	Course Name: Design Patterns/Thinking LAB	**EXPERIMENT NO. 4**	
	Course Code: 20CP210P **Faculty: Dr. Ketan Sabale**	**Branch: CSE**	**Semester: IV**
Submitted by: Rhythm Shah **Roll no:** 22BCP071			

Objective:

- To familiarize students with standard Creational design patterns.

Experiment:

- Explain the prototype design pattern and write a program using any object-oriented programming language to demonstrate the working of prototype design pattern.

Theory:

- Prototype pattern refers to creating duplicate object while keeping performance in mind. This type of design pattern comes under creational pattern as this pattern provides one of the best ways to create an object.
- This pattern involves implementing a prototype interface which tells to create a clone of the current object. This pattern is used when creation of object directly is costly so if don't want to load the data again and again we can use this by implanting this design pattern.
- There are two types of Cloning :
  1) Deep Cloning – All the data changed in one shop will not implement to other shop.
  2) Shallow Cloning – All the data changed in one shop will be changed similarly in second shop.

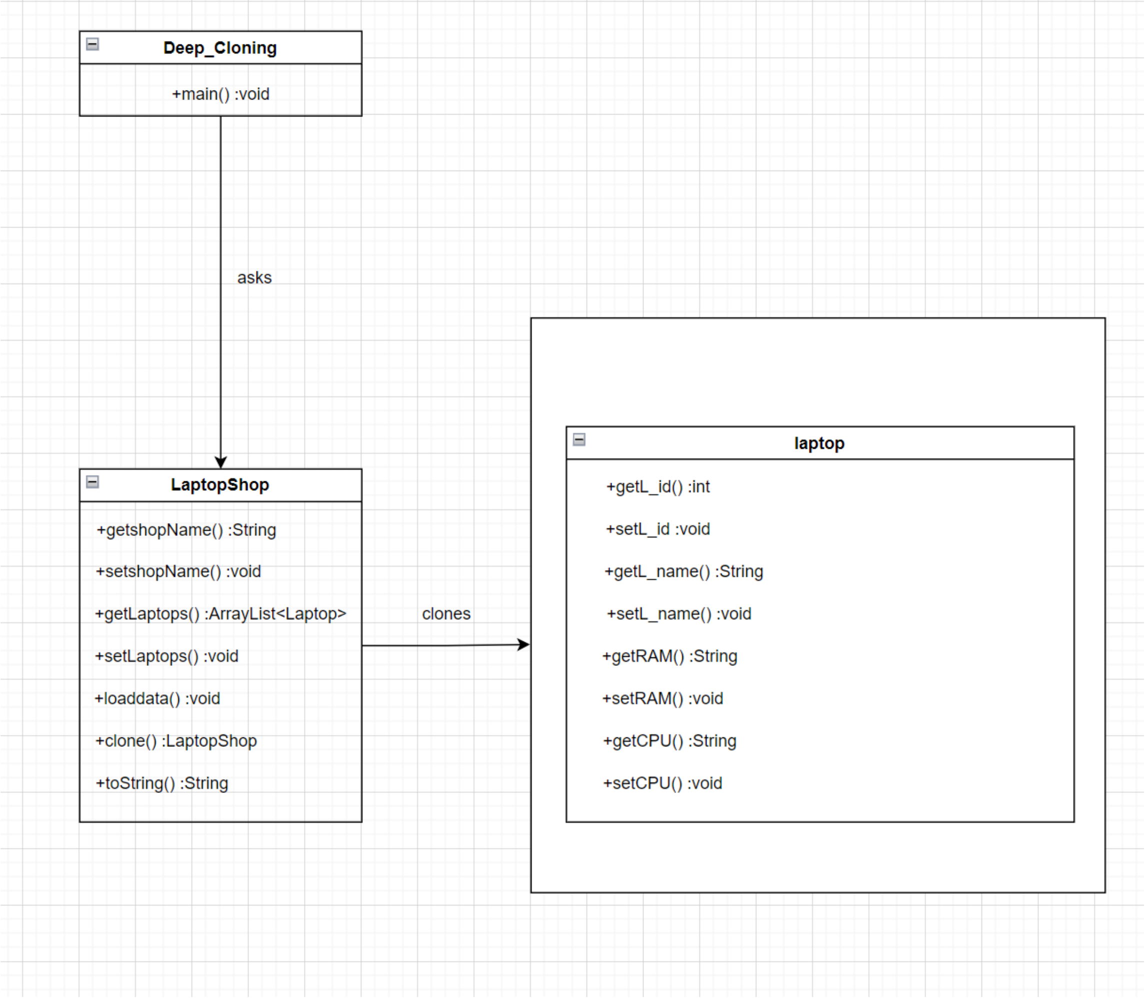
Example(Implementation):

- Here I have created a prototype design pattern for two Laptop Shops. In which I will create 4 methods in the class Laptop which are L_Id, L_Name, RAM and CPU.

- There is one more class which is LaptopShop in which we will implement cloneable and apply get set methods in it for shop name and getLaptops() and a method to load data by loaddata(). We will create a clone of laptopshop in which items which are same we don't need to load the data again in main method.

1)For Deep Cloning

UML Diagram:



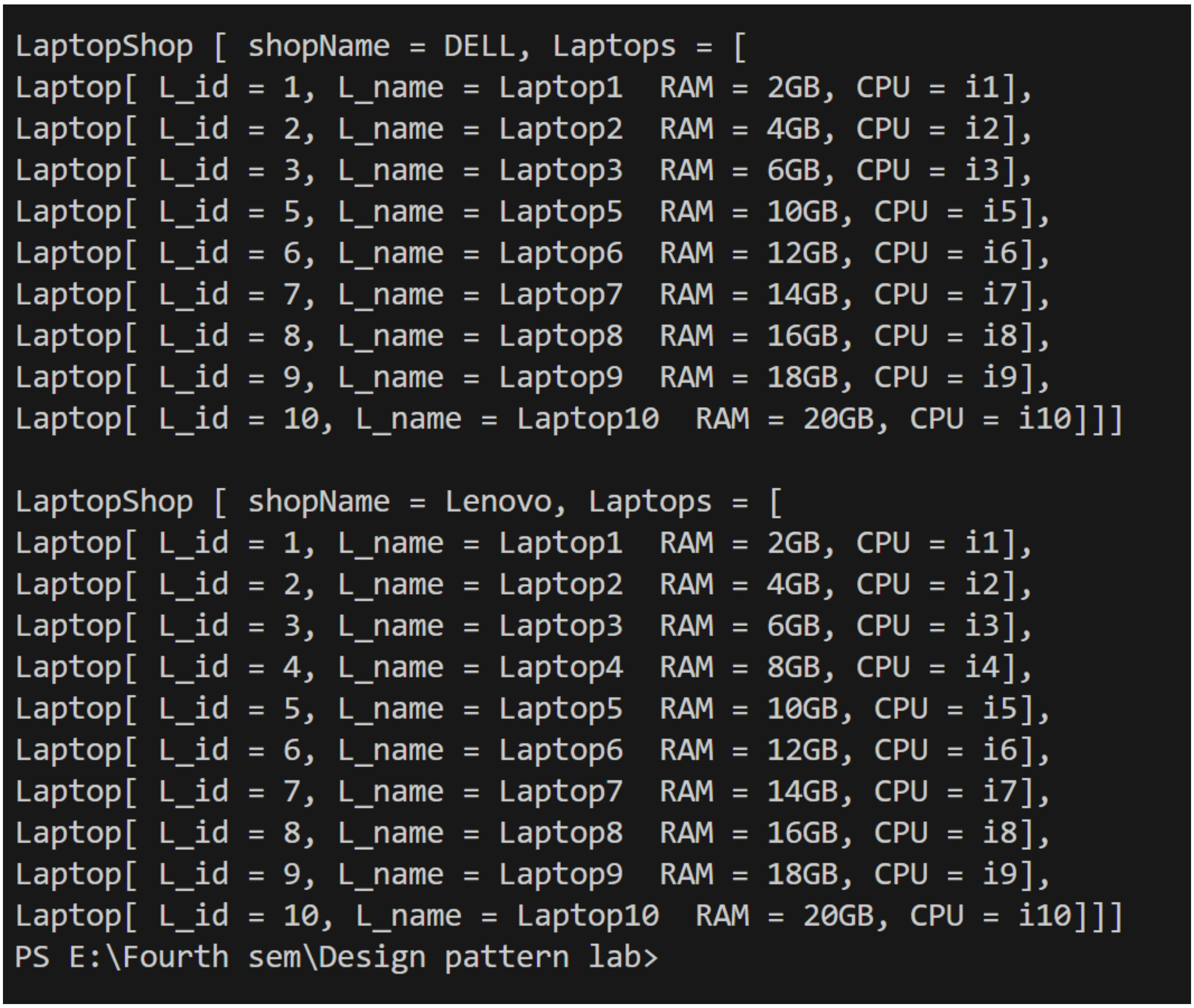
Code:

```
import java.util.ArrayList;
class Laptop {
    private int L_id;
    private String L_name;
    private String RAM;
    private String CPU;

      public int getL_id() {
             return L_id;
     }
      public void setL_id(int L_id) {
             this.L_id = L_id;
     }
      public String getL_name() {
             return L_name;
     }
       public void setL_name(String L_name) {
             this.L_name = L_name;
     }
    public String getRAM() {
        return RAM;
    }
    public void setRAM(String RAM){
    this.RAM = RAM;
    }
    public String getCPU() {
    return CPU;
    }
    public void setCPU(String CPU) {
    this.CPU = CPU;
    }

            public String toString() {
        return " \nLaptop[ L_id = " + L_id + ", L_name = " + L_name + "  RAM = " +
RAM + ", CPU = " + CPU + "]";
            }
}
class LaptopShop implements Cloneable{
            private String shopName;
            ArrayList<Laptop> Laptops = new ArrayList();
            public String getShopName() {
                        return shopName;
            }
            public void setShopName(String shopName) {
```

```
        this.shopName = shopName;
    }
    public ArrayList<Laptop> getLaptops() {
        return Laptops;
    }
    public void setLaptops(ArrayList<Laptop> laptops) {
        this.Laptops = laptops;
    }

    public void loadData() {
        for(int i = 1;i<=10; i++)
        {
            Laptop L = new Laptop();
    String name =  "i" + i;
    String name1 =  2*i + "GB";
    L.setCPU(name);
    L.setRAM(name1);
            L.setL_id(i);
            L.setL_name("Laptop"+i);
            getLaptops().add(L);
        }
    }

    protected LaptopShop clone() throws CloneNotSupportedException {
        LaptopShop shop = new LaptopShop();

        for(Laptop L:getLaptops())
        {
            shop.getLaptops().add(L);
        }
        return shop;
    }

    public String toString() {
        return "LaptopShop [ shopName = "  + shopName + ", Laptops = " +
Laptops + "]";
    }
}
public class Deep_Cloning {

    public static void main(String[] args) throws CloneNotSupportedException {

         LaptopShop ls = new  LaptopShop();
        ls.setShopName("DELL");
        ls.loadData();
   System.out.println();
```

```
                LaptopShop ls1 = ls.clone();
                ls.getLaptops().remove(3);
                ls1.setShopName("Lenovo");
                System.out.println(ls);
                System.out.println();
                System.out.println(ls1);
        }

}
```

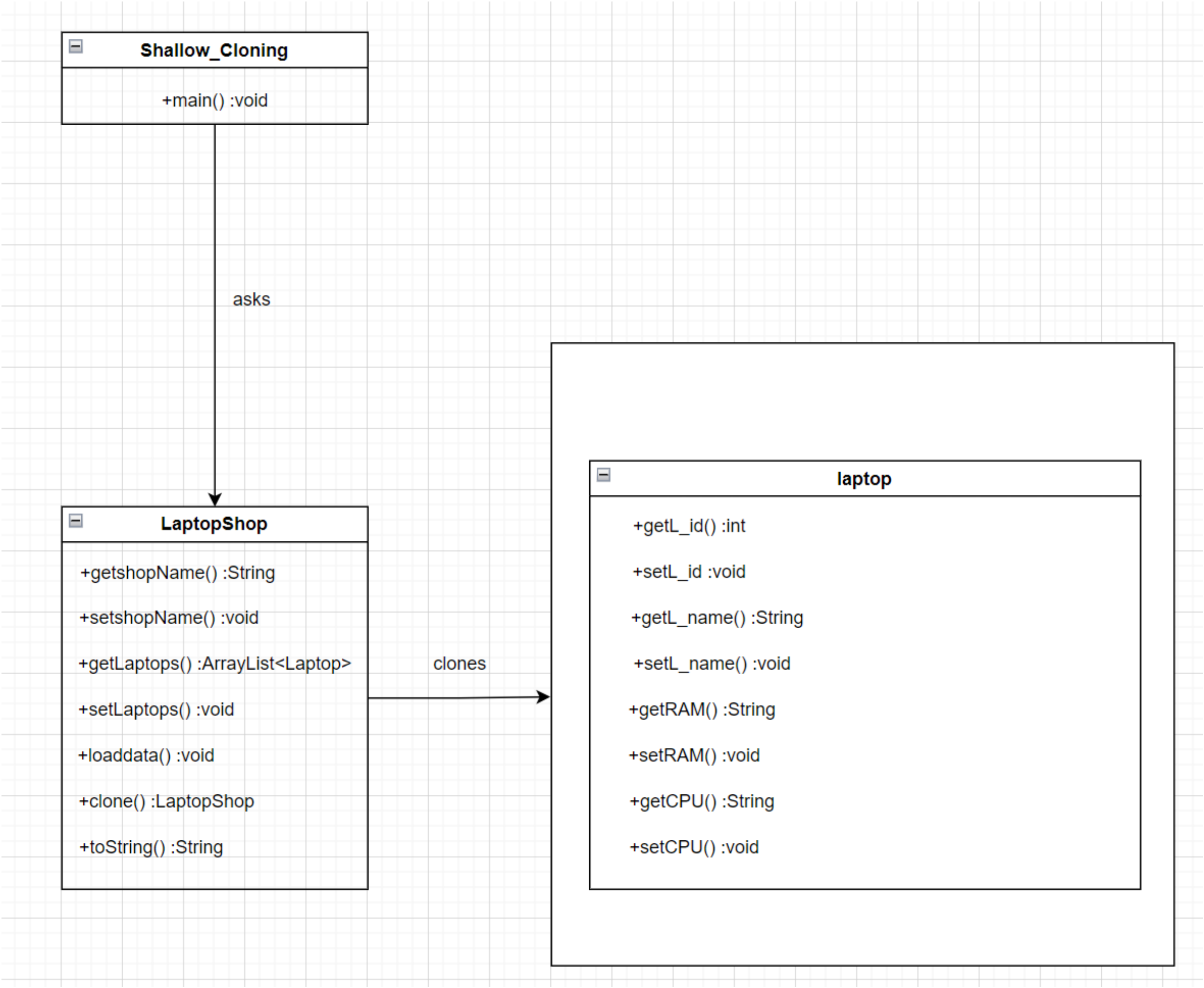
Output



```
LaptopShop [ shopName = DELL, Laptops = [
Laptop[ L_id = 1, L_name = Laptop1  RAM = 2GB, CPU = i1],
Laptop[ L_id = 2, L_name = Laptop2  RAM = 4GB, CPU = i2],
Laptop[ L_id = 3, L_name = Laptop3  RAM = 6GB, CPU = i3],
Laptop[ L_id = 5, L_name = Laptop5  RAM = 10GB, CPU = i5],
Laptop[ L_id = 6, L_name = Laptop6  RAM = 12GB, CPU = i6],
Laptop[ L_id = 7, L_name = Laptop7  RAM = 14GB, CPU = i7],
Laptop[ L_id = 8, L_name = Laptop8  RAM = 16GB, CPU = i8],
Laptop[ L_id = 9, L_name = Laptop9  RAM = 18GB, CPU = i9],
Laptop[ L_id = 10, L_name = Laptop10  RAM = 20GB, CPU = i10]]]

LaptopShop [ shopName = Lenovo, Laptops = [
Laptop[ L_id = 1, L_name = Laptop1  RAM = 2GB, CPU = i1],
Laptop[ L_id = 2, L_name = Laptop2  RAM = 4GB, CPU = i2],
Laptop[ L_id = 3, L_name = Laptop3  RAM = 6GB, CPU = i3],
Laptop[ L_id = 4, L_name = Laptop4  RAM = 8GB, CPU = i4],
Laptop[ L_id = 5, L_name = Laptop5  RAM = 10GB, CPU = i5],
Laptop[ L_id = 6, L_name = Laptop6  RAM = 12GB, CPU = i6],
Laptop[ L_id = 7, L_name = Laptop7  RAM = 14GB, CPU = i7],
Laptop[ L_id = 8, L_name = Laptop8  RAM = 16GB, CPU = i8],
Laptop[ L_id = 9, L_name = Laptop9  RAM = 18GB, CPU = i9],
Laptop[ L_id = 10, L_name = Laptop10  RAM = 20GB, CPU = i10]]]
PS E:\Fourth sem\Design pattern lab>
```

2)For Shallow Cloning

UML Diagram



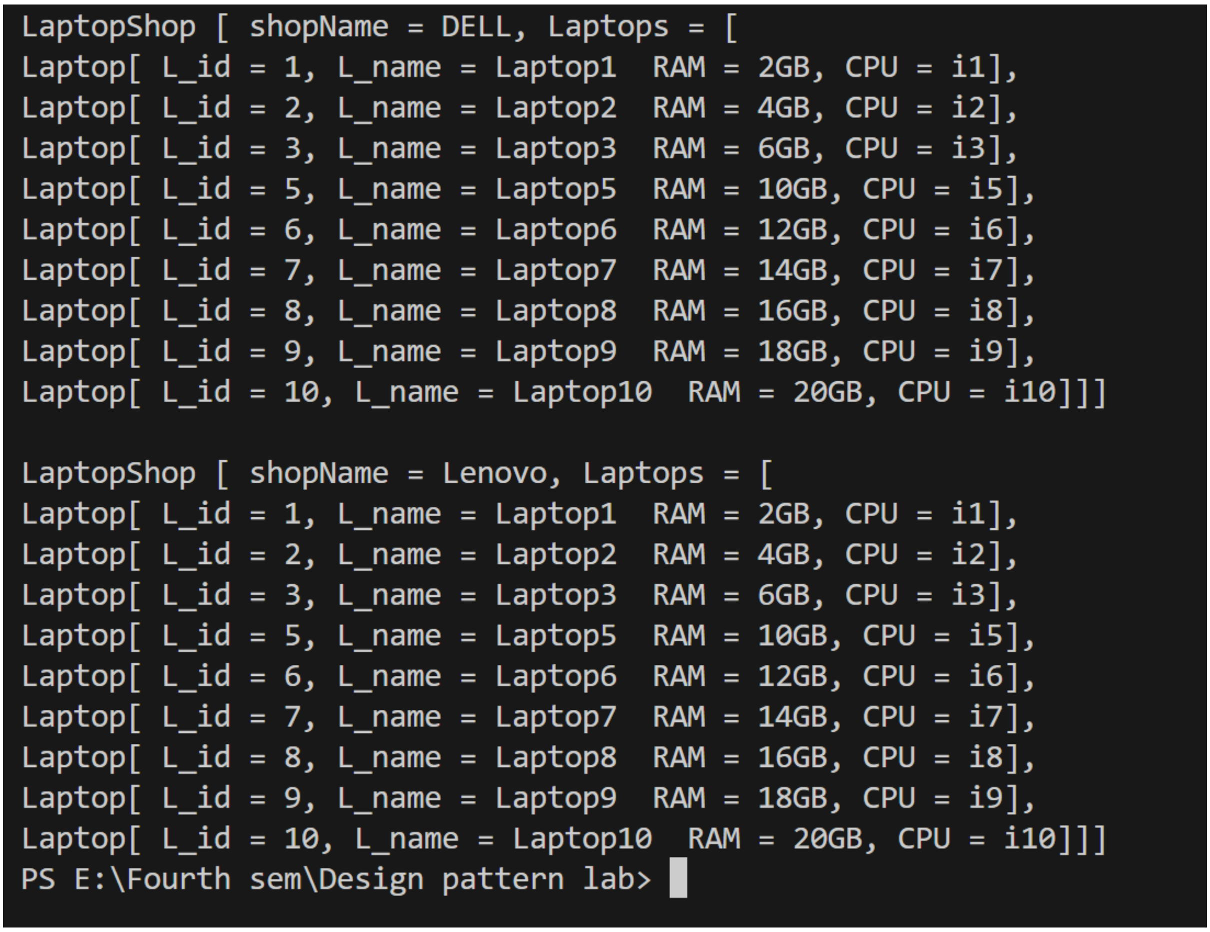
Code

```
import java.util.ArrayList;
class Laptop {
        private int L_id;
        private String L_name;
   private String RAM;
   private String CPU;
        public int getL_id() {
                return L_id;
        }
        public void setL_id(int L_id) {
                this.L_id = L_id;
        }
        public String getL_name() {
                return L_name;
        }
        public void setL_name(String L_name) {
                this.L_name = L_name;
        }
   public String getRAM() {
      return RAM;
   }
   public void setRAM(String RAM){
   this.RAM = RAM;
   }
   public String getCPU() {
   return CPU;
   }
   public void setCPU(String CPU) {
   this.CPU = CPU;
   }

        public String toString() {
      return " \nLaptop[ L_id = " + L_id + ", L_name = " + L_name + "  RAM = " +
RAM + ", CPU = " + CPU + "]";
        }
}
class LaptopShop implements Cloneable{
        private String shopName;
        ArrayList<Laptop> Laptops = new ArrayList();
        public String getShopName() {
                return shopName;
        }
```

```
    public void setShopName(String shopName) {
        this.shopName = shopName;
    }
    public ArrayList<Laptop> getLaptops() {
        return Laptops;
    }
    public void setLaptops(ArrayList<Laptop> laptops) {
        this.Laptops = laptops;
    }

    public void loadData() {
        for(int i = 1;i<=10; i++)
        {
            Laptop L = new Laptop();
    String name =  "i" + i;
    String name1 =  2*i + "GB";
    L.setCPU(name);
    L.setRAM(name1);
            L.setL_id(i);
            L.setL_name("Laptop"+i);
            getLaptops().add(L);
        }
    }

    protected Object clone() throws CloneNotSupportedException {
                return super.clone();
    }
    public String toString() {
        return "LaptopShop [ shopName = "  + shopName + ", Laptops = " +
Laptops + "]";
    }
}
public class Shallow_Cloning {

    public static void main(String[] args) throws CloneNotSupportedException {

         LaptopShop ls = new  LaptopShop();
        ls.setShopName("DELL");
        ls.loadData();
        ls.getLaptops().remove(3);
        System.out.println(ls);

        LaptopShop ls1 = (LaptopShop) ls.clone();
        ls1.setShopName("Lenovo");
        System.out.println();
        System.out.println(ls1);
```

```
            }

        }
```

Output:



```
LaptopShop [ shopName = DELL, Laptops = [
Laptop[ L_id = 1, L_name = Laptop1  RAM = 2GB, CPU = i1],
Laptop[ L_id = 2, L_name = Laptop2  RAM = 4GB, CPU = i2],
Laptop[ L_id = 3, L_name = Laptop3  RAM = 6GB, CPU = i3],
Laptop[ L_id = 5, L_name = Laptop5  RAM = 10GB, CPU = i5],
Laptop[ L_id = 6, L_name = Laptop6  RAM = 12GB, CPU = i6],
Laptop[ L_id = 7, L_name = Laptop7  RAM = 14GB, CPU = i7],
Laptop[ L_id = 8, L_name = Laptop8  RAM = 16GB, CPU = i8],
Laptop[ L_id = 9, L_name = Laptop9  RAM = 18GB, CPU = i9],
Laptop[ L_id = 10, L_name = Laptop10  RAM = 20GB, CPU = i10]]]

LaptopShop [ shopName = Lenovo, Laptops = [
Laptop[ L_id = 1, L_name = Laptop1  RAM = 2GB, CPU = i1],
Laptop[ L_id = 2, L_name = Laptop2  RAM = 4GB, CPU = i2],
Laptop[ L_id = 3, L_name = Laptop3  RAM = 6GB, CPU = i3],
Laptop[ L_id = 5, L_name = Laptop5  RAM = 10GB, CPU = i5],
Laptop[ L_id = 6, L_name = Laptop6  RAM = 12GB, CPU = i6],
Laptop[ L_id = 7, L_name = Laptop7  RAM = 14GB, CPU = i7],
Laptop[ L_id = 8, L_name = Laptop8  RAM = 16GB, CPU = i8],
Laptop[ L_id = 9, L_name = Laptop9  RAM = 18GB, CPU = i9],
Laptop[ L_id = 10, L_name = Laptop10  RAM = 20GB, CPU = i10]]]
PS E:\Fourth sem\Design pattern lab>
```