Name : Anupam Kunwar

Reg : 19BCE1369

CSE1007 – JAVA PROGRAMMING

Lab Exercise on Inheritance

Question 1

Create a TeeShirt class for Toby's Tee Shirt Company. Fields include an order number, size, color, and price. Create set methods for the order number, size, and color and get methods for all four fields. The price is determined by the size: $22.99 for XXL or XXXL, and $19.99 for all other sizes. Create a subclass named CustomTee that descends from TeeShirt and includes a field to hold the slogan requested for the shirt, and include get and set methods this field.

Write an application that creates two objects of each class, and demonstrate that all the methods work correctly. Save the files as TeeShirt.java, CustomTee.java, and DemoTees.java.

Solution :

TeeShirt.java

```
package mypack;

public class TeeShirt{

int ordNo;

String size;

String color;

double price;

public void setOrdNo(int o){

ordNo = o;
```

```
}
public void setSize(String s){
size = s;
}
public void setColor(String c){
color = c;
}
public int getOrdNo(){
return ordNo;
}
public String getSize(){
return size;
}
public String getColor(){
return color;
}
public double getPrice(){
if(size.equals("XXL")||size.equals("XXXL")){
price = 22.99;
}
else{
price = 19.99;
}
return price;
}
}
```

customTees.java

```
package mypack;
public class customTees extends TeeShirt{
String slogan;
public void setSlogan(String s){
slogan = s;
}
public String getSlogan(){
return slogan;
}
}
```

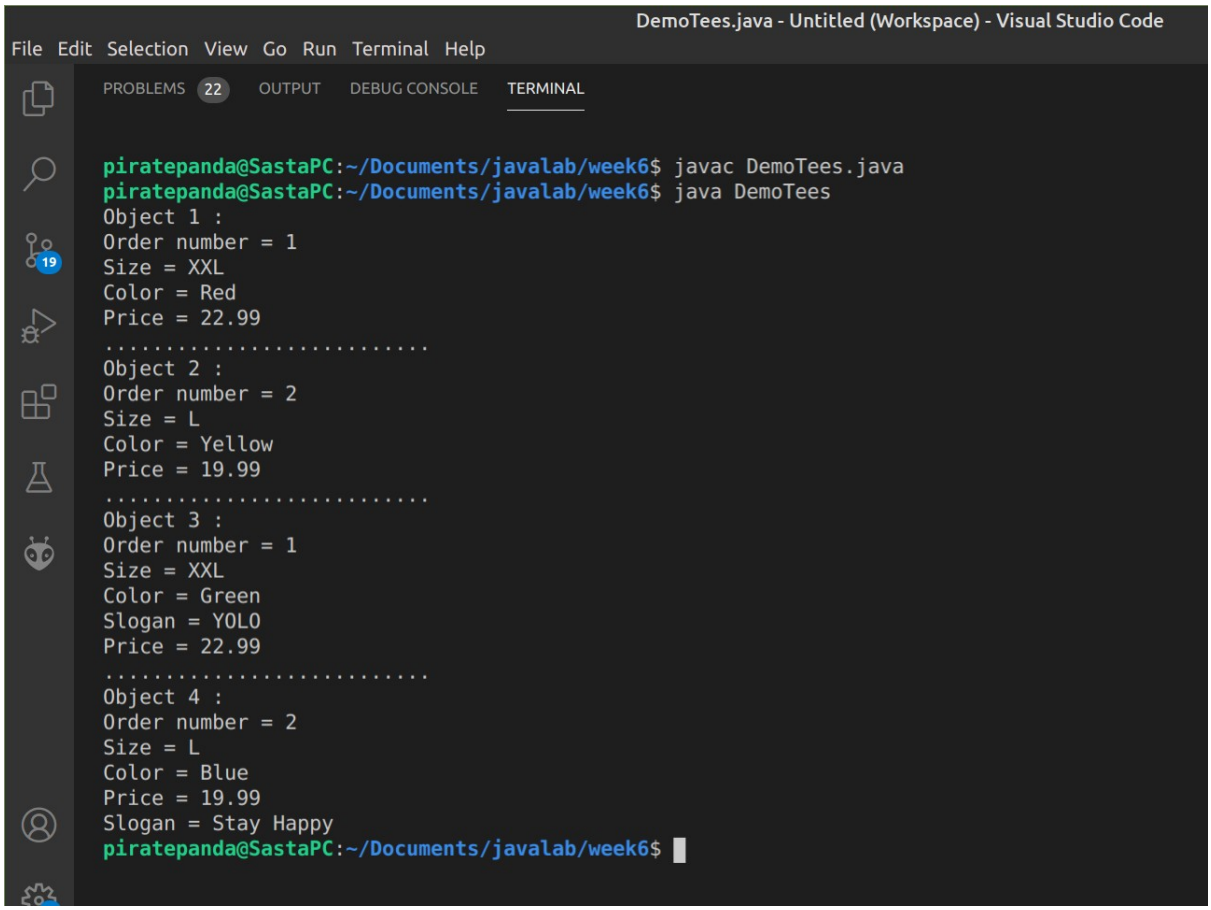
DemoTees.java

```
import mypack.TeeShirt;
import mypack.customTees;
public class DemoTees {
public static void main(String[] args) {
TeeShirt obj1 = new TeeShirt();
TeeShirt obj2 = new TeeShirt();
obj1.setOrdNo(1);
obj1.setSize("XXL");
obj1.setColor("Red");
```

```
obj2.setOrdNo(2);
obj2.setSize("L");
obj2.setColor("Yellow");
System.out.println("Object 1 : ");
System.out.println("Order number = "+obj1.getOrdNo());
System.out.println("Size = "+obj1.getSize());
System.out.println("Color = "+obj1.getColor());
System.out.println("Price = "+obj1.getPrice());
System.out.println(".....");
System.out.println("Object 2 : ");
System.out.println("Order number = "+obj2.getOrdNo());
System.out.println("Size = "+obj2.getSize());
System.out.println("Color = "+obj2.getColor());
System.out.println("Price = "+obj2.getPrice());
System.out.println(".....");
customTees obj3 = new customTees();
customTees obj4 = new customTees();
obj3.setOrdNo(1);
obj3.setSize("XXL");
obj3.setColor("Green");
obj3.setSlogan("YOLO");
obj4.setOrdNo(2);
obj4.setSize("L");
obj4.setColor("Blue");
obj4.setSlogan("Stay Happy");
System.out.println("Object 3 : ");
```

```
System.out.println("Order number = "+obj3.getOrdNo());
System.out.println("Size = "+obj3.getSize());
System.out.println("Color = "+obj3.getColor());
System.out.println("Slogan = "+obj3.getSlogan());
System.out.println("Price = "+obj3.getPrice());
System.out.println(".....");
System.out.println("Object 4 : ");
System.out.println("Order number = "+obj4.getOrdNo());
System.out.println("Size = "+obj4.getSize());
System.out.println("Color = "+obj4.getColor());
System.out.println("Price = "+obj4.getPrice());
System.out.println("Slogan = "+obj4.getSlogan());
}
}
```

Output :



Question 2

The developers of a free online game named Sugar Smash have asked you to develop a class named SugarSmashPlayer that holds data about a single player. The class contains the following fields: the player's integer ID number, a String screen name, and an array of integers that stores the highest score achieved in each of 10 game levels. Include get and set methods for each field. The get and set methods for the scores should each require two parameters —one that represents the score achieved and one that represents the game level to be retrieved or assigned. Display an error message if the user attempts to assign or retrieve a score from a level that is out of range for the array of scores. Additionally, no level except the first one should be set unless the user has earned at least 100 points at each previous level. If a user tries to set a score for a level that is not yet available, issue an error message.

Create a class named PremiumSugarSmashPlayer that descends from SugarSmashPlayer. This class is instantiated when a user pays $2.99 to have access to 40 additional levels of play. As in the free version of the game, a user cannot set a score for a level unless the user has earned at least 100 points at all previous levels.

Create a program that instantiates several objects of each type and demonstrates the methods. Save the files as SugarSmashPlayer.java, PremiumSugarSmashPlayer.java, and DemoSugarSmash.java

Solution :

SugarSmashPlayer.java:

```
public class SugarSmashPlayer {
int id;
String name;
int levels;
int filled=0;
int[] arr;
public SugarSmashPlayer(){
levels=10;
}
public void fixArray(){
arr = new int[levels];
int i=0;
for(i=0;i<levels;i++)
arr[i]=0;
}
public void setId(int i){
```

```
id=i;
}
public void setName(String s){
name=s;
}
public int getId(){
return id;
}
public String getName(){
return name;
}
public void setScore(int score,int level){
if(level>levels||level<0){
System.out.println("Level out of range");
}
else if(level>0){
if(arr[level-1]<100){
System.out.println("Error : Previous level less than 100");
}
else{
arr[level] = score;
filled++;
}
}
else{
arr[level] = score;
```

```
}
}
public void getScore(int level){
if(level>levels||level<0||level>filled){
System.out.println("Level out of range");
}
else{
System.out.println(arr[level]+"\n");
}
}
}
```

PremiumSugarSmashPlayer.java

```
public class PremiumSugarSmashPlayer extends
SugarSmashPlayer {
public PremiumSugarSmashPlayer(){
levels=50;
}
}
```

DemoSugarSmash.java

```
import java.util.Scanner;
public class DemoSugarSmash extends
PremiumSugarSmashPlayer {
public static void main(String[] args) {
Scanner in = new Scanner(System.in);
```

```
int n,score,le;
SugarSmashPlayer obj1 = new SugarSmashPlayer();
obj1.fixArray();
obj1.setId(1234);
obj1.setName("Rohit");
System.out.println("Enter level : ");
le = in.nextInt();
System.out.println("Enter score : ");
score = in.nextInt();
obj1.setScore(score,le);
System.out.println("Name = "+obj1.getName());
System.out.println("ID = "+obj1.getId());
System.out.println("Enter level of which you need to know score : ");
n = in.nextInt();
System.out.print("Score at level "+n+" = ");
obj1.getScore(n);
System.out.println("Premium Player\n==================");
PremiumSugarSmashPlayer obj2 = new
PremiumSugarSmashPlayer();
obj2.fixArray();
obj2.setId(1235);
obj2.setName("Rahul");
System.out.println("Enter level : ");
le = in.nextInt();
System.out.println("Enter score : ");
score = in.nextInt();
```

```
obj2.setScore(score,le);
System.out.println("Name = "+obj2.getName());
System.out.println("ID = "+obj2.getId());
System.out.println("Enter level of which you need to know score : ");
n = in.nextInt();
System.out.print("Score at level "+n+" = ");
obj2.getScore(n);
}
}
```

Output :

Question 3

Create a class named CollegeCourse that includes data fields that hold the department (for example, ENG), the course number (for example, 101), the credits (for example, 3), and the fee for the course (for example, $360). All of the fields are required as arguments to the constructor, except for the fee, which is calculated at $120 per credit hour. Include a display() method that displays the course data. Create a subclass named LabCourse that adds $50 to the course fee. Override the parent class display() method to indicate that the course is a lab course and to display all the data.

Write an application named UseCourse that prompts the user for course information. If the user enters a class in any of the following departments, create a LabCourse: BIO, CHM, CIS, or PHY. If the user enters any other department, create a CollegeCourse that does not include the lab fee. Then display the course data. Save the files as CollegeCourse.java, LabCourse.java, and UseCourse.java.

Solution:

CollegeCourse.java

```
package mypack;
public class CollegeCourse {
String department;
int courseNumber;
int courseFees;
int credits;
public CollegeCourse(String d,int c,int cr){
department = d;
courseNumber = c;
credits = cr;
```

```
}
public void display(){
courseFees = credits*120;
System.out.println("Department = "+department);
System.out.println("Course Number = "+courseNumber);
System.out.println("Course Credits = "+credits);
System.out.println("Course Fees = "+courseFees);
}
}
```

LabCourse.java

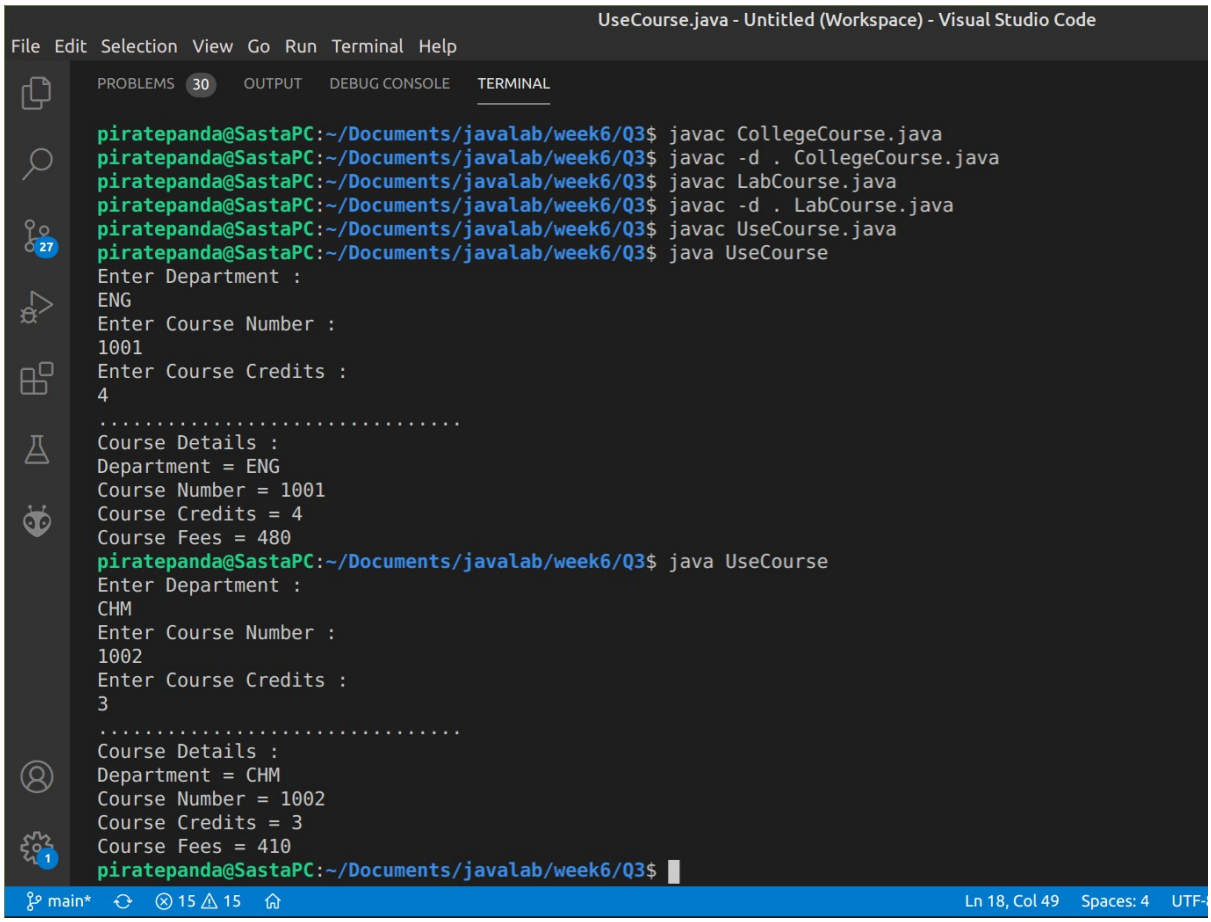
```
package mypack;
import mypack.CollegeCourse;
public class LabCourse extends CollegeCourse{
public LabCourse(String d,int c,int cr){
super(d, c, cr);
}
public void display(){
courseFees = credits*120;
courseFees+=50;
System.out.println("Department = "+department);
System.out.println("Course Number = "+courseNumber);
System.out.println("Course Credits = "+credits);
System.out.println("Course Fees = "+courseFees);
}
}
```

UseCourse.java

```
import java.util.Scanner;
import mypack.CollegeCourse;
import mypack.LabCourse;
public class UseCourse {
public static void main(String[] args) {
Scanner in = new Scanner(System.in);
String d;
int cn,cr,cf;
System.out.println("Enter Department : ");
d = in.next();
System.out.println("Enter Course Number : ");
cn = in.nextInt();
System.out.println("Enter Course Credits : ");
cr = in.nextInt();
System.out.println(".....");

System.out.println("Course Details : ");

if (d.equals("BIO")||d.equals("CHM")||d.equals("CIS")||
d.equals("PHY")) {
LabCourse obj1 = new LabCourse(d, cn, cr);
obj1.display();
}
else{
CollegeCourse obj2 = new CollegeCourse(d, cn, cr);
```

```
obj2.display();
}
}
}
```

Output :



Question 4

Create an abstract NewspaperSubscription class with fields for the subscriber name, address, and rate. Include get and set methods for the name field and get methods for the address and subscription rate; the setAddress() method is abstract. Create two subclasses named PhysicalNewspaperSubscription and OnlineNewspaperSubscription. The parameter for the setAddress() method of the PhysicalNewspaperSubscription class must contain at least one digit; otherwise, an error message is displayed and the subscription rate is set to 0. If the address is valid, the subscription rate is assigned $15. The parameter for the setAddress() method of the OnlineNewspaperSubscription class must contain an at sign (@) or an error message is displayed. If the address is valid, the subscription rate is assigned $9.

Finally, write an application that declares several objects of both subscription subtypes and displays their data fields. Save the files

as NewspaperSubscription.java,
PhysicalNewspaperSubscription.java,
OnlineNewspaperSubscription.
java, and DemoSubscriptions.java.

Solution :

NewspaperSubscription.java

```
package mypack;
public abstract class NewspaperSubscription {
String name;
String address;
double rate;

public String getName() {
return name;
}

public void setName(String n) {
name = n;
}

public String getAddress() {
return address;
}

public double getRate() {
return rate;
```

```
}

public abstract void setAddress(String s);
}
```

OnlineNewspaperSubscription.java

```
package mypack;

public class OnlineNewspaperSubscription extends
NewspaperSubscription {
public void setAddress(String a) {
if (a.contains("@")) {
rate = 9.00;
address=a;
} else {
System.out.println("\nERROR : Invalid address\nRate set to zero");
rate = 0.0;
}
}
}
```

PhysicalNewspaperSubscription.java

```
package mypack;
public class PhysicalNewspaperSubscription extends
NewspaperSubscription {
public void setAddress(String a) {
```

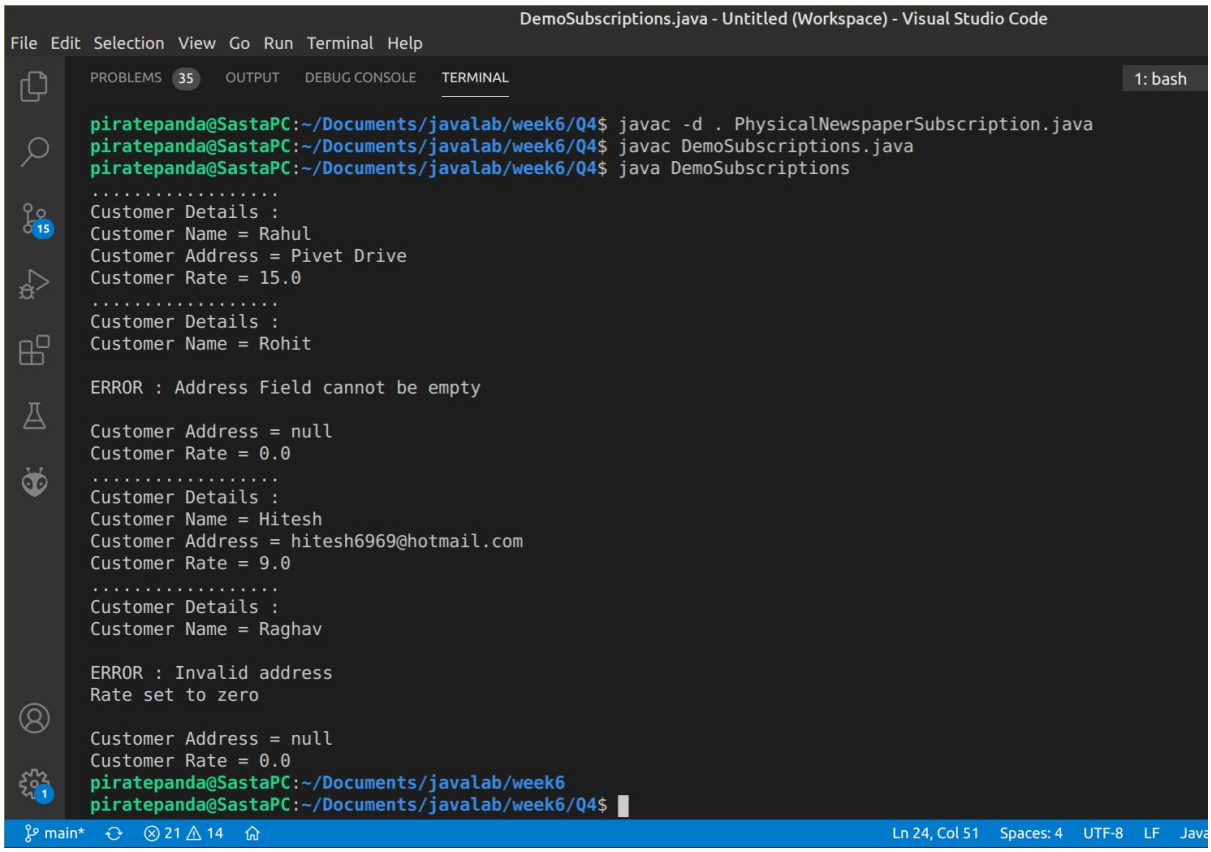
```
if(a.length()==0){
System.out.println("\nERROR : Address Field cannot be empty");
rate=0.0;
}
else{
rate=15;
address=a;
}
}
}
```

DemoSubscriptions.java

```
import mypack.PhysicalNewspaperSubscription;
import mypack.OnlineNewspaperSubscription;
public class DemoSubscriptions {
public static void main(String[] args) {
PhysicalNewspaperSubscription obj1 = new
PhysicalNewspaperSubscription();
PhysicalNewspaperSubscription obj2 = new
PhysicalNewspaperSubscription();
obj1.setName("Rahul");
System.out.println("..................\nCustomer Details : ");
System.out.println("Customer Name = "+obj1.getName());
obj1.setAddress("Pivet Drive");
System.out.println("Customer Address = "+obj1.getAddress());
System.out.println("Customer Rate = "+obj1.getRate());
```

```
obj2.setName("Rohit");
System.out.println("...................\nCustomer Details : ");
System.out.println("Customer Name = "+obj2.getName());
obj2.setAddress("");
System.out.println("Customer Address = "+obj2.getAddress());
System.out.println("Customer Rate = "+obj2.getRate());
OnlineNewspaperSubscription obj3 = new
OnlineNewspaperSubscription();
OnlineNewspaperSubscription obj4 = new
OnlineNewspaperSubscription();
obj3.setName("Hitesh");
System.out.println("...................\nCustomer Details : ");
System.out.println("Customer Name = "+obj3.getName());
obj3.setAddress("hitesh6969@hotmail.com");
System.out.println("Customer Address = "+obj3.getAddress());
System.out.println("Customer Rate = "+obj3.getRate());
obj4.setName("Raghav");
System.out.println("...................\nCustomer Details : ");
System.out.println("Customer Name = "+obj4.getName());
obj4.setAddress("");
System.out.println("Customer Address = "+obj4.getAddress());
System.out.println("Customer Rate = "+obj4.getRate());
}
}
```

Output :



Question 5

Create a class named Blanket with fields for a blanket's size, color, material, and price. Include a constructor that sets default values for the fields as Twin, white, cotton, and $30.00. Include a set method for each of the first three fields. The method that sets size adds $10 to the base price for a double blanket, $25 for a queen blanket, and $40 for a king. The method that sets the material adds $20 to the price for wool and $45 to the price for cashmere. In other words, the price for a king-sized cashmere blanket is $115. Whenever the size or material is invalid, reset the blanket to the default values. Include a
toString() method that returns a description of the blanket. Save the file as Blanket.java.

b. Create a child class named ElectricBlanket that extends Blanket and includes two additional fields: one for the number of heat settings and one for whether the electric blanket has an automatic shutoff feature. Default values are one heat setting and no automatic shutoff. Include get and set methods for the fields. Do not allow the number of settings to be fewer than one or more than five; if it is, use the default setting of 1. Add a $5.75 premium to the price if the blanket has the automatic shutoff feature. Also include a toString() method that calls the parent class toString() method and combines the returned value with data about the new fields to return a complete description of features. Save the file as ElectricBlanket.java.

c. Create an application that declares a blanket of each type and display their details in table format.

Solution :

Blanket.java

```
public class Blanket {
String size, color, material;
double price;

public Blanket() {
size = "Twin";
color = "white";
material = "cotton";
price = 30.00;
}
```

```
public void setSize(String size) {
this.size = size;
if (size.toUpperCase().equals("DOUBLE BLANKET")) {
price += 10;
} else if (size.toUpperCase().equals("QUEEN BLANKET")) {
price += 25;
} else if (size.toUpperCase().equals("KING BLANKET")) {
price += 40;
} else {
System.out.println("Error in setSize() size invalid");
this.size = "Twin";
color = "white";
material = "cotton";
price = 30.00;
}
}

public void setColor(String color) {
if (color.toUpperCase().equals("BLACK")) {
System.out.println("Black colered is not available");
size = "Twin";
color = "white";
material = "cotton";
price = 30.00;
} else {
```

```
this.color = color;
}
}

public void setMaterial(String material) {
this.material = material;
if (material.toUpperCase().equals("WOOL")) {
price += 20;
} else if (material.toUpperCase().equals("CASHMERE")) {
price += 45;
} else {
System.out.println("Error in setMaterial() material invalid");
size = "Twin";
color = "white";
material = "cotton";
price = 30.00;
}
}

public String toString(){
return "\nBlanket Info\nSize: "+size+"\nColor: "+color+"\nMaterial:
"+material+"\nPrice: "+price;
}
}
```

ElectricBlanket.java

```
public class ElectricBlanket extends Blanket{
boolean automaticSwitchOff;
int heatSetting;

public ElectricBlanket(){
super();
automaticSwitchOff = false;
heatSetting = 1;
}

public void setAutomaticSwitchOff(boolean automaticSwitchOff){
this.automaticSwitchOff = automaticSwitchOff;

if (automaticSwitchOff){
price+=5.75;
}
}

public void setHeatSetting(int heatSetting){
if (heatSetting>=1 && heatSetting<=5){
this.heatSetting = heatSetting;
}else{
System.out.println("Not a valid heat setting");
}
}
```

```
public String toString(){
return "\nBlanket Info\nSize: "+size+"\nColor: "+color+"\nMaterial:
"+material+"\nPrice: "+price+"\nAutomatic switch:
"+automaticSwitchOff+"\nHeat Setting: "+heatSetting;
}
}
```

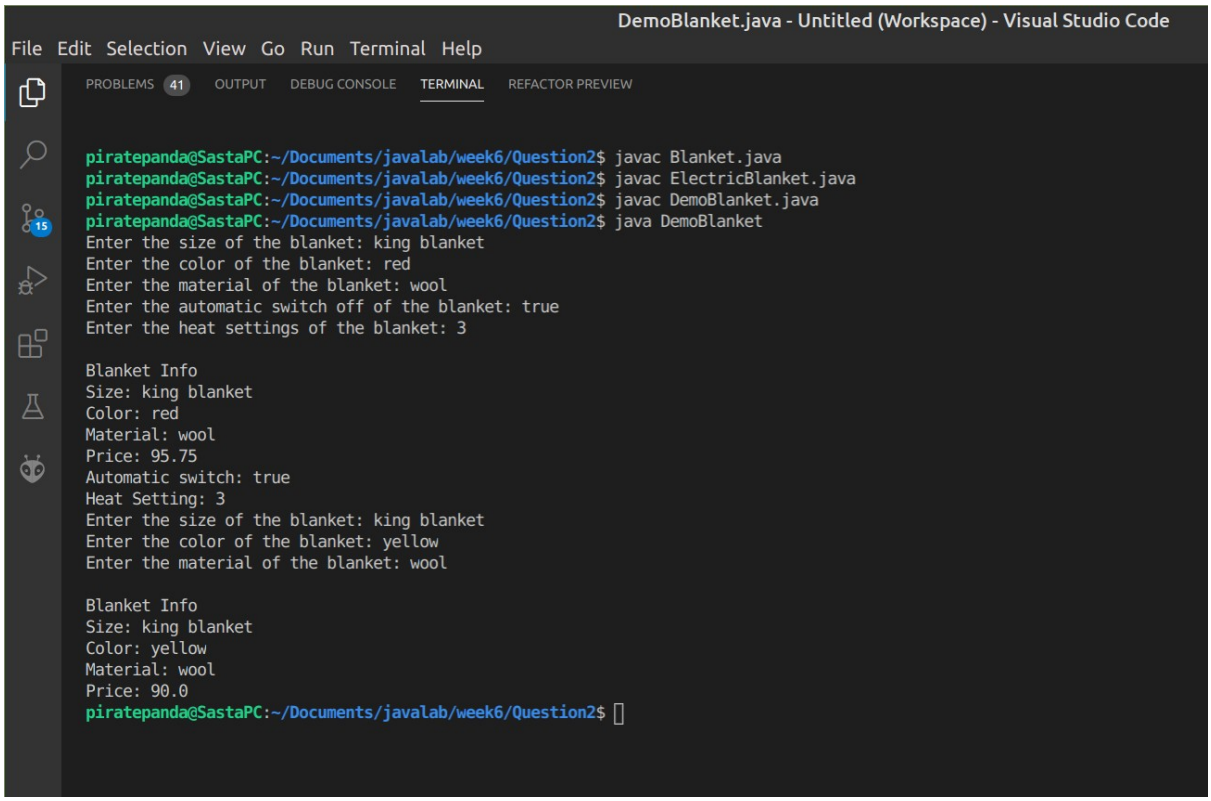
DemoBlanket.java

```
import java.util.Scanner;

public class DemoBlanket {
public static void main(String args[]) {
Scanner input = new Scanner(System.in);
ElectricBlanket obj1 = new ElectricBlanket();

System.out.print("Enter the size of the blanket: ");
String size = input.nextLine();
obj1.setSize(size);

System.out.print("Enter the color of the blanket: ");
String color = input.next();
obj1.setColor(color);

System.out.print("Enter the material of the blanket: ");
String material = input.next();
obj1.setMaterial(material);
```

```
System.out.print("Enter the automatic switch off of the blanket: ");
boolean automaticSwitchOff = input.nextBoolean();
obj1.setAutomaticSwitchOff(automaticSwitchOff);

System.out.print("Enter the heat settings of the blanket: ");
int heatSetting = input.nextInt();
obj1.setHeatSetting(heatSetting);

System.out.println(obj1);

Blanket obj2 = new Blanket();

System.out.print("Enter the size of the blanket: ");
size = input.nextLine();
size = input.nextLine();
obj2.setSize(size);

System.out.print("Enter the color of the blanket: ");
color = input.next();
obj2.setColor(color);

System.out.print("Enter the material of the blanket: ");
material = input.next();
obj2.setMaterial(material);

System.out.println(obj2);
```

```
}

}
```

Output :


DemoBlanket.java - Untitled (Workspace) - Visual Studio Code
File Edit Selection View Go Run Terminal Help
PROBLEMS 41 OUTPUT DEBUG CONSOLE TERMINAL REFACTOR PREVIEW
piratepanda@SastaPC:~/Documents/javalab/week6/Question2$ javac Blanket.java
piratepanda@SastaPC:~/Documents/javalab/week6/Question2$ javac ElectricBlanket.java
piratepanda@SastaPC:~/Documents/javalab/week6/Question2$ javac DemoBlanket.java
piratepanda@SastaPC:~/Documents/javalab/week6/Question2$ java DemoBlanket
Enter the size of the blanket: king blanket
Enter the color of the blanket: red
Enter the material of the blanket: wool
Enter the automatic switch off of the blanket: true
Enter the heat settings of the blanket: 3
Blanket Info
Size: king blanket
Color: red
Material: wool
Price: 95.75
Automatic switch: true
Heat Setting: 3
Enter the size of the blanket: king blanket
Enter the color of the blanket: yellow
Enter the material of the blanket: wool
Blanket Info
Size: king blanket
Color: yellow
Material: wool
Price: 90.0
piratepanda@SastaPC:~/Documents/javalab/week6/Question2$