selenium. It helps make the code  more readable, maintainable, and reusable.

[](https://www.guru99.com/images/AdvanceSelenium/071514_0722_PageObjectM2.png)

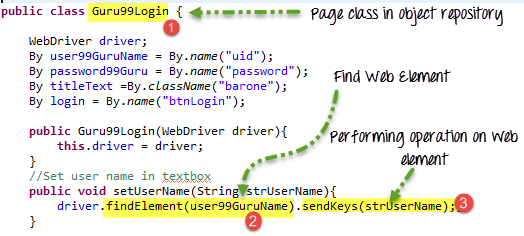
**Advantages of POM**

1. Page Object Design Pattern says operations and flows in the UI should be separated from verification. This concept makes our code cleaner and easy to understand.
2. 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 Page Object Model in Selenium with TestNG/JUnit for functional Testing and at the same time with JBehave/Cucumber for acceptance testing.
3. Code becomes less and optimized because of the reusable page methods in the POM classes.
4. 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()'.

**How to implement POM?**

Simple POM:

It's the basic structure of Page object model framework where all Web Elements of the **AUT** and the method that operate on these Web Elements are maintained inside a class file.A task like verification should be separate as part of Test methods.

[](https://www.guru99.com/images/AdvanceSelenium/071514_0722_PageObjectM4.png)

**Complete Example**

**TestCase:** Go to Guru99 Demo Site.

|  |  |
| --- | --- |
| Step 1) Go to Guru99 Demo Site | [Page Object Model (POM) & Page Factory in Selenium: Complete Tutorial](https://www.guru99.com/images/AdvanceSelenium/071514_0722_PageObjectM5.png) |
| Step 2) In home page check text **"Guru99 Bank"** is present | [Page Object Model (POM) & Page Factory in Selenium: Complete Tutorial](https://www.guru99.com/images/AdvanceSelenium/071514_0722_PageObjectM6.png) |
| Step 3) Login into application | [Page Object Model (POM) & Page Factory in Selenium: Complete Tutorial](https://www.guru99.com/images/AdvanceSelenium/071514_0722_PageObjectM7.png) |
| Step 4) Verify that the Home page contains text as "Manger Id: demo" | [Page Object Model (POM) & Page Factory in Selenium: Complete Tutorial](https://www.guru99.com/images/AdvanceSelenium/071514_0722_PageObjectM8.png) |

Here are we are dealing with 2 pages

1. Login Page
2. Home Page (shown once you login)

Accordingly, we create 2 POM in Selenium classes and one test case for verification where both the classes can be inherited and test case can be performed successfully.

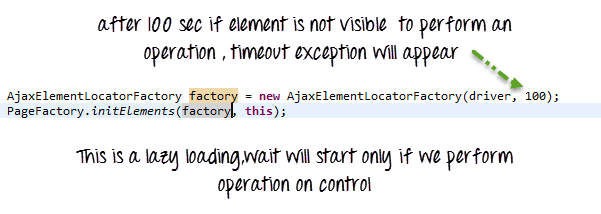
**What is Page Factory in Selenium?**

**Page Factory in Selenium** is an inbuilt Page Object Model framework concept for Selenium WebDriver but it is very optimized. It is used for initialization of Page objects or to instantiate the Page object itself. It is also used to initialize Page class elements without using "FindElement/s."

Here as well, we follow the concept of separation of Page Object Repository and Test Methods. Additionally, with the help of class PageFactory in Selenium, we use annotations **@FindBy** to find WebElement. We use initElements method to initialize web elements

[](https://www.guru99.com/images/AdvanceSelenium/071514_0722_PageObjectM9.png)

**@FindBy** can accept **tagName, partialLinkText, name, linkText, id, css, className, xpath**as attributes.

[](https://www.guru99.com/images/AdvanceSelenium/071514_0722_PageObjectM11.png)

**Summary**

1. Page Object Model in Selenium WebDriver is an Object Repository design pattern.
2. Selenium page object model creates our testing code maintainable, reusable.
3. Page Factory is an optimized way to create object repository in Page Object Model framework concept.