1. DiscountRate(Beauty Parlour) :

```
/**
* Created by etenbrinke on 30/11/15.
* https://www3.ntu.edu.sg/home/ehchua/programming/java/J3f_OOPExercises.html
*
* You are asked to write a discount system for a beauty saloon, which provides services and sells beauty products.
* It offers 3 types of memberships: Premium, Gold and Silver.
* Premium, gold and silver members receive a discount of 20%, 15%, and 10%, respectively, for all services provid
ed.
* Customers without membership receive no discount.
* All members receives a flat 10% discount on products purchased (this might change in future).
* Your system shall consist of three classes: Customer, Discount and Visit, as shown in the class diagram.
* It shall compute the total bill if a customer purchases $x of products and $y of services, for a visit.
* Also write a test program to exercise all the classes.
*/

import java.util.*;
class Customer {

    private String name;
   private boolean member;
   private String memberType;

   public Customer() {
       this.member = false;
   }

   public Customer(String name, boolean member, String memberType) {
       this.name = name;
       this.member = member;
       this.memberType = memberType;
   }

   public String getName() {
       return name;
   }

   public boolean isMember() {
       return member;
   }
```

```
public String getMemberType() {
    return memberType;
}

public void setMemberType(String memberType) {
    this.memberType = memberType;
}

@Override
public String toString() {
    return "Customer{" +
            "name='" + name + '\'' +
            ", member=" + member +
            ", memberType='" + memberType + '\'' +
            '}';
    }
}
class DiscountRate
{

    private static double serviceDiscountPremium = 0.2;
    private static double serviceDiscountGold = 0.15;
    private static double serviceDiscountSilver = 0.1;
    private static double productDiscountPremium = 0.1;
    private static double productDiscountGold = 0.1;
    private static double productDiscountSilver = 0.1;

    public static double getServiceDiscountRate(String type)
    {
        switch (type)
        {
            case "Premium":
            return serviceDiscountPremium;
            case "Gold":
            return serviceDiscountGold;
            case "Silver":
            return serviceDiscountSilver;
            default:
            throw new IllegalArgumentException("wrong service type
specified");
        }
    }

    public static double getProductDiscountRate(String type)
```

```
    {
        switch (type)
        {
            case "Premium":
            return productDiscountPremium;
            case "Gold":
            return productDiscountGold;
            case "Silver":
            return productDiscountSilver;
            default:
            throw new IllegalArgumentException("wrong service type
specified");
        }
    }
class Visit {

      private Customer name;
    private Date date;
    private double serviceExpense;
    private double productExpense;
    public Visit(Customer name, Date date) {
        this.name = name;
        this.date = date;
    }

    public String getCustomerName() {
        return name.getName();
    }

    public double getServiceExpense() {
        return serviceExpense;
}

public void setServiceExpense(double serviceExpense) {
    this.serviceExpense = this.serviceExpense + serviceExpense;
}

public double getProductExpense() {
    return productExpense;
}

public void setProductExpense(double productExpense) {
    this.productExpense = this.productExpense + productExpense;
}
```

```
    public double getTotalExpense() {
        return  (serviceExpense - (serviceExpense *
DiscountRate.getServiceDiscountRate(name.getMemberType()))) +
                (productExpense - (productExpense *
DiscountRate.getProductDiscountRate(name.getMemberType())));

    }

    @Override
    public String toString() {
        return "Visit{" +
                "customer name=" + name.getName() +
                ", customer member=" + name.isMember() +
                ", customer member type=" + name.getMemberType() +
                ", date=" + date +
                ", serviceExpense=" + serviceExpense +
                ", productExpense=" + productExpense +
                '}';
    }
}
public class TestDiscountSystem {

    public static void main (String[] args) {
       Scanner sc=new Scanner(System.in);
       String name1=sc.nextLine();
        boolean member1=sc.nextBoolean();
        sc.nextLine();
        String memberType1=sc.nextLine();
        String name2=sc.nextLine();
        boolean member2=sc.nextBoolean();
        sc.nextLine();
        String memberType2=sc.nextLine();
        double serviceExpense1=sc.nextDouble();
        double productExpense1=sc.nextDouble();
        //double serviceExpense2;
        double productExpense2=sc.nextDouble();
       /*Customer c1 = new Customer("Piet Clerx", true, "Premium");
        Customer c2 = new Customer("Trees Klaas", true, "Silver");*/
        Customer c1=new Customer(name1,member1,memberType1);
        Customer c2=new Customer(name2,member2,memberType2);
        System.out.println(c1.toString());
        System.out.println(c2.toString());
        Visit v1 = new Visit(c1, new Date());
        System.out.println(v1.toString());
```

```
        v1.setProductExpense(productExpense1);
        v1.setServiceExpense(serviceExpense1);
        v1.setProductExpense(productExpense2);
        System.out.println(v1.toString());
        System.out.println("Total expense made by " + v1.getCustomerName() + " =
" + v1.getTotalExpense());
    }
}
```

2. Instruments :

```
/*
 * Create an abstract class Instrument which is having the abstract function play.
 * Create three more sub classes from Instrument which is Piano, Flute, Guitar.
 * Override the play method inside all three classes printing a message
 *    â€œPiano is playing  tan tan tan tan  â€•    for Piano class
 *    â€œFlute is playing  toot toot toot tootâ€•    for Flute class
 *    â€œGuitar is playing  tin  tin  tin â€•    for Guitar class
 * You must not allow the user to declare an object of Instrument class.
 * Create an array of 10 Instruments.
 * Assign different type of instrument to Instrument reference.
 * Check for the polymorphic behavior of  play method.
 * Use the instanceof operator to print that which object stored at which index of
instrument array.
 * */

import java.util.*;
abstract class Instrument
{
    public abstract void Play();
}
class Piano extends Instrument
{
    public void Play()
    {
        System.out.println("Piano is playing tan tan tan tan");
    }
}
class Flute extends Instrument
{
    public void Play()
    {
        System.out.println("Flute is playing toot toot toot toot");
    }
}
```

```
class Guitar extends Instrument
{
	public void Play()
	{
		System.out.println("Guitar is playing tin tin tin ");
	}
}
public class Ans21 /*Eta “Source” class*/
{
	public static void main(String[] args)
	{
		Instrument inst[] = new Instrument[10];
		inst[0] = new Piano();
		inst[1] = new Flute();
		inst[2] = new Guitar();
		inst[3] = new Piano();
		inst[4] = new Flute();
		inst[5] = new Guitar();
		inst[6] = new Piano();
		inst[7] = new Flute();
		inst[8] = new Guitar();
		inst[9] = new Piano();
		for ( int i = 0 ; i < inst.length ; i++ )
		{
			if ( inst[i] instanceof Piano )
			{
				System.out.println("Yes, Its Piano");
				inst[i].Play();
			}
			if ( inst[i] instanceof Flute )
			{
				System.out.println("Yes, Its Flute");
				inst[i].Play();
			}
			if ( inst[i] instanceof Guitar )
			{
				System.out.println("Yes, Its Guitar");
				inst[i].Play();
			}
		}
	}
}
```

3. Rail Compartment :

```
/*
 * Create an abstract class Compartment to represent a rail coach. Provide an abstract
 * function notice in this class. Derive FirstClass, Ladies, General, Luggage classes
 * from the compartment class. Override the notice function in each of them to print
notice
 * suitable to the type of the compartment.
 * Create a class TestCompartment . Write main function to do the following:
 * Declare an array of Compartment of size 10.
 * Create a compartment of a type as decided by a randomly generated integer in the
range 1 to 4.
 * Check the polymorphic behavior of the notice method.
 * */

abstract class Compartment
{
	abstract void notice();
}
class FirstClass extends Compartment
{
	void notice()
	{
		System.out.println("Its FIRSTCLASS");
	}
}
class Ladies extends Compartment
{
	void notice()
	{
		System.out.println("Its LADIES Compartment");
	}
}
class General extends Compartment
{
	void notice()
```

```
	{
		System.out.println("Its GENERAL Compartment");
	}
}
class Luggage extends Compartment
{
	void notice()
	{
		System.out.println("Its LUGGAGE");
	}
}
public class TestCompartment /*Eta “Source” class*/
{
	public static void main(String[] args)
	{
		Compartment c[] = new Compartment[10];
		double i = Math.random()*5;
		int x = (int)i;
		System.out.println(x);
		switch(x)
		{
			case 1:
			c[0] = new FirstClass();
			c[0].notice();
			break;
			case 2:
			c[1] = new Ladies();
			c[1].notice();
			break;
			case 3:
			c[2] = new General();
			c[2].notice();
			break;
			case 4:
			c[3] = new Luggage();
			c[3].notice();
			break;
			default: System.out.println("Invalid Choice");
		}
	}
}
```

4. Check two persons are same :

```
import java.util.*;
public class personSame /*Eta “Source” class*/
{
	public static void main(String[] args)
	{
		Scanner s=new Scanner(System.in);
		String name=s.nextLine();
		int a=s.nextInt();
		String g=s.nextLine();
		String name1=s.nextLine();
		int a1=s.nextInt();
		String g1=s.nextLine();
		if(name.equals(name1) && a==a1 && g.equals(g1))
		{
		   System.out.println("The persons are same...");
		}
		else
		{
		    System.out.println("The persons are different...");
		}
	}
}
```

5. Registered customer (Inheritance-Aggregation) :

```
/*registered customer*/
/*"Cust" should be replaced with "Customer" during exam*/
import java.util.*;
class Address
{
   String l1,l2,city,pin;
   Address(String a,String b,String c,String d)
   {
      l1=a;
      l2=b;
      city=c;
      pin=d;
   }

   void setl1(String x)
   {
     this.l1=x;
   }
```

```
String getl1()
{
   return this.l1;
}

void setl2(String x)
{
   this.l2=x;
}
String getl2()
{
   return this.l2;
}

void setcity(String x)
{
   this.city=x;
}
String getcity()
{
   return this.city;
}

void setpin(String x)
{
   this.pin=x;
}
String getpin()
{
   return this.pin;
}
}
class Cust /*Eta "Customer" hbe.....*/
{
   String custid, custname;
   Address address;
   Cust(String custid, String custname, Address address)
   {
      this.custid=custid;
      this.custname=custname;
      this.address=address;
   }

   String getcustid()
```

```
{
   return this.custid;
}
String getcustname()
{
   return this.custname;
}

String getl1()
{
   return this.address.l1;
}
String getl2()
{
   return this.address.l2;
   }
   String getcity()
   {
   return this.address.city;
   }
   String getpin()
   {
   return this.address.pin;
   }
}

class RegCustomer extends Cust /*Etao "Customer" hbe.....*/
{
   double fees;
   RegCustomer(String custid, String custname, Address address,double fees)
   {
    super(custid,custname,address);
    this.fees=fees;
}

void setcustid(String x)
{
   this.custid=x;
}
void setcustname(String x)
{
   this.custname=x;
}
void setfees(double x)
{
```

```
    this.fees=x;
}
void setl1(String x)
{
    this.address.l1=x;
}
void setl2(String x)
{
    this.address.l2=x;
}
void setcity(String x)
{
    this.address.city=x;
}
    void setpin(String x)
    {
        this.address.pin=x;
    }

    double getfees()
    {
        return this.fees;
    }

    void display()
    {
        System.out.println("Customer Id :"+this.custid+"\nCustomer Name
:"+this.custname+"\nCustomer fees :"+this.fees);
        System.out.println("Address 1 :"+this.address.l1+"\nAddress 2
:"+this.address.l2+"\nCity :"+this.address.city);
        System.out.println("Pin :"+this.address.pin);
    }
}
public class rg
{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        String l1=sc.nextLine();
        String l2=sc.nextLine();
        String city=sc.nextLine();
        String pin=sc.nextLine();
        Address a=new Address(l1,l2,city,pin);
        String custId=sc.nextLine();
        String custName=sc.nextLine();
```

```
        double fees=sc.nextDouble();
        RegCustomer ob=new RegCustomer(custId,custName,a,fees);
        ob.display();
    }
}
```

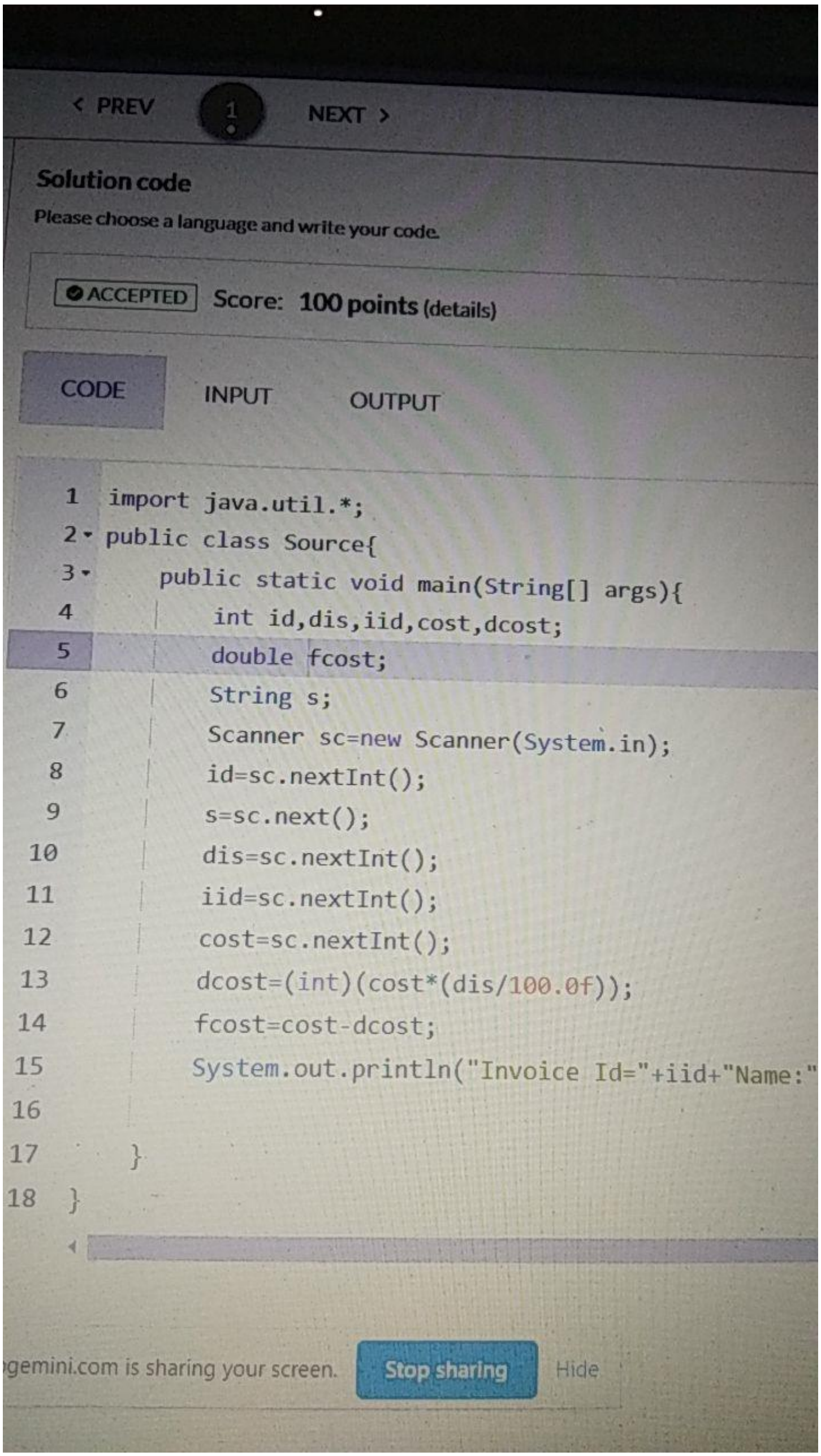
6. Batsman and Bowler :

```java
import java.util.Scanner;
interface IPlayer {
    void play();
}
class Batsman implements IPlayer {
    public void play() {
        System.out.println("Batsman is batting");
    }
}
class Bowler implements IPlayer {
    public void play() {
        System.out.println("Bowler is bowling");
    }
}
class Coach {
    private IPlayer player;
    void setplayer(IPlayer player) {
        this.player=player;
    }
    String coach() {
        Scanner sc=new Scanner(System.in);
        String s=sc.next();
        //String d;
        if(s=="Batsman") {
        Batsman b= new Batsman();
        b.play();

      } else if(s=="Bowler") {
        Bowler bo=new Bowler();
        bo.play();

```

```
19      }
20      String coach() {
21          Scanner sc=new Scanner(System.in);
22          String s=sc.next();
23          //String d;
24          if(s=="Batsman") {
25              Batsman b= new Batsman();
26              b.play();
27
28          } else if(s=="Bowler") {
29              Bowler bo=new Bowler();
30              bo.play();
31
32          } else {
33              return "Invalid Input";
34          }
35          return "";
36      }
37  }
38  class Source {
39      public static void main(String args[]) {
40          Coach c= new Coach();
41          c.coach();
42      }
43  }
```

1 revision found for this solution.

7. Customer and Invoice :



8. Customer and Address :

CODE INPUT OUTPUT Java 8

```
1  import java.io.*;
2  import java.util.*;
3  import java.text.*;
4  import java.math.*;
5  import java.util.regex.*;
6
7  // Class name should be "Source",
8  // otherwise solution won't be accepted
9  public class Source {
10     public static void main(String[] args) {
11
12         Scanner s = new Scanner(System.in);
13          String a= s.nextLine();// Declare the variable
14         System.out.println("Employee Id :"+a);
15         String b= s.nextLine();
16         System.out.println("Employee Name :"+b);
17         String c= s.nextLine();
18         System.out.println("Address 1 :"+c);
19         String d= s.nextLine();
20          System.out.println("Address 2 :"+d);
21          String e= s.nextLine();
22          System.out.println("City :"+e);
23          String f= s.nextLine();
24          System.out.println("Pin :"+f);
25     }
26 }
27
28
```

9. Fund Transfer :

```
import java.util.*;
import java.lang.*;
class Account1
{
    String id;
    String name;
    int balance =0;

    Account1(String id,String name)
    {
        this.id=id;
        this.name=name;
    }
    Account1(String id,String name,int balance)
    {
        this.id=id;
        this.name=name;
        this.balance=balance;
    }

    String getID(){
        return this.id;
    }

    String getName(){
        return this.name;
    }

    int getBalance(){
        return this.balance;
    }

    int credit(int amount){
        this.balance=amount+this.balance;
       return this.balance;
    }

    int debit(int amount){
        if(amount<=this.balance)
            this.balance=this.balance - amount;
        else
            System.out.print("Amount exceeded balance");
```

```
    return this.balance;
}

int transferTo(Account1 another,int amount){
    if(amount<balance){
        this.debit(amount);
        another.credit(amount);
    }
    else
        System.out.println("Insufficient Balance");

    return this.balance;
}
    public String toString() {
        return String.format("Account[id=%s,name=%s,balance=%d]",
id,name,balance);
    }
}
public class TestMain /*Eta “Source” class*/
{
   public static void main(String[] args) {
      // Test constructor and toString()
         Scanner sc=new Scanner(System.in);
         String id1=sc.nextLine();
         String name1=sc.nextLine();
         int balance1=sc.nextInt();
         sc.nextLine();
         String id2=sc.nextLine();
         String name2=sc.nextLine();
         int amount1=sc.nextInt();
         int amount2=sc.nextInt();
         int amount3=sc.nextInt();
      Account1 a1 = new Account1(id1,name1,balance1);
      System.out.println(a1.toString());
      Account1 a2 = new Account1(id2,name2);
      System.out.println(a2.toString());
      a1.credit(amount1);
      a1.debit(amount2);
      a1.transferTo(a2,amount3);
      System.out.println(a1.toString());
      System.out.println(a2.toString());
   }
}
```

10. Book and Author :

```
import java.util.*;
import java.lang.*;
class Account1
{
    String id;
    String name;
    int balance =0;

    Account1(String id,String name)
    {
        this.id=id;
        this.name=name;
    }
    Account1(String id,String name,int balance)
    {
        this.id=id;
        this.name=name;
        this.balance=balance;
    }

    String getID(){
        return this.id;
    }

    String getName(){
        return this.name;
    }

    int getBalance(){
        return this.balance;
    }

    int credit(int amount){
        this.balance=amount+this.balance;
       return this.balance;
    }

    int debit(int amount){
        if(amount<=this.balance)
            this.balance=this.balance - amount;
        else
            System.out.print("Amount exceeded balance");
```

```
    return this.balance;
}

int transferTo(Account1 another,int amount){
    if(amount<balance){
        this.debit(amount);
        another.credit(amount);
    }
    else
        System.out.println("Insufficient Balance");

    return this.balance;
}
    public String toString() {
        return String.format("Account[id=%s,name=%s,balance=%d]",
id,name,balance);
    }
}
public class TestMain {
    public static void main(String[] args) {
        // Test constructor and toString()
            Scanner sc=new Scanner(System.in);
            String id1=sc.nextLine();
            String name1=sc.nextLine();
            int balance1=sc.nextInt();
            sc.nextLine();
            String id2=sc.nextLine();
            String name2=sc.nextLine();
            int amount1=sc.nextInt();
            int amount2=sc.nextInt();
            int amount3=sc.nextInt();
        Account1 a1 = new Account1(id1,name1,balance1);
        System.out.println(a1.toString());
        Account1 a2 = new Account1(id2,name2);
        System.out.println(a2.toString());
        a1.credit(amount1);
        a1.debit(amount2);
        a1.transferTo(a2,amount3);
        System.out.println(a1.toString());
        System.out.println(a2.toString());
    }
}
```

Link :
https://github.com/etenbrinke/JavaTraining/tree/master/src/www3/ntu/edu

PDF File :
file:///C:/Users/NAVONIL/Downloads/kupdf.net_corejavaday2assignments.pdf
(**Whatsapp**)