Experiment – 10

Aim –Flywieght design pattern (Structural pattern)

Concept -**Flyweight** is a structural design pattern that lets you fit more objects into the available amount of RAM by sharing common parts of state between multiple objects instead of keeping all of the data in each object.

Recipe.java

```
public interface Recipe {
    void food();
 }
```

Pizza.java

```
public class Pizza implements Recipe {
    private String toppings;


    public Pizza(String toppings){
       this.toppings = toppings;
    }

    @Override
    public void food() {
       System.out.println("Pizza with " +toppings);
    }
}
```

PizzaFactory.java

```
import java.util.HashMap;

public class PizzaFactory {

   private static final HashMap hm = new HashMap();

   public static Recipe getPizza(String toppings) {
      Pizza p = (Pizza)hm.get(toppings);

      if(p == null) {
         p = new Pizza(toppings);
          hm.put(toppings, p);
         System.out.println("Creating Pizza of toppings :   " + toppings);
       }
      return p;
   }
}
```

Demo.java

```
public class Demo {
    private static final String choice[] = { "Mushrooms", "Peppers",
"Pepperoni" };
    public static void main(String[] args) {

       for(int i=0; i < 10; ++i) {
          Pizza p = (Pizza)PizzaFactory.getPizza(getRandomChoice());
          p.food();
       }
    }
    private static String getRandomChoice() {
       return choice[(int)(Math.random()*choice.length)];
    }
 }
```

OUTPUT :

```
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community
Creating Pizza of toppings :  Peppers
Pizza with Peppers
Creating Pizza of toppings :  Mushrooms
Pizza with Mushrooms
Pizza with Peppers
Pizza with Peppers
Pizza with Mushrooms
Pizza with Mushrooms
Creating Pizza of toppings :  Pepperoni
Pizza with Pepperoni
Pizza with Pepperoni
Pizza with Peppers
Pizza with Pepperoni
```