Java Programming
(CSE-1007)

ASSIGNMENT - 4

Name: Mathew Jerry Meleth
Reg No: 17BIT0050
Slot: L19+L20

SCENARIO – I

Write a program to demonstrate the knowledge of students in File handling. Eg., Define a class ‘Donor’ to store the below mentioned details of a blood donor. Name, age, Address, Contact number, blood group, date of last donation Create ‘n’ objects of this class for all the regular donors at Vellore. 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

BRIEF ABOUT YOUR APPROACH:

In this program we define a class called 'Donor' and we store the details of the patients in Name, Age, Address, Contact Number, Blood Group and the date of last donation as mentioned in the question. After this we create the Main Class 'Donorblood' and create n objects and we write those into a file.
Then we read the objects from the file and display the details according to the given details

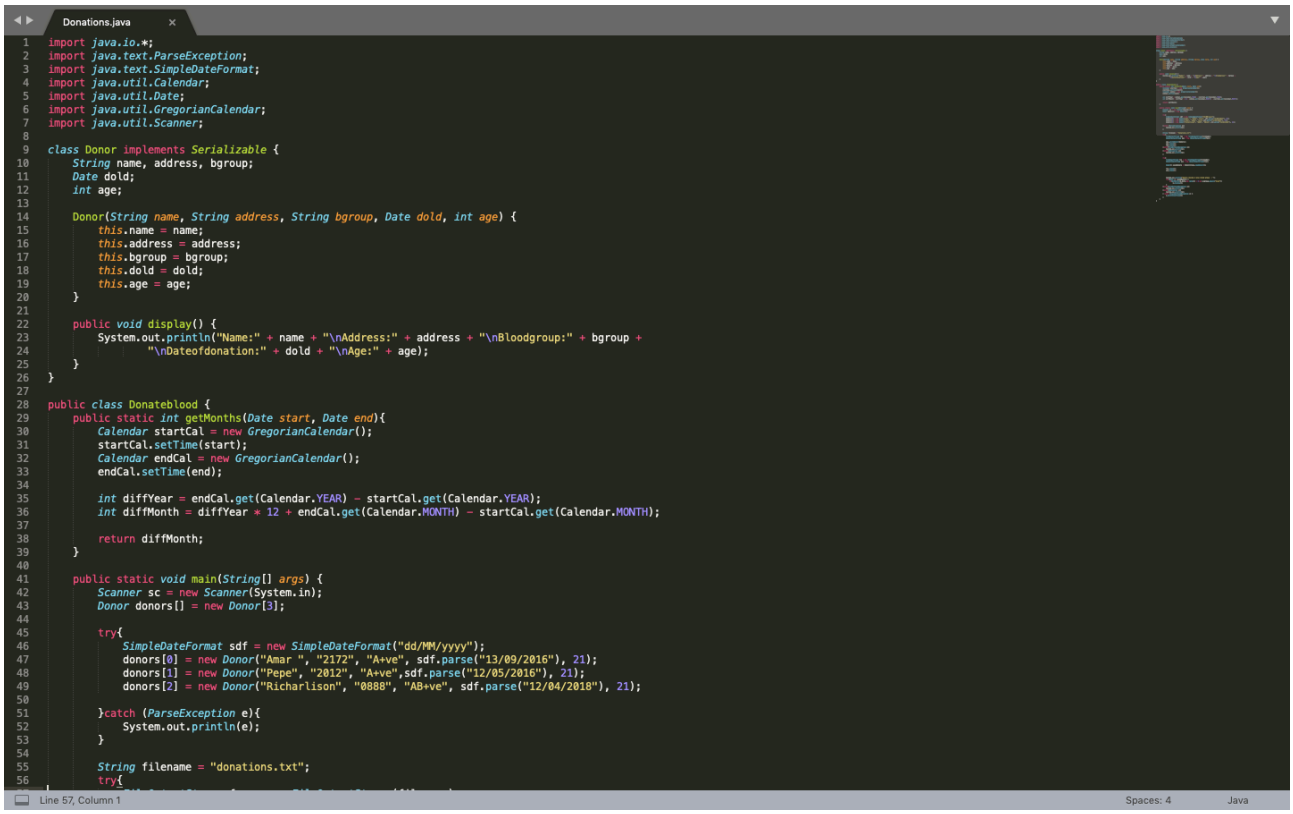
SOURCE CODE:

```
import java.io.*;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.GregorianCalendar;
import java.util.Scanner;

class Donor implements Serializable {
   String name, address, bgroup;
   Date dold;
   int age;

   Donor(String name, String address, String bgroup, Date dold, int age) {
      this.name = name;
      this.address = address;
      this.bgroup = bgroup;
      this.dold = dold;
      this.age = age;
   }

   public void display() {
      System.out.println("Name:" + name + "\nAddress:" + address + "\nBloodgroup:" + bgroup +
            "\nDateofdonation:" + dold + "\nAge:" + age);
   }
}

public class Donateblood {
   public static int getMonths(Date start, Date end){
      Calendar startCal = new GregorianCalendar();
      startCal.setTime(start);
      Calendar endCal = new GregorianCalendar();
      endCal.setTime(end);

      int diffYear = endCal.get(Calendar.YEAR) - startCal.get(Calendar.YEAR);
```

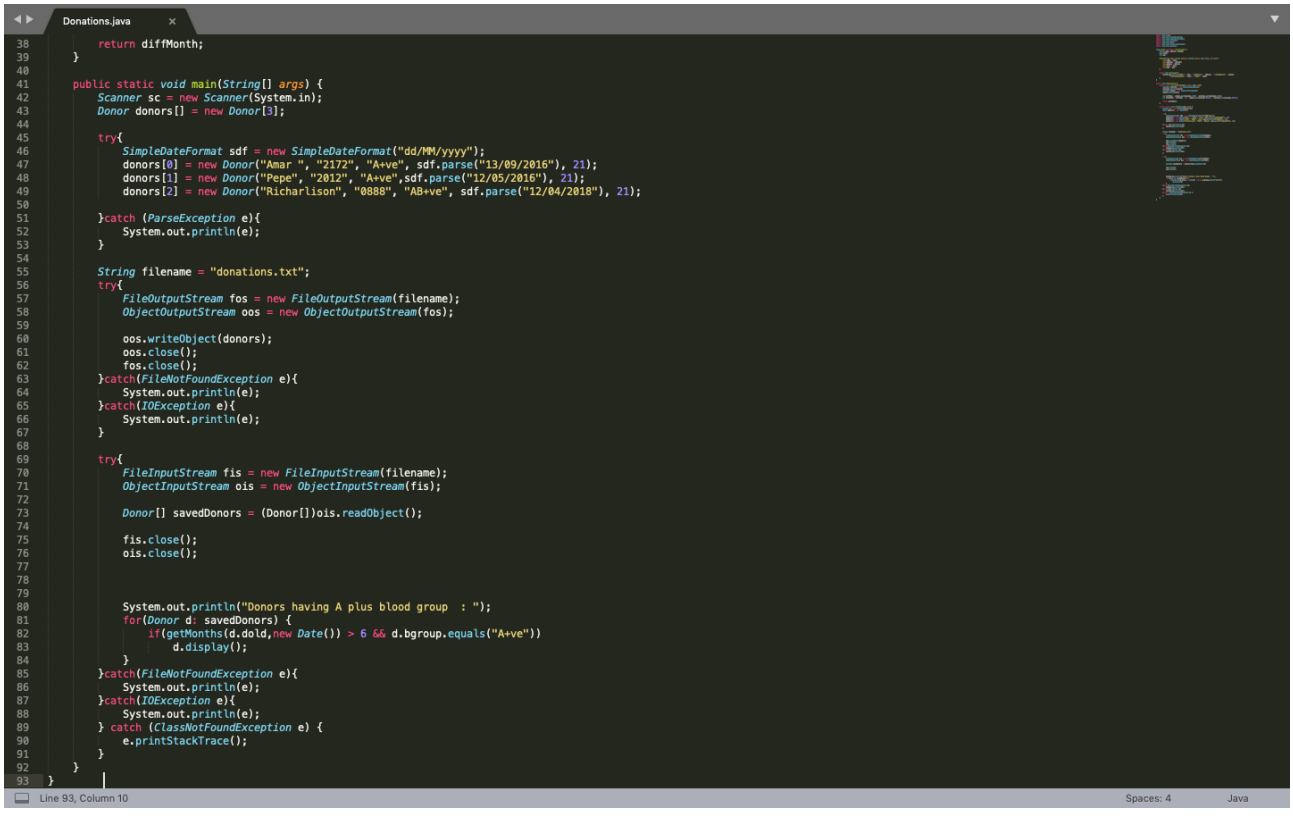
```
        int diffMonth = diffYear * 12 + endCal.get(Calendar.MONTH) -
startCal.get(Calendar.MONTH);

        return diffMonth;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Donor donors[] = new Donor[3];

        try{
            SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
            donors[0] = new Donor("Amar ", "2172", "A+ve", sdf.parse("13/09/2016"), 21);
            donors[1] = new Donor("Pepe", "2012", "A+ve",sdf.parse("12/05/2016"), 21);
            donors[2] = new Donor("Richarlison", "0888", "AB+ve", sdf.parse("12/04/2018"), 21);

        }catch (ParseException e){
            System.out.println(e);
        }

        String filename = "donations.txt";
        try{
            FileOutputStream fos = new FileOutputStream(filename);
            ObjectOutputStream oos = new ObjectOutputStream(fos);

            oos.writeObject(donors);
            oos.close();
            fos.close();
        }catch(FileNotFoundException e){
            System.out.println(e);
        }catch(IOException e){
            System.out.println(e);
        }

        try{
            FileInputStream fis = new FileInputStream(filename);
            ObjectInputStream ois = new ObjectInputStream(fis);

            Donor[] savedDonors = (Donor[])ois.readObject();

            fis.close();
            ois.close();



            System.out.println("Donors having A plus blood group  : ");
            for(Donor d: savedDonors) {
                if(getMonths(d.dold,new Date()) > 6 && d.bgroup.equals("A+ve"))
                    d.display();
            }
        }catch(FileNotFoundException e){
            System.out.println(e);
```

```
        }catch(IOException e){
            System.out.println(e);
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
    }
}
```

EXECUTION:

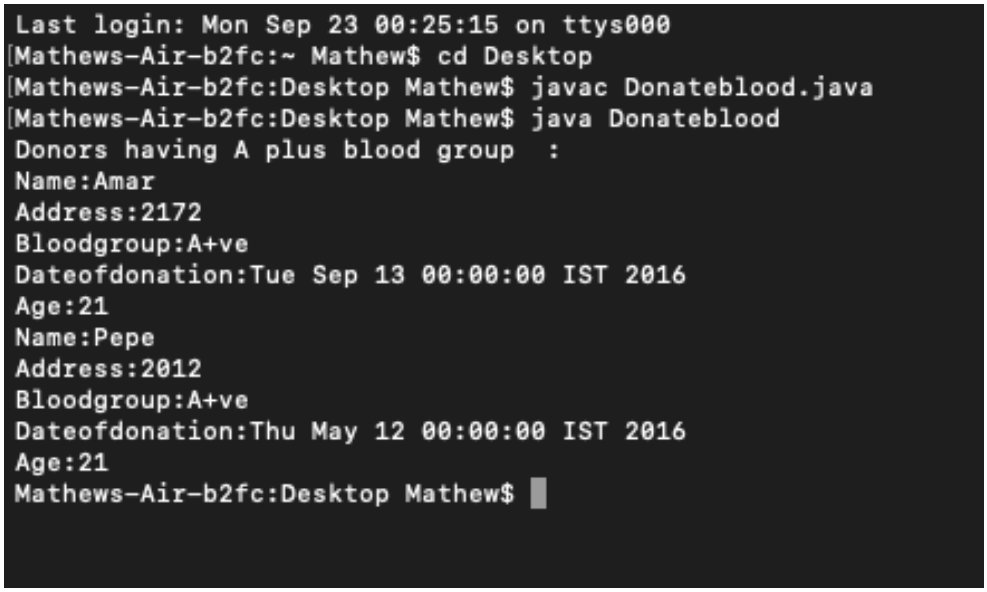


```
import java.io.*;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.GregorianCalendar;
import java.util.Scanner;

class Donor implements Serializable {
    String name, address, bgroup;
    Date dold;
    int age;

    Donor(String name, String address, String bgroup, Date dold, int age) {
        this.name = name;
        this.address = address;
        this.bgroup = bgroup;
        this.dold = dold;
        this.age = age;
    }

    public void display() {
        System.out.println("Name:" + name + "\nAddress:" + address + "\nBloodgroup:" + bgroup +
                "\nDateofdonation:" + dold + "\nAge:" + age);
    }
}

public class Donateblood {
    public static int getMonths(Date start, Date end){
        Calendar startCal = new GregorianCalendar();
        startCal.setTime(start);
        Calendar endCal = new GregorianCalendar();
        endCal.setTime(end);

        int diffYear = endCal.get(Calendar.YEAR) - startCal.get(Calendar.YEAR);
        int diffMonth = diffYear * 12 + endCal.get(Calendar.MONTH) - startCal.get(Calendar.MONTH);

        return diffMonth;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Donor donors[] = new Donor[3];

        try{
            SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
            donors[0] = new Donor("Amar ", "2172", "A+ve", sdf.parse("13/09/2016"), 21);
            donors[1] = new Donor("Pepe", "2012", "A+ve",sdf.parse("12/05/2016"), 21);
            donors[2] = new Donor("Richarlison", "0888", "AB+ve", sdf.parse("12/04/2018"), 21);

        }catch (ParseException e){
            System.out.println(e);
        }

        String filename = "donations.txt";
        try{
```

```
        return diffMonth;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Donor donors[] = new Donor[3];

        try{
            SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
            donors[0] = new Donor("Amar ", "2172", "A+ve", sdf.parse("13/09/2016"), 21);
            donors[1] = new Donor("Pepe", "2012", "A+ve",sdf.parse("12/05/2016"), 21);
            donors[2] = new Donor("Richarlison", "0888", "AB+ve", sdf.parse("12/04/2018"), 21);

        }catch (ParseException e){
            System.out.println(e);
        }

        String filename = "donations.txt";
        try{
            FileOutputStream fos = new FileOutputStream(filename);
            ObjectOutputStream oos = new ObjectOutputStream(fos);

            oos.writeObject(donors);
            oos.close();
            fos.close();
        }catch(FileNotFoundException e){
            System.out.println(e);
        }catch(IOException e){
            System.out.println(e);
        }

        try{
            FileInputStream fis = new FileInputStream(filename);
            ObjectInputStream ois = new ObjectInputStream(fis);

            Donor[] savedDonors = (Donor[])ois.readObject();

            fis.close();
            ois.close();



            System.out.println("Donors having A plus blood group  : ");
            for(Donor d: savedDonors) {
                if(getMonths(d.dold,new Date()) > 6 && d.bgroup.equals("A+ve"))
                    d.display();
            }
        }catch(FileNotFoundException e){
            System.out.println(e);
        }catch(IOException e){
            System.out.println(e);
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
    }
}
```

RESULT



SCENARIO – II

Write a program to demonstrate the knowledge of students in JDBC. Eg: Create a student table with fields roll number, name, percentage. Insert values in the table. Display all the details of the student table in a tabular format on the screen

BRIEF ABOUT YOUR APPROACH:

In this program, we create a table Roll Number, the name and the percentage and then insert values into the table using the JDBC Functions and then we display the output in the table format.

SOURCE CODE:

```
import java.sql.*;
public class Mathew
{
 public static void main(String a[]) throws Exception
 {
  Connection con;
  Statement r2;
  ResultSet r;
  Class.forName("com.mysql.jdbc.Driver");
  con=DriverManager.getConnection("jdbc:mysql://localhost/
student","root","");
  r2=con.createStatement();
  r2.executeUpdate("create table student(rollno int
primary key,name text,per int)");
  r2.executeUpdate("insert into student values(1,'Harry',82)");
  r2.executeUpdate("insert into student values(2,'Ron',78)");
  r2.executeUpdate("insert into student values(3,'Hermione',90)");
  r=r2.executeQuery("select * from student");
  while(r.next())
   System.out.println("\nRoll no="+r.getInt(1)+"\nname="
+r.getString(2)+"\npercentage="+r.getInt(3));
 }
}
```

EXECUTION:

Mathew.java

```
import java.sql.*;
public class Mathew
{
 public static void main(String a[]) throws Exception
 {
  Connection con;
  Statement r2;
  ResultSet r;
  Class.forName("com.mysql.jdbc.Driver");
  con=DriverManager.getConnection("jdbc:mysql://localhost/
student","root","");
  r2=con.createStatement();
  r2.executeUpdate("create table student(rollno int
primary key,name text,per int)");
  r2.executeUpdate("insert into student values(1,'Harry',82)");
  r2.executeUpdate("insert into student values(2,'Ron',78)");
  r2.executeUpdate("insert into student values(3,'Hermione',90)");
  r=r2.executeQuery("select * from student");
  while(r.next())
   System.out.println("\nRoll no="+r.getInt(1)+"\nname="
+r.getString(2)+"\npercentage="+r.getInt(3));
  }
}
```

Line 23, Column 2 Spaces: 2 Java

RESULT

```
Last login: Mon Sep 23 00:58:41 on ttys000
[Mathews-Air-b2fc:~ Mathew$ cd Desktop
[Mathews-Air-b2fc:Desktop Mathew$ javac Mathew.java
[Mathews-Air-b2fc:Desktop Mathew$ java Mathew.java
 |1          |Suraj      |90                    |
 |2          |Katrina    |72                    |
 |3          |Hrithik    |95                    |
Mathews-Air-b2fc:Desktop Mathew$
```

SCENARIO – III

3. Design and develop a java program that reads a filename from keyboard and display the number of characters, lines, and words in the file.

BRIEF ABOUT YOUR APPROACH:

In this program, we create a separate file called File.txt which we import it into the java program and take the text inside the File.txt as an input. Then we initiliaze counters and using different methods we count the number of characters, lines and words in the file and print it as the output.

SOURCE CODE:

```
import java.io.*;

public class WordCountInFile
{
   public static void main(String[] args) throws IOException
   {
      File file = new ("File.txt");
      FileInputStream fileStream = new FileInputStream(file);
      InputStreamReader input = new InputStreamReader(fileStream);
      BufferedReader reader = new BufferedReader(input);

      String line;


      int countWord = 0;
      int sentenceCount = 0;
      int characterCount = 0;
      int paragraphCount = 1;
      int whitespaceCount = 0;


      while((line = reader.readLine()) != null)
      {
         if(line.equals(""))
         {
            paragraphCount++;
         }
         if(!(line.equals("")))
```

```
        {

            characterCount += line.length();

            // \\s+ is the space delimiter in java
            String[] wordList = line.split("\\s+");

            countWord += wordList.length;
            whitespaceCount += countWord -1;

            // [!?.:]+ is the sentence delimiter in java
            String[] sentenceList = line.split("[!?.:]+");

            sentenceCount += sentenceList.length;
        }
    }

    System.out.println("Total word count = " + countWord);
    System.out.println("Total number of sentences = " + sentenceCount);
    System.out.println("Total number of characters = " + characterCount);
    System.out.println("Number of paragraphs = " + paragraphCount);
    System.out.println("Total number of whitespaces = " + whitespaceCount);
  }
}
```

EXECUTION:

WordCountInFile.java

```java
import java.io.*;

public class WordCountInFile
{
    public static void main(String[] args) throws IOException
    {
        File file = new ("File.txt");
        FileInputStream fileStream = new FileInputStream(file);
        InputStreamReader input = new InputStreamReader(fileStream);
        BufferedReader reader = new BufferedReader(input);

        String line;

        // Initializing counters
        int countWord = 0;
        int sentenceCount = 0;
        int characterCount = 0;
        int paragraphCount = 1;
        int whitespaceCount = 0;

        // Reading line by line from the
        // file until a null is returned
        while((line = reader.readLine()) != null)
        {
            if(line.equals(""))
            {
                paragraphCount++;
            }
            if(!(line.equals("")))
            {

                characterCount += line.length();

                // \\s+ is the space delimiter in java
                String[] wordList = line.split("\\s+");

                countWord += wordList.length;
                whitespaceCount += countWord -1;

                // [!?.:]+ is the sentence delimiter in java
                String[] sentenceList = line.split("[!?.:]+");

                sentenceCount += sentenceList.length;
            }
        }

        System.out.println("Total word count = " + countWord);
        System.out.println("Total number of sentences = " + sentenceCount);
        System.out.println("Total number of characters = " + characterCount);
        System.out.println("Number of paragraphs = " + paragraphCount);
        System.out.println("Total number of whitespaces = " + whitespaceCount);
    }
}
```

Line 1, Column 1; Detect Indentation: Setting indentation to 4 spaces Spaces: 4 Java

RESULT:

```
Last login: Mon Sep 23 00:26:24 on ttys000
[Mathews-Air-b2fc:~ Mathew$ cd Desktop
[Mathews-Air-b2fc:Desktop Mathew$ javac WordCountInFile.java
[Mathews-Air-b2fc:Desktop Mathew$ java WordCountFile
Error: Could not find or load main class WordCountFile
Caused by: java.lang.ClassNotFoundException: WordCountFile
[Mathews-Air-b2fc:Desktop Mathew$ javac WordCountInFile.java
[Mathews-Air-b2fc:Desktop Mathew$ java WordCountInFile
Total word count = 5
Total number of sentences = 10
Total number of characters = 70
Mathews-Air-b2fc:Desktop Mathew$
```