**Page object Design Pattern**

Page object model is used to make the code reusable. It helps in code readability.

Instead of writing code wot identify element, open a browser and perform test cases in a single file.

We can use pom to create multiple classes and use them in the test case.

Let us take an example:

Public class tesing{

@Test

Public static void main(String [] args){

WebDriver driver = new EdgeDriver();

System.setProperty(“webdriver.edge.driver”,”<path of the driver>”);

Driver.get(“https://www.google.co.in”);

Driver.findElement(By.xpath(“<path of xpath>”)).sendKeys(“<search>”);

Driver.findElement(By.xpath(“<path of xpath>”)).submit();

}

}

The above program is consists of opeing a web browser and testing on web elements.

By using pom the above code will be converted into following one:

Public class google\_page{

WebDriver driver;

Public google\_search(WebDriver driver) {

This.driver;

}

By search=By.xpath(“<path of search box>”);

Public WebElement search(){

Return driver.findElement(search);

}

}

Then we create another class for testing

Public class test\_search{

@Test

Public void test(){

System.setProperty(“webdriver.edge.driver”,”<path of the driver>”);

WebDriver driver=new EdgeDriver();

Google\_page gp = new Google\_page(driver);

Gp.search().sendKeys(“<element to be searched>”);

Gp.search().submit();

}

}