01. College Admission

```
import java.util.Scanner;

public class Main
{
    String name;
    char gender;
    int mark_HSC, mark_SSLC, Tmark_HSC, Trmark_SSLC;
    float mark_Engineering;

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

                System.out.println("Applicant name");
                String name=sc.nextLine();

                System.out.println("Marks obtained in HSC");
                int mark_HSC=sc.nextInt();

                System.out.println("Total possible marks in HSC");
                int Tmark_HSC=sc.nextInt();

                System.out.println("Engineering cutoff mark");
                float mark_Engineering=sc.nextFloat();

                System.out.println("Marks obtained in SSLC");
                int mark_SSLC=sc.nextInt();

                System.out.println("Total possible marks in SSLC");
                int Trmark_SSLC=sc.nextInt();

                System.out.println("Gender");
                char gender=sc.next().charAt(0);

                System.out.println("Your Application has been Submitted
Successfully");
                System.out.println("The name of the applicant: "+name);
                System.out.println("Engineering Cutoff: "+mark_Engineering);
                System.out.println("Applicant gender: "+gender);
                System.out.println("All the best for your Career");

        }
}
```

02. Ludo King

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

                System.out.println("Enter Alex points");
                int pa=sc.nextInt();
                if (pa<0 || pa>50){
                    System.out.println(pa+" is an invalid number");
                    System.exit(1);
                }

                System.out.println("Enter Nikil points");
                int pn=sc.nextInt();
                if(pn<0 || pn>50){
                    System.out.println(pn+" is an invalid number");
                    System.exit(1);
                }

                System.out.println("Enter Sam points");
                int ps=sc.nextInt();
                if(ps<0 || ps>50){
                    System.out.println(ps+" is an invalid number");
                    System.exit(1);
                }

                if(pa>pn && pa>ps){
                    System.out.println("Alex scored "+pa+" points and won the
game");
                }
                else if (pn>pa && pn>ps){

                    System.out.println("Nikil scored "+pn+" points and won the
game");
                }
                else{
                    System.out.println("Sam scored "+ps+" points and won the
game");
                }

        }
}
```

03. Sim Card

```
import java.util.Scanner;

public class Main {
        public static void main(String[] args) {
                Scanner sc = new Scanner(System.in);

                System.out.println("Enter the phone number");
                long phn=sc.nextLong();

                int odd=0,even=0;
                long temp=phn,rem=0;

                while(temp>0){
                   rem=temp%10;
                   if(rem%2==0){
                       even+=rem;
                   }
                   else{
                       odd+=rem;
                   }
                   temp/=10;
                }
                if(odd>even){
                    System.out.println("Sum of odd is greater than sum of even");
                }
                else if (odd<even){
                    System.out.println("Sum of even is greater than sum of odd");
                }
                else{
                    System.out.println("Sum of odd and even are equal");
                }

        }
}
```

04. Oxygen Plants

```
import java.util.Scanner;
import java.util.Formatter;
import java.*;
public class Main {
        public static void main(String [] args)
        {
                Scanner sc=new Scanner(System.in);

                System.out.println("Enter the floor area of the room(m*m)");
                double l=sc.nextDouble();
                double b=sc.nextDouble();
                System.out.println("Enter the plant area of a single plant(in
cm2)");
                double area=sc.nextInt();

                double a=l*b;
                double bd=area/10000;
                double Tplant=a/bd;

                double rem=Tplant%10;
                Tplant-=rem;
                double oxygen=Tplant*0.9;

            String poxygen=String.format("%.02f",oxygen);
            String pl=String.format("%.02f",l);
            String pb=String.format("%.02f",b);
            String pTplant=String.format("%.0f",Tplant);


                System.out.printf("Total plants placed on floor area "+pl+"*"+pb+"
is "+pTplant+" plants produces "+poxygen+" litres of oxygen in a day");
        }

}
```

05. Sum of Ten

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

                System.out.println("Enter the number");
                int num=sc.nextInt();
                int sum=0;
                for (int i=num; i<=num+9; i++){
                    sum+=i;
                }
                System.out.println("The sum of ten numbers is "+sum);
        }
}
```

06. Electricity Board

//CustomerDetails

```
public class CustomerDetails {

        private String customerId;
        private String customerName;
        private long phoneNumber;
        private String city;
        private double unitConsumed;
        private double costPerUnit;

        public void setCustomerId(String customerId){
            this.customerId=customerId;
        }
        public void setCustomerName(String customerName){
            this.customerName=customerName;
        }
        public void setPhoneNumber(long phoneNumber){
            this.phoneNumber=phoneNumber;
        }
        public void setCity(String city){
            this.city=city;
        }
        public void setUnitConsumed(double unitConsumed){
            this.unitConsumed=unitConsumed;
        }
        public void setCostPerUnit(double costPerUnit){
            this.costPerUnit=costPerUnit;
        }
        public String getCustomerId(){
            return customerId;
        }
        public String getCustomerName(){
            return customerName;
        }
        public long getPhoneNumber(){
            return phoneNumber;
        }
        public String getCity(){
            return city;
        }
        public double getUnitConsumed(){
            return unitConsumed;
        }
        public double getCostPerUnit(){
            return costPerUnit;
        }
```

```
        public CustomerDetails(String customerId, String customerName, long
phoneNumber, String city, double unitConsumed, double costPerUnit){

            this.customerId=customerId;
            this.customerName=customerName;
            this.phoneNumber=phoneNumber;
            this.city=city;
            this.unitConsumed=unitConsumed;
            this.costPerUnit=costPerUnit;

        }

        public double calculateElectricityBill(){
            return(unitConsumed*costPerUnit);
        }
}

//================================================//

import java.util.Scanner;

public class Main {

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

                CustomerDetails c = new
CustomerDetails("Sample","Sample",123456789,"Sample",12.00,13.00);
                System.out.println("Enter Customer Id");
                c.setCustomerId(sc.nextLine());
                System.out.println("Enter Customer Name");
                c.setCustomerName(sc.nextLine());
                System.out.println("Enter Phone Number");
                c.setPhoneNumber(sc.nextLong());
                System.out.println("Enter City");
                c.setCity(sc.next());
                System.out.println("Enter Units Consumed");
                c.setUnitConsumed(sc.nextDouble());
                System.out.println("Enter Cost per Units");
                c.setCostPerUnit(sc.nextDouble());
                double amount=c.calculateElectricityBill();
                System.out.printf("Amount to be paid is Rs.%.2f",amount);

        }

}
```

07. Game Card Points

```
//CardPoints

public class CardPoints {

    private int cardId;
    private String holderName;
    private int balancePoints;

    public void setCardId(int cardId){
        this.cardId=cardId;
    }

    public int getCardId(){
        return cardId;
    }

    public void setHolderName(String holderName){
        this.holderName=holderName;
    }
    public String getHolderName(){
        return holderName;
    }
        public void setBalancePoints(int balancePoints){
            this.balancePoints=balancePoints;
        }
        public int getBalancePoints(){
            return balancePoints;
        }

        public boolean withdrawPoints(int points) {
                if(balancePoints<points){
                    System.out.println("Sorry!!! No enough points");
                    return false;
                }else{
                    int rem=balancePoints-points;
                   balancePoints=rem;
                   System.out.printf("Balance points after used:%d\n",rem);
                    return true;
                }
        }
}

//=====================================================//

import java.util.Scanner;

public class GameCardDetails {
```

```
    public CardPoints getCardDetails()
    {
        Scanner sc = new Scanner(System.in);
        CardPoints cp=new CardPoints();
        int cardId;
        String holderName;
        int balancePoints;
        System.out.println("Enter card id");
        cardId=sc.nextInt();
        System.out.println("Enter card holder name");
        holderName=sc.next();

        do{
            System.out.println("Enter balance points");
            balancePoints = sc.nextInt();
            if(balancePoints<=0){
                System.out.println("Balance points should be positive");
            }
        }while(balancePoints<=0);

        cp.setCardId(cardId);
        cp.setHolderName(holderName);
        cp.setBalancePoints(balancePoints);
        return cp;
    }
    public int getPointUsage()
    {
        Scanner sc = new Scanner(System.in);
        int points;
        do{
            System.out.println("Enter points should be used");
            points =sc.nextInt();
            if(points<=0){
                System.out.println("Points should be positive");
            }
        }while(points<=0);
        return points;
    }

    public static void main(String[] arg)
    {
        CardPoints cp = new CardPoints();
        GameCardDetails cd=new GameCardDetails();
        cp=cd.getCardDetails();
        int points=cd.getPointUsage();
        cp.withdrawPoints(points);
    }
}
```

08. Movie Ticket – Static

```
import java.util.Scanner;

public class Main
{
    static int availableTickets;
    public static void main(String[] arg)
    {
        Scanner sc=new Scanner(System.in);
        int n,nt;
        String name="";
        int ticketid, price;
        System.out.println("Enter movie name");
        name=sc.next();
        System.out.println("Enter no of bookings");
        n=sc.nextInt();
        System.out.println("Enter the available tickets");
        availableTickets=sc.nextInt();

        for (int i=0; i<n;i++ ){
            System.out.println("Enter the ticketid");
            ticketid=sc.nextInt();
            System.out.println("Enter the price");
            price=sc.nextInt();
            System.out.println("Enter the no of tickets");
            nt=sc.nextInt();

            Ticket o1=new Ticket();
            o1.setTicketId(ticketid);
            o1.setPrice(price);
            o1.setAvailableTickets(availableTickets);
            System.out.println("Available tickets: "+availableTickets);
            if(availableTickets>=nt){
                System.out.println("Total amount: "+o1.calculateTicketCost(nt));
                availableTickets=availableTickets-nt;
                if(availableTickets!=0){
                    System.out.println("Available ticket after booking:
"+availableTickets);
                }else{
                    System.out.println("House full");
                    break;
                }
            }
            else{
                System.out.println("Tickets are not available");
            }
        }
    }
}
```

```
//============================================//

public class Ticket
{
    private int ticketid;
    private int price;
    private static int availableTickets;

    public void setTicketId(int ticketid){
        this.ticketid=ticketid;
    }
    public int getTicketId(){
        return ticketid;
    }
    public void setPrice(int price){
        this.price=price;
    }
    public int getPrice(){
        return price;
    }
    public void setAvailableTickets(int availableTickets){
        this.availableTickets=availableTickets;
    }
    public int getAvailableTickets(){
        return availableTickets;
    }

    public int calculateTicketCost(int nooftickets)
    {
        if(availableTickets>=nooftickets){
            availableTickets=availableTickets - nooftickets;
            return (nooftickets*price);
        }
        else if (availableTickets==0){
            return -1;
        }
        else if (availableTickets< nooftickets){
            return -1;
        }
        return 0;
    }
}
```

09. Doctor Details

```
public class Doctor {

        private String doctorId;
        private String doctorName;
        private String specialization;
        private Hospital hospital;

        public Doctor(String doctorId, String doctorName, String specialization,
Hospital hospital){
            this.doctorId=doctorId;
            this.doctorName=doctorName;
            this.specialization=specialization;
            this.hospital=hospital;
        }

        public void setDoctorId(String doctorId){
            this.doctorId=doctorId;
        }

        public String getDoctorId(){
            return doctorId;
        }

        public void setDoctorName(String doctorName){
            this.doctorName=doctorName;
        }
        public String getDoctorName(){
            return doctorName;
        }

        public void setSpecialization(String specialization){
            this.specialization=specialization;
        }
        public String getSpecialization(){
            return specialization;
        }

        public void setHospital(Hospital hospital){
            this.hospital=hospital;
        }
        public Hospital getHospital(){
            return hospital;
        }
}

//================================================//
```

```
public class Hospital {

        private String hospitalName;
        private long contactNumber;
        private String city;

        public Hospital(String hospitalName, long contactNumber, String city){
            this.hospitalName=hospitalName;
            this.contactNumber=contactNumber;
            this.city=city;
        }

        public String getHospitalName(){
            return hospitalName;
        }
        public void setHospitalName(String hospitalName){
            this.hospitalName=hospitalName;
        }
        public long getContactNumber(){
            return contactNumber;
        }
        public void setContactNumber(long contactNumber){
            this.contactNumber=contactNumber;
        }
        public String getCity(){
            return city;
        }
        public void setCity(String city){
            this.city=city;
        }
}

//============================================================//

import java.util.Scanner;

public class Main {

        public static Doctor createDoctorDetails()
    {
        Scanner sc=new Scanner(System.in);
        String dname, spec, did, hname, city;
        long pnumber;
        System.out.println("Enter Hospital Name");
        hname=sc.next();
        System.out.println("Enter Contact Number");
        pnumber=sc.nextLong();
        System.out.println("Enter City");
        city=sc.next();
```

```
        Hospital hos = new Hospital(hname,pnumber,city);
        System.out.println("Enter Doctor Id");
        did=sc.next();
        System.out.println("Enter Doctor Name");
        dname=sc.next();
        System.out.println("Enter Specialization");
        spec=sc.next();

        Doctor d = new Doctor(did,dname,spec,hos);
        return d;
    }

        public static void main(String[] arg)
    {
       Scanner sc=new Scanner(System.in);
       Doctor d = createDoctorDetails();

       System.out.println("Doctor id: "+d.getDoctorId());
       System.out.println("Doctor name: "+d.getDoctorName());
       System.out.println("Specialization: "+d.getSpecialization());
       System.out.println("Hospital Name: "+d.getHospital().getHospitalName());
       System.out.println("Contact Number: "+d.getHospital().getContactNumber());
       System.out.println("City: "+d.getHospital().getCity());
    }

}
```

10. Incredible Toys

```
public class CustomerDetails {

        private String customerId;
        private String customerName;
        private long phoneNumber;
        private String emailId;
        private String toyType;
        private double price;

        public CustomerDetails(String customerId, String customerName, long
phonenumber,
        String emailId, String toyType, double price){
            this.emailId=emailId;
            this.toyType=toyType;
            this.customerId=customerId;
            this.customerName=customerName;
            this.phoneNumber=phoneNumber;
            this.price=price;
        }

        public double calculateDiscount() {
                String type =this.toyType;
                double discount=0;
                if(type.equalsIgnoreCase("SoftToys")){
                    discount=5;
                }else if (type.equalsIgnoreCase("FidgetToys")){
                    discount=10;
                }else if (type.equalsIgnoreCase("SensoryToys")){
                    discount=15;
                }else if (type.equalsIgnoreCase("Puzzles")){
                    discount=20;
                }
                discount=((this.price)*discount)/100;
                double cost = this.price-discount;
                return cost;
        }

        public String getCustomerId(){
            return customerId;
        }

        public void setCustomerId(String customerId){
            this.customerId=customerId;
        }

        public String getCustomerName(){
            return customerName;
        }
```

```
    public void setCustomerName(String customerName){
        this.customerName=customerName;
    }

    public long getPhoneNumber(){
        return phoneNumber;
    }

    public void setPhoneNumber(long phoneNumber){
        this.phoneNumber=phoneNumber;
    }
    public String getEmailId(){
        return emailId;
    }

    public void setEmailId(String emailId){
        this.emailId=emailId;
    }

    public String getToyType(){
        return toyType;
    }

    public void setToyType(String toyType){
       this.toyType=toyType;
    }

    public double getPrice(){
        return price;
    }

    public void setPrice(double price){
       this.price=price;
    }

public boolean validateNum(String str){
    boolean result =str.matches("[0-9]+");
    return result;
}

public boolean validateCustomerId(){
    String[] data=customerId.split("/");
    if(data.length==3){
        if(data[0].equalsIgnoreCase("Incredible")){
            if(data[1].length()==3){
            boolean check =validateNum(data[1]);
            if(check == true){
                if(data[2].length()==4){
                    boolean check1 =validateNum(data[2]);
```

```
                        if(check1==true){
                            return true;
                            }
                        }
                    }
                }
            }
        }
        return false;
    }
}

//=====================================================//

import java.util.Scanner;

public class Main {

        public static void main(String[] args){

            Scanner sc = new Scanner(System.in);

            System.out.println("Enter Customer Id");
            String cid=sc.next();
            System.out.println("Enter Customer Name");
            String name=sc.next();
            System.out.println("Enter Phone Number");
            long phone=sc.nextLong();
            System.out.println("Enter Email Id");
            String email=sc.next();
            System.out.println("Enter type");
            String type=sc.next();
            System.out.println("Enter Price");
            double price = sc.nextDouble();

            CustomerDetails cd = new
CustomerDetails(cid,name,phone,email,type,price);
            if(cd.validateCustomerId()==false){
                System.out.println("Provide a proper Customer Id");
                return;
            }
            System.out.printf("Amount to be paid by the Customer
%.2f\n",cd.calculateDiscount());
        }
}
```

11. PIN Number

```
import java.util.Scanner;

public class Main{

    public static void main(String[] args){

       Scanner sc=new Scanner(System.in);

       int n,x,c;
       System.out.println("Enter the total number of PIN numbers");
       n=sc.nextInt();
       if(n>0){
           int[] arr=new int[n];
           System.out.println("Enter the PIN numbers");
           for(int i=0; i<n; i++){
               arr[i]=sc.nextInt();
               if(arr[i]>0){
                   x=arr[i];
                   c=0;
                   while(x!=0){
                       x/=10;
                       ++c;
                   }
                   if(c<4 || c>4){
                       System.out.println(arr[i]+" is an invalid PIN number");
                       System.exit(0);
                   }}
                   else{
                       System.exit(0);
                   }
               }
               int flag=0,m=0;
               for(int k=0; k<n;k++){
                   int one = (arr[k]/1000)%10;
                   int two = (arr[k]/100)%10;
                   int three =(arr[k]/10)%10;
                   int four =arr[k]%10;

                   if((one%2)!=0 && (two%2)==0 && ( three==2 || three==3 ||
three==5 || three==7) && (four==4 || four==6 || four==8|| four==9)){
                       if(flag==0){
                           System.out.println("Valid PIN numbers are");
                       }
                       System.out.println(arr[k]);
                       flag=1;
                       m++;
                   }
               }
```

```
                if(m<1){
                    System.out.println("All these "+n+" numbers are not a valid PIN
number");
                }

            }

            else{
                System.out.println(+n+" is an invalid number");
            }
        }
    }
```

12. Resort booking

```
import java.util.Scanner;

public class Main{

    public static void Check(int adult, int child, int day){
        if(adult<0){
            System.out.println("Invalid input for number of adults");
            System.exit(1);
        }
        else if (child<0){
            System.out.println("Invalid input for number of children");
            System.exit(1);
        }
        else if (day<=0){
            System.out.println("Invalid input for number of days");
            System.exit(1);
        }
    }

    public static void CalCost(String name, int period, int child, int day){
        int total=((period*1000)+(child*650))*day;
        System.out.println(name+" your booking is confirmed and the total cost is
Rs "+total);

    }

    public static void main(String[] args){

       Scanner sc=new Scanner(System.in);

       String input=sc.next();
       String[] str=input.split(":");
       String name=str[0];
       int adults=Integer.parseInt(str[1]);
       int childs=Integer.parseInt(str[2]);
       int days=Integer.parseInt(str[3]);

       Check(adults,childs,days);
       CalCost(name,adults,childs,days);
    }
}
```

13. Find the winner

```
import java.util.Scanner;

public class Main{
    public static boolean flag=false;

    public static int findWinner(Float[] sum){
        int index=0;
        float fastest=sum[0];
        for (int i=1; i<sum.length;i++ ){
            if(sum[i] < fastest){
                fastest=sum[i];
                index=i;
            }
        }
        return index;
    }

    public static void main(String[] args){

       Scanner sc=new Scanner(System.in);
       System.out.println("Enter the number of teams");
       int no_of_teams=sc.nextInt();
       if(no_of_teams>1){
           Float[] sum = new Float[no_of_teams];
           String[] teams = new String[no_of_teams];
           System.out.println("Enter the details");
           for (int i=0; i<no_of_teams; i++){
               teams[i]=sc.next();
               String[] td=teams[i].split(":");
               if(Float.parseFloat(td[1])<1.00 || Float.parseFloat(td[2])<1.00 ||
Float.parseFloat(td[3])<1.00 || Float.parseFloat(td[4])<1.00){
                   System.out.println("Invalid number");
                   flag = false;
                   break;
               }
               else{
                   flag=true;

sum[i]=Float.parseFloat(td[1])+Float.parseFloat(td[2])+Float.parseFloat(td[3])+Floa
t.parseFloat(td[4]);
               }
           }
           if(flag){
               int winnerIndex=findWinner(sum);
               System.out.print(teams[winnerIndex].split(":")[0]+" team wins the
race in");
               System.out.printf(" %.2f ",sum[winnerIndex]);
               System.out.print("minutes");
```

```
            }
        }else{
            System.out.println("Invalid input");
        }
    }
}
```

14. Fishing competition

```
import java.util.Scanner;

public class Main{

    public static int Points(int age, int big, int medium, int small, int count){
        int total=0;
        if(age<18){
            System.out.println(age+" is an invalid age");
            System.exit(1);
        }
        else if(count<0){
            System.out.println(count+" is an invalid input");
            System.exit(1);
        }
        else if(big<0){
            System.out.println(big+" is an invalid input");
            System.exit(1);
        }
        else if(medium<0){
            System.out.println(medium+" is an invalid input");
            System.exit(1);
        }
        else if(small<0){
            System.out.println(small+" is an invalid input");
        }
        else{
            total=(big*10)+(medium*6)+(small*3);
        }
        return total;
    }

    public static void main(String[] args){

       Scanner sc=new Scanner(System.in);
       System.out.println("Enter the details");
       String str=sc.next();
       String[] details=new String[4];
       details=str.split(":");
       String name=details[0];
       int age=Integer.parseInt(details[1]);
       int big=Integer.parseInt(details[2]);
       int medium=Integer.parseInt(details[3]);
       int small=Integer.parseInt(details[4]);
       int count=big+medium+small;
       int x=Points(age,big,medium,small,count);
       System.out.println(name+" scored "+x+" points");

    }
```

```
}
```

15. Two arrays game

```
import java.util.Scanner;

public class Main{

    public static int[] Calculate(int[] a, int[] b, int Size){
        int[] finalArray=new int[Size];

        for ( int i=0; i<Size; i++){
            finalArray[i]=a[i]+b[i];
            i++;
        }
        for ( int j=1; j<Size; j++){
            finalArray[j]=a[j]-b[j];
            j++;
        }
        return finalArray;
    }

    public static void main(String[] args){

       Scanner sc=new Scanner(System.in);

       System.out.println("Enter the size for the first array");
       int Size=sc.nextInt();
       if(Size<=0){
           System.out.println("Invalid array size");
           System.exit(1);
       }

       System.out.println("Enter the elements for the first array");
       int[] farray=new int[Size];
       for ( int i=0; i<Size; i++){
           farray[i]=sc.nextInt();
       }

       System.out.println("Enter the size for the second array");
       int Size2=sc.nextInt();
       if(Size2<=0){
           System.out.println("Invalid array size");
           System.exit(1);
       }

       if(Size!=Size2){
           System.out.println("Both array size are not same");
           System.exit(1);
       }

       System.out.println("Enter the elements for the second array");
```

```
        int[] sarray=new int[Size2];
        for ( int i=0; i<Size2; i++){
            sarray[i]=sc.nextInt();
        }
        int[] x=Calculate(farray,sarray,Size);
        System.out.println("The elements of the third array");
        for (int i=0; i<x.length; i++){
            System.out.println(x[i]);
        }
    }
}
```

16. Disney Tourism

```
//BoatHouseBooking

public class BoatHouseBooking extends Booking{

        int noOfDays;
        String foodType;
        public BoatHouseBooking(String customerName, String cityName, String
phoneNumber, int noOfPeople, int noOfDays, String foodType){
            super(customerName,cityName,phoneNumber,noOfPeople);
            this.noOfDays=noOfDays;
            this.foodType=foodType;
        }

        public double calculateTotalAmount() {

                return foodType.toLowerCase().equals("nonveg") ? noOfPeople*800 +
noOfDays*3000 + 500 : noOfPeople*800 + noOfDays*3000 + 250;
        }


}

//=============================================//

//BoatRideBooking
public class BoatRideBooking extends Booking{

        private float noOfHours;
        private String guide;

        public BoatRideBooking(String customerName, String cityName, String
phoneNumber, int noOfPeople, float noOfHours, String guide){
            super(customerName,cityName,phoneNumber,noOfPeople);
            this.noOfHours=noOfHours;
            this.guide=guide;
        }

        public double calculateTotalAmount() {
                return guide.toLowerCase().equals("yes") ? noOfPeople*80 +
noOfHours*300 + 150 : noOfPeople*80 + noOfHours*300;
        }
}

//=============================================//

//Booking

import java.util.*;
```

```
public abstract class Booking {

        protected String customerName;
        protected String cityName;
        protected String phoneNumber;
        protected int noOfPeople;

        Booking(String customerName, String cityName, String phoneNumber, int
noOfPeople){
            this.customerName=customerName;
            this.cityName=cityName;
            this.phoneNumber=phoneNumber;
            this.noOfPeople=noOfPeople;
        }

        public String getCustomerName(){
            return customerName;
        }
        public void setCustomerName(String customerName){
            this.customerName=customerName;
        }

        public String getCityName(){
            return cityName;
        }
        public void setCityName(String cityName){
            this.cityName=cityName;
        }

        public String getPhoneNumber(){
            return phoneNumber;
        }
        public void setPhoneNumber(String phoneNumber){
            this.phoneNumber=phoneNumber;
        }

        public int getNoOfPeople(){
            return noOfPeople;
        }
        public void setNoOfPeople(int noOfPeople){
            this.noOfPeople=noOfPeople;
        }

        public abstract double calculateTotalAmount();
}

//=============================================//

//UserInterface
```

```
import java.util.Scanner;

public class UserInterface {

	public static void main(String[] args) {

		Scanner sc = new Scanner(System.in);

		System.out.println("Enter the Customer Name");
		String cname=sc.nextLine();
		System.out.println("Enter the City name");
		String cityName=sc.nextLine();
		System.out.println("Enter the phone number");
		String phoneNumber=sc.nextLine();
		System.out.println("Enter number of people");
		int noOfPeople=Integer.parseInt(sc.nextLine());
		System.out.print("Enter the option\n1. Boat House Booking\n2. Boat
Ride Booking\n");

		int choice=Integer.parseInt(sc.nextLine());
		int days=0;

		if(choice==1){
		    System.out.println("Enter number of days");
		    days=Integer.parseInt(sc.nextLine());
		    System.out.println("Enter food type (Veg/NonVeg)");
		    String foodType=sc.nextLine();

		    BoatHouseBooking bh = new BoatHouseBooking(cname, cityName,
phoneNumber, noOfPeople, days, foodType);
		    System.out.println("Your booking has been confirmed pay
Rs."+bh.calculateTotalAmount());
		}
		else{
		    int noOfHours=0;
		    String guide="";
		    System.out.println("Enter number of hours");
		    noOfHours=Integer.parseInt(sc.nextLine());
		    System.out.println("Do you want guide (Yes/No)");
		    guide=sc.nextLine();

		    BoatRideBooking br = new BoatRideBooking(cname, cityName,
phoneNumber, noOfPeople, noOfHours, guide);
		    System.out.println("Your booking has been confirmed pay
Rs."+br.calculateTotalAmount());
		}
	}

}
```

17. Vivek Furnitures - Polymorphism

```
//Bero

public abstract class Bero {

    protected String beroType;
    protected String beroColour;
    protected double price;

        Bero(String beroType, String beroColour){
            this.beroType=beroType;
            this.beroColour=beroColour;
        }

        public String getBeroType(){
            return beroType;
        }
        public void setBeroType(String beroType){
            this.beroType=beroType;
        }

        public String getBeroColour(){
            return beroColour;
        }
        public void setBeroColour(String beroColour){
            this.beroColour=beroColour;
        }

        public double getPrice(){
            return price;
        }
        public void setPrice(double price){
            this.price=price;
        }

        public abstract void calculatePrice();
}

//=========================================//


public class CustomerDetails {

    private String customerName;
    private long phoneNumber;
    private String address;

    public CustomerDetails(String customerName, long phoneNumber, String address){
        this.customerName=customerName;
```

```
        this.phoneNumber=phoneNumber;
        this.address=address;
    }

    public String getCustomerName(){
        return customerName;
    }
    public void setCustomerName(String customerName){
        this.customerName=customerName;
    }

    public long getPhoneNumber(){
        return phoneNumber;
    }
    public void setPhoneNumber(long phoneNumber){
        this.phoneNumber=phoneNumber;
    }

    public String getAddress(){
        return address;
    }
    public void setAddress(String address){
        this.address=address;
    }
}

//=============================================//

public class Discount {

        public double calculateDiscount(Bero bObj) {
            double discount=0;
                if(bObj instanceof SteelBero){
                    discount=.10*bObj.getPrice();
                }
                else if (bObj instanceof WoodenBero){
                    discount=.15*bObj.getPrice();
                }
                return discount;
        }
}

//=================================================//


public class SteelBero extends Bero{

        private int beroHeight;

        public int getBeroHeight(){
```

```
            return beroHeight;
        }
        public void setBeroHeight(int beroHeight){
            this.beroHeight=beroHeight;
        }

        public SteelBero(String beroType,String beroColour,int beroHeight){
            super(beroType,beroColour);
            this.beroHeight=beroHeight;
        }

        public void calculatePrice() {
                double totalPrice=0;
                if(beroHeight==3){
                    totalPrice=5000;
                }
                else if (beroHeight==5){
                    totalPrice=8000;
                }
                else if (beroHeight==7){
                    totalPrice=10000;
                }
            setPrice(totalPrice);
        }

}

//=============================================//


public class WoodenBero extends Bero{

        private String woodType;

        public WoodenBero(String beroType, String beroColour, String woodType){
            super(beroType,beroColour);
            this.woodType=woodType;
        }

        public void setWoodType(String woodType){
            this.woodType=woodType;
        }
        public String getWoodType(){
            return woodType;
        }

        public void calculatePrice() {
            double totalPrice=0;
            if(woodType.equals("Ply Wood")){
                totalPrice=15000;
```

```
            }
            else if (woodType.equals("Teak Wood")){
                totalPrice=12000;
            }
            else if (woodType.equals("Engineered Wood")){
                totalPrice=10000;
            }
                setPrice(totalPrice);
        }

}

//=========================================//

import java.util.Scanner;

public class Main {

        public static void main(String[] args) {

                double TotalPrice=0;
                Discount d=new Discount();
                Scanner sc = new Scanner(System.in);

                System.out.println("Enter Customer Name");
                String cname=sc.nextLine();
                System.out.println("Enter Phone Number");
                long phno=Long.parseLong(sc.nextLine());
                System.out.println("Enter address");
                String ads=sc.nextLine();
                System.out.println("Enter Bero Type");
                String btype=sc.nextLine();
                System.out.println("Enter Bero Colour");
                String bColour=sc.nextLine();

                if(btype.equals("Wooden Bero")){
                    System.out.println("Enter Wood Type");
                    String wType=sc.nextLine();

                    WoodenBero wb = new WoodenBero(btype,bColour,wType);

                    wb.calculatePrice();
                    TotalPrice= wb.getPrice()-d.calculateDiscount(wb);

                    System.out.printf("Amount needs to be paid
Rs.%.2f",TotalPrice);
                }

                else if (btype.equals("Steel Bero")){
                    System.out.println("Enter Bero Height");
```

```
                    int height=Integer.parseInt(sc.nextLine());

                    SteelBero sb =new SteelBero(btype,bColour,height);

                    sb.calculatePrice();
                    TotalPrice=sb.getPrice()-d.calculateDiscount(sb);

                    System.out.printf("Amount needs to be paid
Rs.%.2f",TotalPrice);

                }
        }
}
```

18. Departmental Store – Interface

```
public interface BonusPoints {

        double calculateBonusPoints();

}

//==================================//


public class CustomerDetails implements BonusPoints, DoorDelivery{

        private String customerName;
        private String phoneNumber;
        private String streetName;
        private double billAmount;
        private int distance;

        public CustomerDetails(String customerName, String phoneNumber, String
streetName, double billAmount, int distance){
            this.customerName=customerName;
            this.phoneNumber=phoneNumber;
            this.streetName=streetName;
            this.billAmount=billAmount;
            this.distance=distance;
        }

        public String getCustomerName(){
            return customerName;
        }
        public void setCustomerName(String customerName){
            this.customerName=customerName;
        }

        public String getPhoneNumber(){
            return phoneNumber;
        }
        public void setPhoneNumber(String phoneNumber){
            this.phoneNumber=phoneNumber;
        }

        public String getStreetName(){
            return streetName;
        }
        public void setStreetName(String streetName){
            this.streetName=streetName;
        }

        public double getBillAmount(){
```

```
            return billAmount;
        }
        public void setBillAmount(double billAmount){
            this.billAmount=billAmount;
        }

        public int getDistance(){
            return distance;
        }
        public void setDistance(int distance){
            this.distance=distance;
        }

        public double calculateBonusPoints() {
                if(billAmount>=250){
                    return billAmount/10;
                }
                return 0;
        }

        public double deliveryCharge() {
                if(distance>=25){
                    return distance*8;
                }
                else if (distance >= 15 && distance <25){
                    return distance*5;
                }
                return distance*2;
        }
}

//=============================================//


public interface DoorDelivery {

        double deliveryCharge();
}

//=============================================//

import java.util.Scanner;

public class Main {
        public static void main(String[] args) {
                Scanner sc = new Scanner (System.in);

                System.out.println("Enter the customer name");
                String cname=sc.nextLine();
                System.out.println("Enter the phone number");
```

```
                String pno=sc.nextLine();
                System.out.println("Enter the street name");
                String streetname=sc.nextLine();
                System.out.println("Enter the bill Amount");
                double billAmount=Double.parseDouble(sc.nextLine());
                System.out.println("Enter the distance");
                int distance=Integer.parseInt(sc.nextLine());
                CustomerDetails cd=new
CustomerDetails(cname,pno,streetname,billAmount,distance);
                System.out.println("Customer name "+cd.getCustomerName());
                System.out.println("phone number "+cd.getPhoneNumber());
                System.out.println("Street name "+cd.getStreetName());
                System.out.println("Bonus points "+cd.calculateBonusPoints());
                System.out.println("Delivery charge "+cd.deliveryCharge());
        }
}
```

19. College Fee – Abstract Class

```
public class DayScholar extends Student{

        private int busNumber;
        private float distance;

        public DayScholar(int studentId, String studentName, String department,
String gender, String category, double collegeFee, int busNumber, float distance){
            super(studentId,studentName,department,gender,category,collegeFee);
            this.busNumber=busNumber;
            this.distance=distance;
        }

        public int getBusNumber(){
            return busNumber;
        }
        public void setBusNumber(int busNumber){
            this.busNumber=busNumber;
        }

        public float getDistance(){
            return distance;
        }
        public void setDistance(float distance){
            this.distance=distance;
        }

        public double calculateTotalFee() {

                int busFee=0;
                if(distance>30 && distance <=40){
                    busFee=28000;
                }
                else if (distance>20 && distance<=30){
                    busFee=20000;
                }
                else if(distance >10 && distance <= 20){
                    busFee=12000;
                }
                else{
                    busFee=6000;
                }
                return (collegeFee+busFee);
        }
}

//=============================================//
```

```
public abstract class Student {

        protected int studentId;
        protected String studentName;
        protected String department;
        protected String gender;
        protected String category;
        protected double collegeFee;

        public Student(int studentId, String studentName, String department, String
gender, String category, double collegeFee){
            this.studentId=studentId;
            this.studentName=studentName;
            this.department=department;
            this.gender=gender;
            this.category=category;
            this.collegeFee=collegeFee;
        }

        public int getStudentId(){
            return studentId;
        }

        public void setStudentId(int studentId){
            this.studentId=studentId;
        }

        public String getStudentName(){
            return studentName;
        }

        public void setStudentName(String studentName){
            this.studentName=studentName;
        }

        public String getDepartment(){
            return department;
        }

        public void setDepartment(String department){
            this.department=department;
        }

        public String getGender(){
            return gender;
        }
        public void setGender(String gender){
            this.gender=gender;
        }
```

```
        public String getCategory(){
            return category;
        }
        public void setCategory(String category){
            this.category=category;
        }

        public double getCollegeFee(){
            return collegeFee;
        }
        public abstract double calculateTotalFee();

}

//=========================================//

import java.util.Scanner;

public class UserInterface {

        public static void main(String[] args) {

                Scanner sc = new Scanner(System.in);

        System.out.println("Enter Student Id");
        int studentId= Integer.parseInt(sc.nextLine());
        System.out.println("Enter Student name");
        String name=sc.nextLine();
        System.out.println("Enter Department name");
        String deptName=sc.nextLine();
        System.out.println("Enter gender");
        String gender=sc.nextLine();
        System.out.println("Enter category");
        String category=sc.nextLine();
        System.out.println("Enter College fee");
        double collegeFee=Double.parseDouble(sc.nextLine());

        if(category.equals("DayScholar")){
            System.out.println("Enter Bus number");
            int busNumber=Integer.parseInt(sc.nextLine());
            System.out.println("Enter the distance");
            int distance=Integer.parseInt(sc.nextLine());

            DayScholar dayScholar=new DayScholar(studentId, name, deptName, gender,
category, collegeFee, busNumber, distance);
            System.out.println("Total College fee is
"+dayScholar.calculateTotalFee());
        }
        else{
            System.out.println("Enter the room number");
```

```
            int roomNumber=Integer.parseInt(sc.nextLine());
            System.out.println("Enter the Block name");
            char blockName=sc.nextLine().charAt(0);
            System.out.println("Enter the room type");
            String roomType=sc.nextLine();

            Hosteller hosteller=new
Hosteller(studentId,name,deptName,gender,category,collegeFee,roomNumber,blockName,r
oomType);
            System.out.println("Total College fee is
"+hosteller.calculateTotalFee());
        }

        }

}

//=====================================//

public class Hosteller extends Student{

    private int roomNumber;
    private char blockName;
    private String roomType;

    public Hosteller(int studenId,String studentName, String department, String
gender, String category, double collegeFee, int roomNumber, char blockName, String
roomType){
        super(studenId,studentName,department,gender,category,collegeFee);
        this.roomNumber=roomNumber;
        this.blockName=blockName;
        this.roomType=roomType;
    }

    public int getRoomNumber(){
        return roomNumber;
    }
    public void setRoomNumber(int roomNumber){
        this.roomNumber=roomNumber;
    }

    public char getBlockName(){
        return blockName;
    }
    public void setBlockName(char blockName){
        this.blockName=blockName;
    }
    public String getRoomType(){
        return roomType;
    }
```

```
    public void setRoomType(String roomType){
        this.roomType=roomType;
    }

    public double calculateTotalFee(){
        int roomFee=0;
        int hostelFee=0;
        if(blockName=='A'){
            hostelFee=60000;
            if(roomType.equals("AC")){
                roomFee=8000;
            }
        }
        else if (blockName=='B'){
            hostelFee=50000;
            if(roomType.equals("AC")){
                roomFee=5000;
            }
        }
        else if (blockName=='C'){
            hostelFee=40000;
            if(roomType.equals("AC")){
                roomFee=2500;
            }
        }
        return collegeFee+hostelFee+roomFee;
    }
}
```

20. Endowment plan – Inheritance

```
public class EducationalEndowment extends Endowment{

    private String educationalInstitution;
    private String educationalDivision;

    public EducationalEndowment(String endowmentId, String holderName, String
endowmentType, String registrationDate, String educationalInstitution, String
educationalDivision){
        super(endowmentId,holderName,endowmentType, registrationDate);
        this.educationalInstitution=educationalInstitution;
        this.educationalDivision=educationalDivision;
    }

    public String getEducationalInstitution(){
        return educationalInstitution;
    }
    public void setEducationalInstitution(String educationalInstitution){
        this.educationalInstitution=educationalInstitution;
    }

    public String getEducationalDivision(){
        return educationalDivision;
    }
    public void setEducationalDivision(String educationalDivision){
        this.educationalDivision=educationalDivision;
    }

        public double calculateEndowment(){
                int endowmentAmount=0;
                if(educationalDivision.equalsIgnoreCase("School")){
                    endowmentAmount=30000;
                }
                else if (educationalDivision.equalsIgnoreCase("UnderGraduate")){
                    endowmentAmount=60000;
                }
                else{
                    endowmentAmount=90000;
                }
                return endowmentAmount;
        }
}

//================================================//


public abstract class Endowment {

        private String endowmentId;
```

```
        private String holderName;
        private String endowmentType;
        private String registrationDate;

        public Endowment(String endowmentId, String holderName, String
endowmentType, String registrationDate){
            this.endowmentId=endowmentId;
            this.holderName=holderName;
            this.endowmentType=endowmentType;
            this.registrationDate=registrationDate;
        }

        public String getEndowmentId(){
            return endowmentId;
        }
        public void setEndowmentId(String endowmentId){
            this.endowmentId=endowmentId;
        }
        public String getHolderName(){
            return holderName;
        }
        public void setHolderName(String holderName){
            this.holderName=holderName;
        }
        public String getEndowmentType(){
            return endowmentType;
        }
        public void setEndowmentType(String endowmentType){
            this.endowmentType=endowmentType;
        }
        public String getRegistrationDate(){
            return registrationDate;
        }
        public void setRegistrationDate(String registrationDate){
            this.registrationDate=registrationDate;
        }
        public abstract double calculateEndowment();
}

//================================================//


public class HealthEndowment extends Endowment{

        private String healthCareCenter;
        private int holderAge;

        public HealthEndowment(String endowmentId, String holderName, String
endowmentType, String registrationDate, String healthCareCenter, int holderAge){
            super(endowmentId,holderName,endowmentType,registrationDate);
```

```
            this.healthCareCenter=healthCareCenter;
            this.holderAge=holderAge;
        }

        public String getHealthCareCenter(){
            return healthCareCenter;
        }

        public void setHealthCareCenter(String healthCareCenter){
            this.healthCareCenter=healthCareCenter;
        }

        public int getHolderAge(){
            return holderAge;
        }
        public void setHolderAge(int holderAge){
            this.holderAge=holderAge;
        }

        public double calculateEndowment(){
                int endowmentAmount=0;
                if(holderAge<=30){
                    endowmentAmount=120000;
                }
                else if (holderAge>30 && holderAge<60){
                    endowmentAmount=200000;
                }
                else{
                    endowmentAmount=500000;
                }
                return endowmentAmount;
        }
}

//=====================================================//

import java.util.Scanner;

public class UserInterface {

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

                System.out.println("Enter Endowment Id");
                String endowmentId=sc.nextLine();
                System.out.println("Enter Holder Name");
                String holderName=sc.nextLine();
                System.out.println("Enter Endowment Type");
                String endowmentType=sc.nextLine();
                System.out.println("Enter Registration Date");
```

```
                String registrationDate=sc.nextLine();

                if(endowmentType.equalsIgnoreCase("Educational")){
                    System.out.println("Enter Educational Institution");
                    String educationalInstitution=sc.nextLine();
                    System.out.println("Enter Educational Division");
                    String educationalDivision=sc.nextLine();
                    EducationalEndowment educationalEndowment = new
EducationalEndowment(endowmentId,holderName,endowmentType,registrationDate,educatio
nalInstitution,educationalDivision);
                    System.out.println("Endowment Amount
"+educationalEndowment.calculateEndowment());
                }
                else{
                    System.out.println("Enter Health Care Center");
                    String healthCenter=sc.nextLine();
                    System.out.println("Enter Holder Age");
                    int holderAge=Integer.parseInt(sc.nextLine());
                    HealthEndowment healthEndowment=new
HealthEndowment(endowmentId,holderName,endowmentType,registrationDate,healthCenter,
holderAge);
                    System.out.println("Endowment Amount
"+healthEndowment.calculateEndowment());
                }
        }

}
```

21. Auditing

```
import java.util.ArrayList;

public interface EmployeeAudit {
    public ArrayList<String> fetchEmployeeDetails(double salary);
}

//=======================================//

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

public class Main {

        private static Map <String,Double> employeeMap = new
HashMap<String,Double>();

        public Map<String, Double> getEmployeeMap() {
                return employeeMap;
        }

        public void setEmployeeMap(Map<String, Double> employeeMap) {
                this.employeeMap = employeeMap;
        }

        public void addEmployeeDetails(String employeeName, double salary)
        {
                employeeMap.put(employeeName,salary);
        }

        public static EmployeeAudit findEmployee()
        {
                ArrayList<String> name= new ArrayList<String>();

                EmployeeAudit employeeAudit = (search)->{
                    for (Map.Entry<String,Double> i:employeeMap.entrySet() )
                    if(i.getValue()<=search){
                        name.add(i.getKey());
                    }
                    return name;
                };

                return employeeAudit;
        }

        public static void main(String[] args)
        {
```

```
                Main emp = new Main();
                Scanner sc=new Scanner(System.in);
                int choice=0;
                do{
                    System.out.println("1. Add Employee details");
                    System.out.println("2. Find Employee details");
                    System.out.println("3. Exit");
                    System.out.println("Enter the choice");
                    choice=Integer.parseInt(sc.nextLine());

                    switch(choice){
                        case 1:
                            System.out.println("Enter the Employee name");
                            String name=sc.nextLine();
                            System.out.println("Enter the Employee Salary");
                            double salary=Double.parseDouble(sc.nextLine());
                            emp.addEmployeeDetails(name,salary);
                            break;

                        case 2:
                            System.out.println("Enter the salary to be searched");
                            double search=Double.parseDouble(sc.nextLine());
                            ArrayList<String>
nameList=findEmployee().fetchEmployeeDetails(search);
                            if(nameList.isEmpty()){
                                System.out.println("No employee found");
                            }
                            else{
                                System.out.println("Employee List");
                                for(String empName: nameList){
                                    System.out.println(empName);
                                }
                            }
                            break;

                        default:
                            break;
                }}while(choice!=3);

                System.out.println("Let's complete the session");

        }
}
```

22. Number Category

```
public interface NumberCategory{
        public boolean checkNumberCategory(int num1,int num2);
}

//=======================================//

import java.util.*;

public class NumberCategoryUtility{

    static int findFactor(int n){
        int i;
        int sum=0;
        for ( i=1; (i*i)<n; i++){
            if(n%i==0){
                sum+=i;
            }
        }
        if(i-(n/i)==1){
            i--;
        }
        for (;i>1; i--){
            if(n%i==0){
                sum+=(n/i);
            }
        }
        return sum;
    }

    public static boolean isPalindrome(int num){
        String n=String.valueOf(num);
        int i=0;
        int j=n.length()-1;

        while(i<j){
            if(n.charAt(i)==n.charAt(j)){
                i++;
                j--;
                continue;
            }return false;
        }return true;
    }

    public static NumberCategory checkAmicable(){
        NumberCategory amicable=((number1,number2)->{
            int n1=findFactor(number1);
            int n2=findFactor(number2);
            if(number1==n2 && number2==n1){
```

```
                return true;
            }
            return false;
        });
        return amicable;
    }

    public static NumberCategory checkPalindrome(){
        NumberCategory
palindrome=(((number1,number2)->isPalindrome(number1*number2)));
        return palindrome;
    }

    public static void main(String [] args)
    {
        Scanner sc=new Scanner(System.in);
        int num1=Integer.parseInt(sc.nextLine());
        int num2=Integer.parseInt(sc.nextLine());
        boolean isAmicable=checkAmicable().checkNumberCategory(num1,num2);
        boolean isPalindrome=checkPalindrome().checkNumberCategory(num1,num2);

        if(isAmicable){
            System.out.println("The numbers are amicable");
        }
        else{
            System.out.println("The numbers are not amicable");
        }
        if(isPalindrome){
            System.out.println("Product do produces a palindrome");
        }
        else{
            System.out.println("Product does not produce a palindrome");
        }
    }
}
```

23. Travel Agency

```
public interface CommissionInfo{
        public double calculateCommissionAmount(Ticket ticketObj);
}

//=======================================//

public class Ticket {

    private long pnrNo;
    private String passengerName;
    private int seatNo;
    private String classType;
    private double ticketFare;

    public long getPnrNo() {
        return pnrNo;
    }

    public void setPnrNo(long pnrNo) {
            this.pnrNo = pnrNo;
    }

    public String getPassengerName() {
        return passengerName;
    }

    public void setPassengerName(String passengerName) {
            this.passengerName = passengerName;
    }

    public int getSeatNo() {
            return seatNo;
    }

    public void setSeatNo(int seatNo) {
        this.seatNo = seatNo;
    }

    public String getClassType() {
        return classType;
    }

    public void setClassType(String classType) {
        this.classType = classType;
    }

    public double getTicketFare() {
        return ticketFare;
```

```
    }

    public void setTicketFare(double ticketFare) {
            this.ticketFare = ticketFare;
    }

    public Ticket(long pnrNo, String passengerName, int seatNo, String classType,
double ticketFare){
            this.pnrNo=pnrNo;
            this.passengerName=passengerName;
            this.seatNo=seatNo;
            this.classType=classType;
            this.ticketFare=ticketFare;
    }
}

//=====================================//

import java.util.*;

public class UserInterface{

    public static CommissionInfo generateCommissionObtained(){
        CommissionInfo commissionInfo=(ticketObj-> {
            double commissionAmt=0;

if(ticketObj.getClassType().equalsIgnoreCase("sl")||ticketObj.getClassType().equals
IgnoreCase("2s")){
                commissionAmt+=60;
            }
            else{
                commissionAmt+=100;
            }
            return commissionAmt;
        });
        return commissionInfo;
    }

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

        System.out.println("Enter the no of passengers");
        int count=Integer.parseInt(sc.nextLine());
        Ticket[] tickets=new Ticket[count];
        for (int i=1; i<=count; i++){
            Ticket ticket;
            System.out.printf("Details of Passenger %d:\n",i);
            System.out.println("Enter the pnr no:");
            long pnrNo = Long.parseLong(sc.nextLine());
```

```
            System.out.println("Enter passenger name:");
            String passengerName=sc.nextLine();
            System.out.println("Enter seat no:");
            int setSeatNo=Integer.parseInt(sc.nextLine());
            System.out.println("Enter class type:");
            String setClassType=sc.nextLine();
            System.out.println("Enter ticket fare:");
            double setTicketFare=Double.parseDouble(sc.nextLine());
            tickets[i-1]=new
Ticket(pnrNo,passengerName,setSeatNo,setClassType,setTicketFare);
        }
        System.out.println("Commission Obtained");
        double commission=0;
        for (int i=0; i<tickets.length;i++){

commission+=generateCommissionObtained().calculateCommissionAmount(tickets[i]);
        }
        System.out.printf("Commission obtained per each person:
Rs.%.2f",commission);
    }
}
```

24. Water Distributor

```
public class Container {

    private String distributorName;
    private int volume;
    private int count;

    public String getDistributorName() {
        return distributorName;
    }
    public void setDistributorName(String distributorName) {
            this.distributorName = distributorName;
    }
    public int getVolume() {
            return volume;
    }
    public void setVolume(int volume) {
            this.volume = volume;
    }
    public int getCount() {
            return count;
    }
    public void setCount(int count) {
            this.count = count;
    }
    public Container(String distributorName, int volume, int count) {
            super();
            this.distributorName = distributorName;
            this.volume = volume;
            this.count = count;
    }

}

//============================================//


public interface DiscountInfo {
        public double calculatePayableAmount(Container containerObj);
}

//============================================//

import java.util.Scanner;
public class UserInterface {

    static boolean validate(int count, int vol){
        if(count>=100 && vol ==10 || count>=100 && vol ==25)
        return true;
```

```
        else if (vol==10 || vol==25)
        return true;
    return false;
    }

        public static DiscountInfo generateBillAmount() {
            DiscountInfo discountInfo = (containerObj->{
                double priceTen=20;
                double priceFive=50;
                double amt=0;
                if(containerObj.getCount()>=100){
                    if(containerObj.getVolume()==10){
                    amt=(containerObj.getCount()*priceTen);
                    amt=amt-(amt*.1);
                }
                else if (containerObj.getVolume()==25){
                    amt=(containerObj.getCount()*priceFive);
                    amt=amt-(amt*.15);
                }
               }
               else if (containerObj.getVolume()==10){
                   return containerObj.getCount()*priceTen;
               }
               else if(containerObj.getVolume()==25){
                   return containerObj.getCount()*priceFive;
            }
                return amt;
        });
        return discountInfo;
        }

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

                System.out.println("Enter the name of the distributor");
                String distributorName=sc.nextLine();
                System.out.println("Enter the volume of the container(in litre)");
                int litre=Integer.parseInt(sc.nextLine());
                System.out.println("Enter the no of containers");
                int count=Integer.parseInt(sc.nextLine());

                Container container =new Container(distributorName, litre, count);
                if(validate(count,litre)){
                    double
amount=generateBillAmount().calculatePayableAmount(container);
                    System.out.println("Generated Bill Amount");
                    System.out.println("Distributor name:
"+container.getDistributorName());
                    System.out.printf("Amount to be paid: Rs.%.2f",amount);
                }
```

```
                else{
                    System.out.println("There is no Discount");
                }

        }
}
```

25. College Account

```
public interface TuitionFee{

        public int calculateTuitionFees(String courseType, int basicFee, int
noOfSemesters);
}

//==================================//

import java.util.Scanner;

public class UserInterface{

    public static TuitionFee generateFeeReceipt() {
        TuitionFee tuitionFee = (courseType,basicFee,noOfSemesters) ->{
            if(courseType.equalsIgnoreCase("SelfFinance")){
                return ((basicFee * noOfSemesters)+ 50000);
            }
            return basicFee*noOfSemesters;
        };
        return tuitionFee;
    }

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

            System.out.println("Enter registration number");
            int regNo=Integer.parseInt(sc.nextLine());
            System.out.println("Enter student name");
            String studName=sc.nextLine();
            System.out.println("Enter no of semesters");
            int semesters=Integer.parseInt(sc.nextLine());
            System.out.println("Enter basic fee");
            int basicFee = Integer.parseInt(sc.nextLine());
            System.out.println("Course type");
            String courseType=sc.nextLine();

            int tuitionFee =
generateFeeReceipt().calculateTuitionFees(courseType,basicFee,semesters);
            System.out.println("Fees Receipt");
            System.out.println("Registration number: "+regNo);
            System.out.println("Student name: "+studName);

            if(courseType.equalsIgnoreCase("regular"))
                System.out.println("Tuition fee for regular student: "+tuitionFee);
           else System.out.println("Tuition fee for selfFinance student:
"+tuitionFee);
    }
```

```
}
```

26. Vehicle Capacity Calculator

```
public class PetrolOverflowException extends Exception {
    public PetrolOverflowException(String s){
        super(s);
    }
}

//=============================================//

import java.util.Scanner;

public class UserInterface extends Validator {

        public static void main(String[] args) {
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter the petrol in the vehicle in liters");
                int availablePetrol = sc.nextInt();
                System.out.println("Enter the petrol to be filled in liters");
                int fillingPetrol = sc.nextInt();
                int petrol = availablePetrol + fillingPetrol;

                try {
                    validatePetrolUsage(petrol);
                } catch(PetrolOverflowException e) {
                    System.out.println(e.getMessage());
                }
        }
}

//=============================================//

public class Validator {
        public static boolean validatePetrolUsage(int petrol) throws
PetrolOverflowException {
                boolean flag = false;

                if (petrol <= 120) {
                        flag = true;
                        System.out.println("petrol Tank will be sufficient");
                } else {
                    throw new PetrolOverflowException("Petrol Tank Overflow");
                }

                return flag;

        }
}
```

27. Array Manipulation - Use try with multi catch

```
import java.util.*;
import java.io.*;

public class Main
{
        public String getDuplicateElement()
    {
        Scanner sc =new Scanner(System.in);
        String str="";
        try{
            System.out.println("Enter the size of an array");

            int n=sc.nextInt();
            int p[]=new int[n];
            System.out.println("Enter the array elements");
            for (int i=0; i<n; i++)
            p[i]=sc.nextInt();

            System.out.println("Enter the position of the element to be
replicated");
            int index=sc.nextInt();
            for (int i=0; i<n; i++)
            str = str+" "+p[i];
            return "The array elements are"+str+" "+p[index];
        }
        catch(ArrayIndexOutOfBoundsException e){
            return("Array index is out of range");
        }
        catch(InputMismatchException e){
            return("Input was not in the correct format");
        }
        catch(NegativeArraySizeException e){
            return("Array size should be positive");
        }
       // return null;
    }

        public static void main(String[] args){
        System.out.println(new Main().getDuplicateElement());
    }
}
```

28. Telecom Regulatory Authority

```
public class MaximumDataUsageException extends Exception{
    public MaximumDataUsageException(String s){
        super(s);
    }
}

//=======================================//

import java.util.Scanner;

public class UserInterface extends Validator{
        public static void main(String[] args) {
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter the data usage in Mb");
                int data = sc.nextInt();
                try {
                    validateDataUsage(data);
                } catch(MaximumDataUsageException e) {
                    System.out.println(e.getMessage());
                }
        }
}

//=======================================//

public class Validator {
        public static boolean validateDataUsage(int data) throws
MaximumDataUsageException {
                boolean flag = false;
                if (data <= 1024) {
                        flag = true;
                        System.out.println("There is sufficient data for usage");
                } else {
                    throw new MaximumDataUsageException("You need to pay extra
charges");

                }
                return flag;
        }

}
```

29. String Extraction

```
import java.util.Scanner;

public class Main {

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

       System.out.println("Enter the String");
       String s=sc.nextLine();
       System.out.println("Enter first index");
       int start=sc.nextInt();
       System.out.println("Enter second index");
       int end =sc.nextInt();
       Main main=new Main();
       System.out.println(main.extractString(s,start,end));
    }

        public String extractString(String s,int start,int end)
    {
        int length=s.length();
        try{
            if(start<0 || end>=length){
                throw new StringIndexOutOfBoundsException();
            }
            else{
                return s.substring(start,end)+".Thanks for using the application.";
            }
        }
        catch(Exception e){
            return "Extraction of String using the given index is not
possible.Thanks for using the application.";
        }
       // return null;
    }

}
```

30. Campus Radio Frequency

```
public class StationNotAvailableException extends Exception{
        public StationNotAvailableException(String s){
            super(s);
        }
}

//=======================================//

import java.util.Scanner;

public class UserInterface extends Validator{

        public static void main(String[] args) {
                Scanner sc = new Scanner(System.in);
                System.out.println("Scan the radio station");
                float freq = sc.nextFloat();

                try {
                    validateStation(freq);
                } catch(StationNotAvailableException e) {
                    System.out.println(e.getMessage());
                }

        }

}

//=========================================//

public class Validator {

        public static boolean validateStation(float freq) throws
StationNotAvailableException {
                boolean flag = false;

                if (freq == 91.2f || freq == 93.5f || freq == 98.9f || freq ==
109.4f){
                    flag=true;
                    System.out.println("Radio Station on!");
                }

                else{
                        throw new StationNotAvailableException("Radio Station not
available");
                }
                        return flag;

        }
```

```
}
```

31. Stock List

```
import java.util.*;
import java.util.Scanner;

public class UserInterface{

    public static void main(String[] args){

        Scanner sc=new Scanner(System.in);

        List<String> company=new ArrayList<String>();

        System.out.println("Enter number of stocks to add");

        int num=Integer.parseInt(sc.nextLine());

        for (int i=0; i<num; i++){
            company.add(sc.nextLine());

        }

        System.out.println(company);
    }
}
```

32. Babitha’s App

```
import java.util.*;

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

                Scanner sc=new Scanner(System.in);

                System.out.println("Enter the paragraph typed");
                String paragraph=sc.nextLine();
                String[] text=paragraph.toLowerCase().split("[;:.?!@#$%, ]+");

                Map<String,Integer> words=new HashMap<String,Integer>();
                LinkedHashMap<String,Integer> wordSort= new LinkedHashMap<>();

                for(String str:text){
                    if(!words.containsKey(str)){
                        words.put(str,1);
                    }
                    else{
                        int count = words.get(str);
                        words.put(str, count+1);
                    }
                }
                int total=0,num=0;


words.entrySet().stream().sorted(Map.Entry.comparingByKey()).forEachOrdered(x->
wordSort.put(x.getKey(),x.getValue()));

                for(Map.Entry<String,Integer> entry:wordSort.entrySet()){
                    num=entry.getValue();
                    total+=num;
                }

                System.out.println("Total number of words "+total);
                System.out.println("Words with the count");

                wordSort.entrySet().forEach(entry -> {
                    System.out.println(entry.getKey()+" - "+entry.getValue());
                    });
        }
}
```

33. Plip Event

```
import java.util.Scanner;

public class Main {
        public static void main(String args[]){
                Scanner sc=new Scanner(System.in);

                System.out.println("Enter the number of students");
                int n=sc.nextInt();

                StudentUtility stu=new StudentUtility();
                for(int i=0; i<n;i++){
                    sc.nextLine();
                    System.out.println("Enter the student name");
                    String name=sc.next();
                    System.out.println("Enter the score");
                    double score = sc.nextDouble();
                    stu.addStudentDetails(name,score);
                }
                int count=stu.filterStudentDetails();
                if(count>0){
                    for(int j=0;j<count;j++){
                        System.out.println("Count is "+count);
                    }
                }
                else{
                    System.out.println("No students found");
                }
        }
}

//=============================================//

import java.util.*;

public class StudentUtility {
        private Map<String,Double> studentMap = new HashMap<String,Double>();

        public Map<String, Double> getStudentMap() {
                return studentMap;
        }

        public void setStudentMap(Map<String, Double> studentMap) {
                this.studentMap = studentMap;
        }

        public void addStudentDetails(String studentName,double score){
                studentMap.put(studentName,score);
        }
```

```
        public int filterStudentDetails(){

                int Count=0;
                for(Map.Entry<String, Double> i:studentMap.entrySet())
                if(i.getValue()>90)
                Count++;
                return Count;
        }
}
```

34. Top Tier Motors

```
import java.util.Scanner;

public class Main {
        public static void main(String args[]){
                Scanner sc=new Scanner(System.in);
                //Fill the code here
                //From here to
                VehicleUtility vehicleUtility=new VehicleUtility();
                System.out.println("Enter the number of vehicles");
                int nbr=Integer.parseInt(sc.nextLine());
                for (int i=1; i<=nbr; i++){
                    System.out.println("Enter the vehicle name and price of Vehicle
"+i);
                    String vehicle = sc.nextLine();
                    double price = Double.parseDouble(sc.nextLine());
                    vehicleUtility.addVehiclePriceDetails(vehicle,price);
                }

                String decision=null;
                System.out.println("Enter the vehicle name to be searched");
                String vehicleName=sc.nextLine();
                do{
                    double discount =
vehicleUtility.calculateCostAfterDiscount(vehicleName);
                    if(discount>0){
                        System.out.println("Price after discount for
"+vehicleName+" is "+discount);
                    }
                    else if (discount<=0){
                        System.out.println(vehicleName+" is not available
currently");
                    }
                    System.out.println("Do you want to continue (Y/N)");
                    decision=sc.next();
                   if(decision.equalsIgnoreCase("Y")){
                        System.out.println("Enter the vehicle name to be
searched");
                        sc.nextLine();
                        vehicleName=sc.nextLine();
                        continue;
                    }
                    else{
                        break;
                    }
                }while(decision!="N");
                System.out.println("Thank you for using the Application");
        }
}
```

```
//=========================================//

import java.util.HashMap;
import java.util.Map;

public class VehicleUtility {
        private Map<String, Double> vehicleMap = new HashMap<String, Double>();

        public Map<String, Double> getVehicleMap() {
                return vehicleMap;
        }

        public void setVehicleMap(Map<String, Double> vehicleMap) {
                this.vehicleMap = vehicleMap;
        }

        // This method should add the vehicleName as key and the price of the
        // vehicle as value into a Map
        public void addVehiclePriceDetails(String vehicleName, double price) {
                // fill the code
                vehicleMap.put(vehicleName, price);
        }

        // This method should calculate the discount and return the selling price
        // after the discount for the vehicle name passed as an argument.
        public double calculateCostAfterDiscount(String vehicleName) {
        //from here
                // fill the code
                try{
                    if(vehicleName.contains("TVS")){
                        return vehicleMap.get(vehicleName)*0.90;
                    }
                    else if (vehicleName.contains("Honda")){
                        return vehicleMap.get(vehicleName)*0.95;
                    }
                    else if (vehicleName.contains("Yamaha")){
                        return vehicleMap.get(vehicleName)*0.93;
                    }
                    else{
                        return 0;
                    }
                }
                catch(NullPointerException e){
                    return -1;
                }
        }
}
```