Lab 4

Q1. Write a class called Phone that contains instance data that represents the make, model, and year of the phone. Define the Phone constructor to initialize these values. Include getter and setter methods for all instance data, and a toString method that returns a one-line description of the phone. Add a method called is Obsolete that returns a boolean indicating if the phone is obsolete (if it is more than 10 years old). Create a driver class called PhoneCheck, whose main method instantiates and updates several Phone objects.

**Solution :**

Phone class

import java.util.Scanner;

class Phone {

    private String make;

    private String model;

    private int year;

    public Phone(String make, String model, int year)

    {

        this.make = make;

        this.model = model;

        this.year = year;

    }

    public String getMake()

    {

        return make;

    }

    public String getModel()

    {

        return model;

    }

    public int getYear()

    {

        return year;

    }

    public void setMake(String make)

    {

        this.make=make;

    }

    public void setModel(String model)

    {

        this.model=model;

    }

    public void setYear(int year)

    {

        this.year= year;

    }

    public String toString()

    {

        return " make = "+make+" model = "+model +" year = " +year;

    }

    public boolean isObselete(int curr\_year)

    {

        return (curr\_year-this.year)>10;

    }

Driver Class :

import java.util.Scanner;

public class MainDriver

{

    public static void main(String[] args) {

        Phone obj2 = new Phone("Apple", "Iphone 20", 2027);

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

        Scanner inp= new Scanner(System.in);

        System.out.println("Enter the make, model and year");

        String make\_inp= inp.nextLine();

        String model\_inp= inp.nextLine();

        int year\_inp= inp.nextInt();

        obj2.setMake(make\_inp);

        obj2.setModel(model\_inp);

        obj2.setYear(year\_inp);;

        System.out.println("Make = " +obj2.getMake());

        System.out.println("Model = " +obj2.getModel());

        System.out.println("Year = " +obj2.getYear());

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

        System.out.println("Enter the current year");

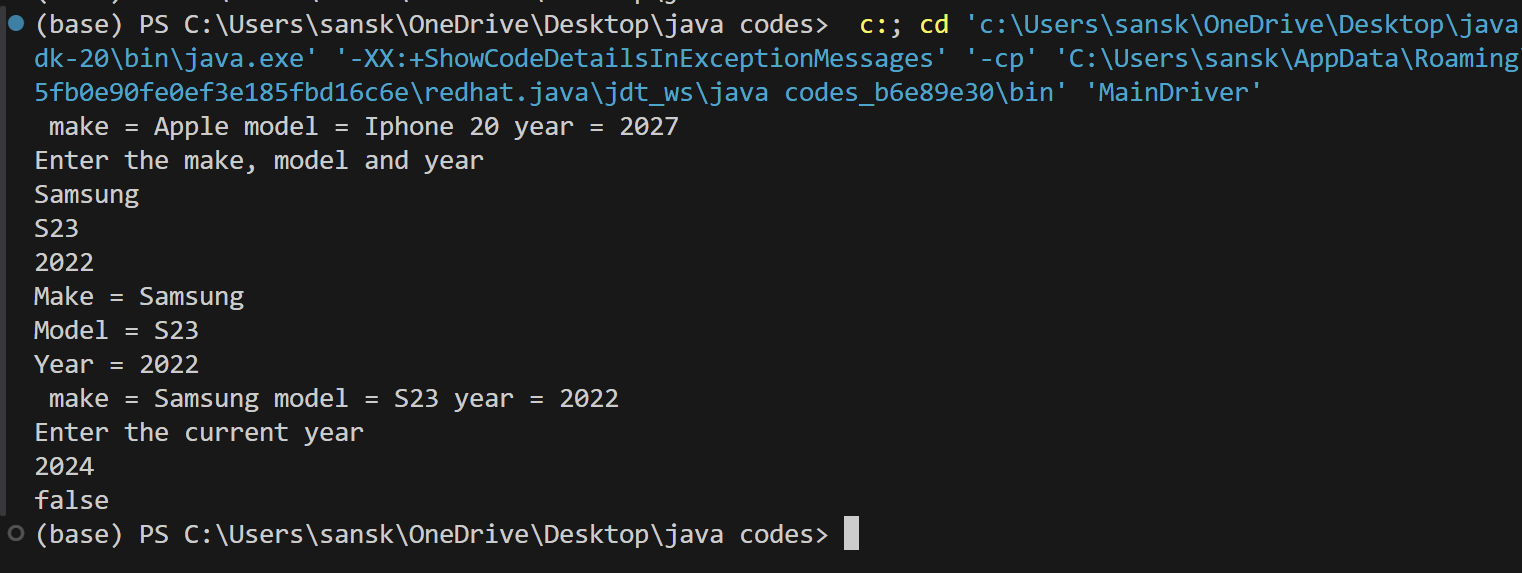
        int curr\_year\_inp= inp.nextInt();

        System.err.println(obj2.isObselete(curr\_year\_inp));

    }

}

**Output :**

****

2. A small airline has just purchased a computer for its new automated reservation system. You have been asked to develop the new system. Write an application to assign seats on each flight of the airline's (capacity: 10 seats) Your application should display the following alternatives: Please type1 for FirstClass and Please type2 for Economy. If the user types 1, your application should assign a seat in the first class section(1-5). If user type 2, your application should assign a seat in the economy section(6-10). Your application should then display a boarding pass indicating the person's seat number and its class. Use 1D array of type Boolean type to represent the seating chart of the plane. Initialize it to false indicating all the seats are empty. The change its value according to assignment. Your application should never assign a seat that has been already assigned.

import java.util.Scanner;

public class airline {

    public static void main(String[] args) {

        System.out.println("Enter class choice : \n 1. First class \n 2. Economy class");

        Scanner inp= new Scanner(System.in);

        int choice = inp.nextInt();

        boolean arr[ ] = new boolean[10];

        switch(choice){

            case 1:

                for(int i=0;i<5;i++)

                {

                    if(arr[i]==false)

                    {

                        System.out.println("Seat alloted in First class is = " +(i+1));

                        arr[i]=true;

                        break;

                    }

                    else

                    {

                        System.out.println("No seat is vacant");

                    }

                }

                break;

            case 2:

                for(int i=5;i<10;i++)

                {

                    if(arr[i]==false)

                    {

                        System.out.println("Seat alloted in Economy class is = " +(i+1));

                        arr[i]=true;

                        break;

                    }

                    else

                    {

                        System.out.println("No seat is vacant");

                    }

                }

                break;

            default:

                System.out.println("Enter the correct choice");

            // inp.close();

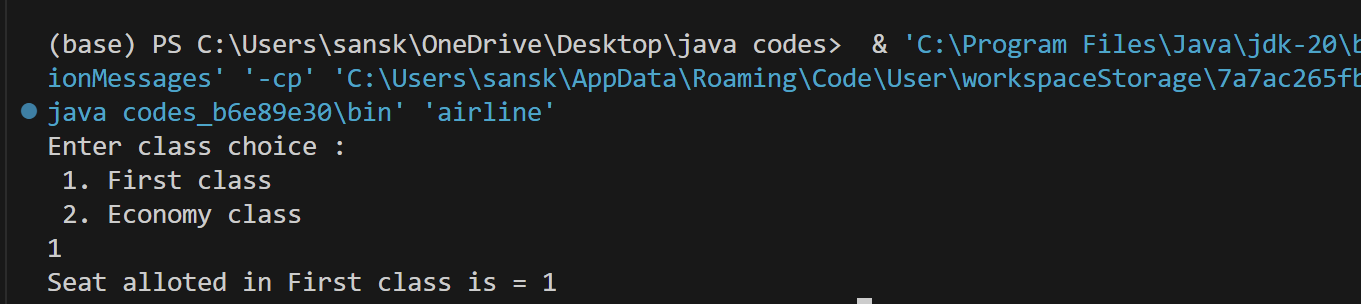
        }

        inp.close();

    }

}

**Output :**

****

Q3. Make a vector class, in which take a input of a n size and then insert two additional elements. Print the final Output.

import java.util.Vector;

public class VectorExample{

    public static void main(String[] args) {

        Vector<Integer> v = new Vector<>(3, 2);

        v.addElement(1);

        v.addElement(2);

        v.addElement(3);

        v.insertElementAt(0, 1);

        v.removeElementAt(2);

        for (int i : v) {

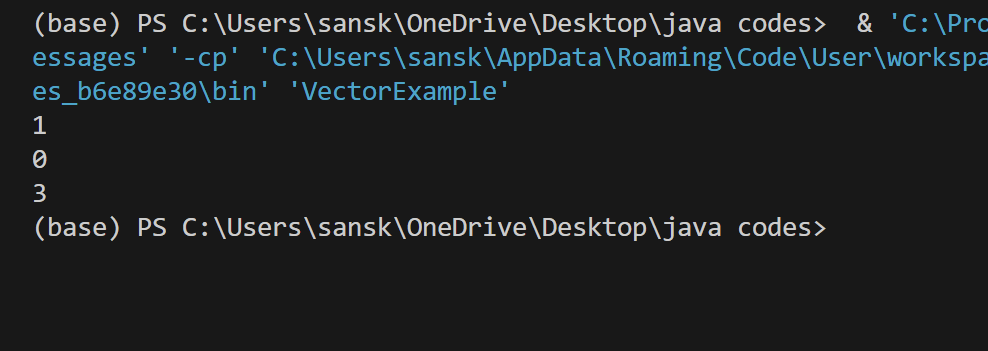
            System.out.println(i);

        }

    }

}

**Output :**

****