Mentor : Gladden Rumao
Topic : Singleton Design Pattern
Date : 22-07-2023

Implementation of Singleton Design Pattern :

Class Pizza (Singleton Class - Lazy way of creating Object - Synchronized for thread safety.

```
public class Pizza {
    private static Pizza pizza;
    //creating object with constructor
    private Pizza(){
        //constructor
    }
    //Lazy way of creating singleton object
    public static Pizza getPizza(){
        synchronized(Pizza.class){
            if( pizza==null ){
                pizza = new Pizza();
            }
        }
        return pizza;
    }
}
```

Class Cake (Singleton Class - Eager way of creating Object)

```java
public class Cake {
    private static Cake cake = new Cake();
    private Cake(){
        //constructor
    }
    //Eager way of creating singleton object
    public static Cake getCake(){
        return cake;
    }
}
```

Class Testing for main method :

```java
public class Testing {
    public static void main(String[] args) {
        //Pizza pizza1 = new Pizza();
//can't create object with constructor as private
//create object with method
        Pizza pizza1 = Pizza.getPizza();
        System.out.println(pizza1.hashCode());
        Pizza pizza2 = Pizza.getPizza();
        System.out.println(pizza2.hashCode());

        Cake cake1 = Cake.getCake();
        System.out.println(cake1.hashCode());
        Cake cake2 = Cake.getCake();
        System.out.println(cake2.hashCode());
    }
}
```

[redacted]