**Slip26**

[October 03, 2024](https://nilambariblogfortybsc-cs.blogspot.com/2024/10/slip26.html)

 Q1) Define a Item class (item\_number, item\_name, item\_price). Define a default and parameterized constructor. Keep a count of objects created. Create objects using parameterized constructor and display the object count after each object is created.(Use static member and method). Also display the contents of each object. [10 marks]

class Item {

    int ino;

    String iname;

    double iprice;

    static int count = 0;

    public Item()

    {

        count++;

    }

    public Item(int ino, String iname, double iprice)

    {

        this.ino = ino;

        this.iname = iname;

        this.iprice = iprice;

      count++;

    }

    public static int getObjectCount()

    {

        return count;

    }

   public void displayItemDetails()

   {

        System.out.println("Item Details:");

        System.out.println("Item Number: " + ino);

        System.out.println("Item Name: " + iname);

        System.out.println("Item Price: $" + iprice);

        System.out.println();

    }

}

class Slip26\_1

{

    public static void main(String[] args)

    {

        Item item1 = new Item(101, "Laptop", 850.00);

        item1.displayItemDetails();

        System.out.println("Objects created: " + Item.getObjectCount());

        System.out.println();

        Item item2 = new Item(102, "Smartphone", 650.00);

        item2.displayItemDetails();

        System.out.println("Objects created: " + Item.getObjectCount());

        System.out.println();

        Item item3 = new Item(103, "Tablet", 300.00);

        item3.displayItemDetails();

        System.out.println("Objects created: " + Item.getObjectCount());

    }

}

Q2) Define a class ‘Donor’ to store the below mentioned details of a blood donor. name, age, address, contactnumber, bloodgroup, date of last donation. Create ‘n’ objects of this class for all the regular donors at Pune. Write these objects to a file. Read these objects from the file and display only those donors’ details whose blood group is ‘A+ve’ and had not donated for the recent six months. [20 marks]

import java.io.\*;

import java.util.Calendar;

import java.util.Date;

class Donor implements Serializable {

    String name,addr,conNo,blg;

    int age;

    Date lddate;

    public Donor(String name,int age,String addr,String conNo,String blg, Date lddate)

    {

        this.name = name;

        this.age = age;

        this.addr = addr;

        this.conNo = conNo;

        this.blg = blg;

        this.lddate = lddate;

    }

    public String getBloodGroup() {

        return blg;

    }

    public Date getDateOfLastDonation() {

        return lddate;

    }

    public void display() {

        System.out.println("Name: " + name);

        System.out.println("Age: " + age);

        System.out.println("Address: " + addr);

        System.out.println("Contact Number: " + conNo);

        System.out.println("Blood Group: " + blg);

        System.out.println("Date of Last Donation: " + lddate);

        System.out.println();

    }

    public boolean isEligibleForDonation() {

        Calendar currentDate = Calendar.getInstance();

        Calendar donationDate = Calendar.getInstance();

        donationDate.setTime(lddate);

        donationDate.add(Calendar.MONTH, 6); // Add 6 months to the last donation date

        return currentDate.after(donationDate);

    }

    public static void writeDonorsToFile(Donor[] donors, String fileName) throws IOException {

        try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(fileName))) {

            // Writing each donor object individually

            for (Donor donor : donors) {

                oos.writeObject(donor);

            }

        }

    }

    public static Donor readDonorFromFile(ObjectInputStream ois) throws IOException, ClassNotFoundException {

        return (Donor) ois.readObject();

    }

    public static Date createDate(int year, int month, int day)

    {

        Calendar cal = Calendar.getInstance();

        cal.set(Calendar.YEAR, year);

        cal.set(Calendar.MONTH, month - 1); // Months are 0-based in Calendar

        cal.set(Calendar.DAY\_OF\_MONTH, day);

        return cal.getTime();

    }

}

class Slip26\_2

{

public static void main(String[] args) {

        Donor[] donors = new Donor[5];

        donors[0] = new Donor("John Doe", 35, "123 Main St", "9876543210", "A+ve", Donor.createDate(2022, 1, 15));

        donors[1] = new Donor("Jane Smith", 28, "456 Elm St", "9876543211", "B+ve", Donor.createDate(2023, 3, 10));

        donors[2] = new Donor("Mike Johnson", 45, "789 Oak St", "9876543212", "A+ve", Donor.createDate(2021, 12, 1));

        donors[3] = new Donor("Emily Davis", 40, "101 Pine St", "9876543213", "O+ve", Donor.createDate(2023, 2, 20));

        donors[4] = new Donor("Chris Brown", 38, "202 Maple St", "9876543214", "A+ve", Donor.createDate(2023, 5, 18));

        String fileName = "donors.dat";

        try {

            Donor.writeDonorsToFile(donors, fileName);

            try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(fileName))) {

                System.out.println("Donors with blood group A+ve who have not donated in the last 6 months:");

                for (int i = 0; i < donors.length; i++) {

                    Donor donor = Donor.readDonorFromFile(ois);

                    if (donor.getBloodGroup().equals("A+ve") && donor.isEligibleForDonation()) {

                        donor.display();

                    }

                }

            }

        } catch (IOException | ClassNotFoundException e) {

            e.printStackTrace();

        }

    }

}