# 20MCA132 OBJECT ORIENTED PROGRAMMING LAB

*Lab Report Submitted By*

**SALINI K B**

**Reg. No.: AJC21MCA-2091**

*In Partial fulfillment for the Award of the Degree Of*

**MASTER OF COMPUTER APPLICATIONS (2 Year)**
**(MCA)**

**APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**


**AMAL JYOTHI COLLEGE OF ENGINEERING**
**KANJIRAPPALLY**

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovapally, Kanjirappally, Kottayam, Kerala – 686518]

**2021-2023**

## DEPARTMENT OF COMPUTER APPLICATIONS

## AMAL JYOTHI COLLEGE OF ENGINEERING
## KANJIRAPPALLY

# CERTIFICATE

This is to certify that the lab report, "**20MCA132 OBJECT PROGRAMMING LAB"** is the bonafide work of **SALINI K B (AJC21MCA-2091)** in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2021-23.

**MS. Gloriya Mathew**
**Lab In-Charge**

**Rev.Fr.Dr.Rubin Thottupuram Jose**
**Head of the Department**

**Internal Examiner**

**External Examiner**

# CONTENT

| | | | |
|---|---|---|---|
| **10** | Area of different shapes using overloaded functions | **17-05-2022** | **31-32** |
| **11** | Create a class 'Employee' with data members Empid, Name, Salary, Address and constructors to initialize the data members. Create another class 'Teacher' that inherit the properties of class employee and contain its own data members department, Subjects taught and constructors to initialize these data members and also include display function to display all the data members. Use array of objects to display details of N teachers. | **17-04-2022** | **33-35** |
| **12** | Create a class 'Person' with data members Name, Gender, Address, Age and a constructor to initialize the data members and another class 'Employee' that inherits the properties of class Person and also contains its own data members like Empid, Company_name, Qualification, Salary and its own constructor. Create another class 'Teacher' that inherits the properties of class Employee and contains its own data members like Subject, Department, Teacherid and also contain constructors and methods to display the data members. Use array of objects to display details of N teachers | **24-05-2022** | **36-38** |
| **13** | Write a program has class Publisher, Book, Literature and Fiction. Read the information and print the details of books from either the category, using inheritance. | **17-05-2022** | **39-42** |
| **14** | Create classes Student and Sports. Create another class Result inherited from Student and Sports. Display the academic and sports score of a student. | **17-05-2022** | **43-46** |
| **15** | Create an interface having prototypes of functions area() and perimeter(). Create two classes Circle and Rectangle which implements the above interface. Create a menu driven program to find area and perimeter of objects.. | **24-05-2022** | **47-49** |

| | | | |
|---|---|---|---|
| **16** | Prepare bill with the given format using calculate method from interface. | **24-05-2022** | **50-52** |
| **17** | Create a Graphics package that has classes and interfaces for figures Rectangle, Triangle, Square and Circle. Test the package by finding the area of these figures | **31-05-2022** | **53-55** |
| **18** | Write a user defined exception class to authenticate the user name and password. | **31-05-2022** | **56-58** |
| **19** | Find the average of N positive integers, raising a user defined exception for each negative input. | **31-05-2022** | **59-61** |
| **20** | Define 2 classes; one for generating Fibonacci numbers and other for displaying even numbers in a given range. Implement using threads. (Runnable Interface). | **31-05-2022** | **62-64** |
| **21** | Program to create a generic stack and do the Push and Pop operations. | **31-05-2022** | **65-67** |
| **22** | Maintain a list of Strings using ArrayList from collection framework, perform built-in operations | **31-05-2022** | **68-70** |
| **23** | Program to demonstrate the creation of queue object using the PriorityQueue class | **07-06-2022** | **71-73** |
| **24** | Program to demonstrate the addition and deletion of elements in deque. | **07-06-2022** | **74-75** |
| **25** | Write a Java program to compare two hash set | **07-06-2022** | **76-78** |

| | | | |
|---|---|---|---|
| **26** | Program to demonstrate the working of Map interface by adding, changing and removing Elements. | **07-06-2022** | **79-81** |
| **27** | Program to find maximum of three numbers using AWT. | **09-06-2022** | **82-83** |
| **28** | Implement a simple calculator using AWT components | **09-06-2022** | **84-85** |
| **29** | Develop a program to handle all mouse events and window events | **29-05-2022** | **86-87** |
| **30** | Develop a program to handle Key events. | **29-05-2022** | **88-89** |
| **31** | Write a program to write to a file, then read from the file and display the contents on the console. | **30-05-2022** | **90-91** |
| **32** | Write a program to copy one file to another | **30-05-2022** | **92** |
| **33** | Write a program that reads from a file having integers. Copy even numbers and odd numbers to separate files. | **30-05-2022** | **93** |

**Name: salini k.b**

**Roll No:33**

**Batch: MCA-B**

**Date:26/03/2022**

## Experiment no:1

**Aim:** Define a class 'product' with data members pcode, pname and price. Create 3 objects of the class and find the product having the lowest price.

**Source code:**

```
class Product
{
 String pcode,pname;
 double price;
 void details()
    {
    System.out.println("Product Details");
    System.out.println("PCode:"+pcode);
    System.out.println("PName:"+pname);
    System.out.println("Price:"+price);
    }
}
public class ProductDetails
    {
    public static void main (String args[])
    {
```

```
Product p1 = new Product();
p1.pcode = "THN2134PI";
p1.pname = "REDALME";
p1.price = 11000;
System.out.println("\nProduct1:");
p1.details();

Product p2 = new Product();
p2.pcode = "WECV872WE";
p2.pname = "Bluetooth";
p2.price = 1500;
System.out.println("\nProduct2:");
p2.details();

Product p3 = new Product();
p3.pcode = "QOI43DT6";
p3.pname = "SMART WATCH";
p3.price = 1000;
System.out.println("\nProduct3:");
p3.details();

if(p1.price<p2.price&&p1.price<p3.price)
{
```

```
        System.out.println("\n\nProduct with lowest price is:");
        p1.details();
       }
       else if(p2.price<p3.price)
       {
        System.out.println("\n\nProduct with lowest price is:");
        p2.details();
       }
       else
       {
        System.out.println("\n\nProduct with lowest price is:");
        p3.details();
       }
    }
}
```

## Output:

```
C:\Users\Student.U33\Documents\salu>javac ProductDetails.java

C:\Users\Student.U33\Documents\salu>java ProductDetails

Product1:
Product Details
PCode:weN2134PI
PName:REdME
Price:1000.0

Product2:
Product Details
PCode:hECV872WE
PName:wifi
Price:100.0

Product3:
Product Details
PCode:OI43T6
PName: WATCH
Price:100.0


Product with lowest price is:
Product Details
PCode:OI43T6
PName: WATCH
Price:100.0

C:\Users\Student.U33\Documents\salu>
```

**Name: salini K.b**

**Roll No:33**

**Batch: MCA-B**

**Date:26/03/2022**

## Experiment No.: 2

## Aim

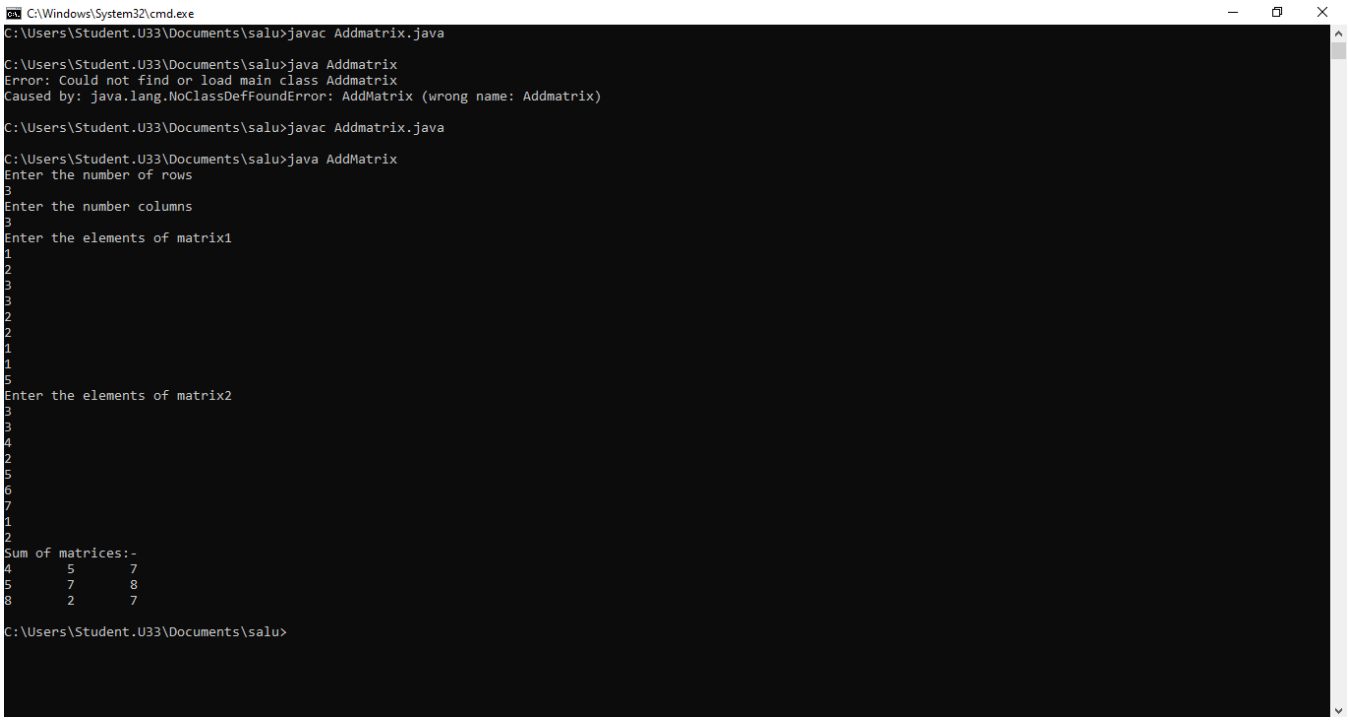
Read 2 matrices from the console and perform matrix addition.

## Procedure

```
import java.util.*;
class AddMatrix
{
public static void main(String args[])
{
int row, col,i,j;
Scanner in = new Scanner(System.in);
System.out.println("Enter the number of rows");
row = in.nextInt();
System.out.println("Enter the number columns");
col = in.nextInt();
int mat1[][] = new int[row][col];
int mat2[][] = new int[row][col];
int res[][] = new int[row][col];
System.out.println("Enter the elements of matrix1");
for ( i= 0 ; i < row ; i++ )
{
for ( j= 0 ; j < col ;j++ )
mat1[i][j] = in.nextInt();
}
System.out.println("Enter the elements of matrix2");
for ( i= 0 ; i < row ; i++ )
{
for ( j= 0 ; j < col ;j++ )
mat2[i][j] = in.nextInt();
}
for ( i= 0 ; i < row ; i++ )
```

```
for ( j= 0 ; j < col ;j++ )
res[i][j] = mat1[i][j] + mat2[i][j] ;
System.out.println("Sum of matrices:-");
for ( i= 0 ; i < row ; i++ )
{
for ( j= 0 ; j < col ;j++ )
System.out.print(res[i][j]+"\t");
System.out.println();
}
}
}
```

**Output Screenshot**



**Name: salini Kb**

**Roll No:33**

**Batch: MCA-B**

**Date:26/03/2022**

**Experiment No.: 3**

**Aim**

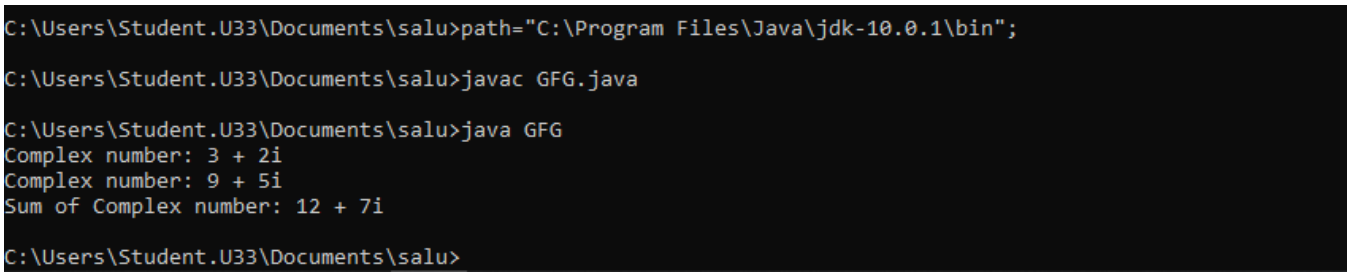
Program to add complex numbers

**Procedure**

```
import java.util.*;
class Complex {
        int real, imaginary;
        Complex(){}
        Complex(int tempReal, int tempImaginary)
        {
                real = tempReal;
                imaginary = tempImaginary;
        }
        Complex addComp(Complex C1, Complex C2)
        {
                Complex temp = new Complex();
                temp.real = C1.real + C2.real;
                temp.imaginary = C1.imaginary + C2.imaginary;
                return temp;
        }
        Complex subtractComp(Complex C1, Complex C2)
        {
                Complex temp = new Complex();
                temp.real = C1.real - C2.real;
                temp.imaginary = C1.imaginary - C2.imaginary;
                return temp;
        }
        void printComplexNumber()
        {
                System.out.println("Complex number: "
                                        + real + " + "
                                        + imaginary + "i");
```

```
        }}
public class GFG {
        public static void main(String[] args)
        {
                Complex C1 = new Complex(3, 2);
                C1.printComplexNumber();
                Complex C2 = new Complex(9, 5);
                C2.printComplexNumber();
                Complex C3 = new Complex();
                C3 = C3.addComp(C1, C2);
                System.out.print("Sum of ");
                C3.printComplexNumber();
        }}
```

**Output Screenshot**



**Name: salini Kb**

**Roll No:33**

**Batch: MCA-B**

**Date:26/03/2022**

## Experiment No.: 4

## Aim

Read a matrix from the console and check whether it is symmetric or not.

## Procedure

```
import java.util.*;
public class Symetric {
	static void checkSymmetric(int mat[][], int row,int col)
	{
		int i, j, flag = 1;
		System.out.println("The matrix formed is:");
		for (i = 0; i < row; i++) {
			for (j = 0; j < col; j++) {
				System.out.print(mat[i][j] + "\t");
			}
			System.out.println("");
		}
		int[][] transpose = new int[row][col];
		for (i = 0; i < row; i++) {
			for (j = 0; j < col; j++) {
				transpose[j][i] = mat[i][j];
			}
		}
		if (row == col) {
			for (i = 0; i < row; i++) {
				for (j = 0; j < col; j++) {
					if (mat[i][j] != transpose[i][j]) {
						flag = 0;
						break;
					}
				}
				if (flag == 0) {
					System.out.print("\nThe matrix is not
symmetric");
					break;
				}
			}
```

```
                    if (flag == 1) {
                        System.out.print("\nThe matrix is symmetric");
                    }
                }
                else {
                    System.out.print("\nThe matrix is not symmetric");
                }
        }
        public static void main(String args[])
        {
                Scanner sc = new Scanner(System.in);
                int i, j, row, col, flag = 1;
                System.out.print("Enter the number of rows:");
                row = sc.nextInt();
                System.out.print("Enter the number of columns:");
                col = sc.nextInt();
                int[][] mat = new int[row][col];
                System.out.println("Enter the matrix elements:");
                for (i = 0; i < row; i++) {
                    for (j = 0; j < col; j++) {
                        mat[i][j] = sc.nextInt();
                    }
                }
                checkSymmetric(mat, row, col);
        }
}
```

**Output Screenshot**

```
C:\Users\Student.U33\Documents\salu>path="C:\Program Files\Java\jdk-10.0.1\bin";

C:\Users\Student.U33\Documents\salu>javac Symetric.java

C:\Users\Student.U33\Documents\salu>java Symetric
Enter the number of rows:2
Enter the number of columns:2
Enter the matrix elements:
1
2
3
1
The matrix formed is:
1       2
3       1

The matrix is not symmetric
C:\Users\Student.U33\Documents\salu>javac Symetric.java
```

```
C:\Users\Student.U33\Documents\salu>javac Symetric.java

C:\Users\Student.U33\Documents\salu>java Symetric
Enter the number of rows:2
Enter the number of columns:2
Enter the matrix elements:
1
1
1
1
The matrix formed is:
1       1
1       1

The matrix is symmetric
C:\Users\Student.U33\Documents\salu>
```

**Name: salini Kb**

**Roll No:33**

**Batch: MCA-B**

**Date:26/03/2022**

## Experiment No.: 5

## Aim

1]Aim:Create cpu with attribute price create innerclass processor (number of core manufacturer and static nested class RAM (Memory ,manufacturer).

Program:

```
class Cpu
{
 int price=7000;
  class Processor
   {
    int nop=10;
            String manf="jack";
   }
   static class Ram
   {
   static String memory="5GB";
   String manf="Antony";
   }
}
public class Computer
{
public static void main (String args[])
{
Cpu obj1=new Cpu();
Cpu.Processor obj2=obj1.new Processor();
Cpu.Ram obj3= new Cpu.Ram();
System.out.println("Cpu price:"+obj1.price);
```

```
System.out.println("Number of processors:"+obj2.nop);

System.out.println("Processor Manufacturer:"+obj2.manf);

System.out.println("Ram memory:"+Cpu.Ram.memory);

System.out.println("Ram manufacturer"+obj3.manf);

}

}
```

```
C:\Users\Student.U33\Documents\salu>javac Computer.java

C:\Users\Student.U33\Documents\salu>java Computer
Cpu price:7000
Number of processors:10
Processor Manufacturer:jack
Ram memory:5GB
Ram manufacturerAntony

C:\Users\Student.U33\Documents\salu>
```

**Name: salini Kb**

**Roll No:33**

**Batch: MCA-B**

**Date:26/03/2022**

## Experiment No.: 6

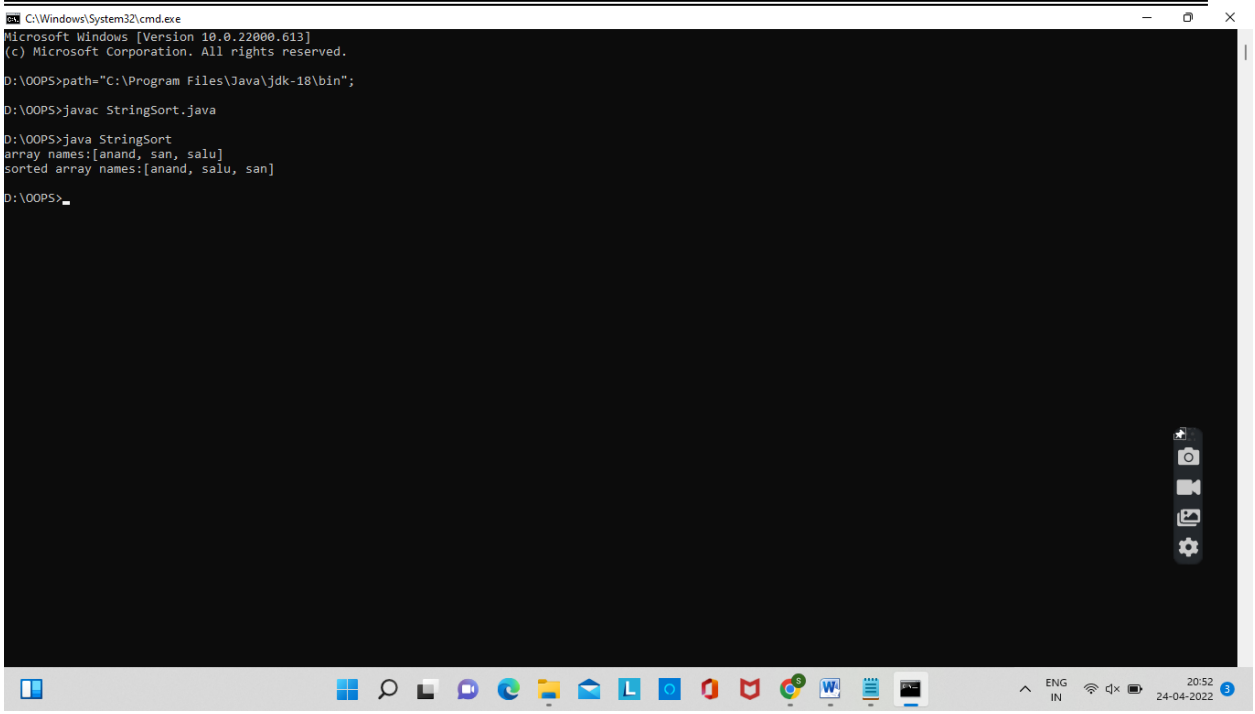
## Aim

Program to Sort strings

## Procedure

```
import java.util.Arrays;
public class StringSort
{
	public static void main(String args[])
	{
		String [] names={"anand","san","salu"};
		System.out.println("array names:"+Arrays.toString(names));

			Arrays.sort(names);
			System.out.println("sorted array
names:"+Arrays.toString(names));
	}
}
```

## Output Screenshot


C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.613]
(c) Microsoft Corporation. All rights reserved.
D:\OOPS>path="C:\Program Files\Java\jdk-18\bin";
D:\OOPS>javac StringSort.java
D:\OOPS>java StringSort
array names:[anand, san, salu]
sorted array names:[anand, salu, san]
D:\OOPS>
ENG
IN
20:52
24-04-2022

## Experiment No.: 7

**Name: Salini kb**

**Roll No:33**

**Batch: MCA-B**

**Date:26/03/2022**

## Aim

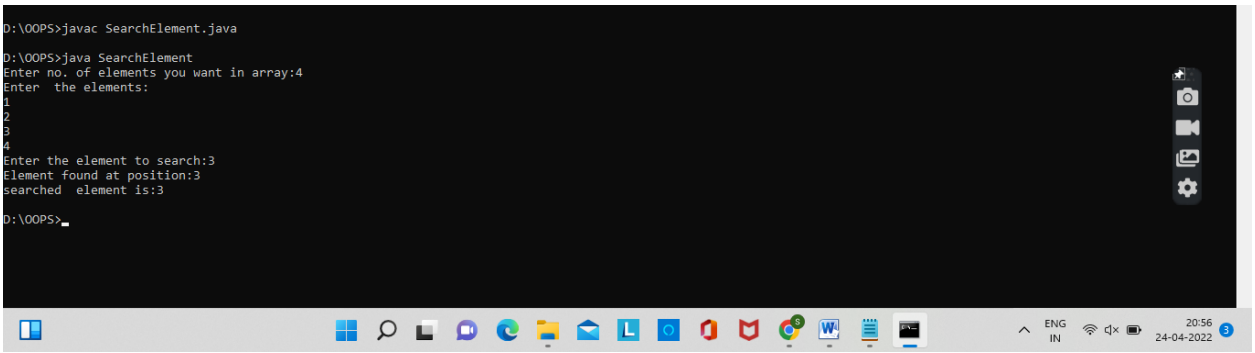
Search an element in an array

## Procedure

```
import java.util.Scanner;
public class SearchElement
{
   public static void main(String[] args)
   {
      int n, element, flag = 0, i = 0;
      Scanner s = new Scanner(System.in);
      System.out.print("Enter no. of elements you want in array:");
      n = s.nextInt();
      int a[] = new int[n];
      System.out.println("Enter  the elements:");
      for(i = 0; i < n; i++)
      {
         a[i] = s.nextInt();
      }
      System.out.print("Enter the element to search:");
      element = s.nextInt();
      for(i = 0; i < n; i++)
      {
         if(a[i] == element)
         {
            flag = 1;
            break;
```

```
                    }
            else
            {
                flag = 0;
            }
        }
        if(flag == 1)
        {
          System.out.println("Element found at position:"+(i +
          1)+""+"\n"+"searched          element is:"+element);
        }
        else
        {
            System.out.println("Element not found");
        }
    }
}
```

## Output Screenshot



## Experiment No.: 8

**Name: salini Kb**

**Roll No:33**

**Batch: MCA-B**

**Date:26/03/2022**

## Aim

Perform string manipulations

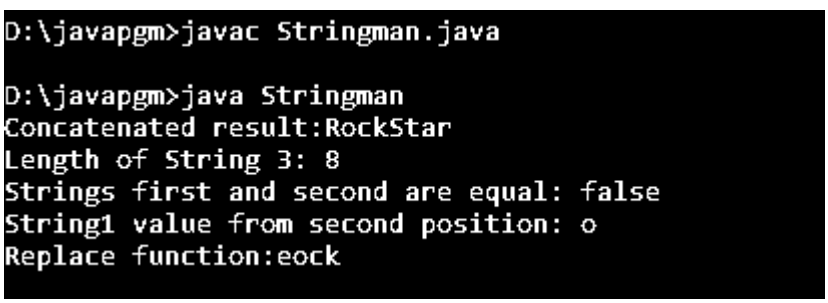
## Procedure

## Procedure

```
import java.util.Scanner;
import java.lang.*;
public class Man
{
public static void main(String [] args)
{
        int a;
       String b,c;
            Scanner sc = new Scanner(System.in);
       System.out.print(" Enter the string : ");
        b = sc.nextLine();
        while(true)
        {
                System.out.println("\nMENU:\n 1.String Length.\n 2.Uppercase.\n
3.Lowercase.\n 4.Concatenate.\n 5.Character index.\n6.Exit.");
                System.out.print("\n Enter your option : ");
                a = sc.nextInt();
                switch(a)
                {
                        case 1:System.out.println(" String length = "+b.length());
                        break;
                        case 2:System.out.println(" String in uppercase = "+b.toUpperCase());
                        break;
```

```
                    case 3:System.out.println(" String in lowercase = "+b.toLowerCase());
                    break;
                    case 4:{ System.out.print(" Enter the string to be concatenate = ");
                    c = sc.next();
                    System.out.println(" Concatenated string = "+b.concat(c));
                    break;
                    }
                    case 5:{ System.out.print(" Enter the Character to be searched in the given
string = ");
                    c = sc.next();
                    System.out.println("The character is found at"+(b.indexOf(c)+1)+ ".");
                    break;
                    }
                    case 6: System.exit(0);
                    }
                }
            }
        }
```

## Output Screenshot



**Name: salini kb**

**Roll No:33**

**Batch: MCA-B**

**Date:26/03/2022**

## Experiment No.: 9

## Aim

9.Program to create a class for Employee having attributes eNo, eName eSalary. Read n employ information and Search for an employee given eNo, using the concept of Array of Objects.

## Procedure

```
import java.util.*;
class Employe1{
    int eno;
    String ename;
    float esalary;
   public void emp1(){
        Scanner s=new Scanner(System.in);
        System.out.println("Enter  employee number:");
       eno=s.nextInt();
       System.out.println("Enter employee name:");
       ename=s.next();
       System.out.println("Enter  employe salary:");
       esalary=s.nextFloat();
 }
public void display(){
      System.out.println("Employe no:"+eno);
      System.out.println("Employe name:"+ename);
      System.out.println("Employe salary:"+esalary);
}
public static void main(String args[]){
```

```
int n;
Scanner sc=new Scanner(System.in);
System.out.println("Enter number of employees");
n=sc.nextInt();
Employe1 obj[]=new Employe1[n];
for(int i=0;i<n;i++)
{
 obj [i]=new Employe1();
 obj[i].emp1();
}
System.out.println(".....Employee Details.....");
for(int i=0;i<n;i++)
{
   obj[i].display();
}
int x;
System.out.println("Enter the number to search employee");
x=sc.nextInt();
int flag=0,i;
for(i=0;i<n;i++)
{
   if(obj[i].eno==x)
     {
        flag=1;
        break;
     }
   else
    {
      flag=0;
    }
}
```

```
  if(flag==1){
      obj[i].display();
  }
  else
    {
      System.out.println("Not found");
     }
 }
}
```

## Output Screenshot

```
D:\javapgm>javac Employe.java

D:\javapgm>java Employe
Enter the number of Employees:3
Enter the Employee Number:100
Enter the Employee Name:riya
Enter the Employee Salary:50000
Enter the Employee Number:101
Enter the Employee Name:Mary
Enter the Employee Salary:60000
Enter the Employee Number:102
Enter the Employee Name:Sona
Enter the Employee Salary:30000
.............Employee Details.............
 Employee Number  : 100
 Employee Name    : riya
 Employee Salary  : 50000.0
 Employee Number  : 101
 Employee Name    : Mary
 Employee Salary  : 60000.0
 Employee Number  : 102
 Employee Name    : Sona
 Employee Salary  : 30000.0
enter the number to search an employee
101
 Employee Number  : 101
 Employee Name    : Mary
 Employee Salary  : 60000.0

D:\javapgm>_
```

**Name: salini kb**

**Roll No:33**

**Batch: MCA-B**

**Date:26/03/2022**

## Experiment No.: 10

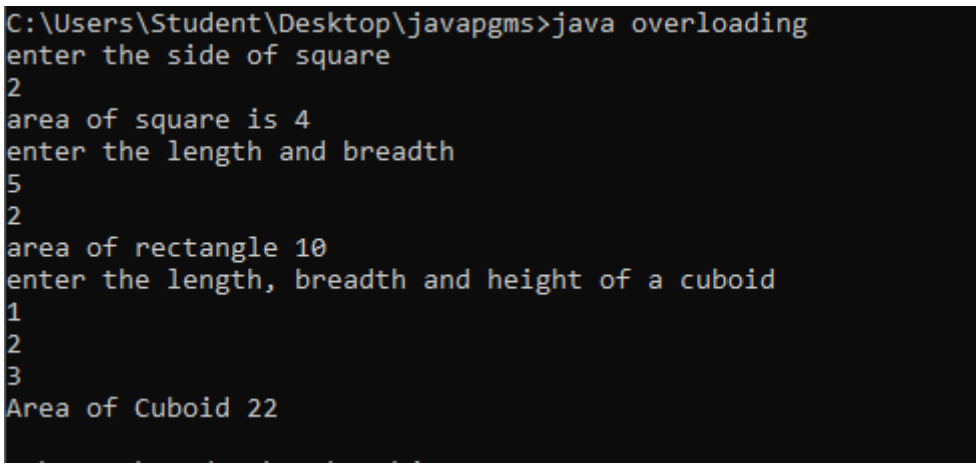
## Aim

Area of different shapes using overloaded functions

```
import java.util.Scanner;

class areaShapes{
    void area(int a){
        System.out.println("area of square is "+a*a);
    }
    void area(int a, int b){
        System.out.println("area of rectangle "+a*b);
    }
    void area(int length, int breadth, int height){
        System.out.println("Area of Cuboid
"+(2*(length*breadth)+2*(length*height)+2*(height*breadth)));

    }
}

public class Shapes {
    public static void main(String[] args)  {
        int a,b,c;
        Scanner s= new Scanner(System.in);
        areaShapes obj=new areaShapes();
        System.out.println("enter the side of square");
        a= s.nextInt();
```

```
        obj.area(a);
        System.out.println("enter the length and breadth");
        a=s.nextInt();
        b=s.nextInt();
        obj.area(a,b);
        System.out.println("enter the length, breadth and height of a cuboid");
        a=s.nextInt();
        b=s.nextInt();
        c=s.nextInt();
        obj.area(a,b,c);
    }
}
```

## Output Screenshot



**Name: salini kb**

**Roll No:33**

**Batch: MCA-B**

**Date:26/03/2022**

## Experiment No.: 11

## Aim

Create a class 'Employee' with data members Empid, Name, Salary, Address and constructors to initialize the data members. Create another class 'Teacher' that inherit the properties of class employee and contain its own data member's department, Subjects taught and constructors to initialize these data members and also include display function to display all the data members. Use array of objects to display details of N teachers.

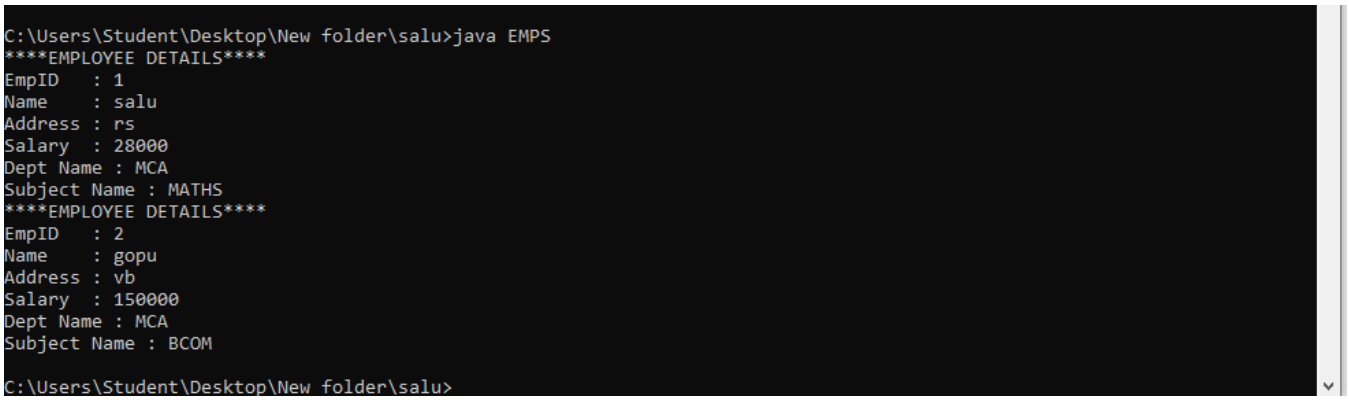
## Procedure

```
class EMPS{

   public static void main(String[] args) {
     Teacher tobj[] = new Teacher[2];
     tobj[0] = new Teacher("101","Rekha","Rosevilla",50000,"MCA","DS");
     tobj[1] = new
Teacher("102","Riya","Deepalayam",110000,"BBA","Commerce");
     tobj[0].display();
     tobj[1].display();
   }
}

class Employees {
   String Empid;
   String Name;
```

```
    String Address;
    int Salary;

    Employees(String id,String name,String addr,int salary){
        this.Empid   = id;
        this.Name    = name;
        this.Address = addr;
        this.Salary  = salary;
    }
    void display(){
        System.out.println("EmpID   : " + this.Empid);
        System.out.println("Name    : " + this.Name);
        System.out.println("Address : " + this.Address);
        System.out.println("Salary  : " + this.Salary);
    }
}


class Teacher extends Employees{
    String Department;
    String Subject;
    Teacher(String id,String name,String addr,int salary,String dept,String subj){
        super(id,name,addr,salary);
        this.Department=dept;
        this.Subject=subj;
    }
    void display(){

        System.out.println("****EMPLOYEE DETAILS****");
        super.display();
```

```
        System.out.println("Dept Name : " + this.Department);
        System.out.println("Subject Name : " + this.Subject);
    }
}
```

## Output Screenshot



**Name: salini kb**

**Roll No:33**

**Batch: MCA-B**

**Date:26/03/2022**

## Experiment No.: 12

## Aim

Create a class 'Person' with data members Name, Gender, Address, Age and a constructor to initialize the data members and another class 'Employee' that inherits the properties of class Person and also contains its own data members like Empid, Company_name, Qualification, Salary and its own constructor. Create another class 'Teacher' that inherits the properties of class Employee and contains its own data members like Subject, Department, Teacherid and also contain constructors and methods to display the data members. Use array of objects to display details of N teachers.

## Procedure

```
import java.util.Scanner;

class Person {
   String name, gender, address;
   int age;

   public Person(String name, String gender, String address, int age) {
      this.name = name;
      this.gender = gender;
      this.address = address;
      this.age = age;
   }
}

class Employee extends Person {

   int empid;
   double salary;
   String company_name, qualification;
```

```
   public Employee(String name, String gender, String address, int age, int
empid, String company_name,
         String qualification, double salary) {
      super(name, gender, address, age);
      this.empid = empid;
      this.company_name = company_name;
      this.qualification = qualification;
      this.salary = salary;
   }
}

class Teacher extends Employee {

   int teacher_id;
   String department, subject;

   public Teacher(String name, String gender, String address, int age, int empid,
String company_name,
         String qualification, double salary, int teacher_id, String department,
String subject) {
      super(name, gender, address, age, empid, company_name, qualification,
salary);
      this.teacher_id = teacher_id;
      this.department = department;
      this.subject = subject;
   }

   void displayDetails(String emp) {
      System.out.println("The name of the " + emp + " is: " + this.name);
      System.out.println("The gender of the " + emp + " is: " + this.gender);
      System.out.println("The address of the " + emp + " is: " + this.address);
      System.out.println("The age of the " + emp + " is: " + this.age);
      System.out.println("The employee ID of the " + emp + " is: " + this.empid);
      System.out.println("The Company name of the " + emp + " is: " +
this.company_name);
      System.out.println("The qualification of the " + emp + " is: " +
this.qualification);
      System.out.println("The salary of the " + emp + " is: " + this.salary);
      System.out.println("The teacher ID of the " + emp + " is: " +
this.teacher_id);
      System.out.println("The department of the " + emp + " is: " +
this.department);
```

```
      System.out.println("The subject of the " + emp + " is: " + this.subject);
   }
}

class arrayMultiLevelInheritance {
   public static void main(String[] args) {
      int empnum;
      Scanner sc = new Scanner(System.in);

      System.out.print("Please enter the number of teacher employees you
want: ");
      empnum = sc.nextInt();

      System.out.println("\n");
      Teacher[] teachers_arr = new Teacher[empnum];

      for (int i = 0; i < empnum; i++) {

         String name, gender, address, company_name, qualification,
department, subject;
         int age, empid, teacher_id;
         double salary;

         System.out.print("Enter the name of the " + (i + 1) + " teacher : ");
         name = sc.next();

         System.out.print("Enter the gender of the " + (i + 1) + " teacher : ");
         gender = sc.next();

         System.out.print("Enter the address of the " + (i + 1) + " teacher : ");
         address = sc.next();

         System.out.print("Enter the age of the " + (i + 1) + " teacher : ");
         age = sc.nextInt();

         System.out.print("Enter the emp ID of the " + (i + 1) + " teacher : ");
         empid = sc.nextInt();

         System.out.print("Enter the company name of the " + (i + 1) + " teacher :
");
         company_name = sc.next();

         System.out.print("Enter the qualification of the " + (i + 1) + " teacher : ");
         qualification = sc.next();
```

```
            System.out.print("Enter the salary of the " + (i + 1) + " teacher : ");
            salary = sc.nextDouble();

            System.out.print("Enter the teacher ID of the " + (i + 1) + " teacher : ");
            teacher_id = sc.nextInt();

            System.out.print("Enter the department of the " + (i + 1) + " teacher : ");
            department = sc.next();

            System.out.print("Enter the subject of the " + (i + 1) + " teacher : ");
            subject = sc.next();

            teachers_arr[i] = new Teacher(name, gender, address, age, empid,
company_name,
                  qualification, salary, teacher_id, department, subject);
            System.out.println("\n");

         }

         for (int i = 0; i < teachers_arr.length; i++) {
            String txt = (i == 0) ? (i + 1) + "st"
                  : ((i == 1) ? (i + 1) + "nd" : ((i == 2) ? (i + 1) + "rd" : (i + 1) + "th"));
            teachers_arr[i].displayDetails(txt);
            System.out.println("\n");
         }

         sc.close();
      }
}
```

**Output Screenshot**

```
C:\Users\Student.U33\Desktop\salu>java arrayMultiLevelInheritance
Please enter the number of teacher employees you want: 5


Enter the name of the 1 teacher : salu
Enter the gender of the 1 teacher : female
Enter the address of the 1 teacher : dgugiow
Enter the age of the 1 teacher : 24
Enter the emp ID of the 1 teacher : 1
Enter the company name of the 1 teacher : ssv
Enter the qualification of the 1 teacher : mtech
Enter the salary of the 1 teacher : 24000
Enter the teacher ID of the 1 teacher : 101
Enter the department of the 1 teacher : cs
Enter the subject of the 1 teacher : maths
```

**Name: salini kb**

**Roll No:33**

**Batch: MCA-B**

**Date:26/03/2022**

## Experiment No.: 13

## Aim

Write a program has class Publisher, Book, Literature and Fiction. Read the information and print the details of books from either the category, using inheritance.

## Procedure

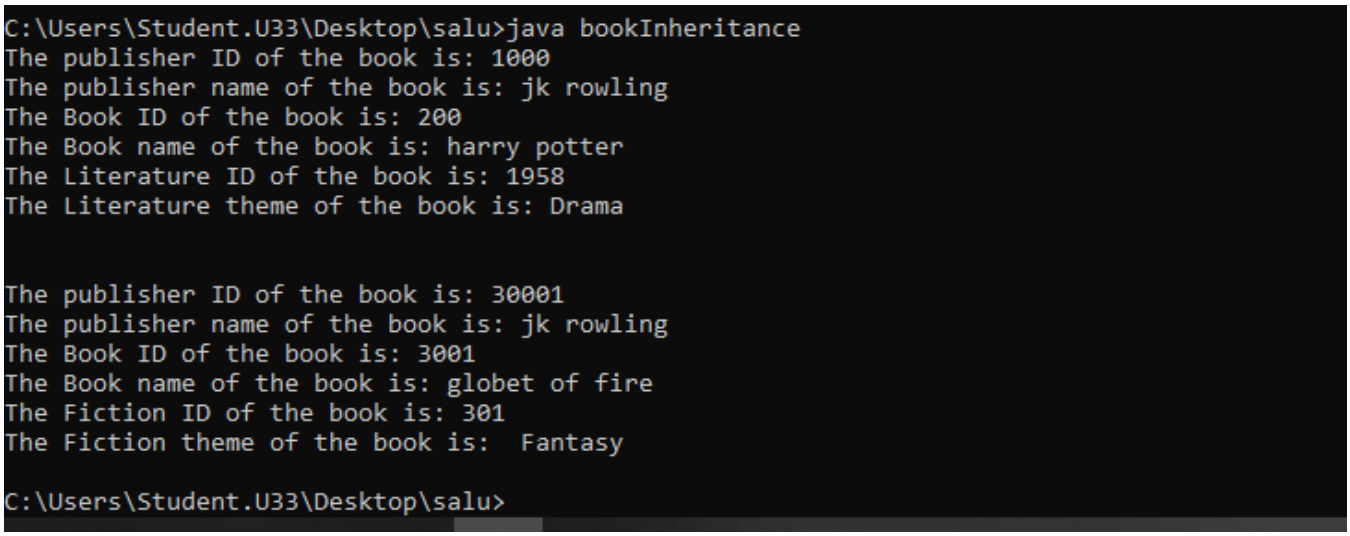
```
import java.util.Scanner;

class Publisher{
   int publisher_id;
   String publisher_name;

   Publisher(int publisher_id, String publisher_name){
      this.publisher_id= publisher_id;
      this.publisher_name= publisher_name;
   }
}

class Book extends Publisher{

   int book_id;
   String book_name;

   Book(int publisher_id, String publisher_name, int book_id, String
book_name) {
      super(publisher_id, publisher_name);
```

```
        this.book_id= book_id;
        this.book_name= book_name;
    }
}

class Literature extends Book{

    int literature_id;
    String literature_theme;

    Literature(int publisher_id, String publisher_name, int book_id, String
book_name, int literature_id, String literature_theme) {
        super(publisher_id, publisher_name, book_id, book_name);
        this.literature_id= literature_id;
        this.literature_theme= literature_theme;
    }

    void displayDetails() {
        System.out.println("The publisher ID of the book is: " + this.publisher_id);
        System.out.println("The publisher name of the book is: " +
this.publisher_name);
        System.out.println("The Book ID of the book is: " + this.book_id);
        System.out.println("The Book name of the book is: " + this.book_name);
        System.out.println("The Literature ID of the book is: " + this.literature_id);
        System.out.println("The Literature theme of the book is: " +
this.literature_theme);
    }

}

class Fiction extends Book{

    int fiction_id;
```

```
   String fiction_theme;

   Fiction(int publisher_id, String publisher_name, int book_id, String
book_name, int fiction_id, String fiction_theme) {
      super(publisher_id, publisher_name, book_id, book_name);
      this.fiction_id= fiction_id;
      this.fiction_theme= fiction_theme;
   }

   void displayDetails() {
      System.out.println("The publisher ID of the book is: " + this.publisher_id);
      System.out.println("The publisher name of the book is: " +
this.publisher_name);
      System.out.println("The Book ID of the book is: " + this.book_id);
      System.out.println("The Book name of the book is: " + this.book_name);
      System.out.println("The Fiction ID of the book is: " + this.fiction_id);
      System.out.println("The Fiction theme of the book is: " +
this.fiction_theme);

   }
}

public class bookInheritance {
   public static void main(String[] args) {
      Literature literature= new Literature(1000,"jk rowling",200,"harry
potter",1958,"Drama");
      Fiction fiction= new Fiction(30001, "jk rowling", 3001, "globet of fire", 301,
" Fantasy");

      literature.displayDetails();
      System.out.println("\n");
      fiction.displayDetails();
```

```
    }
}
```

## Output Screenshot



**Name: salini kb**

**Roll No:33**

**Batch: MCA-B**

**Date:26/03/2022**

## Experiment No.: 14

## Aim

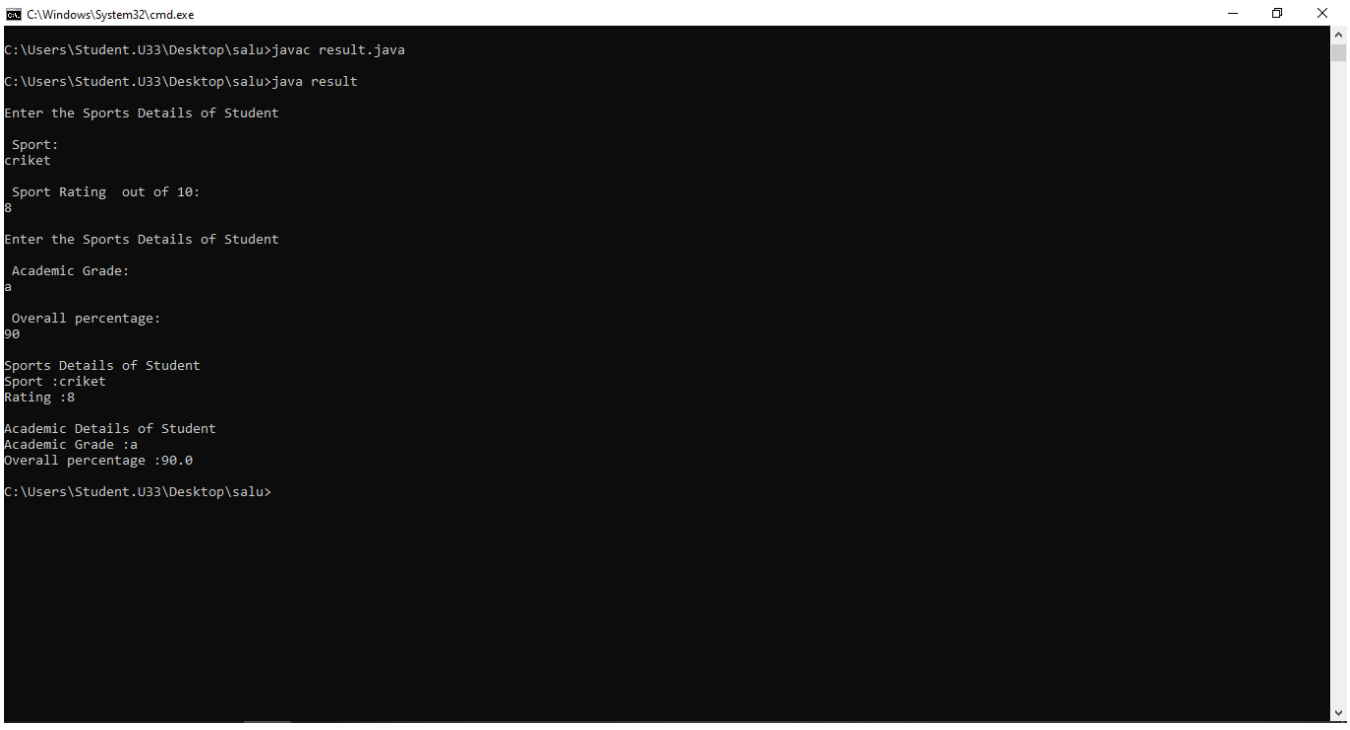
Create classes Student and Sports. Create another class Result inherited from Student and Sports. Display the academic and sports score of a student.

## Procedure

```
import java.util.Scanner;
class sports{
   String sport;
   int Rating;
   sports(String spo, int ra){
      sport = spo;
      Rating = ra;
   }

}
class student extends sports{
   String Grade;
   double Overall_per;
   student(String spo, int ra,String gd, double per ){
      super(spo, ra);
      Grade = gd;
      Overall_per = per;
   }
}
public class result extends student {
   result(String spo, int ra,String gd, double per ){
      super(spo, ra, gd, per);
   }
   void display(){
      System.out.println("\nSports Details of Student");
      System.out.println("Sport :"+sport);
      System.out.println("Rating :"+Rating);
      System.out.println("\nAcademic Details of Student");
```

```
        System.out.println("Academic Grade :"+Grade);
        System.out.println("Overall percentage :"+Overall_per);
    }

    public static void main(String[] args) {
        Scanner sc =new Scanner(System.in);
        System.out.println("\nEnter the Sports Details of Student");
        System.out.println("\n Sport: ");
        String a =sc.next();
        System.out.println("\n Sport Rating  out of 10: ");
        int b =sc.nextInt();
        System.out.println("\nEnter the Sports Details of Student");
        System.out.println("\n Academic Grade: ");
        String c =sc.next();
        System.out.println("\n Overall percentage: ");
        double d =sc.nextDouble();
        sc.close();
        result obj= new result(a,b,c,d);
        obj.display();
    }
}
```

## Output Screenshot



**Experiment No.: 15**

**Name: salini kb**

**Roll No:33**

**Batch: MCA-B**

**Date:26/03/2022**

**Aim**

Create an interface having prototypes of functions area() and perimeter(). Create two classes Circle
and Rectangle which implements the above interface. Create a menu driven program to find area and
perimeter of objects.

**program**

```
import java.util.*;

interface ShapeCalculate{
   double area();
   double perimeter();
}

class Circle implements ShapeCalculate{
   int radius;

   Circle(int radius){
      this.radius= radius;
   }


   public double area() {
      return (3.14*this.radius*this.radius);
   }
```

```
    public double perimeter() {
        return (2*3.14*this.radius);
    }
}

class Rectangle implements ShapeCalculate{
    int length, breadth;

    Rectangle(int length, int breadth){
        this.length= length;
        this.breadth= breadth;
    }


    public double area() {
        return (this.length*this.breadth);
    }


    public double perimeter() {
        return (2*(this.length+this.breadth));
    }
}

public class shapeInterface {
    public static void main(String[] args) {
        Scanner sc= new Scanner(System.in);
        int choice, isexit=0;

        while(true){
            System.out.println("Matheatical Operations: \n1. Area of a
Rectangle.\n2. Perimeter of a Rectangle.\n3. Area of a circle.\n4. Perimeter of
a circle.\n5. Exit\n");
            System.out.print("Please enter the choice: ");
            choice= sc.nextInt();
            switch(choice){
                case 1: {
                    int length, breadth;
                    System.out.print("\nEnter the length of the rectangle: ");
                    length= sc.nextInt();

                    System.out.print("\nEnter the length of the rectangle: ");
                    breadth= sc.nextInt();
```

```
            ShapeCalculate rectangleshape= new Rectangle(length, breadth);
            System.out.println("The area of the mentioned rectangle is :
"+rectangleshape.area()+"sqcm");
            break;
          }

          case 2: {
            int length, breadth;
            System.out.print("\nEnter the length of the rectangle: ");
            length= sc.nextInt();

            System.out.print("\nEnter the length of the rectangle: ");
            breadth= sc.nextInt();

            ShapeCalculate rectangleshape= new Rectangle(length, breadth);
            System.out.println("The perimeter of the mentioned rectangle is :
"+rectangleshape.perimeter()+"cm");
            break;
          }

          case 3: {
            int radius;
            System.out.print("\nEnter the radius of the circle: ");
            radius= sc.nextInt();

            ShapeCalculate circleshape= new Circle(radius);
            System.out.println("The area of the mentioned rectangle is :
"+circleshape.area()+"sqcm");
            break;
          }

          case 4: {
            int radius;
            System.out.print("\nEnter the radius of the circle: ");
            radius= sc.nextInt();

            ShapeCalculate circleshape= new Circle(radius);
            System.out.println("The perimeter of the mentioned rectangle is :
"+circleshape.perimeter()+"cm");
            break;
          }

          case 5: {
            isexit=1;
```

```
                break;
            }
        }

        if(isexit==1){
            break;
        }
    }

  }
}
```

OUTPUT

```
C:\Users\Student.U33\Desktop\salu>javac shapeInterface.java

C:\Users\Student.U33\Desktop\salu>java shapeInterface
Matheatical Operations:
1. Area of a Rectangle.
2. Perimeter of a Rectangle.
3. Area of a circle.
4. Perimeter of a circle.
5. Exit

Please enter the choice: 1

Enter the length of the rectangle: 4

Enter the length of the rectangle: 4
The area of the mentioned rectangle is : 16.0sqcm
Matheatical Operations:
1. Area of a Rectangle.
2. Perimeter of a Rectangle.
3. Area of a circle.
4. Perimeter of a circle.
5. Exit

Please enter the choice: 2

Enter the length of the rectangle: 5

Enter the length of the rectangle: 5
The perimeter of the mentioned rectangle is : 20.0cm
Matheatical Operations:
1. Area of a Rectangle.
2. Perimeter of a Rectangle.
3. Area of a circle.
4. Perimeter of a circle.
5. Exit

Please enter the choice:
```

**Name: salini kb**

**Roll No:33**

**Batch: MCA-B**

**Date:26/03/2022**

## Experiment No.: 16

## Aim

Prepare bill with the given format using calculate method from interface.

Order No.

Date :

Product Id Name Quantity unit price Total

101 A 2 25 50

102 B 1 100 100

Net. Amount 150

## Procedure

```
import java.util.Scanner;
interface calc{
   void calculate();
}
class bill implements calc{
   String date,name,p_id;
   int quantity;
   double unit_price,total,namount=0;
   Scanner sc = new Scanner(System.in);
   public void getdata(){
      System.out.println("\nEnter product id:");
      p_id = sc.nextLine();
      System.out.println("Enter product name:");
```

```
        name = sc.nextLine();
        System.out.println("Enter the Quantity:");
        quantity = sc.nextInt();
        System.out.println("Enter the unit price:");
        unit_price = sc.nextDouble();
    }
    public void calculate(){
        total = quantity * unit_price;
    }
    public void display(){
        System.out.println(p_id+"\t\t"+name+"\t\t"+quantity+"\t\t"+unit_price+"\t"+total);
    }
}
public class BillInterface {
    public static void main(String[] args) {
        int n,i;
        double namount=0,t;
        int ran;
        String date;
        t = Math.random() *1000000;
        ran = (int) t;
        Scanner sc = new Scanner(System.in);
        System.out.println("Order no. #"+ran);
        System.out.print("Enter the date:");
        date = sc.nextLine();
        System.out.println("Enter how many products are there:");
        n = sc.nextInt();
        bill ob[] = new bill[n];
        for(i=0;i<n;i++)
            ob[i] = new bill();
        for(i=0;i<n;i++){
```

```
            ob[i].getdata();
            ob[i].calculate();
        }
        System.out.println("Date:"+date);
        System.out.println("Product Id  Name   Quantity  unit price  Total ");
        System.out.println("---------------------------------------------------------------");
        for(i=0;i<n;i++){
            ob[i].display();
            namount += ob[i].total;
        }
        System.out.println("---------------------------------------------------------------");
        System.out.println("\t\t\t\tNet.Amount\t\t"+ namount);

    }
}
```

## Output Screenshot

```
C:\Users\Student\Desktop\riya\oops>java BillCalc
Order no. #241846
Enter the date:12/03/2022
Enter how many products are there:
2

Enter product id:
101
Enter product name:
Riya
Enter the Quantity:
2
Enter the unit price:
100

Enter product id:
102
Enter product name:
Reena
Enter the Quantity:
3
Enter the unit price:
150
Date:12/03/2022
Product Id  Name   Quantity  unit price  Total
---------------------------------------------------------------
101             Riya            2               100.0   200.0
102             Reena           3               150.0   450.0
---------------------------------------------------------------
                                Net.Amount              650.0
```

**Name: salini kb**

**Roll No:33**

**Batch: MCA-B**

**Date:26/03/2022**

## Experiment No.: 17

## Aim

Create a Graphics package that has classes and interfaces for figures Rectangle, Triangle, Square and Circle. Test the package by finding the area of these figures.

## Procedure

## Circle.java

```
package Graphics;

interface AreaInterface {
    void Area();
}

public class Circle implements AreaInterface{
    double radius;
    public Circle(double radius){
        this.radius= radius;
    }
    public void Area() {
        double area= 3.14 * this.radius * this.radius;
        System.out.println("The area of the given circle is :
"+area);

    }
}
```

Rectangle.java

```
package Graphics;

interface AreaInterface {
    void Area();
}

public class Circle implements AreaInterface{
    double radius;
    public Circle(double radius){
        this.radius= radius;
    }
    public void Area() {
        double area= 3.14 * this.radius * this.radius;
```

```
        System.out.println("The area of the given circle is :
"+area);
    }
}
```

Square.java

```
package Graphics;

interface AreaInterface {
    void Area();
}

public class Square implements AreaInterface{
    double side;
    public Square(double side){
        this.side= side;
    }
    public void Area() {
        double area= this.side * this.side;
        System.out.println("The area of the given square is :
"+area);
    }
}
```

Triangle.java

```
package Graphics;

interface AreaInterface {
    void Area();
}

public class Triangle implements AreaInterface{
    double length, breadth;
    public Triangle(double length, double breadth){
        this.length= length;
        this.breadth= breadth;
    }
    public void Area() {
        double area= 0.5 * this.length * this.breadth;
        System.out.println("The area of the given triangle is :
"+area);
    }
}
```

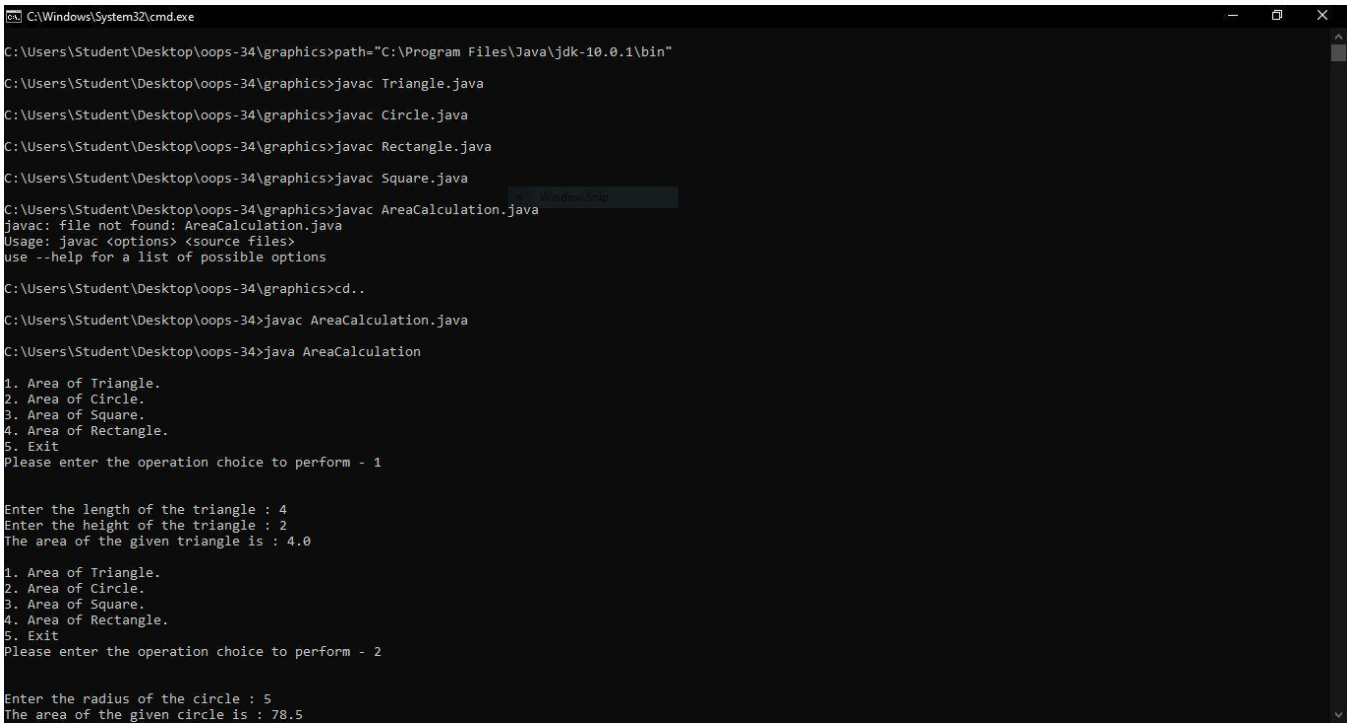
Areacirculation.java

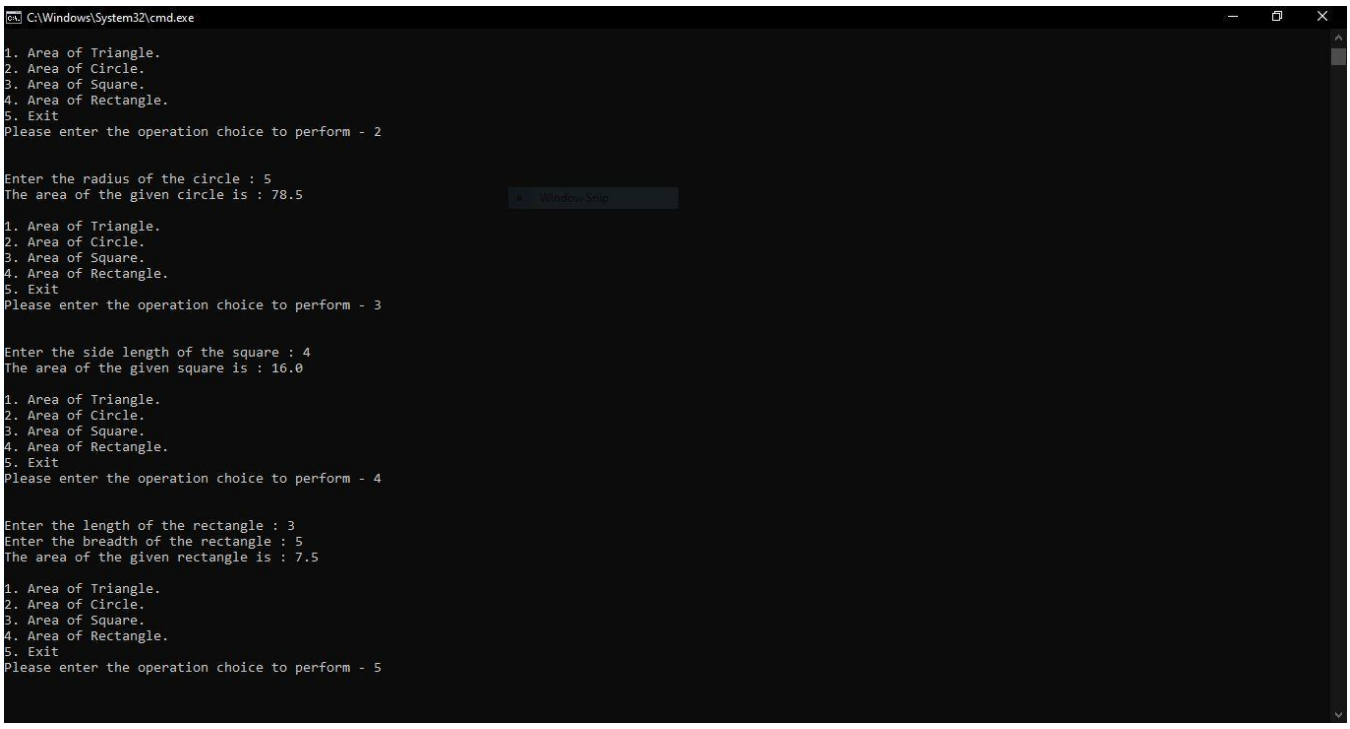
```
import java.util.*;
import Graphics.*;

public class AreaCalculation {
    public static void main(String[] args) {
        Scanner sc= new Scanner(System.in);
        int choice,isexit=0;

        while(isexit==0){
            double length, breadth, side, radius;
```

```
            System.out.println("\n1. Area of Triangle.\n2. Area of
Circle.\n3. Area of Square.\n4. Area of Rectangle.\n5. Exit");
            System.out.print("Please enter the operation choice to
perform - ");
            choice= sc.nextInt();
            System.out.println("\n");

            switch(choice){
                case 1:{
                    System.out.print("Enter the length of the
triangle : ");
                    length= sc.nextDouble();
                    System.out.print("Enter the height of the
triangle : ");
                    breadth= sc.nextDouble();
                    Triangle triangle = new Triangle(length,
breadth);
                    triangle.Area();
                    break;
                }

                case 2:{
                    System.out.print("Enter the radius of the circle
: ");
                    radius= sc.nextDouble();
                    Circle cir= new Circle(radius);
                    cir.Area();
                    break;
                }

                case 3:{
                    System.out.print("Enter the side length of the
square : ");
                    side= sc.nextDouble();
                    Square square= new Square(side);
                    square.Area();
                    break;
                }

                case 4:{
                    System.out.print("Enter the length of the
rectangle : ");
                    length= sc.nextDouble();
                    System.out.print("Enter the breadth of the
rectangle : ");
                    breadth= sc.nextDouble();
                    Rectangle rec= new Rectangle(length, breadth);
                    rec.Area();
                    break;
                }

                case 5:{
                    isexit=1;
                    break;
                }

                default:{
                    break;
```

```
                    }
                }
            }
            sc.close();
        }
}
```

## Output Screenshot



## Experiment No.: 18

**Name: salini kb**

**Roll No:33**

**Batch: MCA-B**

**Date:26/03/2022**

## Aim

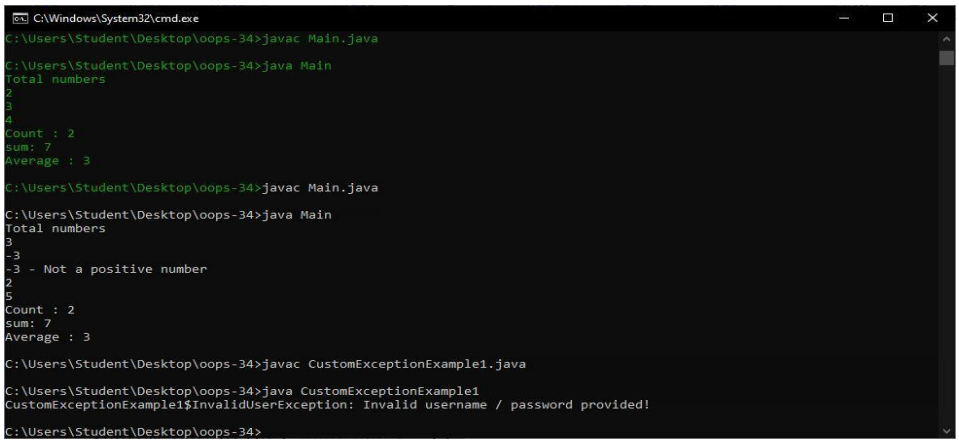
Write a user defined exception class to authenticate the user name and password.

## Procedure

```
public class CustomExceptionExample {


  public static class InvalidUserException extends Exception {

    public InvalidUserException() {
      super("Invalid username / password provided!");
    }

  }

  public static void main(String[] args) {


    String username = "salu";

    String password = "pass";

    try {

      if (username.equals("user") && password.equals("pass")) {

        System.out.println("Authenticated successfully!");

      } else {

        throw new InvalidUserException();
      }
```

```
        } catch (InvalidUserException e) {
            System.out.println(e);
        }
    }
}
```

## Output Screenshot



**Name: salini kb**

**Roll No:33**

**Batch: MCA-B**

**Date:26/03/2022**

## Experiment No.: 19

## Aim

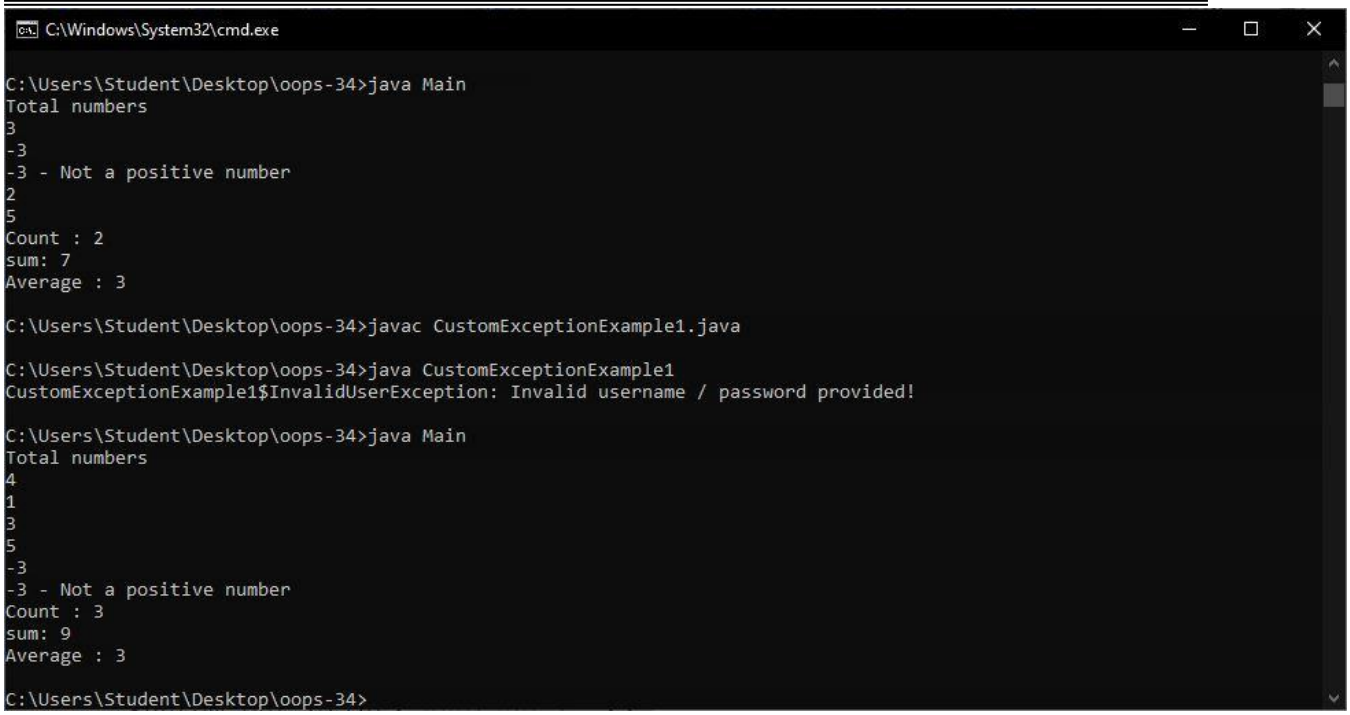
Find the average of N positive integers, raising a user defined exception for each negative input.

## Procedure

```
import java.util.*;

class MyException extends Exception {
   public MyException(String value) {
      super(value);
   }
}

class Main {
   public static void main(String args[]) {
      int totalNums;
      int i;
      int temp, count = 0;
      int sum = 0;

      Scanner sc = new Scanner(System.in);

      System.out.println("Total numbers");
      totalNums = Integer.parseInt(sc.nextLine());
      for (i = 0; i < totalNums; i++) {
         try {
            temp = Integer.parseInt(sc.nextLine());
```

```
            if (temp > 0) {
               sum += temp;
               count += 1;
            } else {
               throw new MyException(Integer.toString(temp));
            }
         } catch (MyException ex) {
            System.out.print(ex.getMessage());
            System.out.println(" - Not a positive number");
         }
      }
      System.out.print("Count : ");
      System.out.println(count);
      System.out.print("sum: ");
      System.out.println(sum);
      System.out.print("Average : ");
      System.out.println(sum / count);

   }
}
```

**Output Screenshot**


C:\Windows\System32\cmd.exe
C:\Users\Student\Desktop\oops-34>java Main
Total numbers
3
-3
-3 - Not a positive number
2
5
Count : 2
sum: 7
Average : 3
C:\Users\Student\Desktop\oops-34>javac CustomExceptionExample1.java
C:\Users\Student\Desktop\oops-34>java CustomExceptionExample1
CustomExceptionExample1$InvalidUserException: Invalid username / password provided!
C:\Users\Student\Desktop\oops-34>java Main
Total numbers
4
1
3
5
-3
-3 - Not a positive number
Count : 3
sum: 9
Average : 3
C:\Users\Student\Desktop\oops-34>

**Experiment No.: 20**

**Aim**

**Name: salini kb**

**Roll No:33**

**Batch: MCA-B**

**Date:26/03/2022**

Define 2 classes; one for generating Fibonacci numbers and other for displaying even numbers in a given range. Implement using threads. (Runnable Interface)

```
import java.util.*;

class fibonacci implements Runnable {
   int l;

   fibonacci(int n) {
      l = n;
   }

   public void run() {
      int c;
      int a = 0, b = 1;
      System.out.print(a + " " + b);
      for (int i = 0; i <= l; i++) {
         c = a + b;
         System.out.print(" " + c);
         a = b;
         b = c;
      }
   }
}
```

```
class even implements Runnable {
   int l;

   even(int n) {
      l = n;
   }

   public void run() {
      for (int i = 0; i <= l; i++) {
         if (i % 2 == 0)
            System.out.print(i + "  ");
      }
      System.out.println("");

   }

}

class My{
   public static void main(String args[]) {
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter Limit :");
      int l = sc.nextInt();
      fibonacci f = new fibonacci(l);
      Thread T1 = new Thread(f);
      T1.start();
      even e = new even(l);
      Thread T2 = new Thread(e);
      T2.start();
   }
}
```

## Output Screenshot

```
C:\Users\Student.U33\Documents\salu>java My
Enter Limit :
4
0  2  4
0 1 1 2 3 5 8
C:\Users\Student.U33\Documents\salu>javac My.java

C:\Users\Student.U33\Documents\salu>java My
Enter Limit :
10
0  2  4  6  8  10
0 1 1 2 3 5 8 13 21 34 55 89 144
C:\Users\Student.U33\Documents\salu>
```

## Experiment No.: 21

**Name: salini kb**

**Roll No:33**

**Batch: MCA-B**

**Date:26/03/2022**

### Aim

Program to create a generic stack and do the Push and Pop operations.

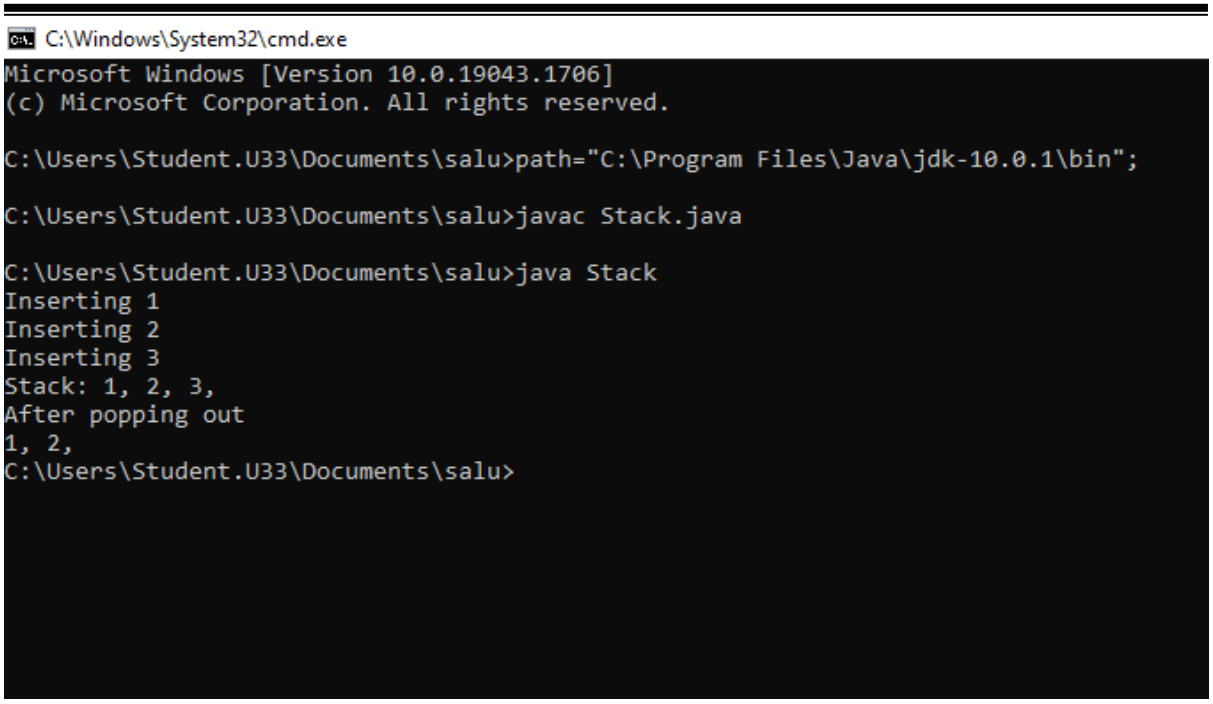
### Procedure

```
class Stack {


 private int arr[];


 private int top;


 private int capacity;


 Stack(int size) {

  arr = new int[size];
  capacity = size;
  top = -1;
 }


 public void push(int x) {
  if (isFull()) {
    System.out.println("Stack OverFlow");


    System.exit(1);
  }
```

```
  System.out.println("Inserting " + x);
  arr[++top] = x;
 }


 public int pop() {


  if (isEmpty()) {
   System.out.println("STACK EMPTY");

   System.exit(1);
  }

  return arr[top--];
 }


 public int getSize() {
  return top + 1;
 }


 public Boolean isEmpty() {
  return top == -1;
 }

 public Boolean isFull() {
  return top == capacity - 1;
 }

 public void printStack() {
  for (int i = 0; i <= top; i++) {
   System.out.print(arr[i] + ", ");
```

```
    }
  }
  public static void main(String[] args) {
    Stack stack = new Stack(5);
    stack.push(1);
    stack.push(2);
    stack.push(3);

    System.out.print("Stack: ");
    stack.printStack();


    stack.pop();
    System.out.println("\nAfter popping out");
    stack.printStack();
  }
}
```

## Output Screenshot


C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19043.1706]
(c) Microsoft Corporation. All rights reserved.
C:\Users\Student.U33\Documents\salu>path="C:\Program Files\Java\jdk-10.0.1\bin";
C:\Users\Student.U33\Documents\salu>javac Stack.java
C:\Users\Student.U33\Documents\salu>java Stack
Inserting 1
Inserting 2
Inserting 3
Stack: 1, 2, 3,
After popping out
1, 2,
C:\Users\Student.U33\Documents\salu>

**Experiment No.: 22**

**Aim**

Maintain a list of Strings using ArrayList from collection framework, perform built-in operations.

**Name:salini kb**

**Roll No:33**

**Batch: MCA-B**

**Date:26/03/2022**

```
import java.util.ArrayList;
import java.util.Collections;

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

      ArrayList<String> data = new ArrayList<String>();

      data.add("A");
      data.add("B");
      data.add("C");
      data.add("D");

      data.set(1, "BB");
      System.out.println(data);


      System.out.println(data.get(0));
      System.out.println(data.get(1));


      data.remove(0);
      System.out.println(data);
```

```
        System.out.println(data.size());


        for (String d : data) {
           System.out.println(d);
        }

        Collections.sort(data);
        System.out.println(data);


        data.clear();
        System.out.println(data);
    }
}
```

**Output Screenshot**

```
C:\Users\Student.U33\Documents\salu>javac Mycls.java

C:\Users\Student.U33\Documents\salu>java Mycls
[A, BB, C, D]
A
BB
[BB, C, D]
3
BB
C
D
[BB, C, D]
[]

C:\Users\Student.U33\Documents\salu>
```

**Name: salini kb**

**Roll No:33**

**Batch: MCA-B**

**Date:26/03/2022**

## Experiment No.: 23

## Aim

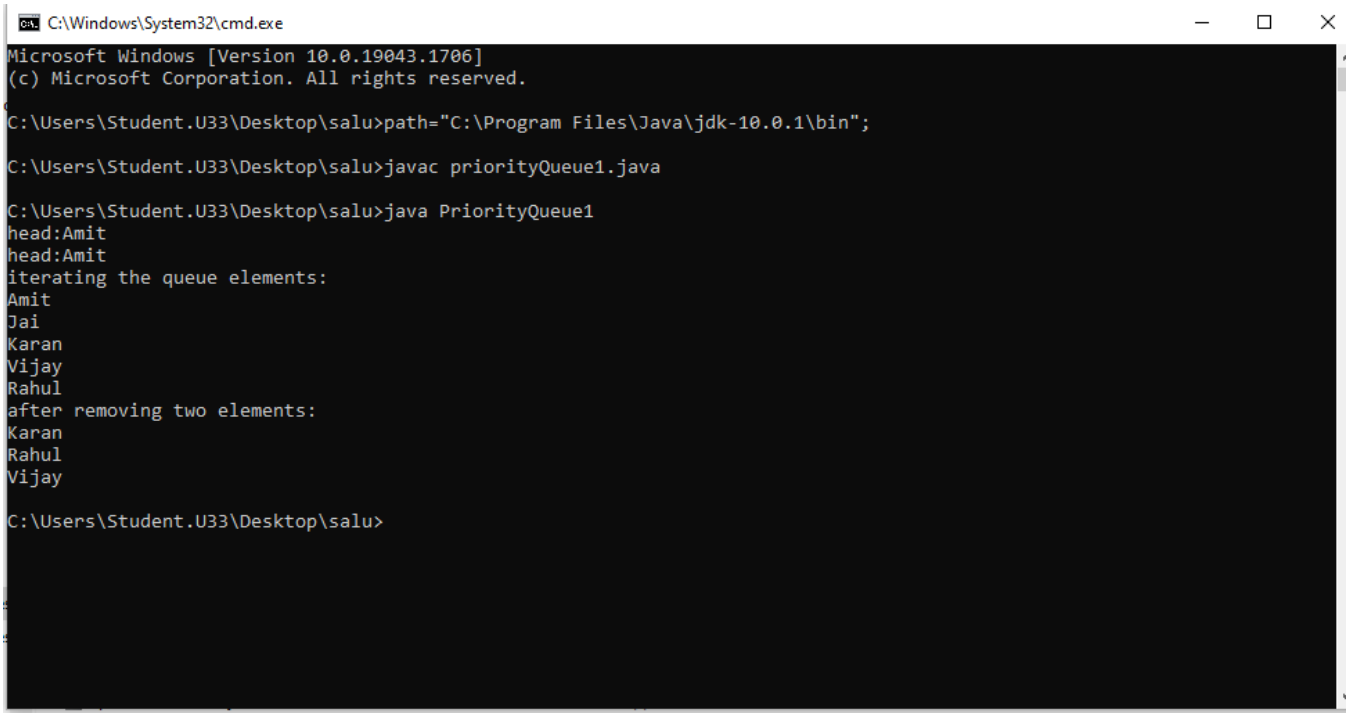
Program to demonstrate the creation of queue object using the PriorityQueue class

## Procedure

```
import java.util.*;
class PriorityQueue1{
public static void main(String args[]){
PriorityQueue<String> queue=new PriorityQueue<String>();
queue.add("Amit");
queue.add("Vijay");
queue.add("Karan");
queue.add("Jai");
queue.add("Rahul");
System.out.println("head:"+queue.element());
System.out.println("head:"+queue.peek());
System.out.println("iterating the queue elements:");
Iterator itr=queue.iterator();
while(itr.hasNext()){
System.out.println(itr.next());
}
queue.remove();
queue.poll();
```

```
System.out.println("after removing two elements:");
Iterator<String> itr2=queue.iterator();
while(itr2.hasNext()){
System.out.println(itr2.next());
}
}
}
```

**Output Screenshot**



**Experiment No.: 24**

**Aim**

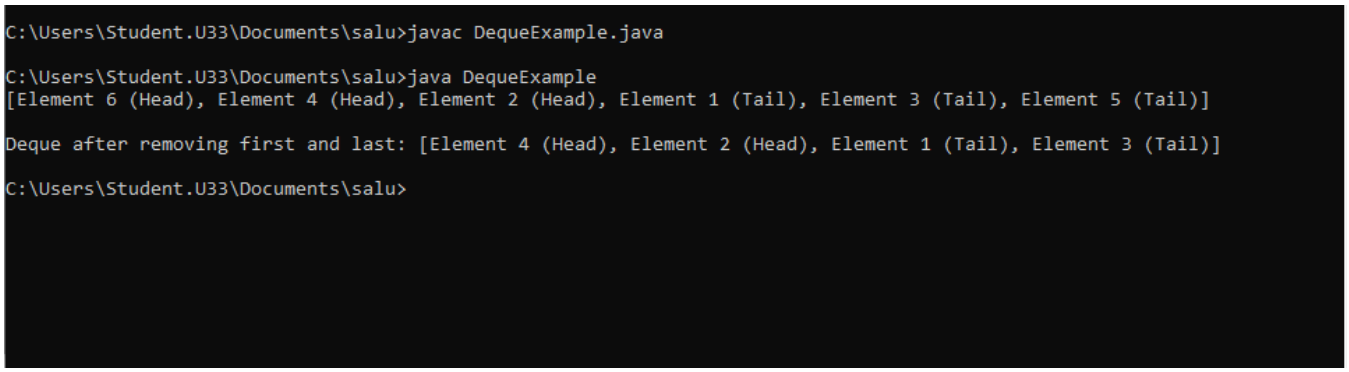
Program to demonstrate the addition and deletion of elements in deque

**Procedure**

```
import java.util.*;

public class DequeExample {
   public static void main(String[] args)
   {
      Deque<String> deque
         = new LinkedList<String>();




      deque.add("Element 1 (Tail)");


      deque.addFirst("Element 2 (Head)");


      deque.addLast("Element 3 (Tail)");


      deque.push("Element 4 (Head)");
```

```
        deque.offer("Element 5 (Tail)");


        deque.offerFirst("Element 6 (Head)");

        System.out.println(deque + "\n");


        deque.removeFirst();
        deque.removeLast();
        System.out.println("Deque after removing "
                        + "first and last: "
                        + deque);
    }
}
```

**Output Screenshot**



## Experiment No.: 25

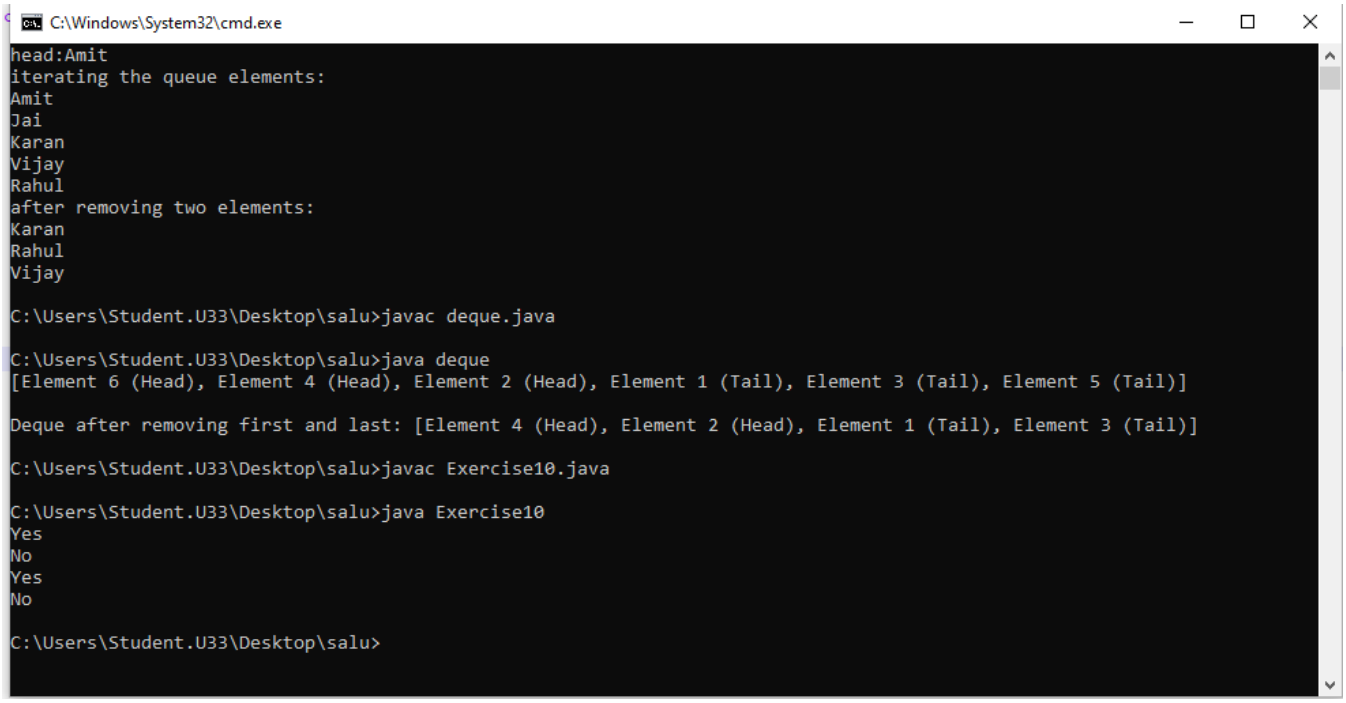
## Aim

Write a Java program to compare two hash set.

## Procedure

```
import java.util.*;
 public class Exercise10 {
 public static void main(String[] args) {

      HashSet<String> h_set = new HashSet<String>();

       h_set.add("Red");
       h_set.add("Green");
       h_set.add("Black");
       h_set.add("White");

       HashSet<String>h_set2 = new HashSet<String>();
       h_set2.add("Red");
       h_set2.add("Pink");
       h_set2.add("Black");
       h_set2.add("Orange");

       HashSet<String>result_set = new HashSet<String>();
      for (String element : h_set){
         System.out.println(h_set2.contains(element) ? "Yes" : "No");
       }
   }
}
```

## Output Screenshot


C:\Windows\System32\cmd.exe
head:Amit
iterating the queue elements:
Amit
Jai
Karan
Vijay
Rahul
after removing two elements:
Karan
Rahul
Vijay
C:\Users\Student.U33\Desktop\salu>javac deque.java
C:\Users\Student.U33\Desktop\salu>java deque
[Element 6 (Head), Element 4 (Head), Element 2 (Head), Element 1 (Tail), Element 3 (Tail), Element 5 (Tail)]
Deque after removing first and last: [Element 4 (Head), Element 2 (Head), Element 1 (Tail), Element 3 (Tail)]
C:\Users\Student.U33\Desktop\salu>javac Exercise10.java
C:\Users\Student.U33\Desktop\salu>java Exercise10
Yes
No
Yes
No
C:\Users\Student.U33\Desktop\salu>

## Experiment No.: 26

**Name: salini kb**

**Roll No:33**

**Batch: MCA-B**

**Date:26/03/2022**

## Aim

Program to demonstrate the working of Map interface by adding, changing and removing elements.

## Procedure

```
import java.util.*;
class HashMapDemo {
    public static void main(String args[]) {
        Map<String, Integer> hm = new HashMap<String, Integer>();
        hm.put("Anu", new Integer(1));
        hm.put("sinu", new Integer(2));
        hm.put("Jinu", new Integer(3));
        for (Map.Entry<String, Integer> me : hm.entrySet()) {
            System.out.print(me.getKey() + ":");
            System.out.println(me.getValue());
        }
    }
}
```

## Output Screenshot

```
D:\Share>javac HashMapDemo.java

D:\Share>java HashMapDemo
Jinu:3
sinu:2
Anu:1
```

## Experiment No.: 27

**Name: salini kb**

**Roll No:33**

**Batch: MCA-B**

**Date:26/03/2022**

## Aim

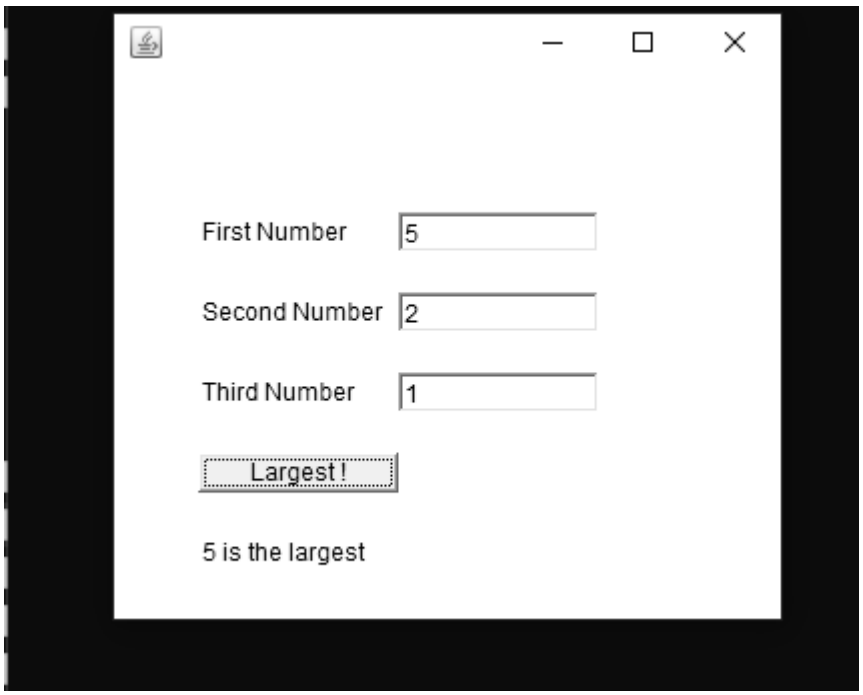
Program to find maximum of three numbers using AWT.

## Procedure

```
import java.awt.*;
import java.awt.event.*;

public class largenum implements ActionListener{
        Frame f=new Frame();
        Label l1=new Label("First Number");
        Label l2=new Label("Second Number");
        Label l3=new Label("Third Number");
        Label res=new Label("Result");
        TextField t1=new TextField();
        TextField t2=new TextField();
        TextField t3=new TextField();
        Button b1=new Button("Largest !");

        largenum(){
                l1.setBounds(50,100,100,20);
                l2.setBounds(50,140,100,20);
                l3.setBounds(50,180,100,20);
                t1.setBounds(150,100,100,20);
                t2.setBounds(150,140,100,20);
                t3.setBounds(150,180,100,20);
                b1.setBounds(50,220,100,20);
                res.setBounds(50,260,100,20);

                f.add(l1);
                f.add(l2);
                f.add(l3);
                f.add(t1);
                f.add(t2);
                f.add(t3);
                f.add(res);
                f.add(b1);

                b1.addActionListener(this);

                f.setLayout(null);
                f.setVisible(true);
                f.setSize(400,400);
        }

        public static void main(String[] args){
                new largenum();
        }
```

```
        public void actionPerformed(ActionEvent e){
                if(e.getSource()==b1){
                        int n1=Integer.parseInt(t1.getText());
                        int n2=Integer.parseInt(t2.getText());
                        int n3=Integer.parseInt(t3.getText());

                        int largeres= (n1 > n2) ? (n1 > n3 ? n1 : n3) : (n2 > n3 ? n2 : n3);
                        res.setText(String.valueOf(largeres)+" is the largest");
                }

        }
}
```

## Output Screenshot



## Experiment No.: 28

## Aim

Implement a simple calculator using AWT components.

**Name: salini kb**

**Roll No:33**

**Batch: MCA-B**

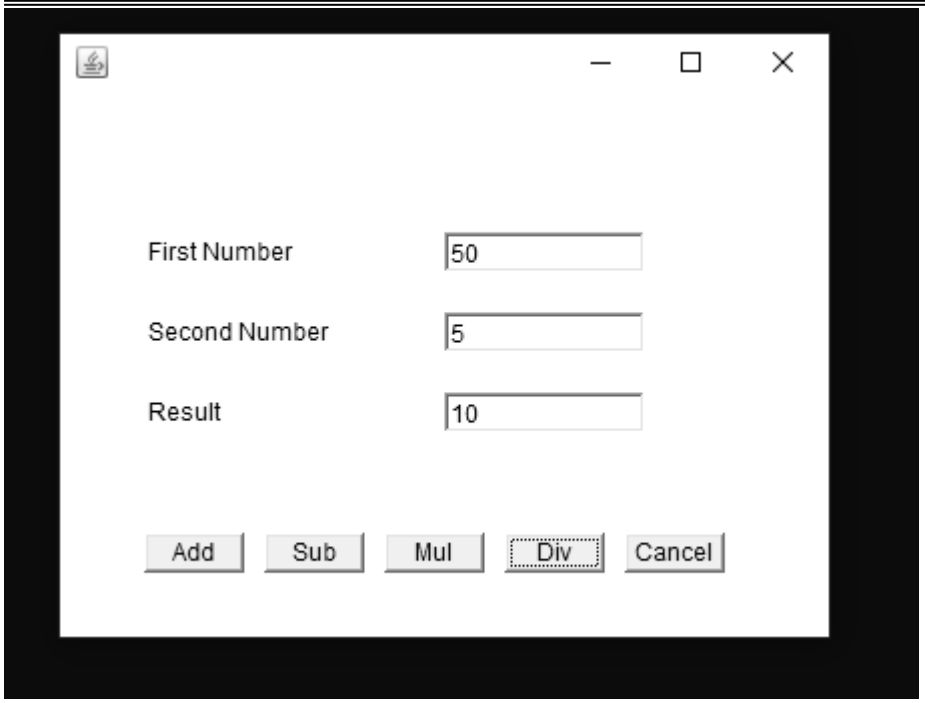
**Date:26/03/2022**

## Procedure

```
import java.awt.*;
import java.awt.event.*;

public class Calculator implements ActionListener
{

Frame f=new Frame();
Label l1=new Label("First Number");
Label l2=new Label("Second Number");
Label l3=new Label("Result");
TextField t1=new TextField();
TextField t2=new TextField();
TextField t3=new TextField();
Button b1=new Button("Add");
Button b2=new Button("Sub");
Button b3=new Button("Mul");
Button b4=new Button("Div");
Button b5=new Button("Cancel");
Calculator()
{

l1.setBounds(50,100,100,20);
l2.setBounds(50,140,100,20);
l3.setBounds(50,180,100,20);
t1.setBounds(200,100,100,20);
t2.setBounds(200,140,100,20);
t3.setBounds(200,180,100,20);
b1.setBounds(50,250,50,20);
b2.setBounds(110,250,50,20);
b3.setBounds(170,250,50,20);
b4.setBounds(230,250,50,20);
b5.setBounds(290,250,50,20);

f.add(l1);
f.add(l2);
f.add(l3);
f.add(t1);
f.add(t2);
f.add(t3);
f.add(b1);
```

```
f.add(b2);
f.add(b3);
f.add(b4);
f.add(b5);
b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);
b4.addActionListener(this);
b5.addActionListener(this);
f.setLayout(null);
f.setVisible(true);
f.setSize(400,350);
}
public void actionPerformed(ActionEvent e)
{
int n1=Integer.parseInt(t1.getText());
int n2=Integer.parseInt(t2.getText());
if(e.getSource()==b1)
{
t3.setText(String.valueOf(n1+n2));
}
if(e.getSource()==b2)
{
t3.setText(String.valueOf(n1-n2));
}
if(e.getSource()==b3)
{
t3.setText(String.valueOf(n1*n2));
}
if(e.getSource()==b4)
{
t3.setText(String.valueOf(n1/n2));
}
if(e.getSource()==b5)
{
System.exit(0);
}
}
public static void main(String...s)
{
new Calculator();
}
}
```

## Output Screenshot


First Number
50
Second Number
5
Result
10
Add
Sub
Mul
Div
Cancel

## Experiment No.: 29

## Aim

Develop a program to handle all mouse events and window events

**Name: salini kb**

**Roll No:33**

**Batch: MCA-B**

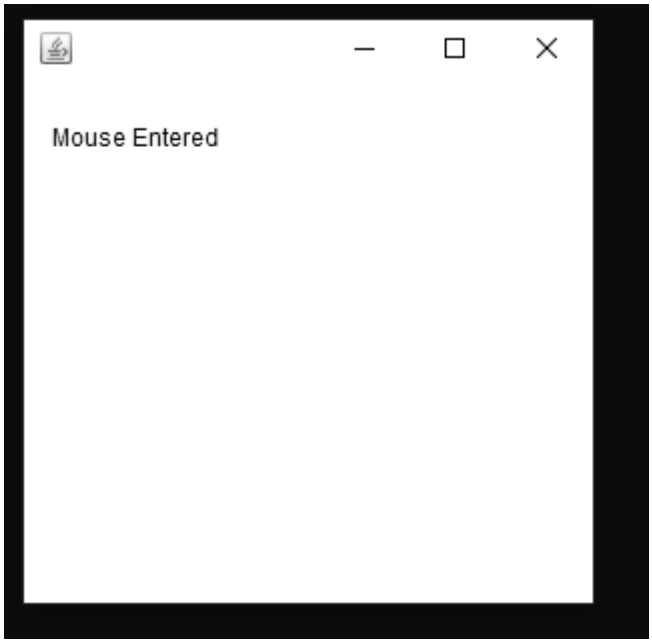
**Date:26/03/2022**

## Procedure

```
import java.awt.*;
import java.awt.event.*;
public class Mouseevents extends Frame implements MouseListener{
   Label l;
   Mouseevents(){
      addMouseListener(this);

      l=new Label();
      l.setBounds(20,50,100,20);
      add(l);
      setSize(300,300);
      setLayout(null);
      setVisible(true);
   }
   public void mouseClicked(MouseEvent e) {
      l.setText("Mouse Clicked");
   }
   public void mouseEntered(MouseEvent e) {
      l.setText("Mouse Entered");
   }
   public void mouseExited(MouseEvent e) {
      l.setText("Mouse Exited");
   }
   public void mousePressed(MouseEvent e) {
      l.setText("Mouse Pressed");
   }
   public void mouseReleased(MouseEvent e) {
      l.setText("Mouse Released");
   }
public static void main(String[] args) {
   new Mouseevents();
```

```
}
}
```

## Output Screenshot



## Experiment No.: 30

## Aim

**Name: salini kb**

**Roll No:33**

**Batch: MCA-B**

**Date:26/03/2022**

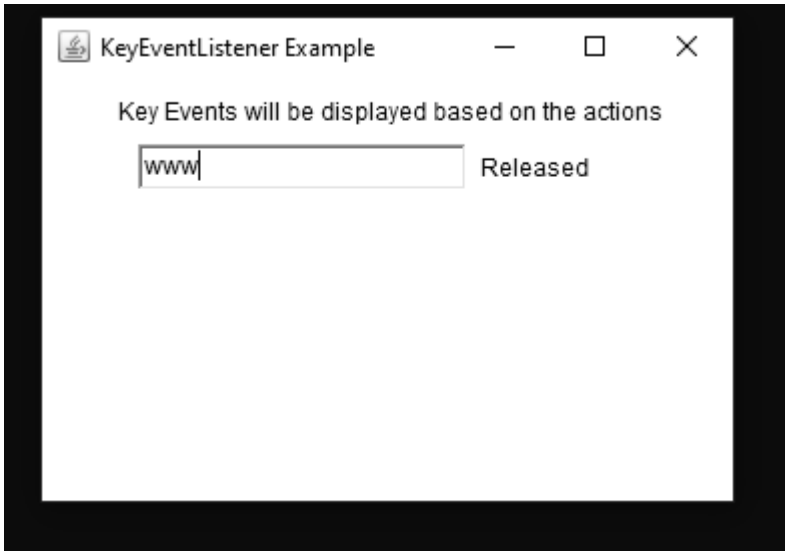
Develop a program to handle Key events.

## Procedure

```
import java.awt.FlowLayout;
import java.awt.Frame;
import java.awt.Label;
import java.awt.TextField;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
public class KE implements KeyListener
{
   Label lb1, lbl2, lb;
   TextField tf1;
   Frame fr;
   String s;
   KE()
   {

      fr = new Frame("KeyEventListener Example");

      lb1= new Label(" Key Events will be displayed based on the actions", Label.CENTER);
      lbl2= new Label();
      lb= new Label();

      tf1 = new TextField(20);
      fr.setLayout(new FlowLayout());

      fr.add(lb1);

      fr.add(tf1);

      fr.add(lbl2);

      tf1.addKeyListener(this);

      fr.setSize(460,250);

      fr.setVisible(true);
   }

   public void keyPressed(KeyEvent ev)
   {
      lbl2.setText(" Key pressed");
```

```
    }

    public void keyReleased(KeyEvent ev)
    {
        lbl2.setText("Released");
    }

    public void keyTyped(KeyEvent ev)
    {
        lbl2.setText("Key is typed");

        fr.setVisible(true);
    }
    public static void main(String[] args)
    {
        new KE();
    }
}
```

## Output Screenshot



## Experiment No.: 31

**Name: salini kb**

**Roll No:33**

**Batch: MCA-B**

**Date:26/03/2022**

## Aim

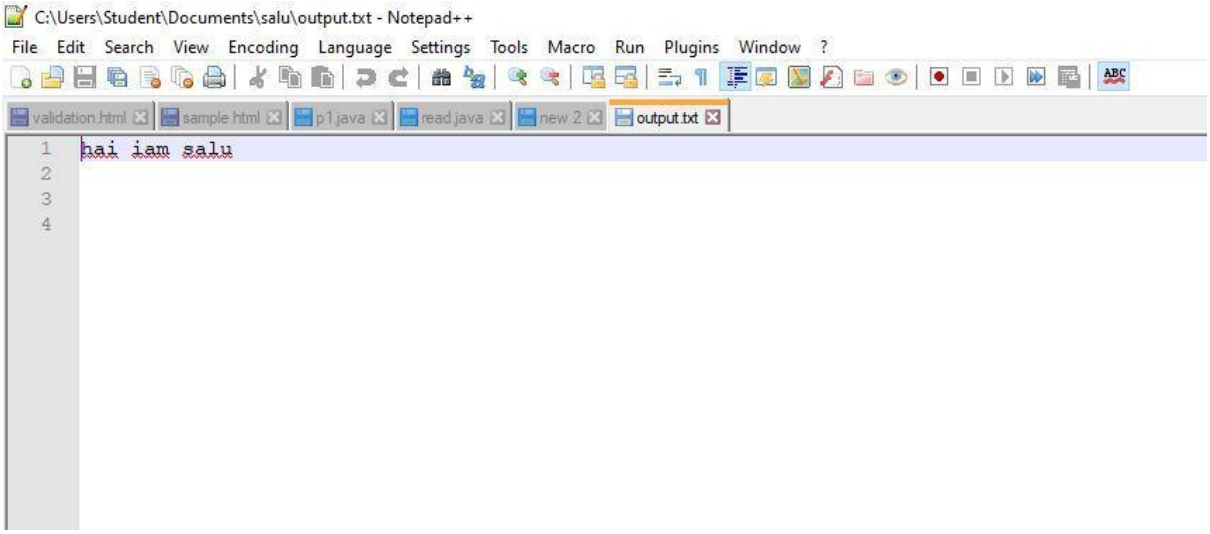
Write a program to write to a file, then read from the file and display the contents on the console.

## Procedure

```
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.*;
import java.util.*;
import java.io.File;
class read {
public static void main(String[] args) {

String var = "";
Scanner scan = new Scanner(System.in);
System.out.println("Enter the text to create file : type ENTER key 3 times to stop");
while (!var.endsWith("\n\n\n"))
var = var + scan.nextLine() + "\n";
try {

File file = new File("output.txt");

FileWriter fw = new FileWriter(file);
fw.write(var);
fw.close();
System.out.println("Reading File content");
```

```
FileReader fr = new FileReader("output.txt");
String str = "";
int i;
while ((i = fr.read()) != -1) {

str += (char) i;
}
System.out.println(str);
fr.close();
} catch (IOException e) {
System.out.println("There are some exception");
}
}
}
```

```
C:\Users\Student\Documents\salu>javac read.java

C:\Users\Student\Documents\salu>java read
Enter the text to create file : type ENTER key 3 times to stop
hai iam salu


Reading File content
hai iam salu
```



## Experiment No.: 32

**Name: salini kb**

**Roll No:33**

**Batch: MCA-B**

**Date:26/03/2022**

## Aim

Write a program to copy one file to another.

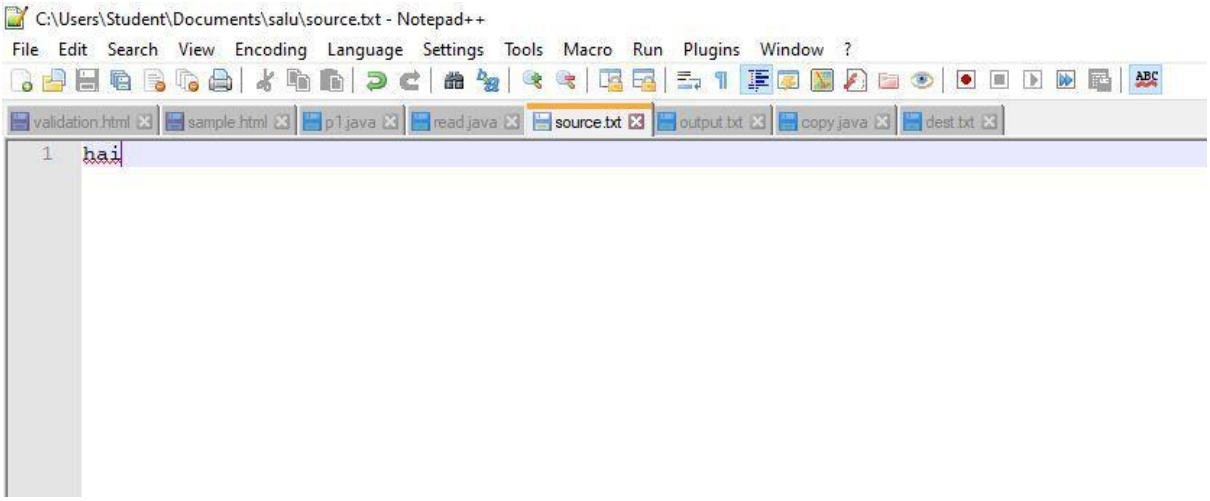
## procedure

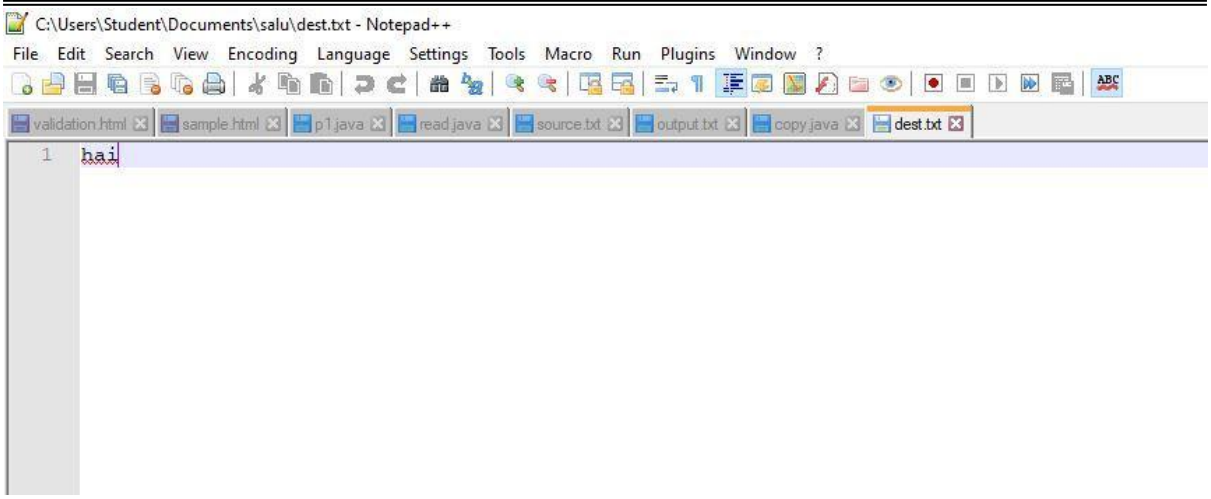
```
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.*;
import java.util.*;
import java.io.File;
public class copy {
public static void main(String[] args) {
Scanner scan=new Scanner(System.in);
System.out.println("Enter the source File Name");
String source=scan.nextLine();
try {
FileReader fr=new FileReader(source);
String str = "";
int i;
System.out.println("Reading from file "+source);
while ((i = fr.read()) != -1) {

str += (char) i;
}
System.out.println(str);
System.out.println("\n Enter the filename to copy");
String destination=scan.nextLine();
```

```
File file=new File(destination);
FileWriter fw = new FileWriter(file);
fw.write(str);
fr.close();
fw.close();
System.out.println("Copied from "+source+" to "+destination+ " Successfully..!");
} catch (Exception e) {

System.out.println("Exception Occured");
}
}
}
```

## output

```
C:\Users\Student\Documents\salu>javac copy.java

C:\Users\Student\Documents\salu>java copy
Enter the source File Name
source.txt
Reading from file source.txt
hai

 Enter the filename to copy
dest.txt
Copied from source.txt to dest.txt Successfully..!

C:\Users\Student\Documents\salu>
```




C:\Users\Student\Documents\salu\dest.txt - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
validation.html
sample.html
p1.java
read.java
source.txt
output.txt
copy.java
dest.txt
1
hai

## Experiment No.: 33

**Name: salini kb**

**Roll No:33**

**Batch: MCA-B**

**Date:26/03/2022**

## Aim

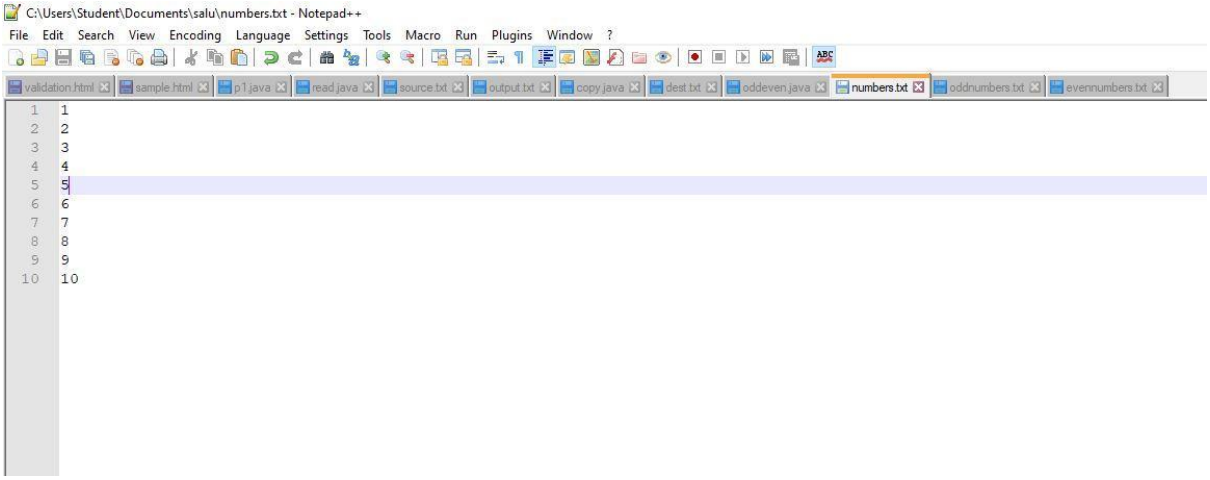
Write a program that reads from a file having integers. Copy even numbers and odd numbers to separate files.

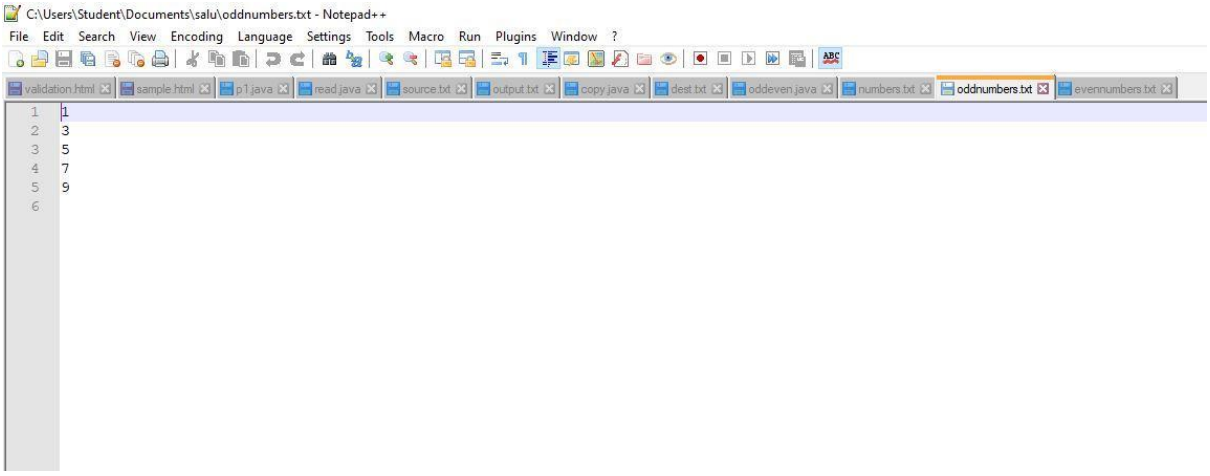
```
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.*;
import java.util.*;
import java.io.File;
public class oddeven {
public static void main(String[] args) {
try {
FileReader fr = new FileReader("numbers.txt");
BufferedReader br = new BufferedReader(fr);
File file1 = new File("oddnumbers.txt");
FileWriter fw1 = new FileWriter(file1);
File file2 = new File("evennumbers.txt");
FileWriter fw2 = new FileWriter(file2);
String num;
while ((num = br.readLine()) != null) {
if (Integer.parseInt(num) % 2 == 0) {
fw2.write(num + "\n");
} else {
fw1.write(num + "\n")
}
}
```

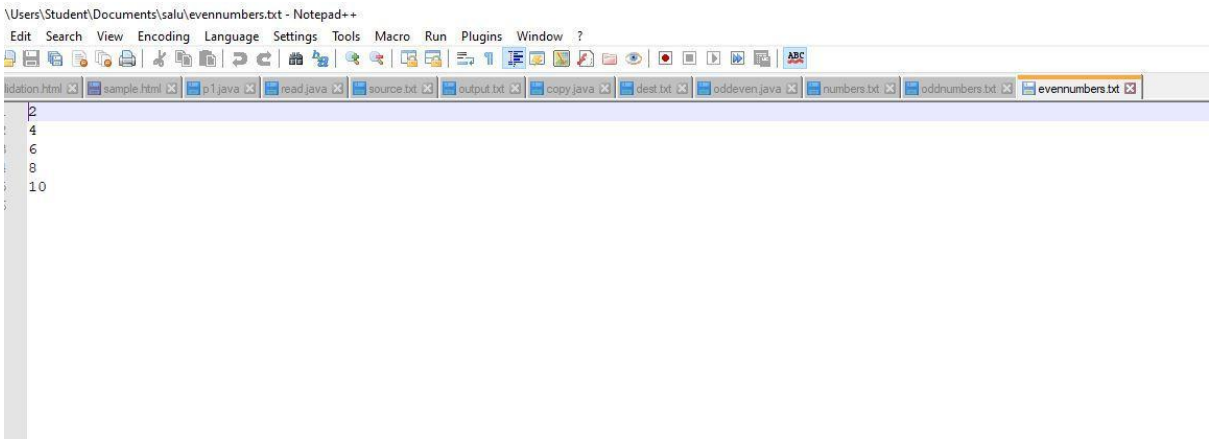
```
fw1.close();

fw2.close();

} catch (Exception e) {


System.out.println("Error");

}

}

}
```

## output

```
C:\Users\Student\Documents\salu>javac oddeven.java

C:\Users\Student\Documents\salu>java oddeven
copied successfully!
```




\Users\Student\Documents\salu\evennumbers.txt - Notepad++
Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
sample.html
p1.java
read.java
source.txt
output.txt
copy.java
dest.txt
oddeven.java
numbers.txt
oddnumbers.txt
evennumbers.txt
2
4
6
8
10