

CSE1007 – JAVA Programming

Lab Exercise on Classes and Inheritance

Question 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
{
    float price, q, cvalue, total;

    Billing (float price)
    {
        this.price=price;
    }

    Billing (float price, float q)
    {
        this.price=price;
        this.q=q;
    }

    Billing (float price, float q, float cvalue)
    {
        this.price=price;
        this.q=q;
        this.cvalue=cvalue;
    }
}
```

```
}

float computeBill(float price)
{
    total=price+(8*price)/100;
    return total;
}

float computeBill(float price, float q)
{
    total=q*price+(8*q*price)/100;
    return total;
}

float computeBill(float price, float q, float cvalue)
{
    total=q*price-cvalue+(8*(q*price-cvalue))/100;
    return total;
}

void print()
{
    System.out.println("Total: "+total);
}

public static void main(String[] args)
{
    Scanner in= new Scanner(System.in);
    int ch;
    while(true)
    {
        System.out.println("\n1.Single 2.Double 3.Triple 4.Exit");
        System.out.print("Enter your choice:: ");
        ch=in.nextInt();
        if(ch==1)
        {
            System.out.print("Enter price of one photo book: ");
            Float p=in.nextFloat();
            Billing b=new Billing(p);
            b.total=b.computeBill(p);
            b.print();
        }
        else if(ch==2)
        {
            System.out.print("Enter price of one photo book: ");
            Float p=in.nextFloat();
            System.out.print("Enter quantity: ");
```

```
        Float q=in.nextFloat();
        Billing b=new Billing(p,q);
        b.total=b.computeBill(p,q);
        b.print();
    }
    else if(ch==3)
    {
        System.out.print("Enter price of one photo book: ");
        Float p=in.nextFloat();
        System.out.print("Enter quantity: ");
        Float q=in.nextFloat();
        System.out.print("Enter coupon value: ");
        Float c=in.nextFloat();
        Billing b=new Billing(p,q,c);
        b.total=b.computeBill(p,q,c);
        b.print();
    }
    else if(ch==4)
        break;
    }
}
```

OUTPUT:

```
C:\Gokul\VIT\SEM-4\CSE1007 - Java\Lab\Lab6>javac Billing.java
C:\Gokul\VIT\SEM-4\CSE1007 - Java\Lab\Lab6>java Billing

1.Single 2.Double 3.Triple 4.Exit
Enter your choice:: 1
Enter price of one photo book: 25
Total: 27.0

1.Single 2.Double 3.Triple 4.Exit
Enter your choice:: 2
Enter price of one photo book: 25
Enter quantity: 2
Total: 54.0

1.Single 2.Double 3.Triple 4.Exit
Enter your choice:: 3
Enter price of one photo book: 25
Enter quantity: 2
Enter coupon value: 5
Total: 48.6

1.Single 2.Double 3.Triple 4.Exit
Enter your choice:: 4
```

Question 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
{
    int ID, age;
    int[] BloodData =new int[2];

    Patient()
    {
        ID=0;
        age=0;
        BloodData[0]=0;
        BloodData[1]=1;
    }

    Patient (int ID)
    {
        this.ID=ID;
    }

    Patient (int ID, int age)
    {
        this.ID=ID;
        this.age=age;
    }

    Patient (int ID, int age, int[] BloodData)
    {
        this.ID=ID;
        this.age=age;
        this.BloodData[0]=BloodData[0];
        this.BloodData[1]=BloodData[1];
    }

    void get (int ID)
    {
```

```
        Scanner in= new Scanner(System.in);
        System.out.print("Enter ID: ");
        this.ID=in.nextInt();
    }

    void get (int ID, int age)
    {
        Scanner in= new Scanner(System.in);
        System.out.print("Enter ID : ");
        this.ID=in.nextInt();
        System.out.print("Enter age: ");
        this.age=in.nextInt();
    }

    void get (int ID, int age, int[] BloodData)
    {
        Scanner in= new Scanner(System.in);
        System.out.print("Enter ID : ");
        this.ID=in.nextInt();
        System.out.print("Enter age: ");
        this.age=in.nextInt();
        System.out.print("Enter Blood Data: ");
        this.BloodData[0]=in.nextInt();
        this.BloodData[1]=in.nextInt();
    }

    void print()
    {
        System.out.println("\nPatient Deatils");
        if(ID!=0)
            System.out.println("ID: "+ID);
        if(age!=0)
            System.out.println("Age: "+age);
        if(BloodData[0]!=0 && BloodData[1]!=0)
            System.out.println("Blood Data: "+BloodData[0]+"-"+BloodData[1]);
    }
}

public class TestPatient
{
    public static void main(String[] args)
    {
        Scanner in= new Scanner(System.in);
        int ch;
        while(true)
```

```
{
    System.out.println("\n1.ID only  2.ID & age  3.ID, age & Blood Data  4.Exit");
    System.out.print("Enter your choice: ");
    ch=in.nextInt();
    if(ch==1)
    {
        Patient p=new Patient();
        p.get(p.ID);
        Patient p1=new Patient(p.ID);
        p1.print();
    }
    else if(ch==2)
    {
        Patient p=new Patient();
        p.get(p.ID, p.age);
        Patient p1=new Patient(p.ID, p.age);
        p1.print();
    }
    else if(ch==3)
    {
        Patient p=new Patient();
        p.get(p.ID, p.age, p.BloodData);
        Patient p1=new Patient(p.ID, p.age, p.BloodData);
        p1.print();
    }
    else if(ch==4)
        break;
    }
}
```

OUTPUT:

```
C:\Gokul\VIT\SEM-4\CSE1007 - Java\Lab\Lab6>javac TestPatient.java

C:\Gokul\VIT\SEM-4\CSE1007 - Java\Lab\Lab6>java TestPatient

1.ID only  2.ID & age  3.ID, age & Blood Data  4.Exit
Enter your choice: 1
Enter ID: 101

Patient Deatils
ID: 101

1.ID only  2.ID & age  3.ID, age & Blood Data  4.Exit
Enter your choice: 2
Enter ID : 101
Enter age: 25

Patient Deatils
ID: 101
Age: 25

1.ID only  2.ID & age  3.ID, age & Blood Data  4.Exit
Enter your choice: 3
Enter ID : 101
Enter age: 25
Enter Blood Data: 15 19

Patient Deatils
ID: 101
Age: 25
Blood Data: 15-19

1.ID only  2.ID & age  3.ID, age & Blood Data  4.Exit
Enter your choice: 4
```

Question 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
{
    long day;
    int noOfDay;
    String nameOfDay;

    Julian (long day)
    {
        this.day=day;
    }

    void findDay()
    {
        noOfDay=(int) (this.day+1)%7;
        String[]
names={"Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday"};
        nameOfDay=names[noOfDay];
    }
}
```



```
void print()
{
    System.out.println("Julian day number "+day+" is a "+nameOfDay+".");
}

public static void main(String[] args)
{
    Scanner in= new Scanner(System.in);
    System.out.println("Enter a Julian day number and press Enter.");
    long d=in.nextLong();
    Julian j=new Julian(d);
    j.findDay();
    j.print();
}
}
```

OUTPUT:

```
C:\Gokul\VIT\SEM-4\CSE1007 - Java\Lab\Lab6>javac Julian.java
C:\Gokul\VIT\SEM-4\CSE1007 - Java\Lab\Lab6>java Julian
Enter a Julian day number and press Enter.
2451545
Julian day number 2451545 is a Saturday.

C:\Gokul\VIT\SEM-4\CSE1007 - Java\Lab\Lab6>java Julian
Enter a Julian day number and press Enter.
2451547
Julian day number 2451547 is a Monday.
```

Question 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
{
    float collarSize, sleeveLen;
    String material;

    Shirt()
    {
        collarSize=0;
        sleeveLen=0;
        material="cotton";
    }

    Shirt(float collarSize, float sleeveLen, String material)
    {
        this.collarSize=collarSize;
        this.sleeveLen=sleeveLen;
        this.material=material;
    }

    void get (float collarSize, float sleeveLen)
    {
        Scanner in= new Scanner(System.in);
        System.out.print("\nEnter collar size : ");
        this.collarSize =in.nextFloat();
        System.out.print("Enter sleeve size: ");
        this.sleeveLen=in.nextFloat();
    }

    void get (float collarSize, float sleeveLen, String material)
    {
        Scanner in= new Scanner(System.in);
        System.out.print("\nEnter collar size : ");
        this.collarSize =in.nextFloat();
        System.out.print("Enter sleeve size: ");
        this.sleeveLen=in.nextFloat();
        System.out.print("Enter material: ");
```

```
        this.material=in.next();
    }

    void print()
    {
        System.out.println("Collor size  : "+collorSize);
        System.out.println("Sleeve length : "+sleeveLen);
        System.out.println("Material    : "+material);
    }
}

public class TestShirt
{
    public static void main(String[] args)
    {
        Scanner in= new Scanner(System.in);
        int ch;

        Shirt sh1=new Shirt();
        sh1.get(sh1.collorSize, sh1.sleeveLen, sh1.material);
        Shirt s1=new Shirt(sh1.collorSize,sh1.sleeveLen,sh1.material);

        Shirt sh2=new Shirt();
        sh2.get(sh2.collorSize, sh2.sleeveLen);
        Shirt s2=new Shirt(sh2.collorSize,sh2.sleeveLen,sh2.material);

        Shirt sh3=new Shirt();
        sh3.get(sh3.collorSize, sh3.sleeveLen, sh3.material);
        Shirt s3=new Shirt(sh3.collorSize,sh3.sleeveLen,sh3.material);

        System.out.println("\nShirt-1 Deatils");
        s1.print();
        System.out.println("\nShirt-2 Deatils");
        s2.print();
        System.out.println("\nShirt-3 Deatils");
        s3.print();

    }
}
```

OUTPUT:

```
C:\Gokul\VIT\SEM-4\CSE1007 - Java\Lab\Lab6>javac TestShirt.java

C:\Gokul\VIT\SEM-4\CSE1007 - Java\Lab\Lab6>java TestShirt

Enter collar size : 10.5
Enter sleeve size: 22.65
Enter material: Polyester

Enter collar size : 11.5
Enter sleeve size: 25.6

Enter collar size : 9.55
Enter sleeve size: 28.56
Enter material: Nylon

Shirt-1 Deatils
Collor size   : 10.5
Sleeve length : 22.65
Material      : Polyester

Shirt-2 Deatils
Collor size   : 11.5
Sleeve length : 25.6
Material      : cotton

Shirt-3 Deatils
Collor size   : 9.55
Sleeve length : 28.56
Material      : Nylon
```

Question 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. Use static block static variable wherever is applicable.

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 TaxPayer
{
    int SSN;
    float Inc;

    TaxPayer(int SSN, float Inc)
    {
        this.SSN=SSN;
        this.Inc=Inc;
    }

    int getSSN()
    {
        return SSN;
    }

    float getInc()
    {
        return Inc;
    }

    void display()
    {
        System.out.println("SSN : "+getSSN());
        System.out.println("Income : "+getInc());
    }
}

public class UseTaxPayer
{
    static int SNO=100000, count=1;
    public static void main(String[] args)
```

```
{
    Scanner in=new Scanner(System.in);
    int n,i;
    float Income;
    System.out.print("\nEnter no: of tax payers: ");
    n=in.nextInt();
    TaxPayer T[]=new TaxPayer[n];

    for(i=0;i<n;i++)
    {
        System.out.print("Enter Yearly Gross Income of Tax payer-"+(count++)+": ");
        Income=in.nextFloat();
        T[i]=new TaxPayer((SNO++),Income);
    }
    for(i=0;i<n;i++)
    {
        System.out.println("\nTax payer-"+(i+1));
        T[i].display();
    }
}
```

OUTPUT:

```
C:\Goku1\VIT\SEM-4\CSE1007 - Java\Lab\Lab6>javac UseTaxPayer.java
C:\Goku1\VIT\SEM-4\CSE1007 - Java\Lab\Lab6>java UseTaxPayer

Enter no: of tax payers: 10
Enter Yearly Gross Income of Tax payer-1: 2500
Enter Yearly Gross Income of Tax payer-2: 7450
Enter Yearly Gross Income of Tax payer-3: 12000
Enter Yearly Gross Income of Tax payer-4: 1500
Enter Yearly Gross Income of Tax payer-5: 89000
Enter Yearly Gross Income of Tax payer-6: 4700
Enter Yearly Gross Income of Tax payer-7: 1000
Enter Yearly Gross Income of Tax payer-8: 289750
Enter Yearly Gross Income of Tax payer-9: 14500
Enter Yearly Gross Income of Tax payer-10: 56300

Tax payer-1
SSN    : 100000
Income : 2500.0

Tax payer-2
SSN    : 100001
Income : 7450.0

Tax payer-3
SSN    : 100002
Income : 12000.0

Tax payer-4
SSN    : 100003
Income : 1500.0

Tax payer-5
SSN    : 100004
Income : 89000.0

Tax payer-6
SSN    : 100005
Income : 4700.0

Tax payer-7
SSN    : 100006
Income : 1000.0

Tax payer-8
SSN    : 100007
Income : 289750.0

Tax payer-9
SSN    : 100008
Income : 14500.0

Tax payer-10
SSN    : 100009
Income : 56300.0
```

Question 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.Scanner;

class TeeShirt
{
    int oNO;
    double price;
    String size, color;

    void setDetails()
    {
        Scanner in= new Scanner(System.in);
        System.out.print("\nEnter order no: ");
        oNO =in.nextInt();
        System.out.print("Enter size: ");
        size=in.next();
        System.out.print("Enter color: ");
        color=in.next();
        setPrice();
    }

    void setPrice()
    {
        if(size.equals("XXL") || size.equals("XXXL"))
            price=22.99;
        else
            price=19.99;
    }

    int getoNO()
    {
        return oNO;
    }
}
```



```
double getprice()
{
    return price;
}

String getsize()
{
    return size;
}

String getcolor()
{
    return color;
}

void printDetails()
{
    System.out.println("Order No : "+getoNO());
    System.out.println("Size   : "+getsize());
    System.out.println("Color  : "+getcolor());
    System.out.println("Price  : "+getprice());
}

public static void main(String[] args)
{
    Scanner in= new Scanner(System.in);
    int ch;

    CustomTee s1=new CustomTee();
    s1.set();
    CustomTee s2=new CustomTee();
    s2.set();
    System.out.println("\nTee Shirt-1 Deatils");
    s1.print();
    System.out.println("\nTee Shirt-2 Deatils");
    s2.print();
}

class CustomTee extends TeeShirt
{
    String slogan;
    void set()
    {
        setDetails();
        Scanner in= new Scanner(System.in);
        System.out.print("Enter Slogan: ");
```

```
        slogan=in.nextLine();
    }

    String getslogan()
    {
        return slogan;
    }

    void print()
    {
        printDetails();
        System.out.println("Slogan : "+getslogan());
    }
}
```

OUTPUT:

```
C:\Gokul\VIT\SEM-4\CSE1007 - Java\Lab\Lab6>javac TeeShirt.java

C:\Gokul\VIT\SEM-4\CSE1007 - Java\Lab\Lab6>java TeeShirt

Enter order no: 151
Enter size: M
Enter color: Red
Enter Slogan: Built for comfort

Enter order no: 210
Enter size: XXXL
Enter color: Blue
Enter Slogan: Style that suits you

Tee Shirt-1 Deatils
Order No : 151
Size      : M
Color     : Red
Price     : 19.99
Slogan    : Built for comfort

Tee Shirt-2 Deatils
Order No : 210
Size      : XXXL
Color     : Blue
Price     : 22.99
Slogan    : Style that suits you
```

Question 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 ID;
    String sname;
    int[] score=new int[100];
    int Level;

    Scanner in=new Scanner(System.in);

    SugarSmashPlayer()
    {
        ID=0;
        sname=null;
        for(int i=0;i<10;i++)
            score[i]=-1;
    }

    void setID(int ID)
    {
        System.out.print("Enter player ID: ");
        this.ID=in.nextInt();
    }
}
```

```
}  
void setsname(String snam)  
{  
    System.out.print("Enter screen name: ");  
    this.sname=in.next();  
}  
int setscore(int Level, int Score)  
{  
    int flag=0;  
    Level=in.nextInt();  
    if(Level>=1 && Level<=10)  
    {  
        if(Level==1)  
        {  
            System.out.print("Enter Score: ");  
            score[Level-1]=in.nextInt();  
        }  
        else  
        {  
            if(score[Level-1]>=100)  
            {  
                System.out.print("Enter Score: ");  
                score[Level-1]=in.nextInt();  
            }  
            else  
            {  
                System.out.println("Not enough points for next level. Kindly Enter again!");  
                flag=1;  
            }  
        }  
    }  
    else  
    {  
        System.out.println("Invalid Game Level. Kindly Enter again!");  
        flag=1;  
    }  
    if(flag==1)  
        return -1;  
    else  
        return 0;  
}  
  
int getID()  
{  
    return ID;  
}  
String getsname()
```

```
{
    return sname;
}
int getscore(int Level)
{
    if(Level>=1 && Level<=10)
        return score[Level-1];
    else
    {
        System.out.println("Invalid Game Level");
        return -1;
    }
}

void display()
{
    int s;
    System.out.println("\nPlayer ID : "+getID());
    System.out.println("Screen Name : "+getsname());
    System.out.println();
    for(int i=0;i<10;i++)
    {
        s=getscore(i+1);
        if(i==9 && s!=-1)
            System.out.println("Level-"+(i+1)+" : "+s);
        else if(i>=0 && s!=-1)
            System.out.println("Level-"+(i+1)+" : "+s);
    }
}

class PremiumSugarSmashPlayer extends SugarSmashPlayer
{
    PremiumSugarSmashPlayer(int id, String name, int A[])
    {
        this.ID=id;
        this.sname=name;
        for(int i=0;i<10;i++)
            this.score[i]=A[i];
        for(int i=10;i<50;i++)
            score[i]=-1;
    }

    int Psetscore(int Level, int Score)
    {
        int flag=0;
        Level=in.nextInt();
```

```
if(Level>=11 && Level<=50)
{
    if(score[Level-1-1]>=100)
    {
        System.out.print("Enter Score: ");
        score[Level-1]=in.nextInt();
    }

    else
    {
        System.out.println("Sorry Not enough points for next level. Kindly Enter again!");
        flag=1;
    }
}
else
{
    System.out.println("Invalid Game Level. Kindly Enter again!");
    flag=1;
}
if(flag==1)
return -1;
else
return 0;
}

int Pgetscore(int Level)
{
    if(Level>=11 && Level<=50)
    return score[Level-1];
    else
    {
        System.out.println("Invalid Game Level");
        return -1;
    }
}

void Pdisplay()
{
    display();
    int s;
    for(int i=10;i<50;i++)
    {
        s=Pgetscore(i+1);
        if(s!=-1)
        System.out.println("Level-"+(i+1)+" : "+s);
    }
}
```

```
}

public class DemoSugarSmash
{
    public static void main(String[] args)
    {
        Scanner in=new Scanner(System.in);
        int i,stat,ch=0;

        SugarSmashPlayer s1=new SugarSmashPlayer();
        s1.setID(s1.ID);
        s1.setsname(s1.sname);
        System.out.println();
        for(i=0;i<10;i++)
        {
            if(i==0)
                System.out.print("Enter Level: ");
            else
            {
                System.out.println("1.Next Level 2.Exit");
                System.out.print("Enter your choice: ");
                ch=in.nextInt();
                if(ch==1)
                    System.out.print("Enter Level: ");
                else
                    break;
            }
            stat=s1.setscore(i+1,s1.score[i]);
            if(stat==-1 && i==1)
                i=i-2;
            else if(stat==-1)
                i--;
        }
        s1.display();

        //Premium Account
        System.out.println("\n\n1.Pay $2.99 and get Premium Access 2.Exit");
        System.out.print("Enter your choice: ");
        int choice=in.nextInt();
        if(ch==2 || choice==2)
        {
            System.out.println("Not enough points in first 10 levels");
            System.exit(0);
        }
    }
}
```

```
PremiumSugarSmashPlayer s2=new PremiumSugarSmashPlayer(s1.ID, s1.sname,  
s1.score);
```

```
    for(i=10;i<50;i++)  
    {  
        if(i==10)  
            System.out.print("Enter Level: ");  
        else  
        {  
            System.out.println("1.Next Level  2.Exit");  
            System.out.print("Enter your choice: ");  
            ch=in.nextInt();  
            if(ch==1)  
                System.out.print((i+1)+"Enter Level: ");  
            else  
                break;  
        }  
        stat=s2.Psetscore(i+1,s2.score[i]);  
        if(stat== -1)  
            i--;  
    }  
    s2.Pdisplay();  
}
```

OUTPUT:

```
C:\Gokul\VIT\SEM-4\CSE1007 - Java\Lab\Lab6>javac DemoSugarSmash.java  
  
C:\Gokul\VIT\SEM-4\CSE1007 - Java\Lab\Lab6>java DemoSugarSmash  
Enter player ID: 101  
Enter screen name: S1  
  
Enter Level: 1  
Enter Score: 25  
1.Next Level  2.Exit  
Enter your choice: 1  
Enter Level: 2  
Not enough points for next level. Kindly Enter again!  
Enter Level: 1  
Enter Score: 105  
1.Next Level  2.Exit  
Enter your choice: 1  
Enter Level: 2  
Enter Score: 104  
1.Next Level  2.Exit  
Enter your choice: 1  
Enter Level: 3  
Enter Score: 500
```



```
1.Next Level  2.Exit
Enter your choice: 1
Enter Level: 4
Enter Score: 250
1.Next Level  2.Exit
Enter your choice: 1
Enter Level: 5
Enter Score: 470
1.Next Level  2.Exit
Enter your choice: 1
Enter Level: 6
Enter Score: 120
1.Next Level  2.Exit
Enter your choice: 1
Enter Level: 7
Enter Score: 780
1.Next Level  2.Exit
Enter your choice: 1
Enter Level: 11
Invalid Game Level. Kindly Enter again!
1.Next Level  2.Exit
Enter your choice: 1
Enter Level: 8
Enter Score: 100
1.Next Level  2.Exit
Enter your choice: 1
Enter Level: 9
Enter Score: 120
1.Next Level  2.Exit
Enter your choice: 1
Enter Level: 10
Enter Score: 400

Player ID   : 101
Screen Name : S1

Level-1   : 105
Level-2   : 104
Level-3   : 500
Level-4   : 250
Level-5   : 470
Level-6   : 120
Level-7   : 780
Level-8   : 100
Level-9   : 120
Level-10  : 400
```

1.Pay \$2.99 and get Premium Access 2.Exit

Enter your choice: 1

Enter Level: 11

Enter Score: 140

1.Next Level 2.Exit

Enter your choice: 1

12Enter Level: 12

Enter Score: 460

1.Next Level 2.Exit

Enter your choice: 2

Player ID : 101

Screen Name : S1

Level-1 : 105

Level-2 : 104

Level-3 : 500

Level-4 : 250

Level-5 : 470

Level-6 : 120

Level-7 : 780

Level-8 : 100

Level-9 : 120

Level-10 : 400

Level-11 : 140

Level-12 : 460