	Course Name: Design Patterns/Thinking LAB	**EXPERIMENT NO. 9**	
	Course Code: 20CP210P **Faculty: Dr. Ketan Sabale**	**Branch: CSE**	**Semester: IV**
Submitted by: Rhythm Shah **Roll no:** 22BCP071			

Objective: To familiarize students with standard Structural design patterns.
Experiment: Explain the Flyweight design pattern and write a program using any object-oriented programming language to demonstrate the working of Flyweight design pattern.

Theory: -

Using a structural design technique called flyweight, you may fit more objects into the RAM that is available by sharing common bits of state rather than storing all of the data in each object.

The key idea behind the Flyweight pattern is to use shared objects to support large numbers of fine-grained objects efficiently.

Implementation: -

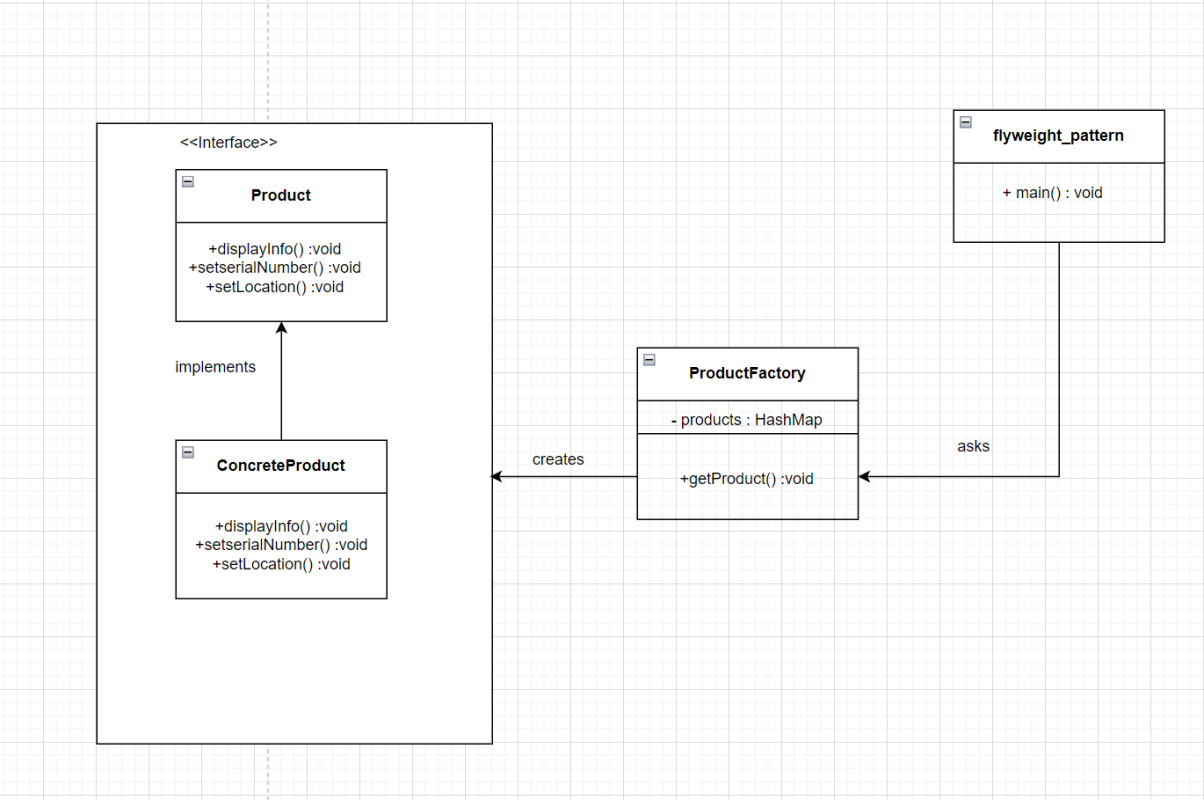
Here an interface product is created which hase three methods which are displayInfo() , setSerialNumber() and setLocation().

Class concrete Product implements Product these class represents individual products.

I have implemented for Inventory Management System in which In an inventory management system for a retail store pattern to manage product

information. Flyweight objects represents shared attributes of products while individual product instances only store unique attributes.

UML Diagram: -



Code: -

```
import java.util.HashMap;

interface Product {
    void displayInfo();
    void setSerialNumber(String serialNumber);
    void setLocation(String location);
}

// Concrete product class representing individual products
class ConcreteProduct implements Product {
    // Unique attributes
    private String key;
    private String serialNumber;
    private String location;
```

```
    public ConcreteProduct(String key) {
        this.key = key;
    }

    public void setSerialNumber(String serialNumber) {
        this.serialNumber = serialNumber;
    }

    public void setLocation(String location) {
        this.location = location;
    }

    public void displayInfo() {
        System.out.println("Product : DisplayInfo() [Key : " + key + ",
serialNumber : " + serialNumber + ", location: " + location + "]");
    }
}

class ProductFactory {
    private static final HashMap<String, Product> productCache = new
HashMap<>();

    public static Product getProduct(String key) {
        Product product = productCache.get(key);
        if (product == null) {
            // Create a new flyweight product
            product = new ConcreteProduct(key);
            productCache.put(key, product);
            System.out.println("Creating product  : " + key);
        }
        return product;
    }
}

// Client code
public class flyweight_pattern {
    public static void main(String[] args) {
        // Simulating products in the inventory
        Product product1 = ProductFactory.getProduct("Electronics");
        Product product2 = ProductFactory.getProduct("Clothing");
        Product product3 = ProductFactory.getProduct("Electronics");
        Product product4 = ProductFactory.getProduct("Clothing");

        // Set serial numbers and locations using the Product interface
        product1.setSerialNumber("0001");
        product1.setLocation("Warehouse A");
```

```
        product2.setSerialNumber("0002");
        product2.setLocation("Warehouse B");

        product3.setSerialNumber("0003");
        product3.setLocation("Warehouse A");

        product4.setSerialNumber("0004");
        product4.setLocation("Warehouse B");

        // Display product information
        product1.displayInfo();
        product2.displayInfo();
        product3.displayInfo();
        product4.displayInfo();
    }
}
```

Output: -

```
Creating product  : Electronics
Creating product  : Clothing
Product : DisplayInfo() [Key : Electronics, serialNumber : 0003, location: Warehouse A]
Product : DisplayInfo() [Key : Clothing, serialNumber : 0004, location: Warehouse B]
Product : DisplayInfo() [Key : Electronics, serialNumber : 0003, location: Warehouse A]
Product : DisplayInfo() [Key : Clothing, serialNumber : 0004, location: Warehouse B]
PS E:\Fourth sem\Design pattern lab>
```