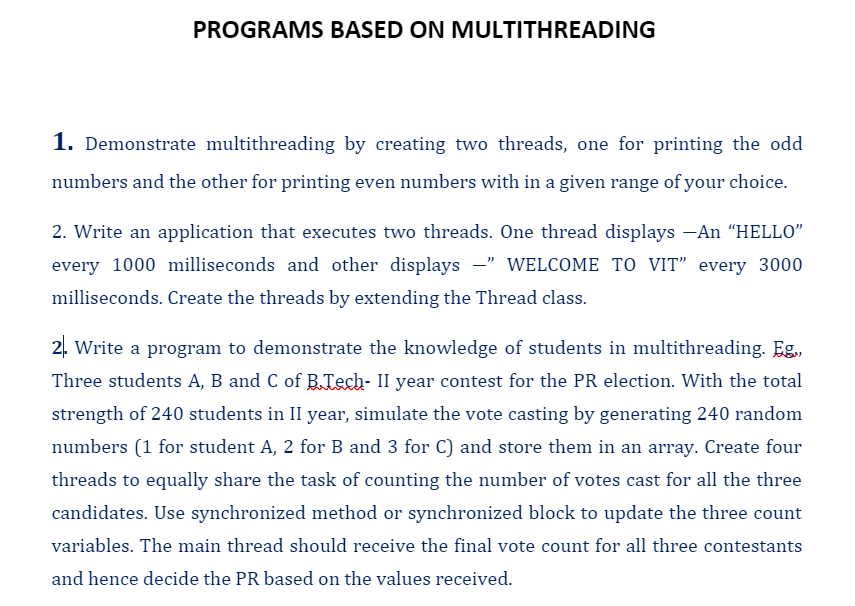
|  |
| --- |
| Photo displaying partial image of two pie charts on a canvas-textured page |
| DIGITAL ASSIGNMENT 5  CSE1007 / JAVA LAB |
| |  |  |  | | --- | --- | --- | | ANISH SHRESTHA | 11/17/21 | 20BCE2893 | |

**DIGITAL ASSIGNMENT -5**

**On Multi-threading**



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) {

                }

            }

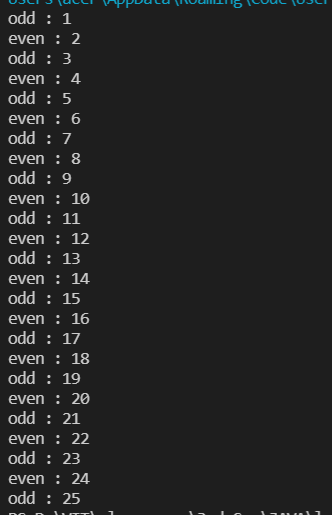
            number += 2;

        }

    }

}

Output:



2.

Code:

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:



3.

Code:

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 {

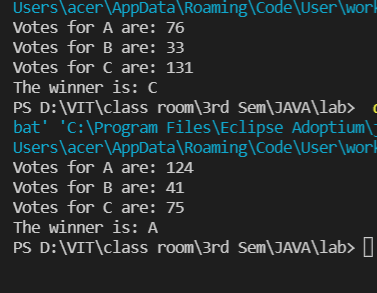
            System.out.println("Tie");

        }

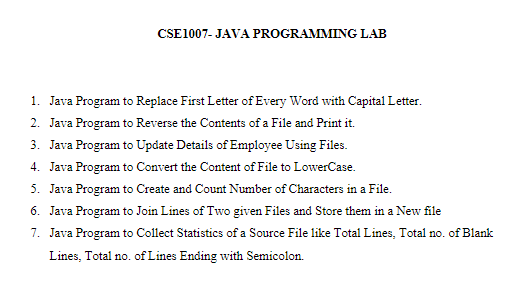
    }

}

Output:



File handling:



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();

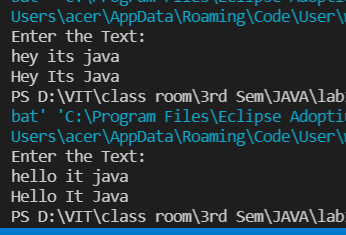
        scan.close();

        linescan.close();

    }

}

Output:



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]);

        }

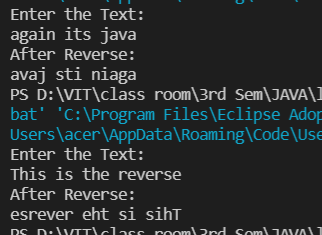
        fr.close();

        scan.close();

    }

}

Output:



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;

    }

    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();

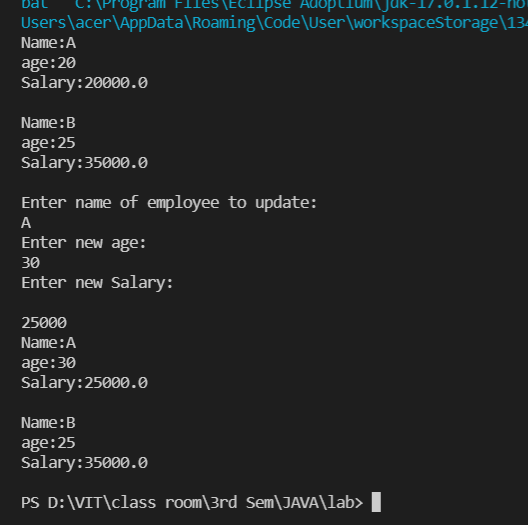
        er2.Display();

        scan.close();

    }

}

Output:



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);

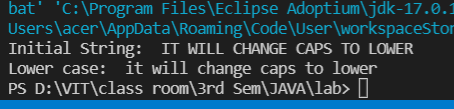
        System.out.println("Lower case:  " + s.toLowerCase());

        fr.close();

    }

}

OUTPUT:



5.

CODE:

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);

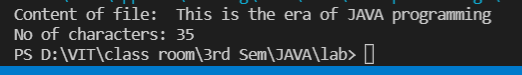
        System.out.println("No of characters: " + count);

        fr.close();

    }

}

Output:



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);

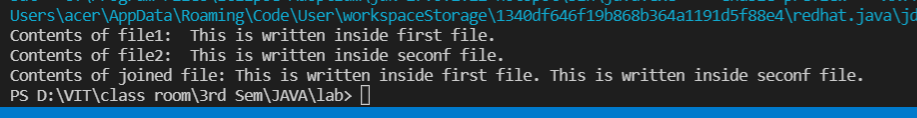
        fr2.close();

        fr3.close();

    }

}

Output:



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);

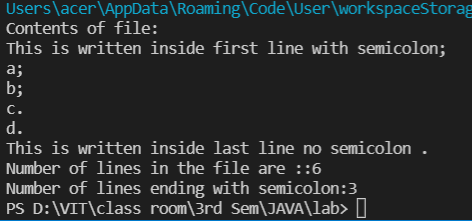
        fr.close();

        fis.close();

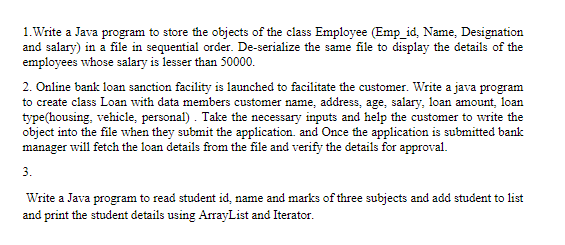
    }

}

Output:



FILE SERIALIZABLE- DESERIALIZABLE:



1.

CODE:

import java.io.\*;

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);

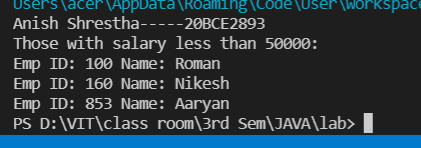
            }

        }

    }

}

Output:



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

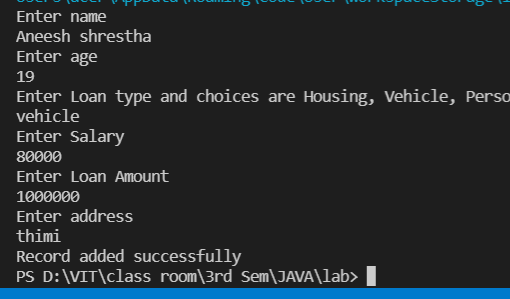
        System.out.println("Record added successfully");

        obj.close();

    }

}

Output:



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.");

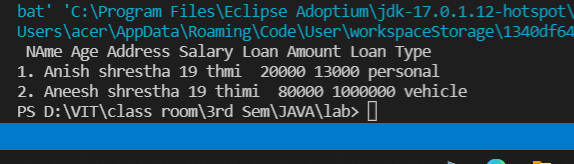
            e.printStackTrace(); // print the stack of exception

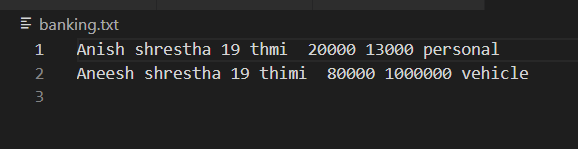
        }

    }

}

Output:





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:

