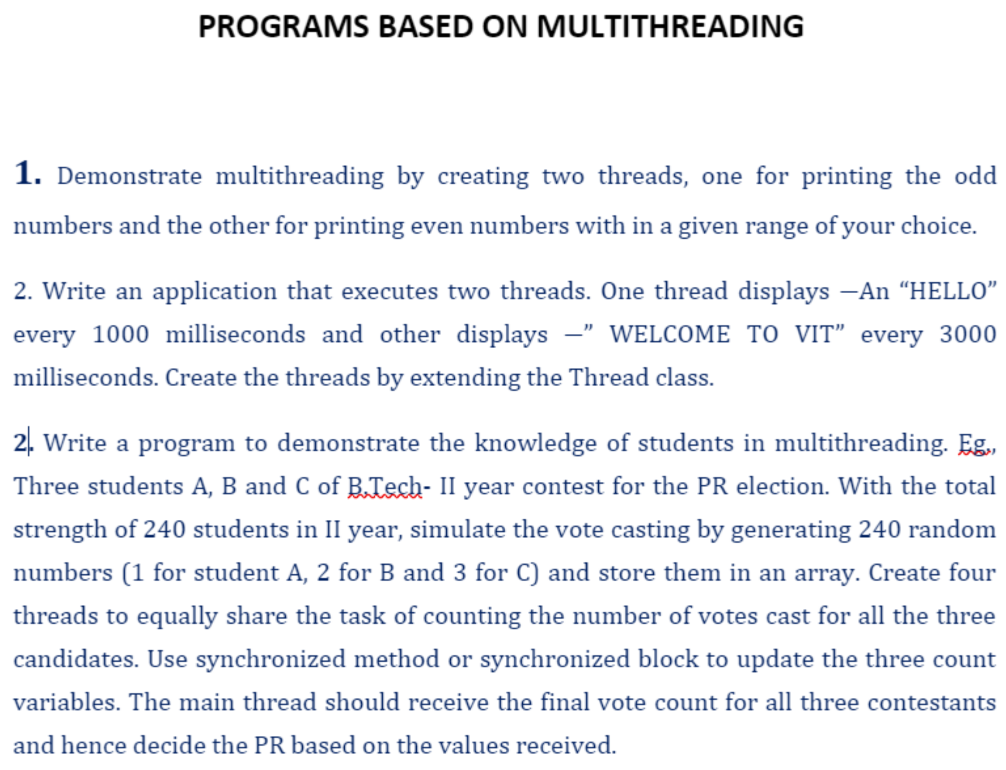
DIGITAL ASSIGNMENT 5

CSE1007 / JAVA LAB

ANISH SHRESTHA 11/17/21 20BCE2893

DIGITAL ASSIGNMENT -5

On Multi-threading



PROGRAMS BASED ON MULTITHREADING

1. Demonstrate multithreading by creating two threads, one for printing the odd numbers and the other for printing even numbers with in a given range of your choice.

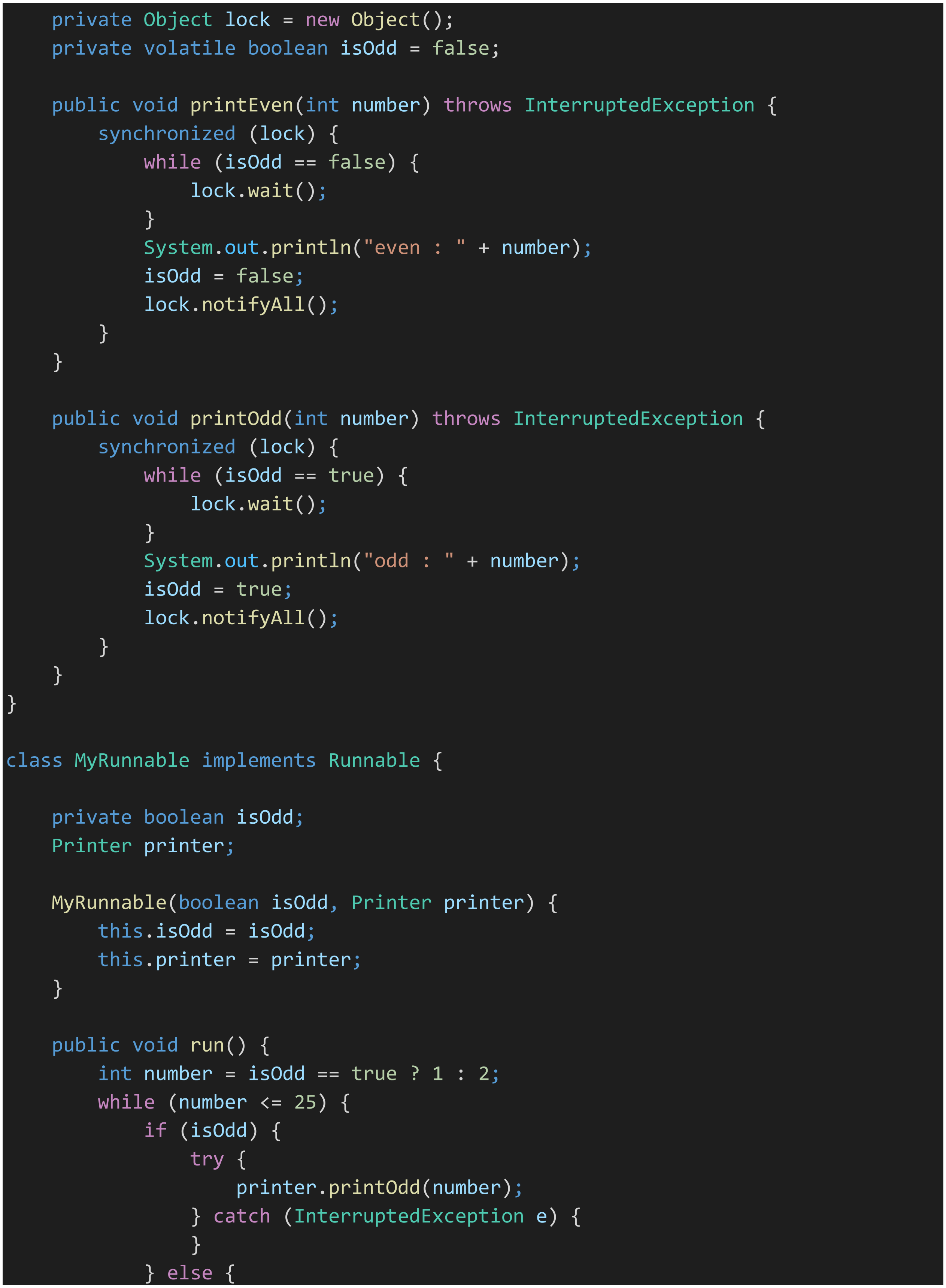
2. Write an application that executes two threads. One thread displays —An "HELLO" every 1000 milliseconds and other displays —" WELCOME TO VIT" every 3000 milliseconds. Create the threads by extending the Thread class.

2. Write a program to demonstrate the knowledge of students in multithreading. Eg., Three students A, B and C of B.Tech- II year contest for the PR election. With the total strength of 240 students in II year, simulate the vote casting by generating 240 random numbers (1 for student A, 2 for B and 3 for C) and store them in an array. Create four threads to equally share the task of counting the number of votes cast for all the three candidates. Use synchronized method or synchronized block to update the three count variables. The main thread should receive the final vote count for all three contestants and hence decide the PR based on the values received.

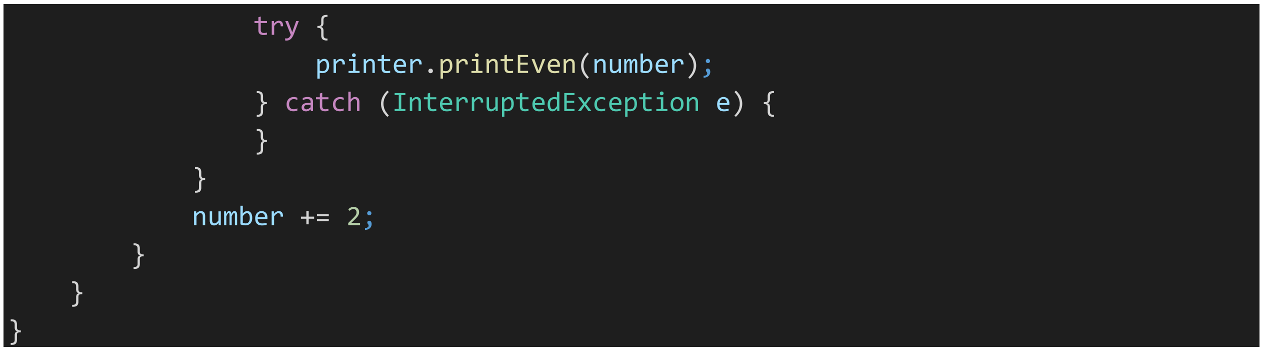
1.

Code:

```
public class OddEvenThreadType2 {
    public static void main(String[] args) {

        Printer printer = new Printer();

        MyRunnable r1 = new MyRunnable(true, printer);// isOdd = true
        Thread t1 = new Thread(r1);
        MyRunnable r2 = new MyRunnable(false, printer);// isOdd = false
        Thread t2 = new Thread(r2);
        t1.start();
        t2.start();
    }
}

class Printer {
```

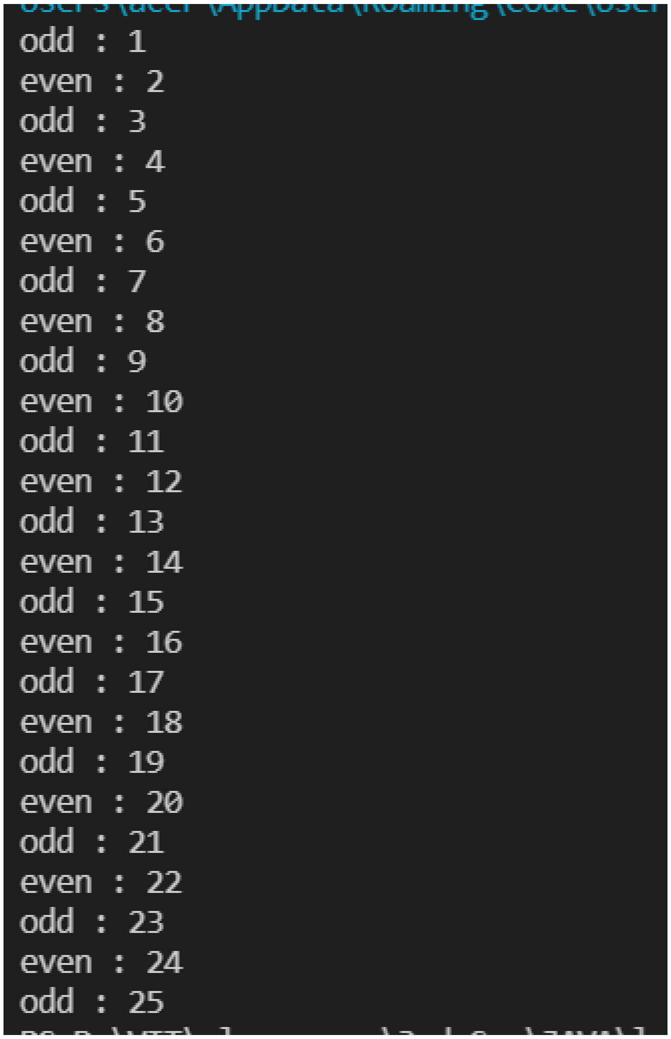


```
    private Object lock = new Object();
    private volatile boolean isOdd = false;

    public void printEven(int number) throws InterruptedException {
        synchronized (lock) {
            while (isOdd == false) {
                lock.wait();
            }
            System.out.println("even : " + number);
            isOdd = false;
            lock.notifyAll();
        }
    }

    public void printOdd(int number) throws InterruptedException {
        synchronized (lock) {
            while (isOdd == true) {
                lock.wait();
            }
            System.out.println("odd : " + number);
            isOdd = true;
            lock.notifyAll();
        }
    }
}

class MyRunnable implements Runnable {

    private boolean isOdd;
    Printer printer;

    MyRunnable(boolean isOdd, Printer printer) {
        this.isOdd = isOdd;
        this.printer = printer;
    }

    public void run() {
        int number = isOdd == true ? 1 : 2;
        while (number <= 25) {
            if (isOdd) {
                try {
                    printer.printOdd(number);
                } catch (InterruptedException e) {
                }
            } else {
```



```
                try {
                    printer.printEven(number);
                } catch (InterruptedException e) {
                }
            }
            number += 2;
        }
    }
}
```

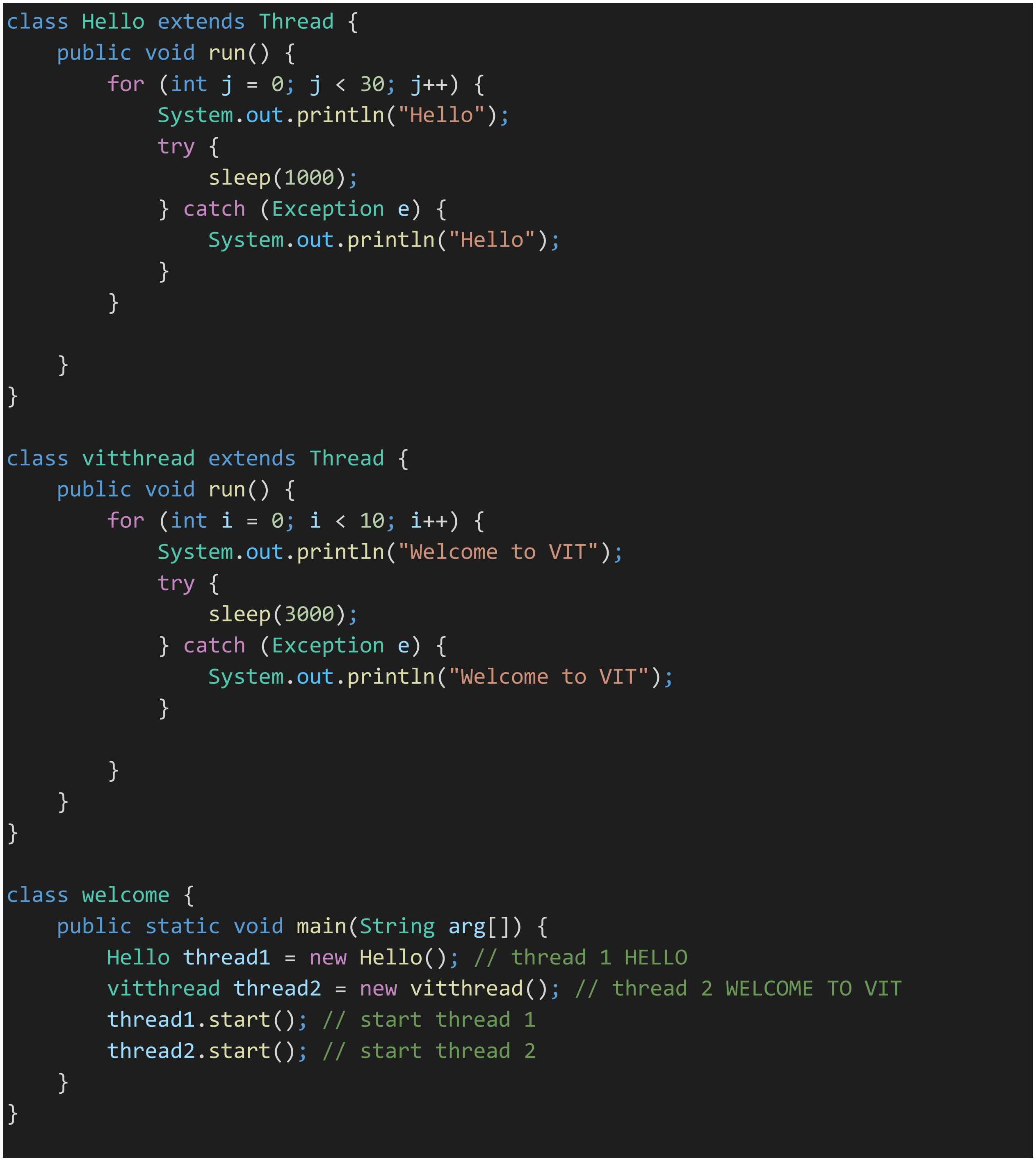
Output:



```
odd : 1
even : 2
odd : 3
even : 4
odd : 5
even : 6
odd : 7
even : 8
odd : 9
even : 10
odd : 11
even : 12
odd : 13
even : 14
odd : 15
even : 16
odd : 17
even : 18
odd : 19
even : 20
odd : 21
even : 22
odd : 23
even : 24
odd : 25
```

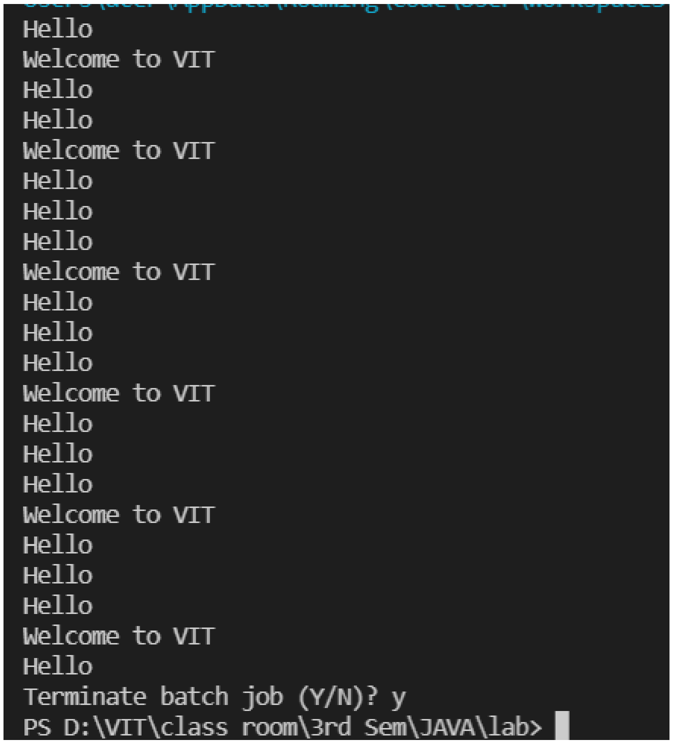
2.

Code:



```java
class Hello extends Thread {
    public void run() {
        for (int j = 0; j < 30; j++) {
            System.out.println("Hello");
            try {
                sleep(1000);
            } catch (Exception e) {
                System.out.println("Hello");
            }
        }

    }
}

class vitthread extends Thread {
    public void run() {
        for (int i = 0; i < 10; i++) {
            System.out.println("Welcome to VIT");
            try {
                sleep(3000);
            } catch (Exception e) {
                System.out.println("Welcome to VIT");
            }

        }
    }
}

class welcome {
    public static void main(String arg[]) {
        Hello thread1 = new Hello(); // thread 1 HELLO
        vitthread thread2 = new vitthread(); // thread 2 WELCOME TO VIT
        thread1.start(); // start thread 1
        thread2.start(); // start thread 2
    }
}
```

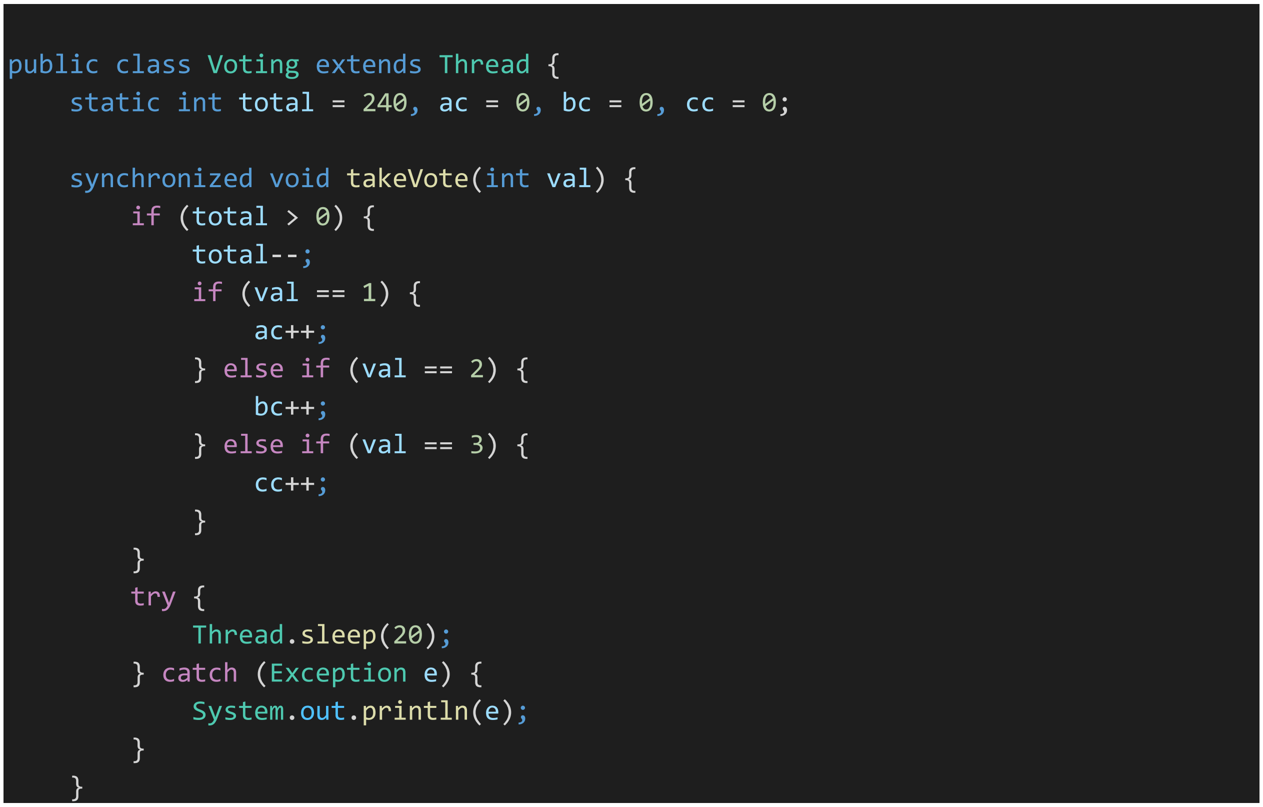
Output:



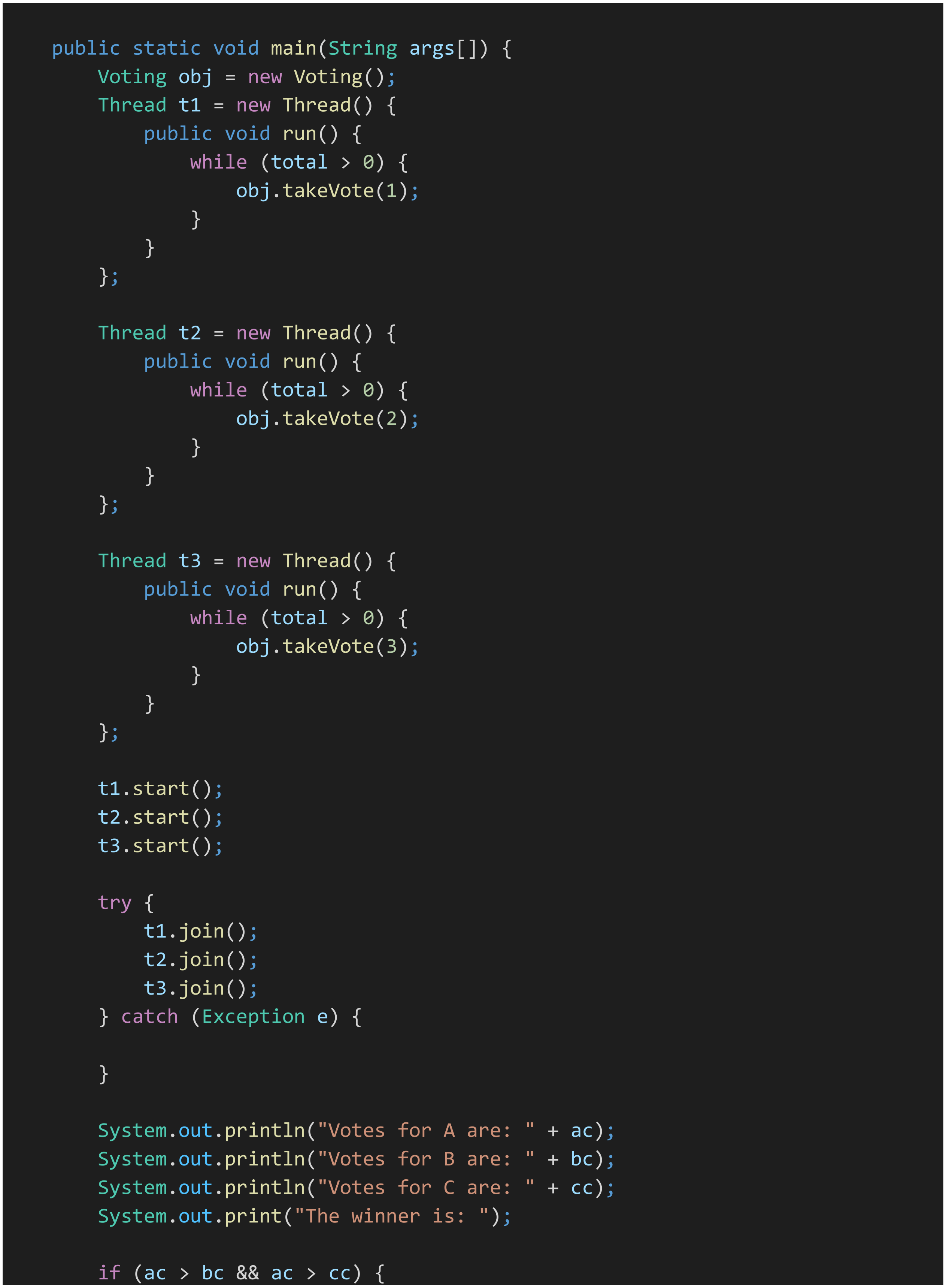
```
Hello
Welcome to VIT
Hello
Hello
Welcome to VIT
Hello
Hello
Hello
Welcome to VIT
Hello
Hello
Hello
Welcome to VIT
Hello
Hello
Hello
Welcome to VIT
Hello
Hello
Hello
Welcome to VIT
Hello
Terminate batch job (Y/N)? y
PS D:\VIT\class room\3rd Sem\JAVA\lab>
```

3.

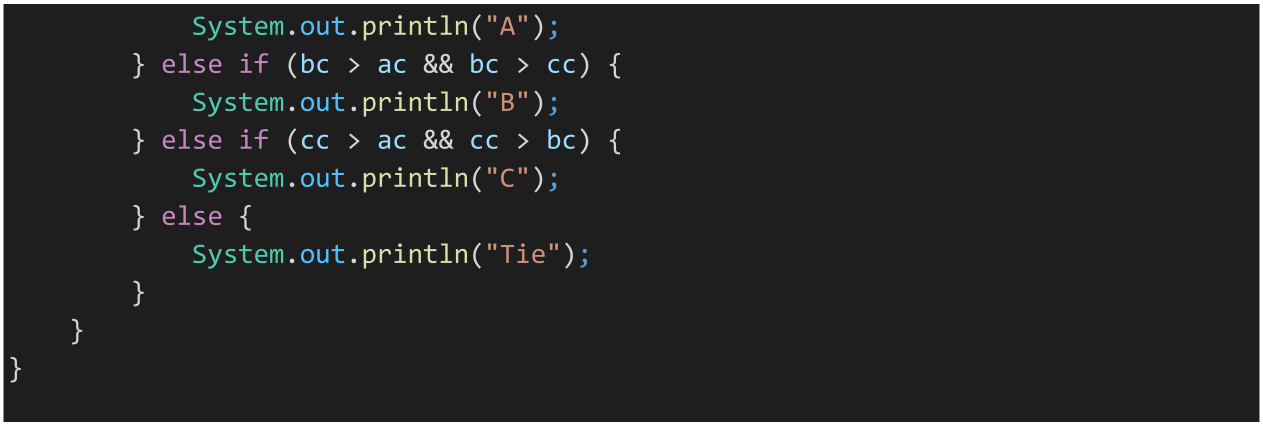
Code:



```java
public class Voting extends Thread {
    static int total = 240, ac = 0, bc = 0, cc = 0;

    synchronized void takeVote(int val) {
        if (total > 0) {
            total--;
            if (val == 1) {
                ac++;
            } else if (val == 2) {
                bc++;
            } else if (val == 3) {
                cc++;
            }
        }
        try {
            Thread.sleep(20);
        } catch (Exception e) {
            System.out.println(e);
        }
    }
```

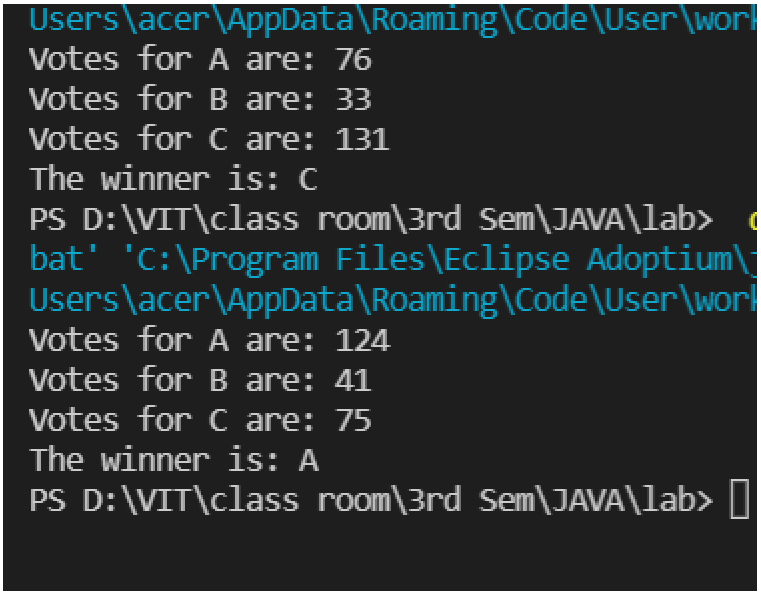


```
    public static void main(String args[]) {
        Voting obj = new Voting();
        Thread t1 = new Thread() {
            public void run() {
                while (total > 0) {
                    obj.takeVote(1);
                }
            }
        };

        Thread t2 = new Thread() {
            public void run() {
                while (total > 0) {
                    obj.takeVote(2);
                }
            }
        };

        Thread t3 = new Thread() {
            public void run() {
                while (total > 0) {
                    obj.takeVote(3);
                }
            }
        };

        t1.start();
        t2.start();
        t3.start();

        try {
            t1.join();
            t2.join();
            t3.join();
        } catch (Exception e) {

        }

        System.out.println("Votes for A are: " + ac);
        System.out.println("Votes for B are: " + bc);
        System.out.println("Votes for C are: " + cc);
        System.out.print("The winner is: ");

        if (ac > bc && ac > cc) {
```



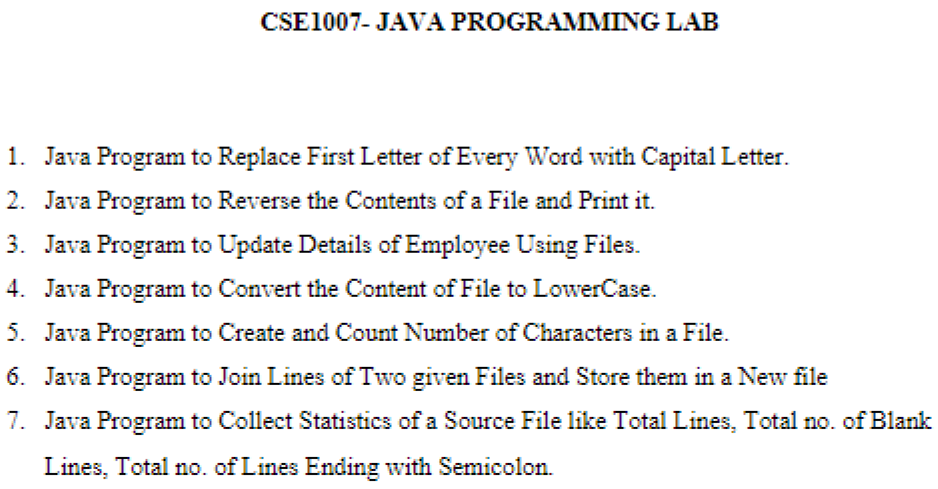
```
            System.out.println("A");
        } else if (bc > ac && bc > cc) {
            System.out.println("B");
        } else if (cc > ac && cc > bc) {
            System.out.println("C");
        } else {
            System.out.println("Tie");
        }
    }
}
```

Output:



```
Users\acer\AppData\Roaming\Code\User\wor
Votes for A are: 76
Votes for B are: 33
Votes for C are: 131
The winner is: C
PS D:\VIT\class room\3rd Sem\JAVA\lab>  c
bat' 'C:\Program Files\Eclipse Adoptium\
Users\acer\AppData\Roaming\Code\User\wor
Votes for A are: 124
Votes for B are: 41
Votes for C are: 75
The winner is: A
PS D:\VIT\class room\3rd Sem\JAVA\lab> 
```

File handling:



CSE1007- JAVA PROGRAMMING LAB

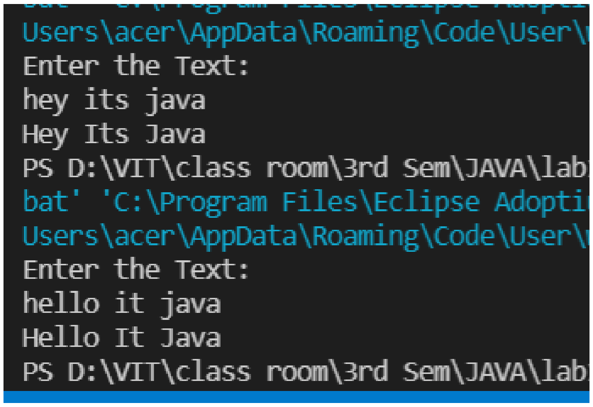
1. Java Program to Replace First Letter of Every Word with Capital Letter.
2. Java Program to Reverse the Contents of a File and Print it.
3. Java Program to Update Details of Employee Using Files.
4. Java Program to Convert the Content of File to LowerCase.
5. Java Program to Create and Count Number of Characters in a File.
6. Java Program to Join Lines of Two given Files and Store them in a New file
7. Java Program to Collect Statistics of a Source File like Total Lines, Total no. of Blank Lines, Total no. of Lines Ending with Semicolon.

1.

Code:

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

public class CapsWord {
    public static void main(String[] args) throws IOException {
        String file = "casefile.txt";
        Scanner scan = new Scanner(System.in);
        FileWriter fw = new FileWriter(file);
        System.out.println("Enter the Text:");
        String input = scan.nextLine();
        fw.write(input);
        fw.close();
        FileReader fr = new FileReader(file);
        int i;
        String s = new String();
        while ((i = fr.read()) != -1)
            s = s + (char) i;
        String result = "";
        Scanner linescan = new Scanner(s);
        while (linescan.hasNext()) {
            String word = linescan.next();
            result = result + Character.toUpperCase(word.charAt(0)) +
word.substring(1) + " ";
        }
        System.out.println(result);
        fr.close();
        scan.close();
        linescan.close();
    }
}
```

Output:

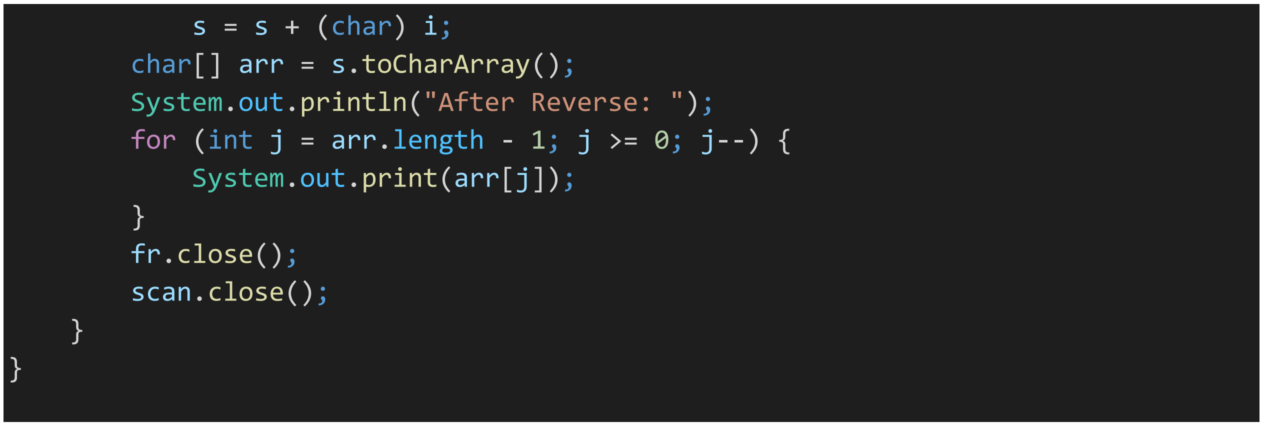


```
Users\acer\AppData\Roaming\Code\User\
Enter the Text:
hey its java
Hey Its Java
PS D:\VIT\class room\3rd Sem\JAVA\lab
bat' 'C:\Program Files\Eclipse Adopti
Users\acer\AppData\Roaming\Code\User\
Enter the Text:
hello it java
Hello It Java
PS D:\VIT\class room\3rd Sem\JAVA\lab
```

2.

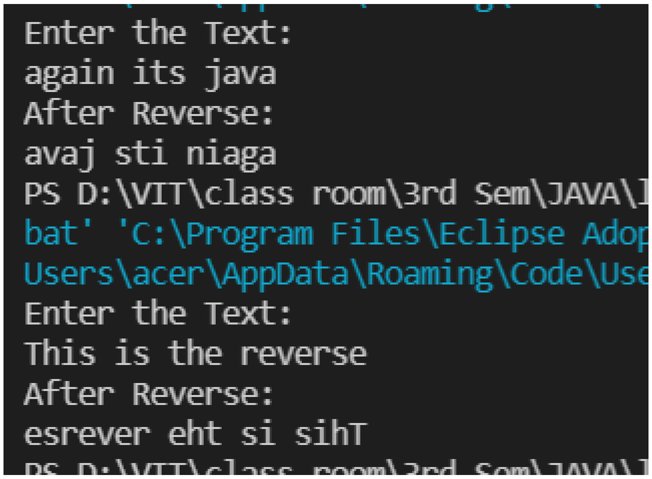
Code:

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

public class Reverse {
    public static void main(String[] args) throws IOException {
        String file = "casefile.txt";
        Scanner scan = new Scanner(System.in);
        FileWriter fw = new FileWriter(file);

        System.out.println("Enter the Text:");
        String input = scan.nextLine();
        fw.write(input);
        fw.close();
        FileReader fr = new FileReader(file);
        int i;
        String s = new String();
        while ((i = fr.read()) != -1)
```



```
            s = s + (char) i;
        char[] arr = s.toCharArray();
        System.out.println("After Reverse: ");
        for (int j = arr.length - 1; j >= 0; j--) {
            System.out.print(arr[j]);
        }
        fr.close();
        scan.close();
    }
}
```

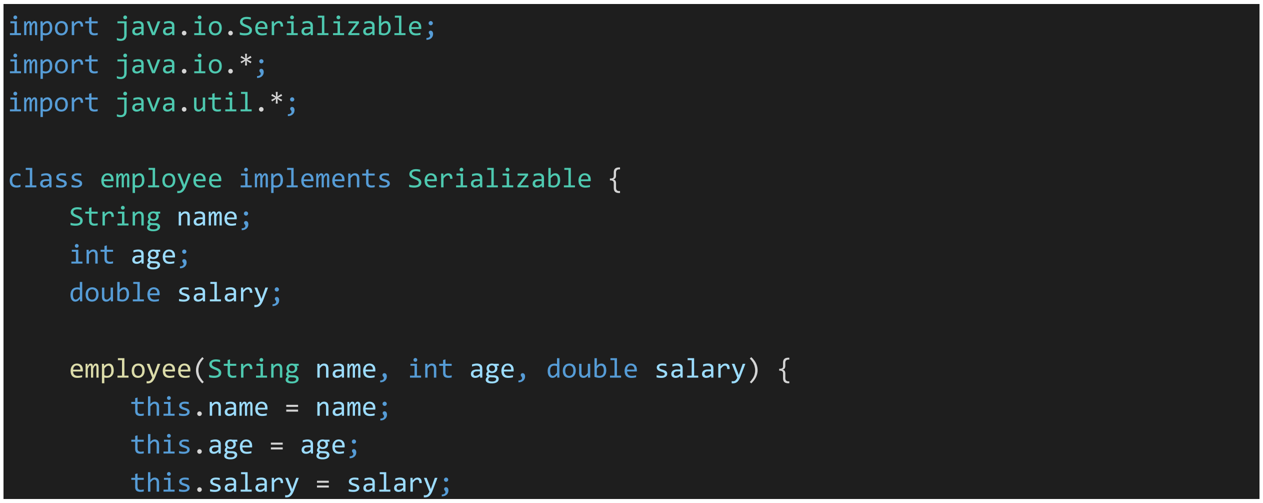
Output:



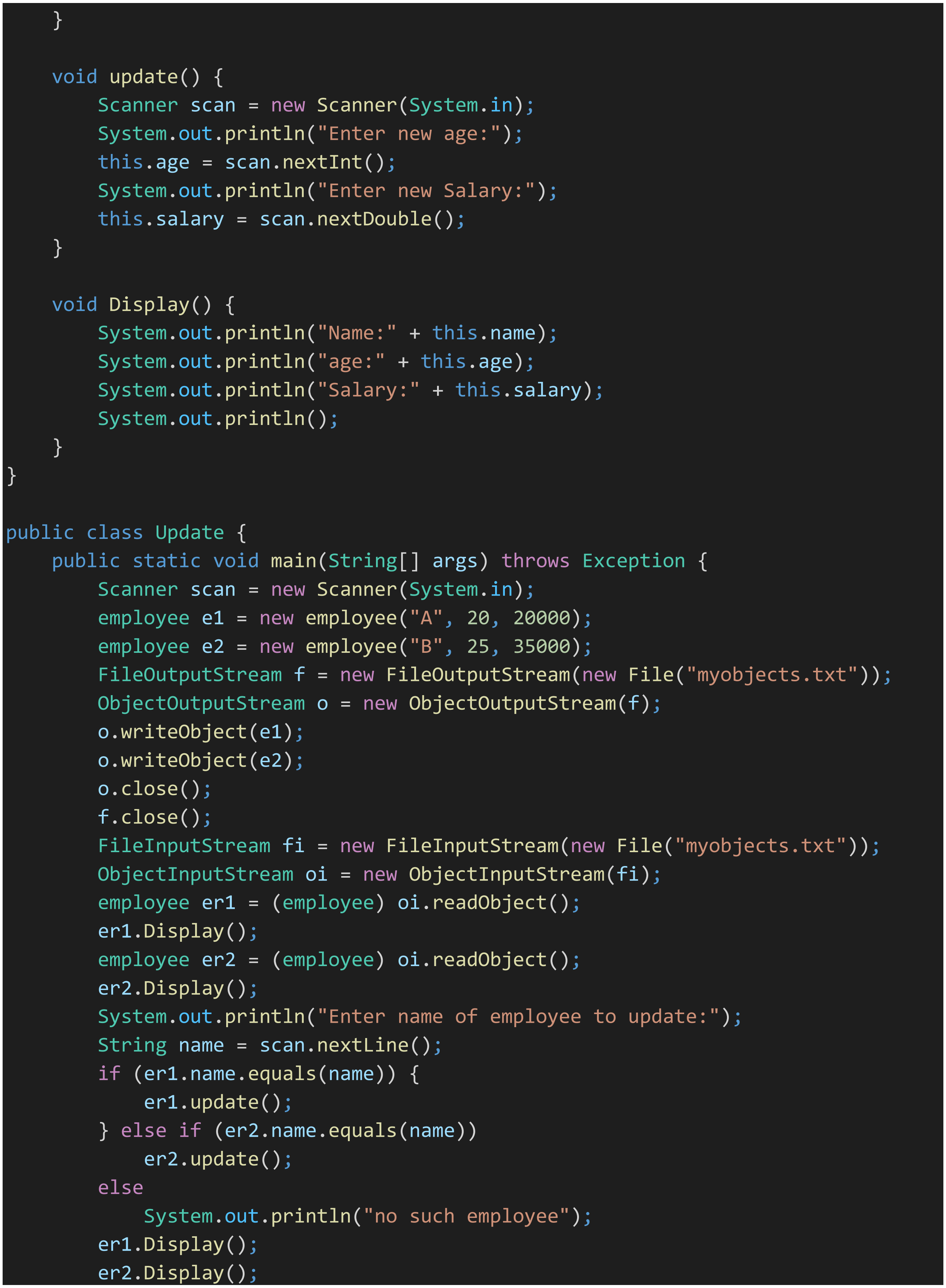
```
Enter the Text:
again its java
After Reverse:
avaj sti niaga
PS D:\VIT\class room\3rd Sem\JAVA\
bat' 'C:\Program Files\Eclipse Adop
Users\acer\AppData\Roaming\Code\Use
Enter the Text:
This is the reverse
After Reverse:
esrever eht si sihT
PS D:\VIT\class room\3rd Sem\JAVA\
```

3.

Code:

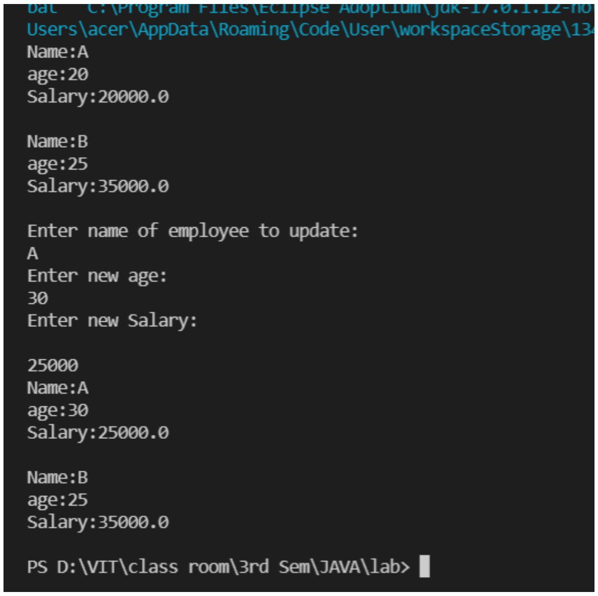


```
import java.io.Serializable;
import java.io.*;
import java.util.*;

class employee implements Serializable {
    String name;
    int age;
    double salary;

    employee(String name, int age, double salary) {
        this.name = name;
        this.age = age;
        this.salary = salary;
```



```java
    }

    void update() {
        Scanner scan = new Scanner(System.in);
        System.out.println("Enter new age:");
        this.age = scan.nextInt();
        System.out.println("Enter new Salary:");
        this.salary = scan.nextDouble();
    }

    void Display() {
        System.out.println("Name:" + this.name);
        System.out.println("age:" + this.age);
        System.out.println("Salary:" + this.salary);
        System.out.println();
    }
}

public class Update {
    public static void main(String[] args) throws Exception {
        Scanner scan = new Scanner(System.in);
        employee e1 = new employee("A", 20, 20000);
        employee e2 = new employee("B", 25, 35000);
        FileOutputStream f = new FileOutputStream(new File("myobjects.txt"));
        ObjectOutputStream o = new ObjectOutputStream(f);
        o.writeObject(e1);
        o.writeObject(e2);
        o.close();
        f.close();
        FileInputStream fi = new FileInputStream(new File("myobjects.txt"));
        ObjectInputStream oi = new ObjectInputStream(fi);
        employee er1 = (employee) oi.readObject();
        er1.Display();
        employee er2 = (employee) oi.readObject();
        er2.Display();
        System.out.println("Enter name of employee to update:");
        String name = scan.nextLine();
        if (er1.name.equals(name)) {
            er1.update();
        } else if (er2.name.equals(name))
            er2.update();
        else
            System.out.println("no such employee");
        er1.Display();
        er2.Display();
        scan.close();
    }

}
```

Output:



```
bat   C:\Program Files\Eclipse Adoptium\jdk-17.0.1.12-ho
Users\acer\AppData\Roaming\Code\User\workspaceStorage\134
Name:A
age:20
Salary:20000.0

Name:B
age:25
Salary:35000.0

Enter name of employee to update:
A
Enter new age:
30
Enter new Salary:

25000
Name:A
age:30
Salary:25000.0

Name:B
age:25
Salary:35000.0

PS D:\VIT\class room\3rd Sem\JAVA\lab> ▌
```

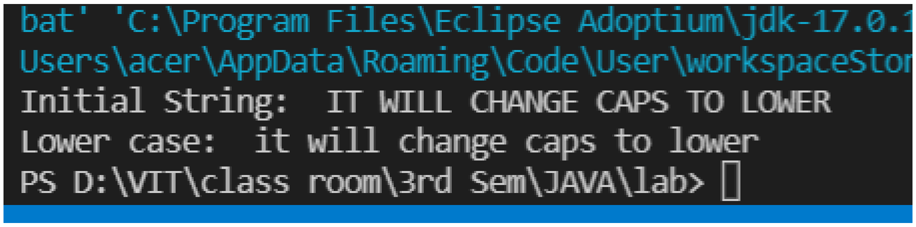
4.

Code:

```
import java.io.*;

public class Lowercase {
    public static void main(String[] args) throws Exception {
        FileWriter writer = new FileWriter("file.txt", true);
        writer.write("IT WILL CHANGE CAPS TO LOWER");
        writer.close();
        FileReader fr = new FileReader("file.txt");
        int i;
        String s = new String();
        while ((i = fr.read()) != -1)
            s = s + (char) i;
        System.out.println("Initial String:  " + s);
        System.out.println("Lower case:  " + s.toLowerCase());
        fr.close();
    }
```

```
}
```

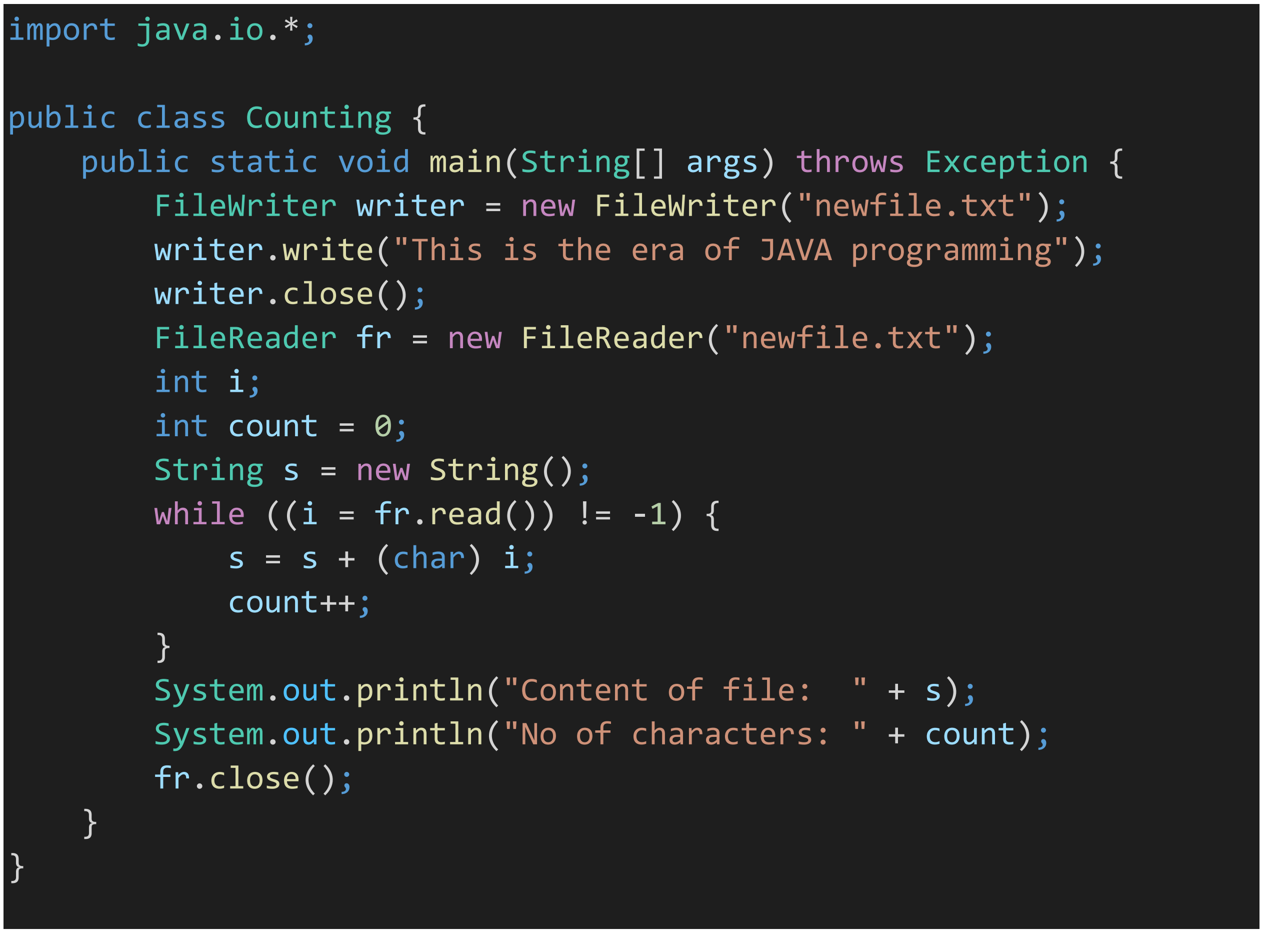
OUTPUT:



```
bat' 'C:\Program Files\Eclipse Adoptium\jdk-17.0.1
Users\acer\AppData\Roaming\Code\User\workspaceStor
Initial String:  IT WILL CHANGE CAPS TO LOWER
Lower case:  it will change caps to lower
PS D:\VIT\class room\3rd Sem\JAVA\lab> 
```

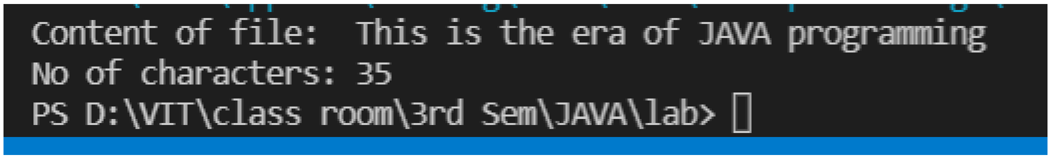
5.

CODE:



```java
import java.io.*;

public class Counting {
    public static void main(String[] args) throws Exception {
        FileWriter writer = new FileWriter("newfile.txt");
        writer.write("This is the era of JAVA programming");
        writer.close();
        FileReader fr = new FileReader("newfile.txt");
        int i;
        int count = 0;
        String s = new String();
        while ((i = fr.read()) != -1) {
            s = s + (char) i;
            count++;
        }
        System.out.println("Content of file:  " + s);
        System.out.println("No of characters: " + count);
        fr.close();
    }
}
```

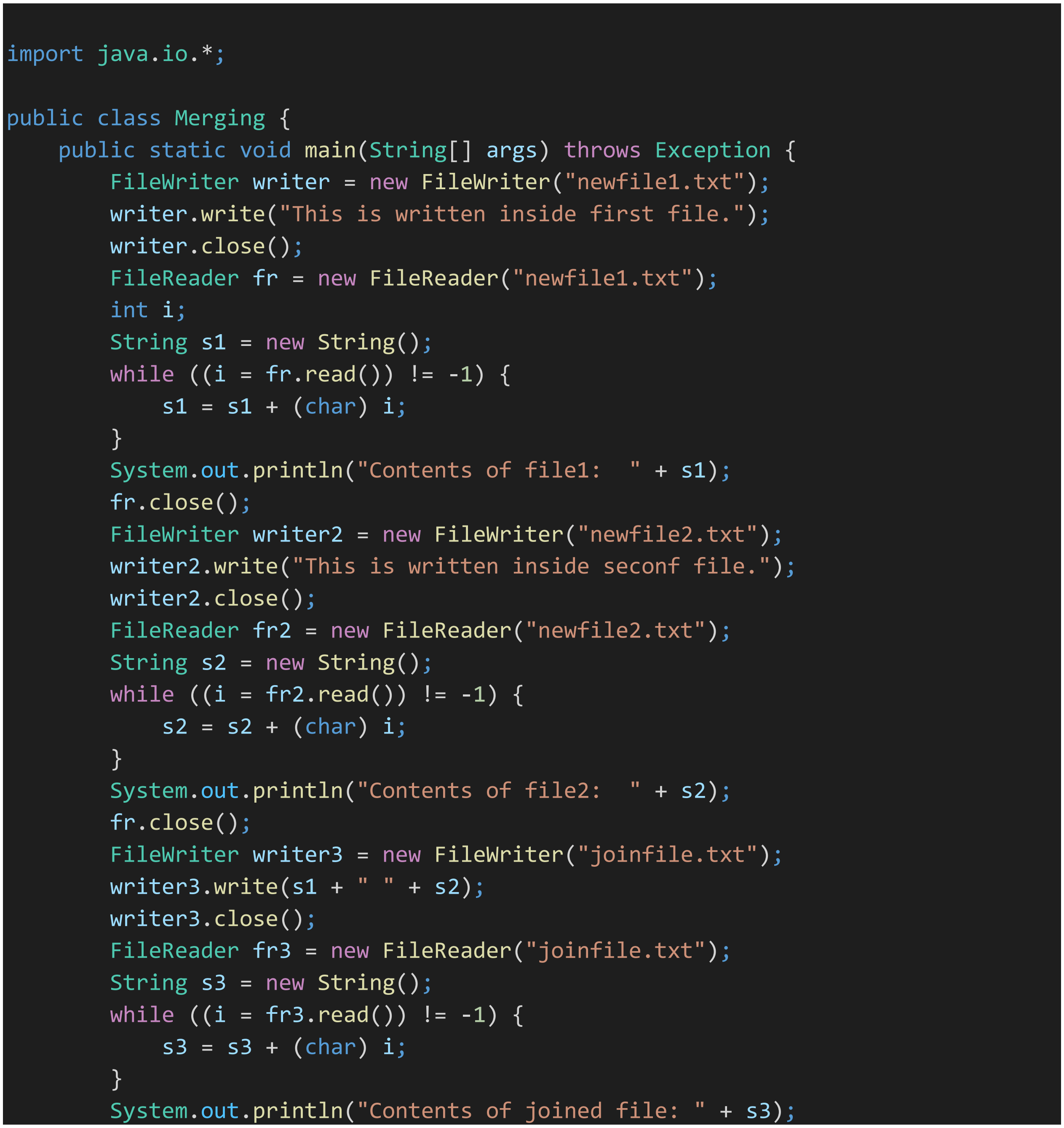
Output:



```
Content of file:  This is the era of JAVA programming
No of characters: 35
PS D:\VIT\class room\3rd Sem\JAVA\lab> 
```

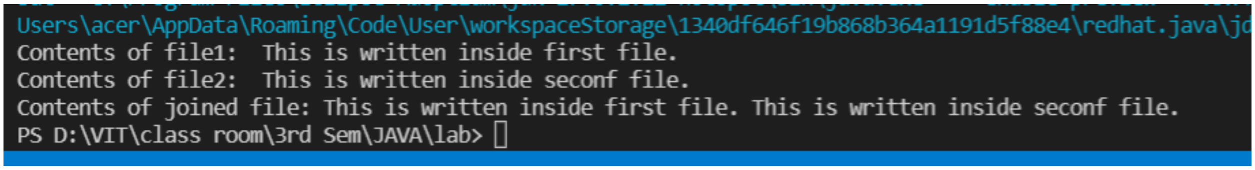
6.

Code:



```
import java.io.*;

public class Merging {
    public static void main(String[] args) throws Exception {
        FileWriter writer = new FileWriter("newfile1.txt");
        writer.write("This is written inside first file.");
        writer.close();
        FileReader fr = new FileReader("newfile1.txt");
        int i;
        String s1 = new String();
        while ((i = fr.read()) != -1) {
            s1 = s1 + (char) i;
        }
        System.out.println("Contents of file1:  " + s1);
        fr.close();
        FileWriter writer2 = new FileWriter("newfile2.txt");
        writer2.write("This is written inside seconf file.");
        writer2.close();
        FileReader fr2 = new FileReader("newfile2.txt");
        String s2 = new String();
        while ((i = fr2.read()) != -1) {
            s2 = s2 + (char) i;
        }
        System.out.println("Contents of file2:  " + s2);
        fr.close();
        FileWriter writer3 = new FileWriter("joinfile.txt");
        writer3.write(s1 + " " + s2);
        writer3.close();
        FileReader fr3 = new FileReader("joinfile.txt");
        String s3 = new String();
        while ((i = fr3.read()) != -1) {
            s3 = s3 + (char) i;
        }
        System.out.println("Contents of joined file: " + s3);
        fr2.close();

        fr3.close();
    }
}
```

Output:

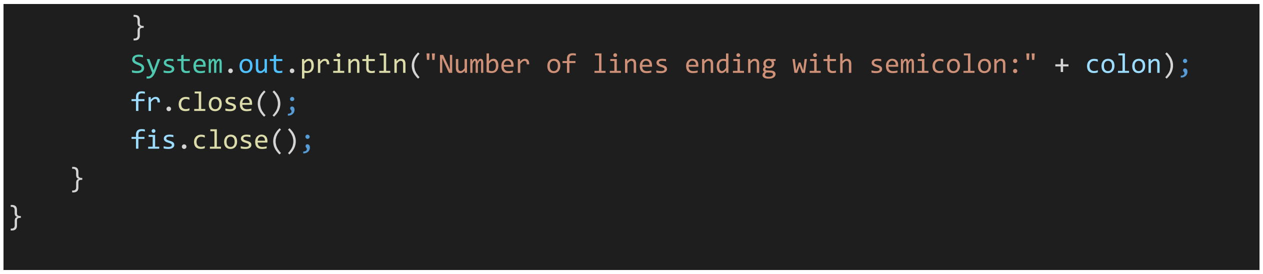


```
Users\acer\AppData\Roaming\Code\User\workspaceStorage\1340df646f19b868b364a1191d5f88e4\redhat.java\jd
Contents of file1:  This is written inside first file.
Contents of file2:  This is written inside seconf file.
Contents of joined file: This is written inside first file. This is written inside seconf file.
PS D:\VIT\class room\3rd Sem\JAVA\lab> 
```

7.

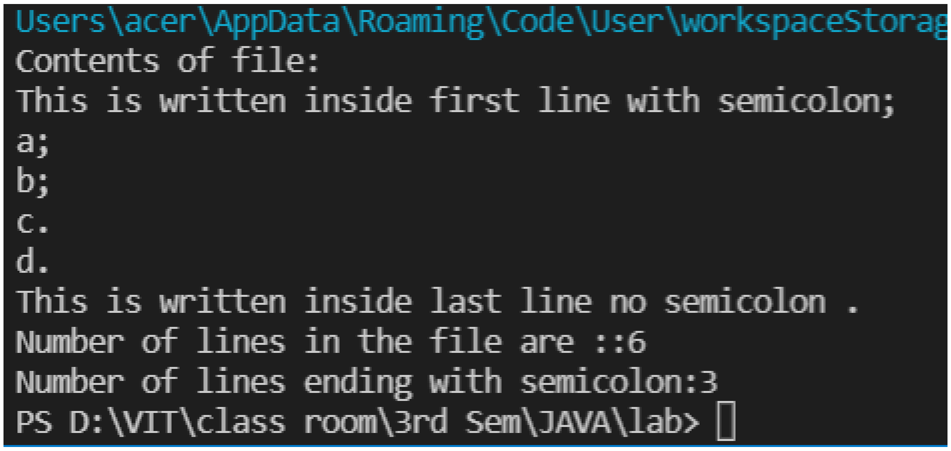
Code:

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileReader;

public class Counter {
    public static void main(String args[]) throws Exception {
        FileReader fr = new FileReader("joinfile.txt");
        int i;
        String s = new String();
        while ((i = fr.read()) != -1)
            s = s + (char) i;
        System.out.println("Contents of file:");
        System.out.println(s);
        File file = new File("joinfile.txt");
        FileInputStream fis = new FileInputStream(file);
        byte[] byteArray = new byte[(int) file.length()];
        fis.read(byteArray);
        String data = new String(byteArray);
        String[] stringArray = data.split("\r\n");
        System.out.println("Number of lines in the file are ::" +
stringArray.length);
        String[] sarr = s.split("\n");
        int colon = 0;
        for (i = 0; i < sarr.length; i++) {
            if (sarr[i].indexOf(';') != -1) {
                colon++;
            }
```



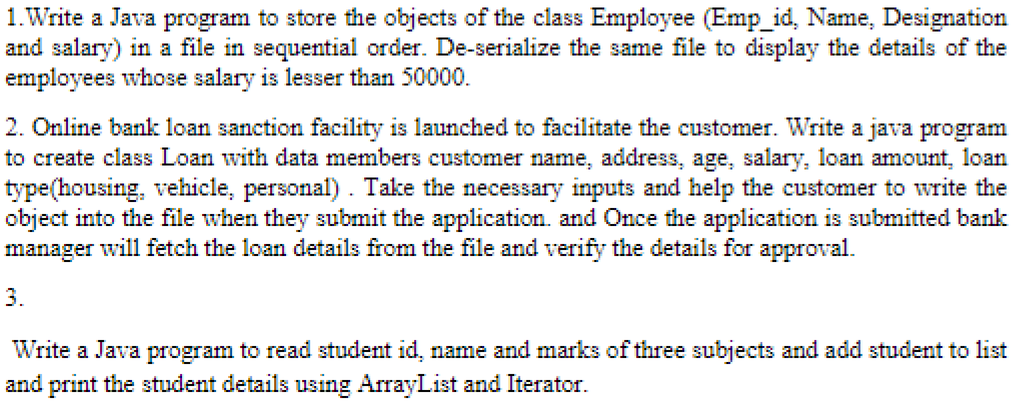
```
        }
        System.out.println("Number of lines ending with semicolon:" + colon);
        fr.close();
        fis.close();
    }
}
```

Output:



```
Users\acer\AppData\Roaming\Code\User\workspaceStorag
Contents of file:
This is written inside first line with semicolon;
a;
b;
c.
d.
This is written inside last line no semicolon .
Number of lines in the file are ::6
Number of lines ending with semicolon:3
PS D:\VIT\class room\3rd Sem\JAVA\lab> 
```

FILE SERIALIZABLE- DESERIALIZABLE:



1.Write a Java program to store the objects of the class Employee (Emp_id, Name, Designation and salary) in a file in sequential order. De-serialize the same file to display the details of the employees whose salary is lesser than 50000.

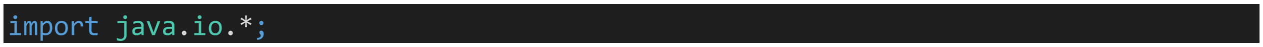
2. Online bank loan sanction facility is launched to facilitate the customer. Write a java program to create class Loan with data members customer name, address, age, salary, loan amount, loan type(housing, vehicle, personal) . Take the necessary inputs and help the customer to write the object into the file when they submit the application. and Once the application is submitted bank manager will fetch the loan details from the file and verify the details for approval.

3.

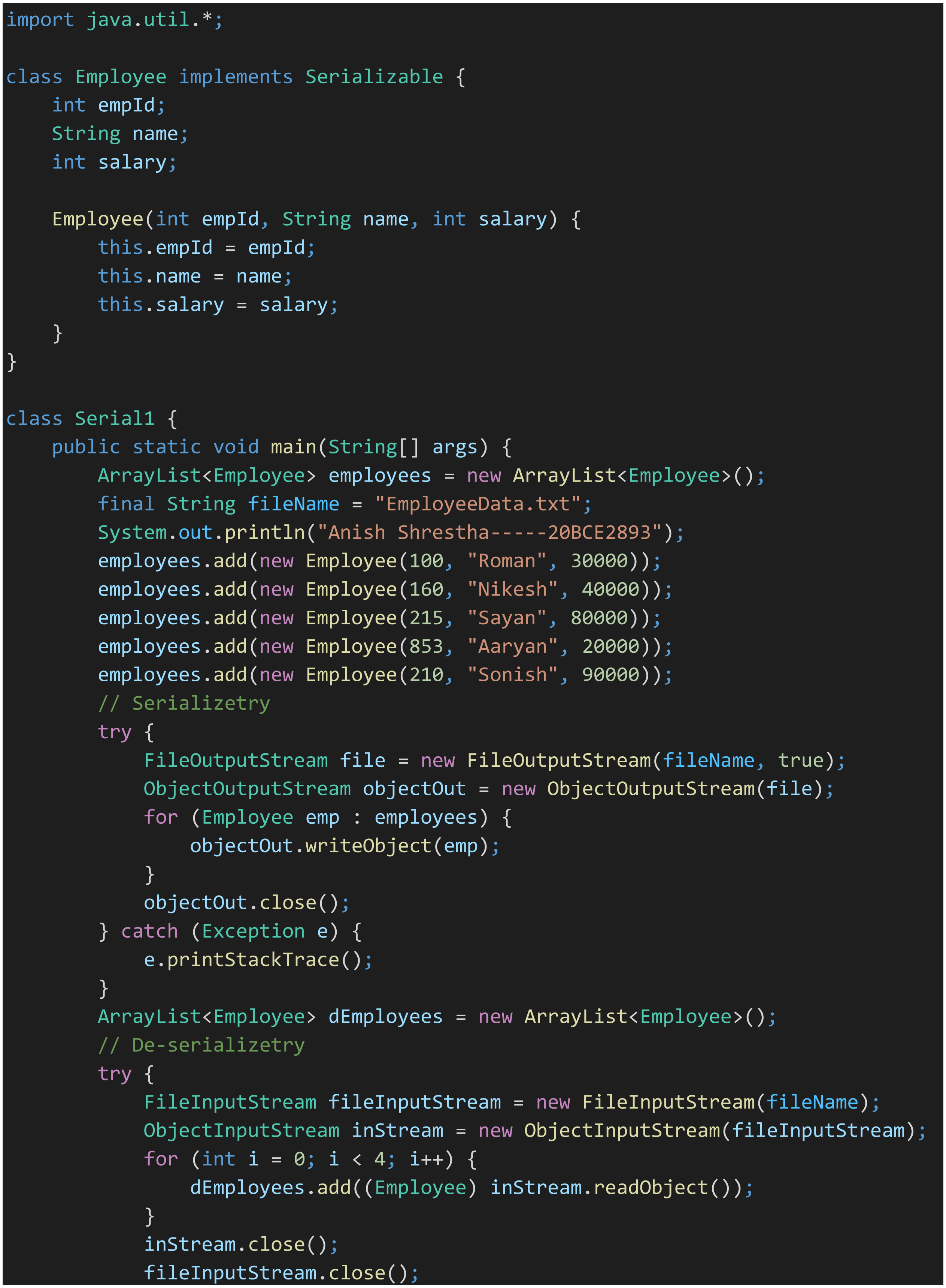
Write a Java program to read student id, name and marks of three subjects and add student to list and print the student details using ArrayList and Iterator.

1.

CODE:

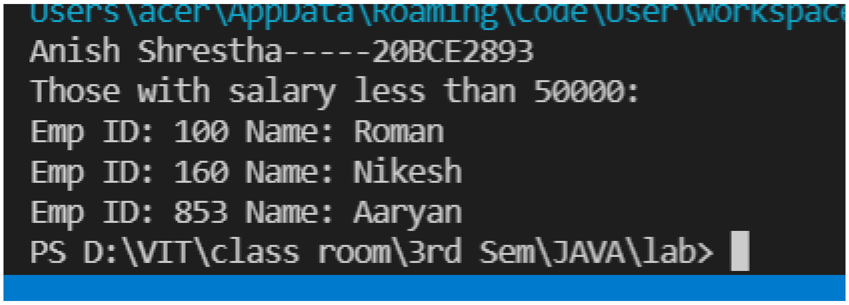


```
import java.io.*;
```



```java
import java.util.*;

class Employee implements Serializable {
    int empId;
    String name;
    int salary;

    Employee(int empId, String name, int salary) {
        this.empId = empId;
        this.name = name;
        this.salary = salary;
    }
}

class Serial1 {
    public static void main(String[] args) {
        ArrayList<Employee> employees = new ArrayList<Employee>();
        final String fileName = "EmployeeData.txt";
        System.out.println("Anish Shrestha-----20BCE2893");
        employees.add(new Employee(100, "Roman", 30000));
        employees.add(new Employee(160, "Nikesh", 40000));
        employees.add(new Employee(215, "Sayan", 80000));
        employees.add(new Employee(853, "Aaryan", 20000));
        employees.add(new Employee(210, "Sonish", 90000));
        // Serializetry
        try {
            FileOutputStream file = new FileOutputStream(fileName, true);
            ObjectOutputStream objectOut = new ObjectOutputStream(file);
            for (Employee emp : employees) {
                objectOut.writeObject(emp);
            }
            objectOut.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
        ArrayList<Employee> dEmployees = new ArrayList<Employee>();
        // De-serializetry
        try {
            FileInputStream fileInputStream = new FileInputStream(fileName);
            ObjectInputStream inStream = new ObjectInputStream(fileInputStream);
            for (int i = 0; i < 4; i++) {
                dEmployees.add((Employee) inStream.readObject());
            }
            inStream.close();
            fileInputStream.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
        System.out.println("Those with salary less than 50000: ");
        for (Employee emp : dEmployees) {
            if (emp.salary < 50000) {
                System.out.println("Emp ID: " + Integer.toString(emp.empId) + " Name: " + emp.name);
            }
        }
    }
}
```

Output:

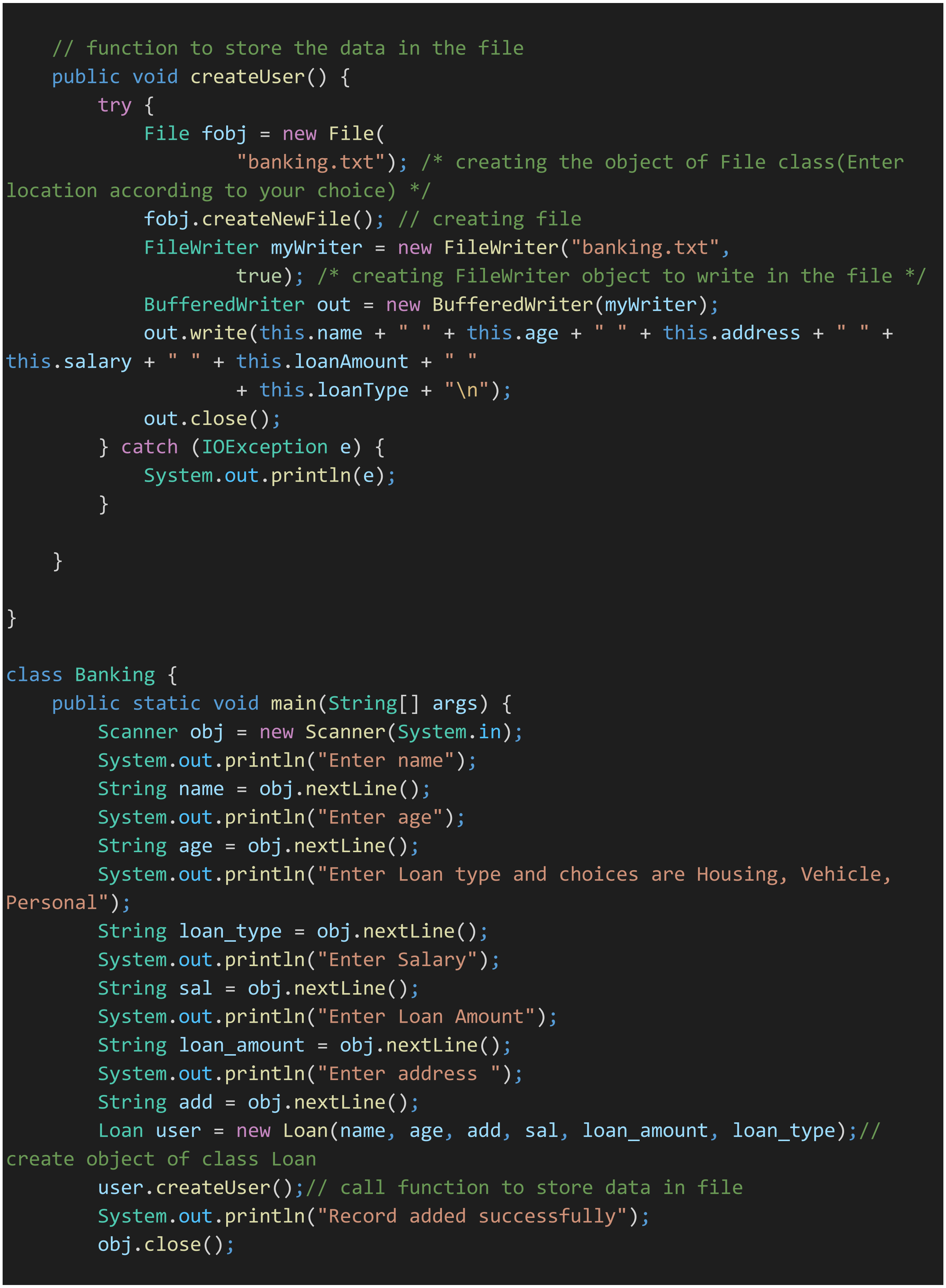


```
Users\acer\AppData\Roaming\Code\User\workspace
Anish Shrestha-----20BCE2893
Those with salary less than 50000:
Emp ID: 100 Name: Roman
Emp ID: 160 Name: Nikesh
Emp ID: 853 Name: Aaryan
PS D:\VIT\class room\3rd Sem\JAVA\lab>
```

2.

Code for client/customer:

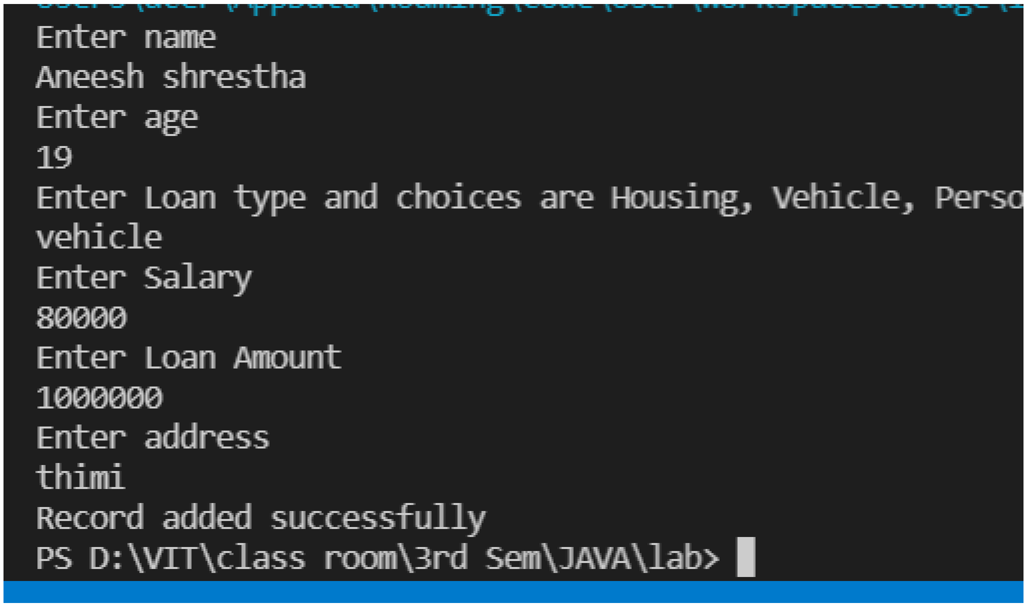
```
import java.io.*;
import java.util.Scanner;

class Loan {
    private String name, age, address, salary, loanAmount, loanType;

    // constructor for variable initialization
    Loan(String n, String age, String address, String salary, String la, String lt) {
        this.name = n;
        this.age = age;
        this.address = address;
        this.salary = salary;
        this.loanAmount = la;
        this.loanType = lt;

    }
```



```
    // function to store the data in the file
    public void createUser() {
        try {
            File fobj = new File(
                    "banking.txt"); /* creating the object of File class(Enter
location according to your choice) */
            fobj.createNewFile(); // creating file
            FileWriter myWriter = new FileWriter("banking.txt",
                    true); /* creating FileWriter object to write in the file */
            BufferedWriter out = new BufferedWriter(myWriter);
            out.write(this.name + " " + this.age + " " + this.address + " " +
this.salary + " " + this.loanAmount + " "
                    + this.loanType + "\n");
            out.close();
        } catch (IOException e) {
            System.out.println(e);
        }

    }

}

class Banking {
    public static void main(String[] args) {
        Scanner obj = new Scanner(System.in);
        System.out.println("Enter name");
        String name = obj.nextLine();
        System.out.println("Enter age");
        String age = obj.nextLine();
        System.out.println("Enter Loan type and choices are Housing, Vehicle,
Personal");
        String loan_type = obj.nextLine();
        System.out.println("Enter Salary");
        String sal = obj.nextLine();
        System.out.println("Enter Loan Amount");
        String loan_amount = obj.nextLine();
        System.out.println("Enter address ");
        String add = obj.nextLine();
        Loan user = new Loan(name, age, add, sal, loan_amount, loan_type);//
create object of class Loan
        user.createUser();// call function to store data in file
        System.out.println("Record added successfully");
        obj.close();
```

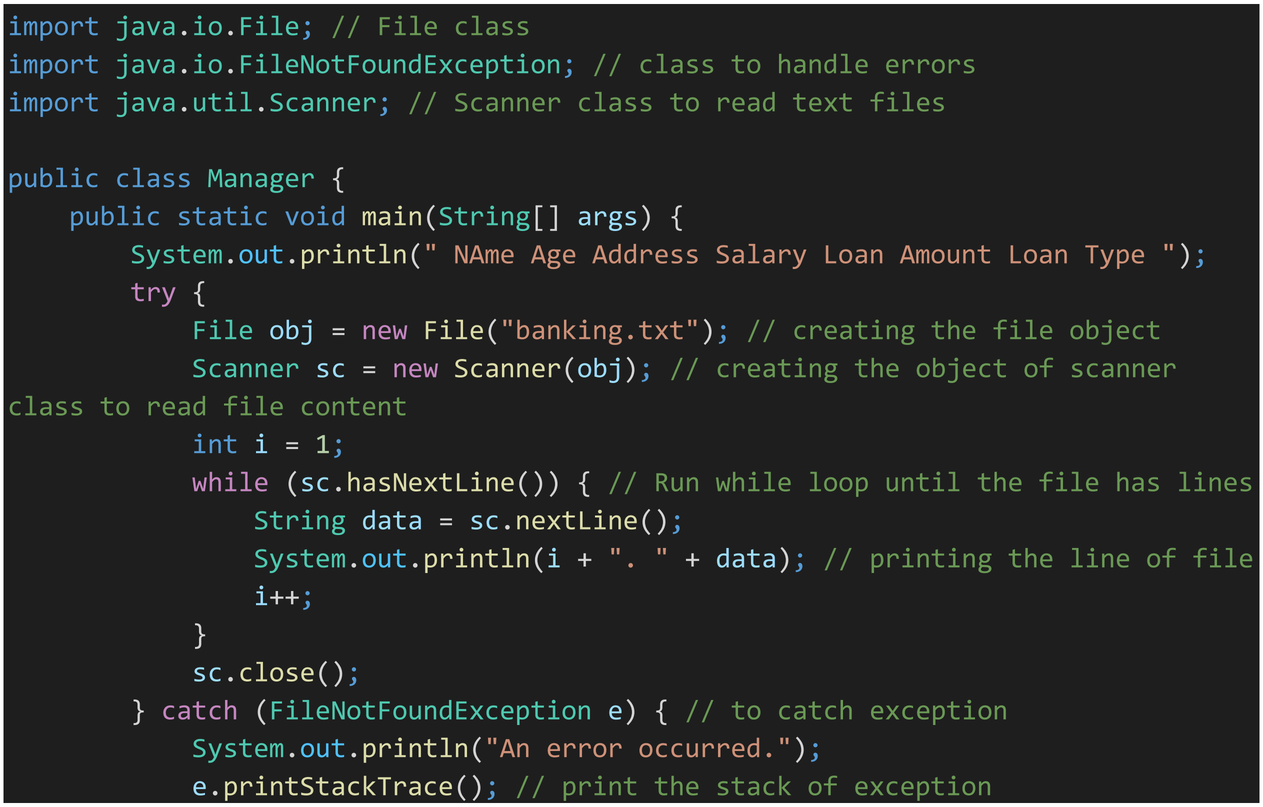
```
    }
}
```

Output:



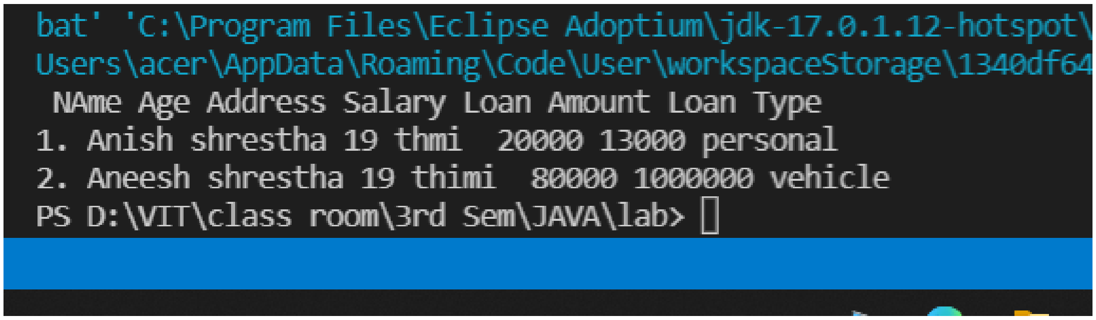
```
Enter name
Aneesh shrestha
Enter age
19
Enter Loan type and choices are Housing, Vehicle, Perso
vehicle
Enter Salary
80000
Enter Loan Amount
1000000
Enter address
thimi
Record added successfully
PS D:\VIT\class room\3rd Sem\JAVA\lab> 
```

Code for Manager:

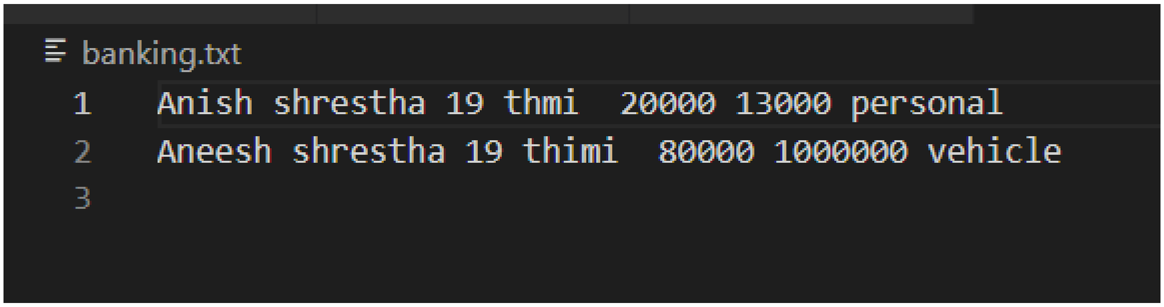


```
import java.io.File; // File class
import java.io.FileNotFoundException; // class to handle errors
import java.util.Scanner; // Scanner class to read text files

public class Manager {
    public static void main(String[] args) {
        System.out.println(" NAme Age Address Salary Loan Amount Loan Type ");
        try {
            File obj = new File("banking.txt"); // creating the file object
            Scanner sc = new Scanner(obj); // creating the object of scanner
class to read file content
            int i = 1;
            while (sc.hasNextLine()) { // Run while loop until the file has lines
                String data = sc.nextLine();
                System.out.println(i + ". " + data); // printing the line of file
                i++;
            }
            sc.close();
        } catch (FileNotFoundException e) { // to catch exception
            System.out.println("An error occurred.");
            e.printStackTrace(); // print the stack of exception
        }
    }
}
```

Output:



```
 bat' 'C:\Program Files\Eclipse Adoptium\jdk-17.0.1.12-hotspot\
 Users\acer\AppData\Roaming\Code\User\workspaceStorage\1340df64
  NAme Age Address Salary Loan Amount Loan Type
 1. Anish shrestha 19 thmi  20000 13000 personal
 2. Aneesh shrestha 19 thimi  80000 1000000 vehicle
 PS D:\VIT\class room\3rd Sem\JAVA\lab> 
```



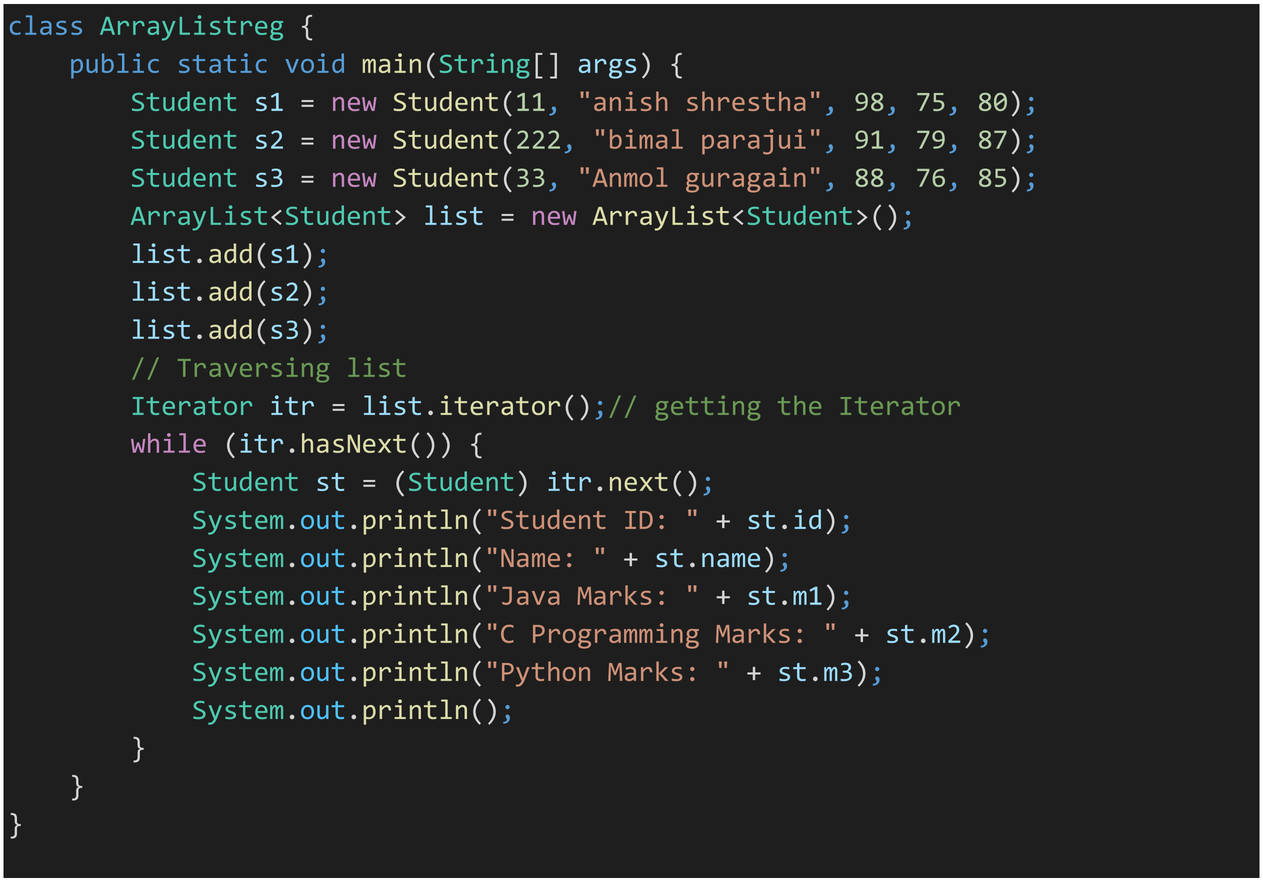
banking.txt

```
Anish shrestha 19 thmi  20000 13000 personal
Aneesh shrestha 19 thimi  80000 1000000 vehicle

```

3.

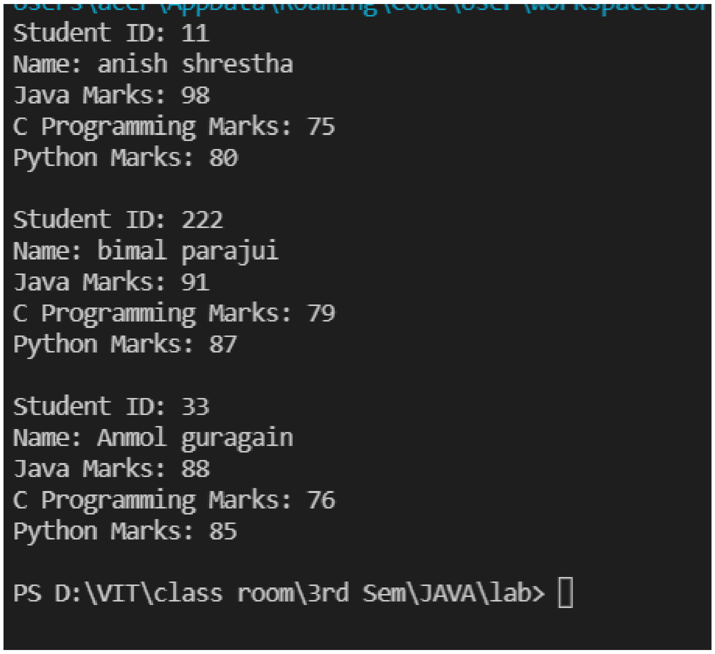
Code:

```
import java.util.*;

class Student {
    int id, m1, m2, m3;
    String name;

    Student(int id, String name, int m1, int m2, int m3) {
        this.id = id;
        this.name = name;
        this.m1 = m1;
        this.m2 = m2;
        this.m3 = m3;
    }
}
```



```
class ArrayListreg {
    public static void main(String[] args) {
        Student s1 = new Student(11, "anish shrestha", 98, 75, 80);
        Student s2 = new Student(222, "bimal parajui", 91, 79, 87);
        Student s3 = new Student(33, "Anmol guragain", 88, 76, 85);
        ArrayList<Student> list = new ArrayList<Student>();
        list.add(s1);
        list.add(s2);
        list.add(s3);
        // Traversing list
        Iterator itr = list.iterator();// getting the Iterator
        while (itr.hasNext()) {
            Student st = (Student) itr.next();
            System.out.println("Student ID: " + st.id);
            System.out.println("Name: " + st.name);
            System.out.println("Java Marks: " + st.m1);
            System.out.println("C Programming Marks: " + st.m2);
            System.out.println("Python Marks: " + st.m3);
            System.out.println();
        }
    }
}
```

Output:



```
Users\acer\AppData\Roaming\Code\User\workspaceStor
Student ID: 11
Name: anish shrestha
Java Marks: 98
C Programming Marks: 75
Python Marks: 80

Student ID: 222
Name: bimal parajui
Java Marks: 91
C Programming Marks: 79
Python Marks: 87

Student ID: 33
Name: Anmol guragain
Java Marks: 88
C Programming Marks: 76
Python Marks: 85

PS D:\VIT\class room\3rd Sem\JAVA\lab> 
```