**Also called page chaining model because all the pages are interconnected to each other):**

* **Page object model is a design pattern, it’s an approach. It’s not frame work.**
* **We will integrate with some Data Driven which is called Data Driven Frame Work.**
* **We will use TestNG for writing the Test cases.**
* **Suppose we have an application with different pages like HOMEPAGE, LOGIN, REGISTRATION, SEARCHPAGE, ADDTOCARTPAGE ETC. So, we have this dynamic page are available.**
* **So, what exactly the concept of POM?**
* **1. For each & every page, we have to create a separate Java Class…..logInPage.java, homepage.java.**
* **If we have 100 pages of application then we have to create 100 pages of java class.**
* **Then 1st we have to design our objects (WebObjects/WebElements).**
* **What are login page elements---user name, password, submit button etc.**
* **Then we have to define Actions/Methods—for that particular’s features---login method—clickOnForgotPassword, clickOnForgotPassword, clickOnSignUpButton etc.**
* **1st layer is my page layer.**
* **Object Repository----collections of all the web objects or web elements. We can use the same object repository for different purpose.**
* **Then we create different Test Layer for each and every class. This Test will be written with the help of TestNG—**
* **Before this layer we have to create TestBase.java---this is the parent class of all the classes.**
* **In TestBaseClass what are the feature are available.**
* **Some different initialization I do over here….I want to initialize my webdriver.**
* **Properties file**
* **Config.properties**

**What is Page Factory?**

**Page Factory is a class provided by Selenium WebDriver to support Page Object Design patterns. In Page Factory, testers use @FindBy annotation.**

**1. The initElements method is used to initialize web elements.**

**2. @FindBy: An annotation used in Page Factory to locate and declare web elements using different**

**locators.**