LAB EXERCISE ON CLASSES AND INHERITANCE

NAME : SHERAL SIMON WASKAR

REG. NO : 20BCE1182

COURSE CODE : - CSE 1007 LAB

COURSE : JAVA PROGRAMMING

LAB SLOT : L13-L14

FACULTY : J V THOMAS ABRAHAM

DATE : 23/02/2022

1) Create a class named Billing that includes three overloaded computeBill() methods for a photo book store.
   • When computeBill() receives a single parameter, it represents the price of one photo book ordered. Add 8% tax, and return the total due.
   • When computeBill() receives two parameters, they represent the price of a photo book and the quantity ordered. Multiply the two values, add 8% tax, and return the total due.
   •When computeBill() receives three parameters, they represent the price of a photo book, the quantity ordered, and a coupon value. Multiply the quantity and price, reduce the result by the coupon value, and then add 8% tax and return the total due. Write a main() method that tests all three overloaded methods. Save the application as Billing.java.

Code:

```
import java.util.Scanner;

class Billing

{


public static double computeBill(double price)

{


double total_due = price + 8*price/100;

return total_due;

}


public static double computeBill(double price ,int quantity)

{


double total_due = price*quantity + 8*price*quantity/100;
```

```
return total_due;
}

public static double computeBill(double price , int coupon , int quantity)
{

double total_due = price*quantity - coupon +  8*price*quantity/100;

return total_due;
}

public static void main(String[] args)
{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the price of a photo book");

double price = sc.nextDouble();

System.out.println("Enter the quatity ordered");

int quantity = sc.nextInt();

System.out.println("Enter the coupon value");

int coupon = sc.nextInt();

System.out.print("The price of 1 photo book is ");

System.out.println(computeBill(price));
```

```
System.out.println("The price of " + quantity + " is " + computeBill(price,quantity));


System.out.println("The price of " + quantity + " after applying coupon of Rs " + coupon + " is "  +
computeBill( price , coupon , quantity));

}

}
```

Output :

```
D:\SEM 4\CSE1007_LAB>javac Billing.java

D:\SEM 4\CSE1007_LAB>java Billing.java
Enter the price of a photo book
26
Enter the quatity ordered
4
Enter the coupon value
50
The price of 1 photo book is 28.08
The price of 4 is 112.32
The price of 4 after applying coupon of Rs 50 is 62.32
```

2)Create a class named Patient that includes an ID number, age, and BloodData. Provide a default constructor that sets the ID number to 0, the age to 0, and the BloodData values to 0 and 1. Create an overloaded constructor that provides values for each field. Also provide get methods for each field. Save the file as Patient.java. Create an application that demonstrates that each method works correctly, and save it as TestPatient.java.

Code :

```
import java.util.Scanner;


class Patient

{


private int IDno;

private int age;

private double BloodData;
```

```
Patient()
{
IDno = 0;
age = 0;
BloodData = 0.0;
}

Patient(int IDno , int age , double BloodData)
{
this.IDno = IDno ;
this.age = age ;
this.BloodData = BloodData;
}

public int getIDno()
{
return IDno;
}

public int getage()
{
return age;

}
public double getBloodData()
{
return BloodData;
}
public String toString()
{
```

```
String str = " The ID number is  " + getIDno() + " Age is " + getage() + " Blood value " +
getBloodData();

return str;

}

}

class TestPatient

{

public static void main(String[] args)

{


Scanner sc = new Scanner(System.in);

System.out.println("Enter your ID number ");

int IDno = sc.nextInt();


System.out.println("Enter your age");

int age = sc.nextInt();


System.out.println("Enter the Blood value in range of 0 to 1");

double BloodData = sc.nextDouble();


Patient p1 = new Patient(IDno , age , BloodData);


Patient p2 = new Patient(1242,20,0.5);

System.out.println(p1.toString());

System.out.println(p2.toString());


}

}
```

Output :

```
D:\SEM 4\CSE1007_LAB>javac TestPatient.java

D:\SEM 4\CSE1007_LAB>java -classpath .;yourjar.jar TestPatient
Enter your ID number
1678
Enter your age
45
Enter the Blood value in range of 0 to 1
0.6
 The ID number is  1678 Age is 45 Blood value 0.6
 The ID number is  1242 Age is 20 Blood value 0.5
```

3) People who deal with historical dates use a number called the Julian day to calculate the number of days between two events. The Julian day is the number of days that have elapsed since January 1, 4713 B.C. For example, the Julian day for October 16, 1956, is 2435763. There are formulas for computing the Julian day from a given date, and vice versa. One very simple formula computes the day of the week from a given Julian day: Day of the week = (Julian day + 1) % 7 where % is the Java modulus operator. This formula gives a result of 0 for Sunday, 1 for Monday, and so on, up to 6 for Saturday. For Julian day 2435763, the result is 2 (Tuesday). Your job is to write a Java application that requests and inputs a Julian day, computes the day of the week using the formula, and then displays the name of the day that corresponds to that number. Your output might look like this: Enter a Julian day number and press Enter. 2451545 Julian day number 2451545 is a Saturday. Enter a Julian day number and press Enter. 2451547 Julian day number 2451547 is a Monday.

Code :

```
import java.util.Scanner;


class Julian

{


public static void calculate(int julianday)

{


int dayofweek = (julianday + 1) % 7;

if (dayofweek == 0)

System.out.println("The Julian day " + julianday + " is a Sunday ");


else if (dayofweek == 1)

System.out.println("The Julian day " + julianday + " is a  Monday");
```

```
else if (dayofweek == 2)
System.out.println("The Julian day " + julianday + " is a Tuesday");

else if (dayofweek == 3)
System.out.println("The Julian day " + julianday + " is a Wednesday ");

else if (dayofweek == 4)
System.out.println("The Julian day " + julianday + " is a Thursday ");

else if(dayofweek == 5)
System.out.println("The Julian day " + julianday + " is a Friday ");

else if(dayofweek == 6)
System.out.println("The Julian day " + julianday + " is a Saturday");

}

public static void main(String[] args)
{

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter a Julian day and press Enter ");


        int julianday = sc.nextInt();

        calculate(julianday);
}
```

}

Output :

```
D:\SEM 4\CSE1007_LAB>javac Julian.java

D:\SEM 4\CSE1007_LAB>java Julian.java
Enter a Julian day and press Enter
2451545
The Julian day 2451545 is a Saturday
```

```
D:\SEM 4\CSE1007_LAB>javac Julian.java

D:\SEM 4\CSE1007_LAB>java Julian.java
Enter a Julian day and press Enter
2451547
The Julian day 2451547 is a  Monday
```

4) Create a class named Shirt with data fields for collar size and sleeve length. Include a constructor that takes arguments for each field. Also include a String class variable named material and initialize it to “cotton”. Write a program named TestShirt to instantiate three Shirt objects with different collar sizes and sleeve lengths and then display all the data, including material, for each shirt.

Code :

```
import java.util.Scanner;

class Shirt

{

private int collarsize;

private double sleevelength;

private String material;


Shirt()

{

collarsize = 0;

sleevelength = 0.0;

material = "cotton";

}
```

```
Shirt(int collarsize , double sleevelength , String material)
{
this.collarsize = collarsize;
this.sleevelength = sleevelength;
this.material = material;
}

void details()
{
Scanner sc = new Scanner(System.in);
collarsize = sc.nextInt();
sleevelength = sc.nextDouble();
material = sc.next();



}
public String toString()
{
return "The collar size is " +  collarsize + "\n" + " The sleeve length is " + sleevelength + "\n" + "The material of the shirt is " + material ;
}
}
class TestShirt
{
public static void main(String[] args)
{

        int collarsize ;
        double sleevelength ;
        String material;
```

```
        Shirt s1 = new Shirt();
        s1.details();
        System.out.println(s1.toString());
        Scanner sc = new Scanner(System.in);
         collarsize = sc.nextInt();
        sleevelength = sc.nextDouble();
        material = sc.next();
        Shirt s2 = new Shirt(collarsize , sleevelength ,material);
        System.out.println(s2.toString());
        Shirt s3 = new Shirt();
        s3.details();
        System.out.println(s3.toString());
}
}
```

Output :

```
D:\SEM 4\CSE1007_LAB>javac TestShirt.java

D:\SEM 4\CSE1007_LAB>java -classpath .;yourjar.jar TestShirt
50
4.7
silk
The collar size is 50
 The sleeve length is 4.7
The material of the shirt is silk
40
3.8
polyester
The collar size is 40
 The sleeve length is 3.8
The material of the shirt is polyester
30
3.5
cotton
The collar size is 30
 The sleeve length is 3.5
The material of the shirt is cotton
```

5) Create a class named TaxPayer. Data fields for TaxPayer include Social Security Number (SSN- use an int for the type), and yearly gross income. Methods include a constructor that requires values for both data fields, and two get methods that return each of the data field. The SSN starts from 100000 and increases by 1 for the next TaxPayer. Write a program named UseTaxPayer that declares an array of 10 TaxPayer objects. Set SSN and gross income. Display the 10 TaxPayer objects

Code :

```
import java.util.Scanner;

class TaxPlayer

{


  int SSN;

  int income;


 TaxPlayer()

{

SSN = 0;

income = 0;

}


TaxPlayer(int SSN , int income)

{


this.SSN = SSN;

this.income = income;

}


public int getSSN()

{


return SSN;

}

public int getincome()
```

```
{
return income;
}



}
class UseTaxPlayer
{
public static void main(String[] args)
{



Scanner sc = new Scanner(System.in);


TaxPlayer tp1 = new TaxPlayer();
System.out.println("Enter the SSN number of 1st person");
tp1.SSN = sc.nextInt();
TaxPlayer tp[] = new TaxPlayer[10];
for (int i=0;i<tp.length;i++)
{
 tp[i] = new TaxPlayer();
System.out.println("Enter the yearly gross income ");
tp[i].income =sc.nextInt();
System.out.println(" SSN of person " + (i+1) + " is " + (tp1.getSSN()+i) + " and yearly gross income is "
+ tp[i].getincome());
}
}
}
```

Output :

```
D:\SEM 4\CSE1007_LAB>java -classpath .;yourjar.jar UseTaxPlayer
Enter the SSN number of 1st person
100000
Enter the yearly gross income
2500000
 SSN of person 1 is 100000 and yearly gross income is 2500000
Enter the yearly gross income
700000
 SSN of person 2 is 100001 and yearly gross income is 700000
Enter the yearly gross income
670000
 SSN of person 3 is 100002 and yearly gross income is 670000
Enter the yearly gross income
980000
 SSN of person 4 is 100003 and yearly gross income is 980000
Enter the yearly gross income
108990
 SSN of person 5 is 100004 and yearly gross income is 108990
Enter the yearly gross income
840000
 SSN of person 6 is 100005 and yearly gross income is 840000
Enter the yearly gross income
567709
 SSN of person 7 is 100006 and yearly gross income is 567709
Enter the yearly gross income
683460
 SSN of person 8 is 100007 and yearly gross income is 683460
Enter the yearly gross income
4500000
 SSN of person 9 is 100008 and yearly gross income is 4500000
Enter the yearly gross income
575788
 SSN of person 10 is 100009 and yearly gross income is 575788
```

6) 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

Code :

```
import java.util.*;


class TeeShirt

{


private int order_no ;

private String size;
```

```
private String color;
private double price;

TeeShirt()
{
order_no = 0;
size = null;
color = null;
price = 0.0;
}
int getorder_no()
{
return order_no;
}
String getsize()
{
return size;
}
String getcolor()
{
return color;
}

void setorder_no(int order_no)
{
this.order_no = order_no;
}
void setsize(String size)
{
this.size = size;
if ( size.equals("XXL") || size.equals("XXXL"))
```

```
price = 22.99;
else
price = 19.99;

}
void setcolor(String color)
{
this.color = color;
}
void display()
{
System.out.print("Order number is " + order_no + " size is " + size + " color is " + color + " price of the TeeShirt is " + price);
}
}


class CustomTee extends TeeShirt
{
private String slogan;
CustomTee()
{
super();
slogan = null;
}

String getslogan()
{
 return slogan;
}
void setslogan(String slogan)
```

```
{
 this.slogan = slogan;
}
void print()
{
display();
System.out.print(" Slogan is " + slogan);
}
}
public class DemoTees
{
public static void main(String[] args)
{
TeeShirt ts = new TeeShirt();

Scanner sc = new Scanner(System.in);
int order_no = sc.nextInt();
String size = sc.next();
String color = sc.next();
ts.setorder_no(order_no);
ts.setsize(size);
ts.setcolor(color);
ts.display();
CustomTee ct = new CustomTee();
System.out.println();
 order_no = sc.nextInt();
size = sc.next();
color = sc.next();
String junk = sc.nextLine();
String slogan = sc.nextLine();
ct.setorder_no(order_no);
```

```
ct.setsize(size);

ct.setcolor(color);

ct.setslogan(slogan);

ct.print();


}

}
```

Output :

```
D:\SEM 4\CSE1007_LAB>javac DemoTees.java

D:\SEM 4\CSE1007_LAB>java -classpath .;yourjar.jar DemoTees
1089
XL
red
Order number is 1089 size is XL color is red price of the TeeShirt is 19.99
1000
XXL
blue
Keep Smiling
Order number is 1000 size is XXL color is blue price of the TeeShirt is 22.99 Slogan is Keep Smiling
```

7) 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

Code:

```
import java.util.Scanner;

class SugarSmashPlayer {
   int pid;
   String sname;
   int[] scores;
```

```
    public SugarSmashPlayer() {
        scores = new int[11];
    }

    public int getPid() {
        return pid;
    }

    public String getSname() {
        return sname;
    }

    public int getScore(int level) {
        return scores[level - 1];
    }

    public void setVals(int pid, String sname) {
        this.pid = pid;
        this.sname = sname;
    }

    public void setScore(int level, int score) {
        if (level <=10) {
            if (scores[level - 1] >= 100 || level == 1)
                scores[level] = score;
            else
                System.out.println("Pass previous level first!!!");
        } else
            System.out.println("Enter valid level!!!");
    }
}
```

```
class PremiumSugarSmashPlayer extends SugarSmashPlayer {
   int[] additionalLevels;
   public void unlock(){
      additionalLevels = new int[40];
   }
   public void setAdditionalScore(int level, int score){
      if(level < 39){
         additionalLevels[level-10] = score;
      }
      else
         System.out.println("Enter valid levels only!!");
   }
}

class DemoSugarSmash {
   public static void main(String[] args) {
      Scanner in = new Scanner(System.in);
      PremiumSugarSmashPlayer player = new PremiumSugarSmashPlayer();
      player.setVals(12, "Faraz");
      System.out.println("Welcome " + player.getSname() + " player id " + player.getPid());
      String ch = "yes";
      while (ch.equals("yes")) {
                              for (int i=1 ;i<=10;i++)
                  {
         System.out.println("\nEnter level and score: ");
         int lev = in.nextInt();
         int sco = in.nextInt();
         player.setScore(lev, sco);

                  }
```

```
        System.out.println("\nWant to pay $2.99 for premium membership? ");
        ch = in.next();
        if (ch.equals("no")) {
           System.out.println("Bye...");
        }
        else{
              System.out.println("\nUnlocking 40 extra levels...");
              player.unlock();
              System.out.println("You are premium player now!!");
           }
        }
     }
  }
```

Output:

```
D:\SEM 4\CSE1007_LAB>javac DemoSugarSmash.java

D:\SEM 4\CSE1007_LAB>java -classpath .;yourjar.jar DemoSugarSmash
```

```
Welcome Faraz player id 12

Enter level and score:
1
405

Enter level and score:
2
504

Enter level and score:
3
560

Enter level and score:
4
670

Enter level and score:
5
305

Enter level and score:
6
367

Enter level and score:
7
975

Enter level and score:
8
266

Enter level and score:
9
156

Enter level and score:
10
754

Want to pay $2.99 for premium membership?
no
Bye...

D:\SEM 4\CSE1007_LAB>
```

With validations

```
D:\SEM 4\CSE1007_LAB>javac DemoSugarSmash.java

D:\SEM 4\CSE1007_LAB>java -classpath .;yourjar.jar DemoSugarSmash
```

```
Welcome Faraz player id 12

Enter level and score:
1
200

Enter level and score:
2
90
You have not cleared this level please try again

Enter level and score:
2
104

Enter level and score:
3
10
You have not cleared this level please try again

Enter level and score:
3
100

Enter level and score:
4
105
```

```
Enter level and score:
5
640

Enter level and score:
6
904

Enter level and score:
7
246

Enter level and score:
8
890

Want to pay $2.99 for premium membership?
no
Bye...

D:\SEM 4\CSE1007_LAB>
```