**Practical**

**JAVA Programming**

**Ashutosh Chandrakant Deshmukh**

**Roll No : 15**

**Division : Technocrats**

1. Print the pattern as follows:

1

212

32123

4321234

543212345

**Code :**

public class PatternPrint2 {

    public static *void* main(String[] *args*) {

        for (*int* i = 1; i <= 5; i++) {

*int* n = 5;

            for (*int* j = 1; j <= n - i; j++) {

                System.out.print(" ");

            }

            for (*int* k = i; k >= 1; k--) {

                System.out.print(k);

            }

            for (*int* l = 2; l <= i; l++) {

                System.out.print(l);

            }

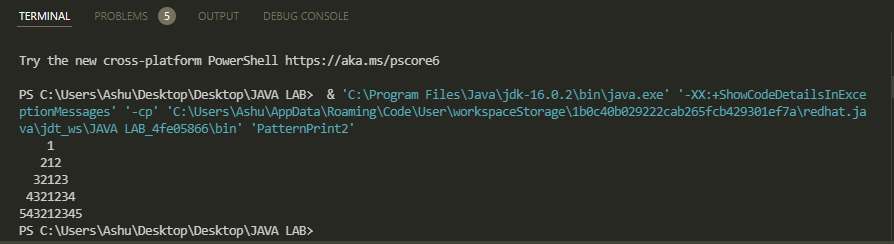
            System.out.println();

        }

    }

}

**Ouput :**



1. Write a program to find out the sum of all digits provided sum should be of single digit. Note: If num=8422 then sumnum=8+4+2+2=16 => 1+6=7

**Code :**

public class SumOfDigits {

    public static *void* main(String[] *args*) {

*int* n = 845;

*int* sum = 0;

        while (n > 0 || sum > 9) {

            if (n == 0) {

                n = sum;

                sum = 0;

            }

            sum = sum + n % 10;

            n = n / 10;

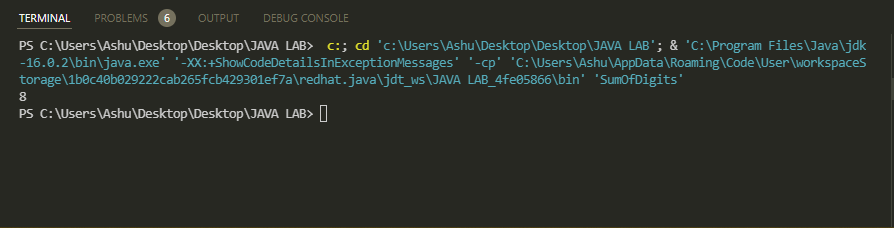
        }

        System.out.println(sum);

    }

}

**Output :**



1. Write a program to calculate the sum of even digits and odd digits. Note: if num=12243 then evensum=2+2+4=8 and oddsum=1+3=4

**Code:**

import java.util.\*;

public class Sumofevenandodd {

    public static *void* main(String[] *args*) {

*int* n;

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the number : ");

        n = sc.nextInt();

        sc.close();

*int* esum = 0;

*int* osum = 0;

*int* rem;

        while (n > 0) {

            rem = n % 10; // for getting last digit

            if (rem % 2 == 0) {

                // System.out.println("Even Number : " + rem);

                esum = esum + rem;

            } else {

                // System.out.println("Odd Number : " + rem);

                osum = osum + rem;

            }

            n = n / 10;

        }

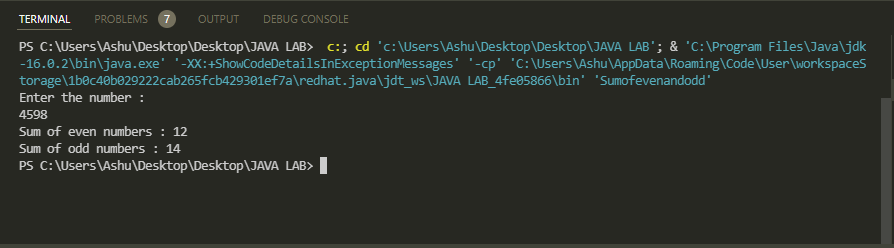
        System.out.println("Sum of even numbers : " + esum);

        System.out.println("Sum of odd numbers : " + osum);

    }

}

**Output :**



1. Create a class of Customer with members Cust\_nm, age, contact num, membership, reward\_points. Demonstrate Constructor Overloading in the same.

**Code :**

class Customer {

*int* cust\_num, cust\_age, contact\_num;

    Customer(*int* *a*, *int* *b*, *int* *c*) {

        cust\_num = *a*;

        cust\_age = *b*;

        contact\_num = *c*;

        System.out.println("Parameterized Constructor Invoked");

    }

    Customer() {

        cust\_num = 1;

        cust\_age = 21;

        contact\_num = 83088787;

        System.out.println("Default Constructor Invoked");

    }

}

public class ConstructorOverloading {

    public static *void* main(String[] *args*) {

        Customer c1 = new Customer(2, 25, 789875);

        System.out.println("Cust Num :" + c1.cust\_num);

        System.out.println("Cust Num :" + c1.cust\_age);

        System.out.println("Cust Num :" + c1.contact\_num);

        System.out.println();

        Customer c2 = new Customer();

        System.out.println("Cust Num :" + c2.cust\_num);

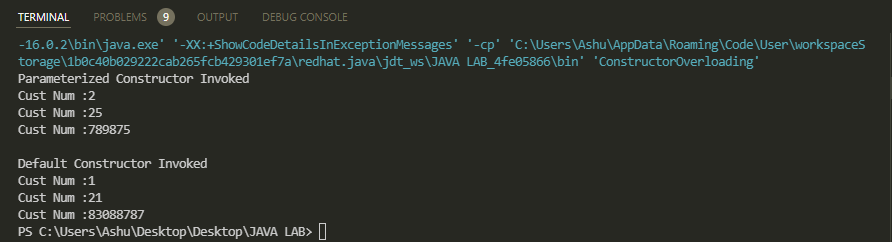
        System.out.println("Cust Num :" + c2.cust\_age);

        System.out.println("Cust Num :" + c2.contact\_num);

    }

}

**Output :**



1. Consider first 15 natural numbers and Find out the sum of prime numbers amongst them.

**Code :**

import java.util.\*;

public class SumOfPrimeNo {

    public static *void* main(String[] *args*) {

*int* n, sum = 0, i = 1, j;

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter n");

        n = sc.nextInt();

*int* temp = n;

        while (n != 0) {

*int* count = 0;

            for (j = 1; j <= i; j++) {

                if (i % j == 0) {

                    count++;

                }

            }

            if (count == 2) {

                sum = sum + i;

                n--;

            }

            i++;

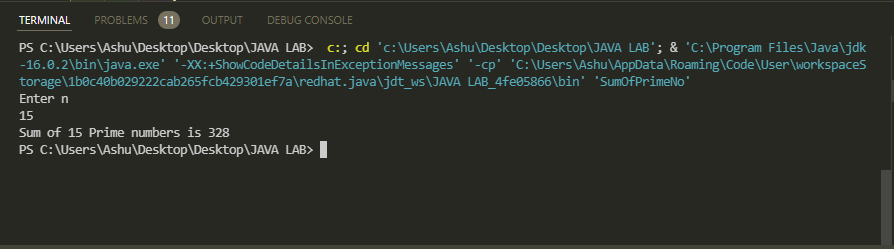
        }

        System.out.println("Sum of " + temp + " Prime numbers is " + sum);

    }

}

**Output :**



1. WAP to find out the area of a parallelogram with breadth and height. Use initialization block to initialize breadth and height.

**Code :**

public class AreaParallelogram {

*int* breadth, height;

    {

        breadth = 5;

        height = 7;

    }

*void* area() {

        System.out.println("Area of parallelogram = " + breadth \* height);

    }

    public static *void* main(String[] *args*) {

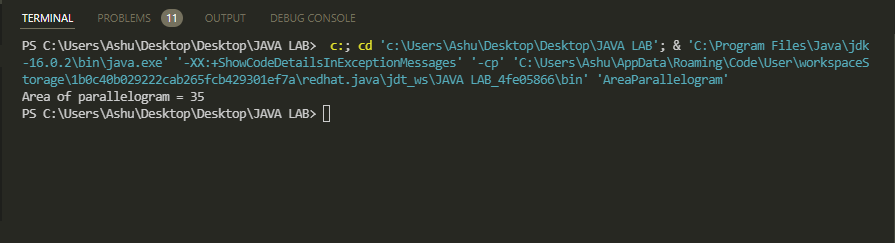
        AreaParallelogram a = new AreaParallelogram();

        a.area();

    }

}

**Output :**



1. Create a class to print the area of a square and a rectangle. Overload the area method for calculating the same.

**Code :**

class Area {

*void* findArea(*double* *side*) {

        System.out.println("Area of the Square: " + *side* \* *side* + " sq");

    }

*void* findArea(*double* *l*, *double* *b*) {

        System.out.println("Area of the rectangle: " + *l* \* *b* + " sq");

    }

}

public class AreaOfSquareRectangle {

    public static *void* main(String[] *args*) {

        Area a1 = new Area();

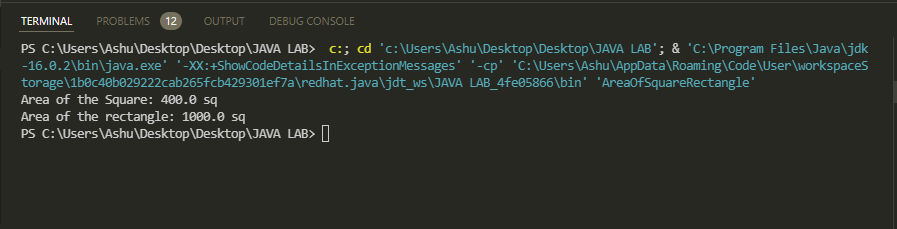
        a1.findArea(20);

        a1.findArea(40, 25);

    }

}

**Output :**



1. We have set of stars. The topmost row has 1 star, the next row down has 2 stars, the next row has 3 stars, and so on. Compute recursively (no loops or multiplication) the total number of stars to form a triangle with the given number of rows.

**Code :**

class PrintStarPatternWithoutLoop {

    // function to print a row

    static *void* printn(*int* *num*) {

        // base case

        if (*num* == 0) // stop recursion when num is 0

            return; // return value null, mean terminate the block

        else {

            printn(*num* - 1); // recursively calling pattern method

            System.out.print(" \* ");

        }

    }

    // function to print the pattern

    static *void* pattern(*int* *n*, *int* *i*) {

        // base case

        if (*n* == 0)

            return;

        printn(*i*);

        System.out.println();

        pattern(*n* - 1, *i* + 1); // recursively calling pattern method

    }

    public static *void* main(String[] *args*) {

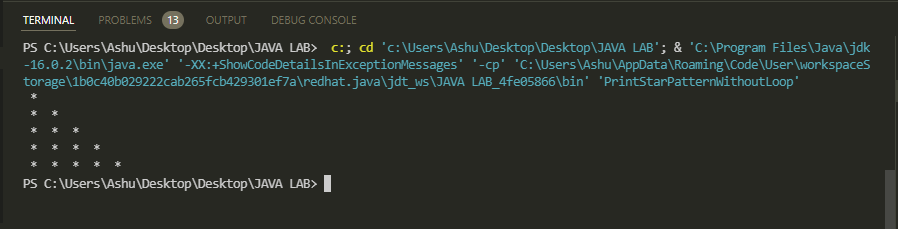
*int* n = 5;

        pattern(n, 1);

    }

}

**Output :**



1. Create a Car class and define nested Engine class. Call void start() method from engine class. Assume suitable fields and methods.

**Code :**

class car {

    class engine {

*void* start() {

            System.out.println("car engine is started succesfully.");

        }

    }

}

public class NestedCarClasseDemon {

    public static *void* main(String[] *args*) {

        car c = new car();

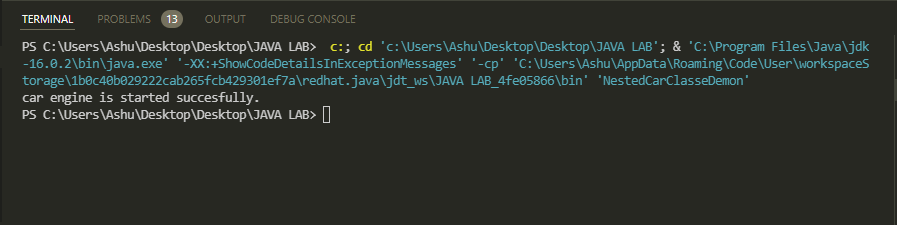
        car.engine e = c.new engine();

        e.start();

    }

}

**Output :**



1. Consider following:

abstract class Book{ String title;

abstract void setTitle(String s);

String getTitle()

{

return title;

}

}

Implement above abstract method in MyBook class. Call the method setTitle(“The Complete Reference”);

**Code :**

abstract class Book {

    String title;

    abstract *void* setTitle(String *s*);

}

class myBook extends Book {

*void* setTitle(String *s*) {

        title = *s*;

    }

    String getTitle() {

        return title;

    }

}

public class AbstractClassBook {

    public static *void* main(String[] *args*) {

        myBook b1 = new myBook();

        b1.setTitle("The Complete Reference");

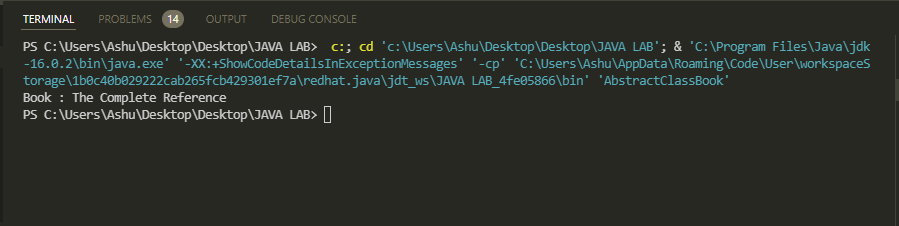
        b1.getTitle();

        System.out.println("Book : " + b1.title);

    }

}

**Output :**



1. Create an abstract class 'Bank' with an abstract method 'getBalance'. Rs.100, Rs.150 and Rs.200 are deposited in banks A, B and C respectively. 'BankA', 'BankB' and 'BankC' are subclasses of class 'Bank', each having a method named 'getBalance'. Call this method by creating an object of each of the three classes.

**Code :**

abstract class Bank {

    abstract *void* getBalance();

}

class BankA extends Bank {

*int* deposited = 100;

    public *void* getBalance() {

        System.out.println(deposited);

    }

}

class BankB extends Bank {

*int* deposited = 150;

    public *void* getBalance() {

        System.out.println(deposited);

    }

}

class BankC extends Bank {

*int* deposited = 200;

    public *void* getBalance() {

        System.out.println(deposited);

    }

}

public class AbstratClassBankDemonstrate {

    public static *void* main(String[] *args*) {

        BankA b1 = new BankA();

        b1.getBalance();

        BankB b2 = new BankB();

        b2.getBalance();

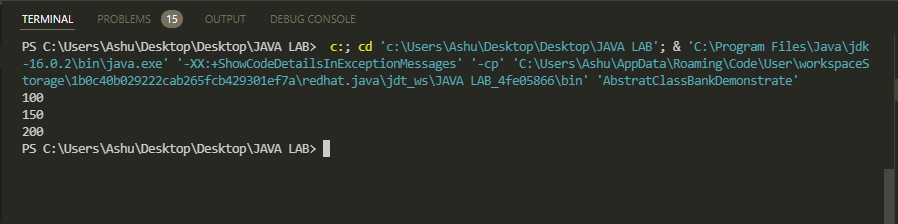
        BankC b3 = new BankC();

        b3.getBalance();

    }

}

**Output :**



1. Write a program to identify the maximum temperature among given 5 cities. (Consider Temperature class with temp variable. Hint: use Static)

**Code :**

import java.util.\*;

public class cityTemperature {

    String city\_nm;

*double* city\_Temp;

    static *double* max\_temp = 0;

    static *double* m\_temp = 0;

*void* getTempData() {

        Scanner in = new Scanner(System.in);

        System.out.print("Enter City Name : ");

        city\_nm = in.next();

        System.out.print("Enter Temperature : ");

        city\_Temp = in.nextDouble();

        System.out.print("");

    }

    public static *void* main(String[] *args*) {

        String city\_nm = "";

*int* i = 0;

        cityTemperature t[] = new cityTemperature[5]; // Created array of objects

        for (i = 0; i < 5; i++) {

            t[i] = new cityTemperature(); // Called Constructor for every object

            t[i].getTempData(); // Called to get city and temperature data

        }

        for (i = 0; i < 5; i++) {

            if (t[i].city\_Temp > max\_temp) {

                max\_temp = t[i].city\_Temp;

                city\_nm = t[i].city\_nm;

            }

        }

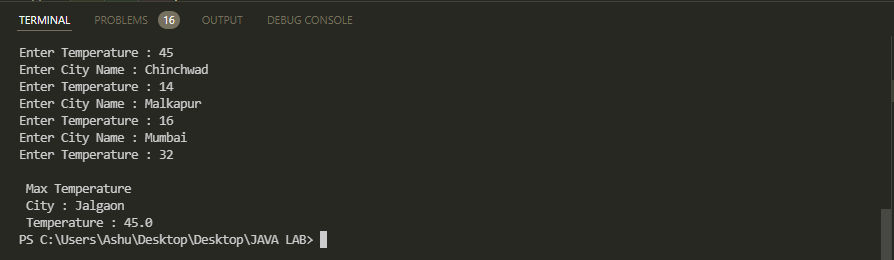
        // To Print The Maximum Temperature city name and temperature

        System.out.println("\n Max Temperature \n City : " + city\_nm + "\n Temperature : " + max\_temp);

    }

}

**Output :**



1. Declare the integer array with 10 numbers. Generate 2 new arrays Prime and NonPrime with prime and non-prime numbers from main array.

**Code :**

public class PrimeNonPrimeArray {

    public static *void* main(String[] *args*) {

*int* arr[] = { 2, 3, 5, 12, 45, 7, 89, 5, 4, 1 };

*int* cnt = 0;

*int* ncnt = 0;

*int* prime[] = new *int*[10];

*int* notprime[] = new *int*[10];

        for (*int* v : arr) {

*int* i, m = 0;

            m = v / 2;

            if (v == 0 || v == 1) {

                continue;

            } else {

                for (i = 2; i <= m;) {

                    if (v % i == 0) {

                        notprime[ncnt] = v;

                        ncnt++;

                        break;

                    } else {

                        prime[cnt] = v;

                        cnt++;

                        break;

                    }

                }

            }

        }

        System.out.print("Prime array : ");

        for (*int* r = 0; r < cnt; r++) {

            System.out.print(prime[r] + " ");

        }

        System.out.println("");

        System.out.print("Not Prime array : ");

        for (*int* r = 0; r < ncnt; r++) {

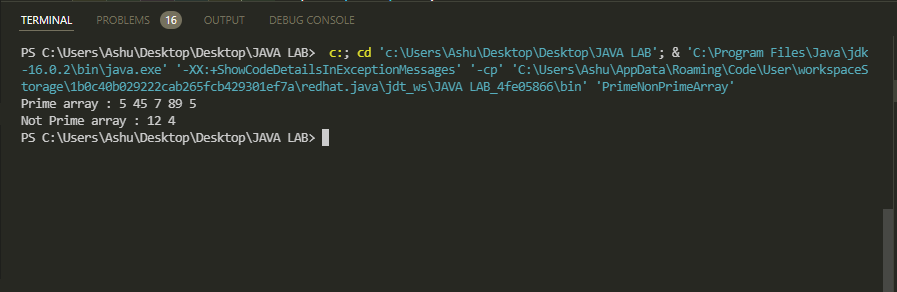
            System.out.print(notprime[r] + " ");

        }

    }

}

**Output :**



1. Write an application to identify and move all 0's to the end of an array. Maintain the sequence of the other (non-zero) array elements.

**Code :**

class MoveZeroToEndArray {

    static *void* moveZerosToEnd(*int* *arr*[], *int* *n*) {

*int* count = 0;

        for (*int* i = 0; i < *n*; i++) {

            if (*arr*[i] != 0)

*arr*[count++] = *arr*[i];

        }

        while (count < *n*)

*arr*[count++] = 0;

    }

    public static *void* main(String[] *args*) {

*int* arr[] = { 1, 9, 8, 4, 0, 0, 2, 7, 0, 6, 0, 9 };

*int* n = arr.length;

        moveZerosToEnd(arr, n);

        System.out.println("Array after pushing zeros to the end: ");

        for (*int* i = 0; i < n; i++) {

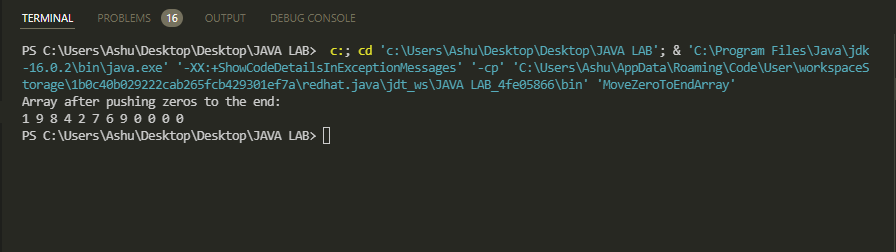
            System.out.print(arr[i] + " ");

        }

    }

}

**Output :**



1. Create Student(Roll, Name, City) array and retrieve the information of student who belongs to City Mumbai.

**Code :**

import java.util.\*;

class mumbaiStudents {

    String name, city;

*int* roll;

*void* setStudent() {

        Scanner in = new Scanner(System.in);

        System.out.print("Enter Name : ");

        name = in.next();

        System.out.print("Enter City : ");

        city = in.next();

        System.out.print("Enter roll : ");

        roll = in.nextInt();

        System.out.print("");

    }

    public static *void* main(String *args*[]) {

        String mCity = "mumbai";

        String cCity = "Mumbai";

        mumbaiStudents obj[] = new mumbaiStudents[2];

        for (*int* i = 0; i < 2; i++) {

            obj[i] = new mumbaiStudents();

            obj[i].setStudent();

        }

        for (*int* i = 0; i < 2; i++) {

            if (obj[i].city.equals(mCity) || obj[i].city.equals(cCity)) {

                System.out.println("Roll no = " + obj[i].roll + " Name = " + obj[i].name + " City = " + obj[i].city);

            } else {

                System.out.println("No Student is From Mumbai");

                break;

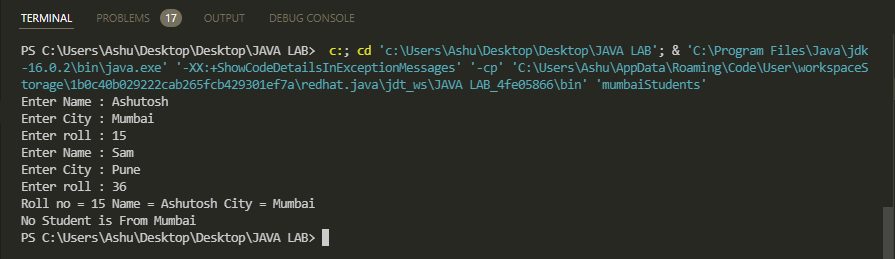
            }

        }

    }

}

**Output :**



1. Write a try-catch block to calculate the factorial of the entered number. An exception should be handled if the entered number is negative. Use the Exception class.

**Code :**

import java.util.Scanner;

class negativeNumberException extends Exception {

    public negativeNumberException(String *s*) {

        super(*s*);

    }

}

public class factorialException {

    public static *void* main(String[] *args*) {

*int* i, fact = 1;

        Scanner sc = new Scanner(System.in);

        try {

            System.out.println("Enter The Number : ");

*int* number = sc.nextInt();

            if (number < 0) {

                throw new negativeNumberException("Negative Number Is Entered");

            } else {

                for (i = 1; i <= number; i++) {

                    fact = fact \* i;

                }

                System.out.println("Factorial of " + number + " is: " + fact);

            }

        } catch (negativeNumberException e) {

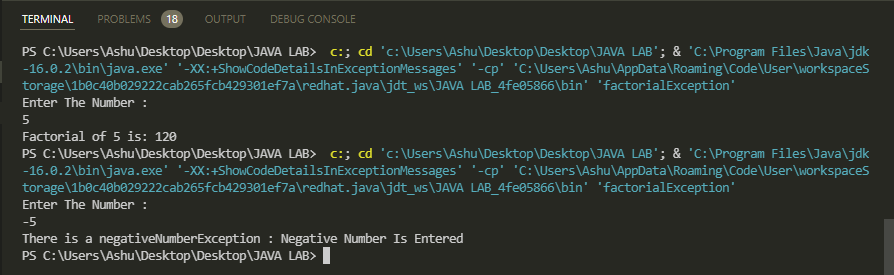
            System.out.println("There is a negativeNumberException : " + e.getMessage());

        }

    }

}

**Output :**



1. Write an application which will throw OverwtProductException if Product weight is above 60kg. (Use User defined exception)

**Code :**

import java.util.Scanner;

class OverwtProductException extends Exception {

    public OverwtProductException(String *s*) {

        super(*s*);

    }

}

public class overWeightException {

    public static *void* main(String *args*[]) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter Weight ");

*int* Weight = sc.nextInt();

        try {

            if (Weight >= 60) {

                throw new OverwtProductException("Weight is Above 60");

            }

            System.out.println("Success, The Weight is Accepted");

        } catch (OverwtProductException e) {

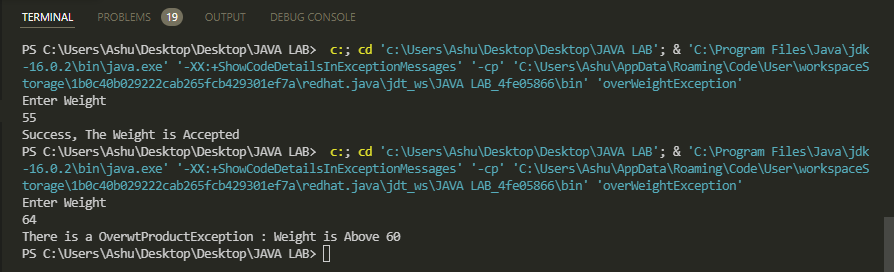
            System.out.println("There is a OverwtProductException : " + e.getMessage());

        }

    }

}

**Output :**



1. WAP to copy contents of one file into another, by removing extra blank spaces & comment marks.

**Code :**

import java.io.\*;

public class copyTwoFileToThird {

    public static *void* main(String[] *args*) throws IOException {

        FileOutputStream fos = new FileOutputStream("file3.txt");

        FileReader fr1 = new FileReader("file1.txt");

        FileReader fr2 = new FileReader("file2.txt");

        BufferedReader br = new BufferedReader(fr1);

        String f2line = br.readLine();

        // copy each line of file1.txt to file3.txt

        while (f2line != null) {

            fos.write(f2line.getBytes());

            f2line.replaceAll("\\s", "");

            f2line = br.readLine();

        }

        br = new BufferedReader(fr2);

        f2line = br.readLine();

        // copy each line of file2.txt to file3.txt

        while (f2line != null) {

            fos.write(f2line.getBytes());

            f2line = br.readLine();

        }

        System.out.println("Merged file1 and file2 to file3");

        fos.close();

        br.close();

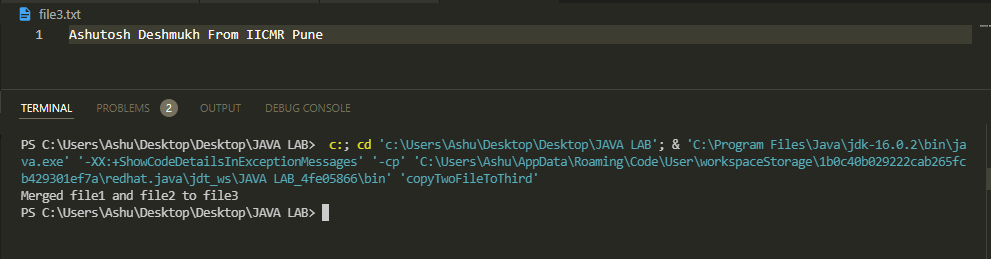
        fr1.close();

        fr2.close();

    }

}

**Output :**



1. Accept the String msg from User and display the string on console by removing duplicate characters.

**Code :**

import java.util.\*;

public class removeDuplicateChar {

    static String removeDuplicate(*char* *str*[], *int* *n*) {

*int* index = 0;

        for (*int* i = 0; i < *n*; i++) {

*int* j;

            for (j = 0; j < i; j++) {

                if (*str*[i] == *str*[j]) {

                    break;

                }

            }

            if (j == i) {

*str*[index++] = *str*[i];

            }

        }

        return String.valueOf(Arrays.copyOf(*str*, index));

    }

    public static *void* main(String[] *args*) {

*char* str[] = "Hello".toCharArray();

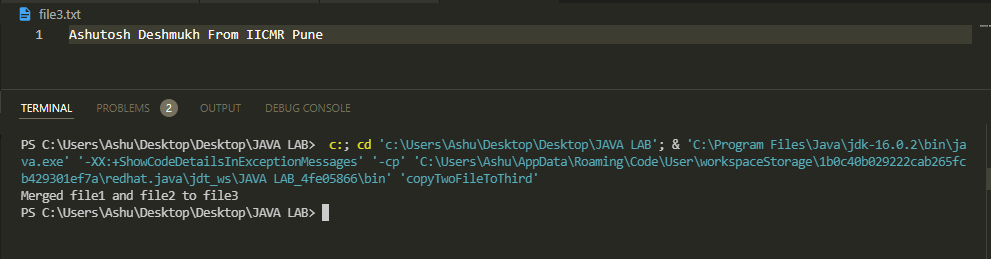
*int* n = str.length;

        System.out.println(removeDuplicate(str, n));

    }

}

**Output :**



1. Write a menu driven java program for the following:
2. Insert
3. Update
4. Delete
5. Search
6. Display
7. Exit

Consider Doctor (dr\_id, dr\_nm, specialization) table for this.

**Code :**

import java.io.DataInputStream;

import java.sql.\*;

public class menuDriven {

    public static *void* main(String[] *args*) {

        try {

            Class.forName("com.mysql.cj.jdbc.Driver");

            Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/IICMR", "root", "");

            Statement stmt = con.createStatement();

*int* opt;

            do {

                System.out.println("Please Select a option from below : ");

                System.out.println("1. Insert ");

                System.out.println("2. Update ");

                System.out.println("3. Delete ");

                System.out.println("4. Search ");

                System.out.println("5. Display ");

                System.out.println("6. Exit ");

                DataInputStream dis = new DataInputStream(System.in);

                opt = Integer.parseInt(dis.readLine());

                switch (opt) {

                    case 1:

                        System.out.println("You have selected " + opt);

                        System.out.println("Insert Doctor Details : ");

                        System.out.print("Enter Doctor ID :");

*int* drid = Integer.parseInt(dis.readLine());

                        System.out.println("Enter Doctor Name : ");

                        String nm = dis.readLine();

                        System.out.println("Enter Specialization : ");

                        String spec = dis.readLine();

                        String str = "INSERT INTO doctor VALUES (" + drid + ",'" + nm + "','" + spec + "')";

*int* result = stmt.executeUpdate(str);

                        System.out.println("Row Inserted..!!!");

                        break;

                    case 2:

                        System.out.print("Enter the Dr. Id: ");

*int* did = Integer.parseInt(dis.readLine());

                        System.out.print("Enter the Doctor Name : ");

                        nm = dis.readLine();

*int* k = stmt.executeUpdate("update doctor set dr\_nm='" + nm + "' where dr\_id=" + did);

                        if (k > 0) {

                            System.out.println("Record Updated..!!");

                        } else {

                            System.out.println("Record Not Found : ");

                        }

                        System.out.println("...............................................");

                        break;

                    case 3:

                        System.out.println("Delete : ");

                        String s = "Select \* from doctor";

                        ResultSet rs = stmt.executeQuery(s);

                        while (rs.next()) {

                            System.out.println("Doctor ID : " + rs.getInt(1));

                            System.out.println("Doctor Name : " + rs.getString(2));

                            System.out.println("Specialization : " + rs.getString(3));

                            System.out.println("");

                        }

                        System.out.println("Enter the ID of the doctor you want to delete : ");

*int* dl = Integer.parseInt(dis.readLine());

                        String strd = "delete from doctor where dr\_id = " + dl + "";

*int* str1 = stmt.executeUpdate(strd);

                        System.out.println("Row Deleted..!!!!");

                        break;

                    case 4:

                        System.out.println("Enter the ID of the doctor you want to search : ");

*int* is = Integer.parseInt(dis.readLine());

                        String sse = "Select \* from doctor where dr\_id =" + is;

                        ResultSet rs4 = stmt.executeQuery(sse);

                        if (rs4.next()) {

                            System.out.println("Doctor ID : " + rs4.getInt(1));

                            System.out.println("Doctor Name : " + rs4.getString(2));

                            System.out.println("Specialization : " + rs4.getString(3));

                            System.out.println("");

                        }

                        break;

                    case 5:

                        String sd = "Select \* from doctor";

                        ResultSet rs1 = stmt.executeQuery(sd);

                        while (rs1.next()) {

                            System.out.println("Doctor ID : " + rs1.getInt(1));

                            System.out.println("Doctor Name : " + rs1.getString(2));

                            System.out.println("Specialization : " + rs1.getString(3));

                            System.out.println("");

                        }

                    case 6:

                        break;

                }

            } while (opt != 6);

        } catch (Exception e) {

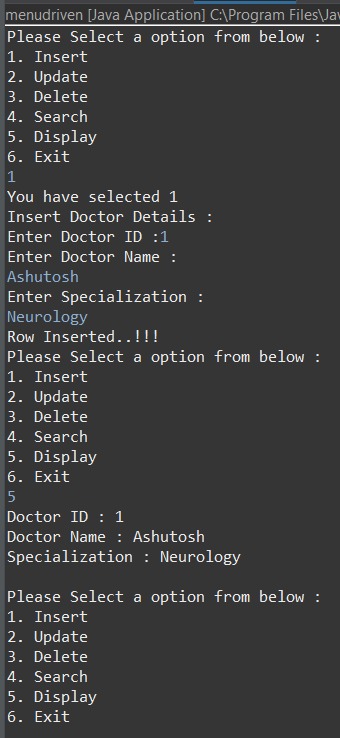
            System.out.println("Error : " + e);

        }

    }

}

**Output :**



1. WAP to accept book number from user. If book is present, display book details & ask user whether he wants to delete the book details. If user says ‘Yes’ delete the record

**Code :**

import java.sql.\*;

import java.io.\*;

public class 22

{

    public static *void* main(*String* *arg*[])

    {

*int* id,phno;

*String* nm,add;

        try        {

*DataInputStream* din=new DataInputStream(System.in);

            System.out.println("Enter information of Customer.");

            System.out.println("Enter ID of Customer.");

            id=Integer.parseInt(din.readLine());

            System.out.println("Enter Name of Customer.");

            nm=din.readLine();

            System.out.println("Enter Address of Customer.");

            add=din.readLine();

            System.out.println("Enter Contact\_No of Customer.");

            phno=Integer.parseInt(din.readLine());

            Class.forName("com.mysql.cj.jdbc.Driver");

*Connection* con=DriverManager.getConnection("jdbc:mysql://localhost:3306/IICMR","root","");

*PreparedStatement* pst=con.prepareStatement("insert into Customervalues values(?,?,?,?)");

            pst.setInt(1,id);

            pst.setString(2,nm);

            pst.setString(3,add);

            pst.setInt(4,phno);

*int* n=pst.executeUpdate();

            if(n>0)

            {

                 System.out.println("Record is Inserted...!!!");

            }

            else

            {

                System.out.println("Error...!!!");

            }

            pst.close();

            con.close();

        }

        catch(*Exception* *e*)

        {

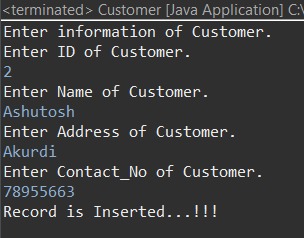
            System.out.println(e);

        }

    }

}

**Output :**

****

1. Write a JDBC program to accept the details of customer (CID, CName, Address, Ph\_No) and store it into the database (Use PreparedStatement interface)

**Code :**

import java.io.DataInputStream;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

public class 23 {

    public static *void* main(*String*[] *args*) throws *Exception* {

*DataInputStream* dis= new DataInputStream(System.in);

        Class.forName("com.mysql.cj.jdbc.Driver");

*Connection* con = DriverManager.getConnection("jdbc:mysql://localhost:3306/book","root","");

*Statement* smt = con.createStatement();

        System.out.println("Enter the name of the book you want to search : ");

*String* nm = dis.readLine();

*String* s="select \* from book where bk\_nm = '"+nm+"'";

*ResultSet* rs = smt.executeQuery(s);

        if(rs.next()==true)

        {

            System.out.println("Do you want to delete the book record : ");

*String* ans=dis.readLine();

            if(ans.equals("yes"))

            {

*String* s1="DELETE FROM `book` WHERE `bk\_nm`='I too had a Love Story'";

*int* i= smt.executeUpdate(s1);

            if(i>0)

            {

                System.out.println(i+" Record Deleted. ");

            }

            }

        }

    }

}

**Output :**

