20MCA132 - OBJECT ORIENTED PROGRAMMING LAB

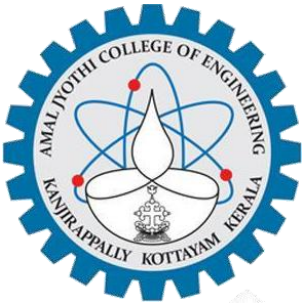
Lab Report Submitted By

SANDRA P M

Reg. No.: AJC21MCA-2092

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. Koovappally, 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 ORIENTED PROGRAMMING LAB"** is the bonafide work of **Sandra P M AJC21MCA-2092** 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 2021-22.

Mrs.Gloriya Mathew
Lab In-Charge

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

Internal Examiner

External Examiner

CONTENT

	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.		
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**.**	**17/05/22**	**27-30**
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.	**17/05/22**	**31-33**

14	Create classes Student and Sports. Create another class Result inherited from Student and Sports. Display the academic and sports score of a student.	**17/05/22**	**34-35**
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.	**24/05/22**	**36-38**
16	Prepare a bill with the given format using the calculate method from the interface.	**24/05/22**	**39-41**
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.	**31/05/22**	**42-47**
18	Write a user defined exception class to authenticate the user name and password.	**31/05/22**	**47-48**
19	Find the average of N positive integers, raising a user defined exception for each negative input.	**31/05/22**	**49-50**
20	Define 2 classes; one for generating Fibonacci numbers and other for displaying even numbers in a given range. Implement using threads. (Runnable Interface)	**31/05/22**	**50-51**
21	Program to create a generic stack and do the Push and Pop operations.	**31/05/22**	**52**
22	Maintain a list of Strings using ArrayList from collection framework, perform built-in operations.	**31/05/22**	**53-55**
23	Program to demonstrate the creation of queue object using the PriorityQueue class	**31/05/22**	**56**
24	Program to demonstrate the addition and deletion of elements in deque	**31/05/22**	**57-62**

25	Write a Java program to compare two hash set	**31/05/22**	**63-64**
26	Program to demonstrate the working of Map interface by adding, changing and removing elements.	**13/06/22**	**65-67**
27	Program to find a maximum of three numbers using AWT.	**09/06/22**	**68-70**
28	Implement a simple calculator using AWT components.	**09/06/22**	**71-75**
29	Develop a program to handle all mouse events and window events	**09/06/22**	**76-77**
30	Develop a program to handle Key events.	**30/05/22**	**78-79**
31	Write a program to write to a file, then read from the file and display the contents on the console.	**30/05/22**	**80-82**
32	Write a program to copy one file to another**.**	**30/05/22**	**83-84**
33	Write a program that reads from a file having integers. Copy even numbers and odd numbers to separate files.	**30/05/22**	**85-86**

Name: Sandra PM
Roll No:34
Batch:MCA
Date:5/04/22

Object oriented programming lab

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 = "13wtno1";
   p1.pname = "camlin notebook";
   p1.price = 45;
   System.out.println("\nProduct1:");
   p1.details();

   Product p2 = new Product();
   p2.pcode = "13wtno2";
   p2.pname = "pinpoint pen";
   p2.price = 10;
```

```
        System.out.println("\nProduct2:");


       p2.details();
       Product p3 = new Product();
        p3.pcode = "13wtno3";
        p3.pname = "classmates notebook";
        p3.price = 50;
        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\Desktop\oops>javac ProductDetails.java

C:\Users\Student\Desktop\oops>java ProductDetails

Product1:
Product Details
PCode:13wtno1
PName:camlin notebook
Price:45.0

Product2:
Product Details
PCode:13wtno2
PName:pinpoint pen
Price:10.0

Product3:
Product Details
PCode:13wtno3
PName:classmates notebook
Price:50.0


Product with lowest price is:
Product Details
PCode:13wtno2
PName:pinpoint pen
Price:10.0

C:\Users\Student\Desktop\oops>
```

Object oriented programming lab

Name: Sandra PM
Roll No:34
Batch:MCA
Date:5/04/22

Experiment No.: 2

Aim

Read 2 matrices from the console and perform matrix addition.

Source code:

```
 import java.util.Scanner;
class AddMatrix {
public static void main (String args[]){
int row,col,i,j;
Scanner in =new Scanner(System.in);
System.out.println("enter the rows");
row =in.nextInt();
System.out.println("enter the 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 in first matrix");

for(i=0;i<row;i++)
{
        for(j=0;j<col;j++)
         mat1[i][j]=in.nextInt();
   System.out.println();
}
System.out.println("enter the elements in second matrix");
for(i=0;i<row;i++)
{
        for(j=0;j<col;j++)
        {
                mat2[i][j]=in.nextInt();
                System.out.println();
        }
        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:

```
C:\Users\Student\Desktop\oops>javac AddMatrix.java

C:\Users\Student\Desktop\oops>java AddMatrix
enter the rows
3
enter the columns
3
enter the elements in first matrix
1 2 3

1 2 3

1 2 3

enter the elements in second matrix
3 4


5 6 7

sum of matrices
4       6       8
1       2       3
1       2       3

C:\Users\Student\Desktop\oops>java AddMatrix
enter the rows
4
```

Name: Sandra P M
Roll No:34
Batch:MCA B
Date:06-04-2022

OBJECT ORIENTED PROGRAMMING LAB

Experiment No.: 3

Aim

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 ComplexNumber {
public static void main(String[] args){
Complex C1 = new Complex(5, 2);
C1.printComplexNumber();
Complex C2 = new Complex(8, 7);
C2.printComplexNumber();
Complex C3 = new Complex();
C3 = C3.addComp(C1, C2);
System.out.print("Sum of ");
C3.printComplexNumber();
}
}
```

Output

```
14 errors

C:\Users\Sandra\OneDrive\Desktop\oops>javac ComplexNumber.java

C:\Users\Sandra\OneDrive\Desktop\oops>java ComplexNumber
Complex number: 5 + 2i
Complex number: 8 + 7i
Sum of Complex number: 13 + 9i

C:\Users\Sandra\OneDrive\Desktop\oops>
```

Name: Sandra P M
Roll No:34
Batch:MCA B
Date:06-04-2022

OBJECT ORIENTED PROGRAMMING LAB

Experiment No.: 3

Aim

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 ComplexNumber {
public static void main(String[] args){
Complex C1 = new Complex(5, 2);
C1.printComplexNumber();
Complex C2 = new Complex(8, 7);
C2.printComplexNumber();
Complex C3 = new Complex();
C3 = C3.addComp(C1, C2);
System.out.print("Sum of ");
C3.printComplexNumber();
}
}
```

Output

```
14 errors

C:\Users\Sandra\OneDrive\Desktop\oops>javac ComplexNumber.java

C:\Users\Sandra\OneDrive\Desktop\oops>java ComplexNumber
Complex number: 5 + 2i
Complex number: 8 + 7i
Sum of Complex number: 13 + 9i

C:\Users\Sandra\OneDrive\Desktop\oops>
```

OBJECT ORIENTED PROGRAMMING

Name: Sandra P M
Roll No:34
Batch: MCA B
Date:06/04/22

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

```
C:\Users\Sandra\OneDrive\Desktop\oops>java Symetric
Enter the number of rows:2
Enter the number of columns:2
Enter the matrix elements:
6 8
5 7
The matrix formed is:
6       8
5       7

The matrix is not symmetric
C:\Users\Sandra\OneDrive\Desktop\oops>java Symetric
Enter the number of rows:3
Enter the number of columns:3
Enter the matrix elements:
2  3 6
3  4 5
6  5 9
The matrix formed is:
2       3       6
3       4       5
6       5       9

The matrix is symmetric
C:\Users\Sandra\OneDrive\Desktop\oops>
```

OBJECT ORIENTED PROGRAMMING LAB

Experiment No.: 5

Name: Sandra P M
Roll No:34
Batch: MCA B
Date:17-05-2022

Aim

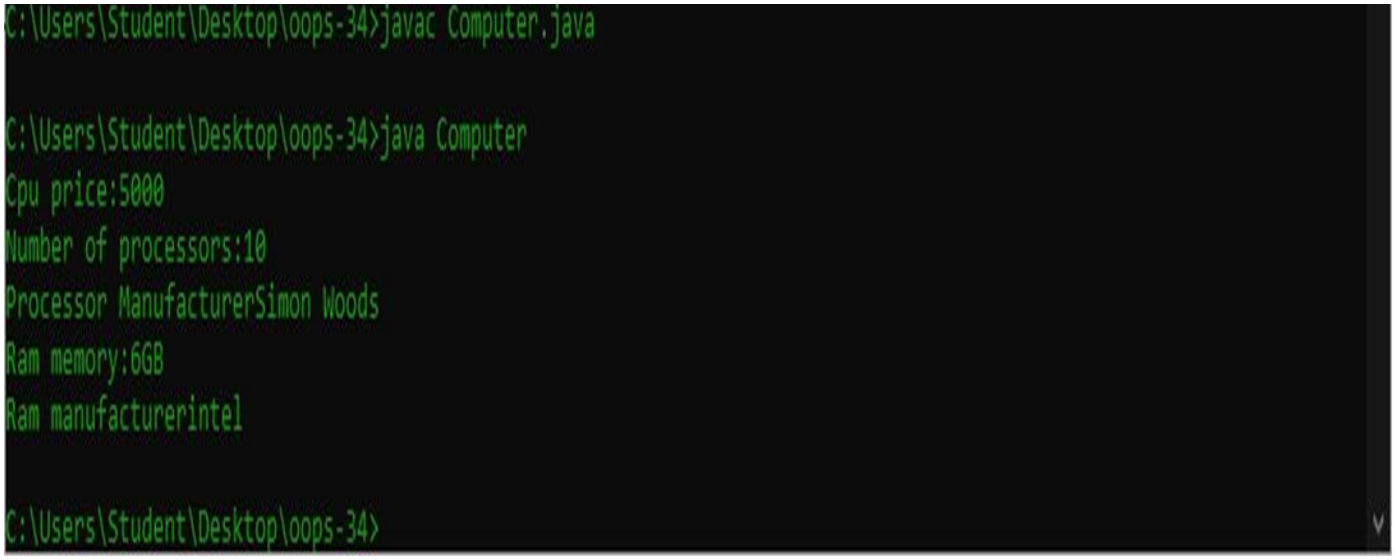
Create CPU with attribute price. Create inner class Processor (no. of cores, manufacturer) and static nested class RAM (memory, manufacturer). Create an object of CPU and print information of Processor and RAM.

Procedure

```
class Cpu
{
 int price=5000;
  class Processor
   {
    int nop=10;
          String manf="Simon Woods";
   }
   static class Ram
   {
   static String memory="6GB";
   String manf="intel";
   }
}
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);
```

```
}
}
```

Output


C:\Users\Student\Desktop\oops-34>javac Computer.java
C:\Users\Student\Desktop\oops-34>java Computer
Cpu price:5000
Number of processors:10
Processor ManufacturerSimon Woods
Ram memory:6GB
Ram manufacturerintel
C:\Users\Student\Desktop\oops-34>

OBJECT ORIENTED PROGRAMMING LAB

Name: Sandra PM
Roll No:34
Batch: MCA b
Date:22/04/2022

Experiment No.: 6

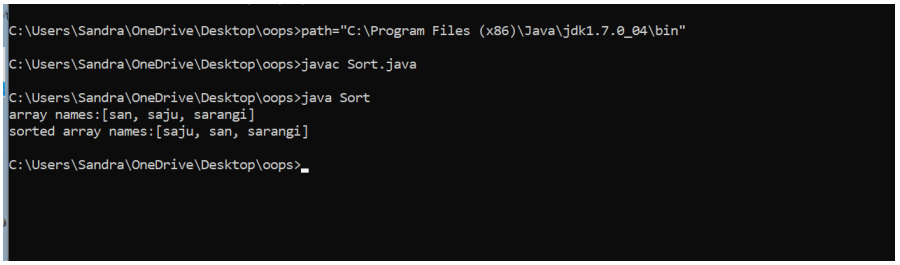
Aim

Program to Sort strings

Procedure

```
import java.util.Arrays;
 public class Sort{

        public static void main(String args[])
        {
                String [] names={"san","saju","sarangi"};
                System.out.println("array names:"+Arrays.toString(names));

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

Output Screenshot



Name: Sandra PM
Roll No:34
Batch: MCA B
Date:22/04/2022

OBJECT ORIENTED PROGRAMMING LAB

Experiment No.: 7

Aim

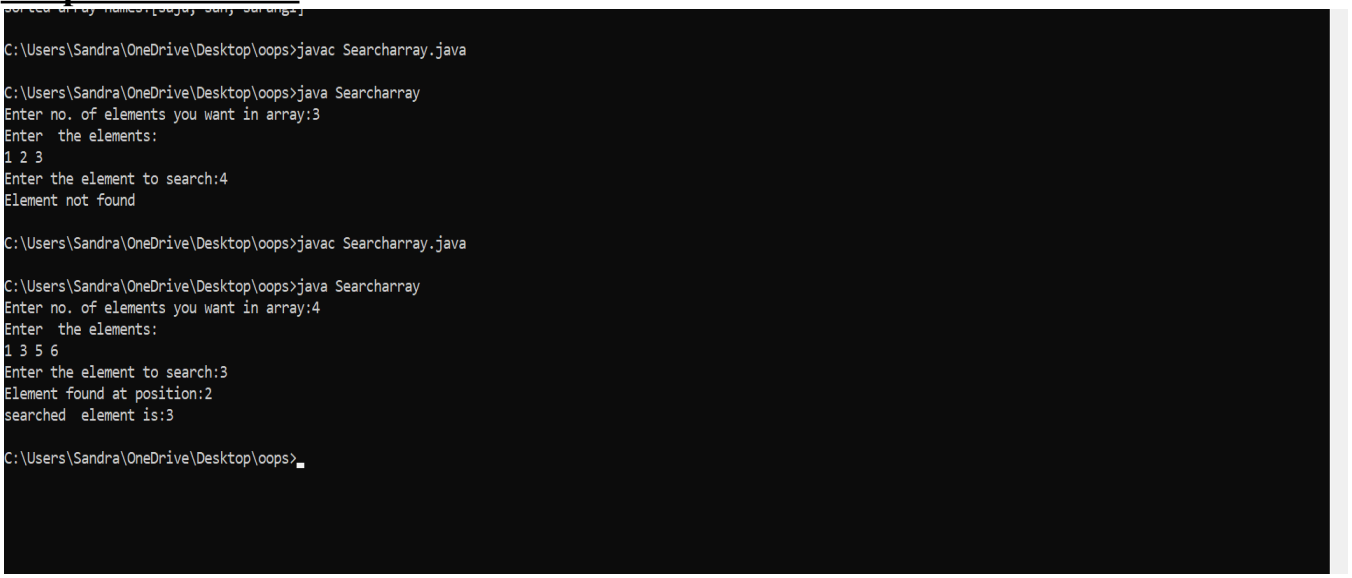
Search an element in an array.

Procedure

```
import java.util.Scanner;
public class Searcharray
{
 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



Name:Sandra P M
Roll No:34
Batch:MCA B
Date:22/04/22

OBJECT ORIENTED PROGRAMMING LAB

Experiment No.: 8

Aim

Perform string manipulations

Procedure

```
public class StringManipulation {

   public static void main(String[] args){
      String str1= "This is my first job ", str2="and I like it.";

      System.out.println("The string 01 is : "+str1+"\nString 02 is : "+str2);
      String strconcat= str1+str2;
      System.out.println("\nThe concatenation of two strings is : "+strconcat);

      String strUppercase= str1.toUpperCase();
      System.out.println("\nNormal String to uppercase string is : "+strUppercase);

      String strLowercase= str2.toLowerCase();
      System.out.println("\nNormal String to lowercase string is : "+strLowercase);

      String strsubString= str1.substring(5);
      System.out.println("\nSubstring of the string is : "+strsubString);

      String strtrim= str1.trim();
      System.out.println("\nString trim is given by : "+strtrim);

      boolean strcontains= str1.contains("my");
      System.out.println("\nCheck if the string 01 contains -'my' : "+strcontains);

      int strlength= str2.length();
      System.out.println("\nThe length of the string 02 is : "+strlength);
```

```
    }
}
```

Output

```
C:\Users\Student\Desktop\oops-34>javac StringManipulation.java

C:\Users\Student\Desktop\oops-34>java StringManipulation
The string 01 is : This is  first oops programming
String 02 is : and I like it.

The concatenation of two strings is : This is  first oops programming and I like it.

Normal String to uppercase string is : THIS IS  FIRST OOPS PROGRAMMING

Normal String to lowercase string is : and i like it.

Substring of the string is : is  first oops programming

String trim is given by : This is  first oops programming

Check if the string 01 contains -'my' : false

The length of the string 02 is : 14

C:\Users\Student\Desktop\oops-34>
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Sandra P M
Roll No:34
Batch : MCA B
Date:17/05/22

Experiment No.: 9

Aim

To create a class for Employee having attributes eNo, eName, eSalary. Read Nemployee information and Search for an employee given eNo, using the concept of Array of Objects.

Procedure

```
import java.util.*;
public class Employee {
int eNo;
String eName;
int eSalary;
public void read(){
Scanner sc= new Scanner(System.in);
System.out.print("Enter ID : ");
eNo = Integer.parseInt(sc.nextLine());
System.out.print("Enter Name : ");
eName = sc.nextLine();
System.out.print("Enter monthly salary : ");
eSalary = Integer.parseInt(sc.nextLine());
}
public void display(){
System.out.println("Name : "+ eName );
}

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

```
int i,n=3;
int No;
Employee emp[] = new Employee[n];
for(i=0;i&lt;n;i++){
emp[i] = new Employee();
emp[i].read();


}
System.out.println("Search");
Scanner s= new Scanner(System.in);
System.out.print("Enter ID : ");
No = s.nextInt();
for(i=0;i&lt;n;i++){
if(emp[i].eNo == No){
emp[i].display();
break;
}

}
}
}
```

Output

```
Enter ID : 1
Enter Name : jomol
Enter monthly salary : 12222
Enter ID : 2
Enter Name : rosna
Enter monthly salary : 12333
Enter ID : 3
Enter Name : ann
Enter monthly salary : 34444
Search
Enter ID : 1
Name : jomol
```

Object oriented programming lab

Name: sandra pm
Roll No:34
Batch: mca-b
Date:17-5-2022

Experiment No.: 10

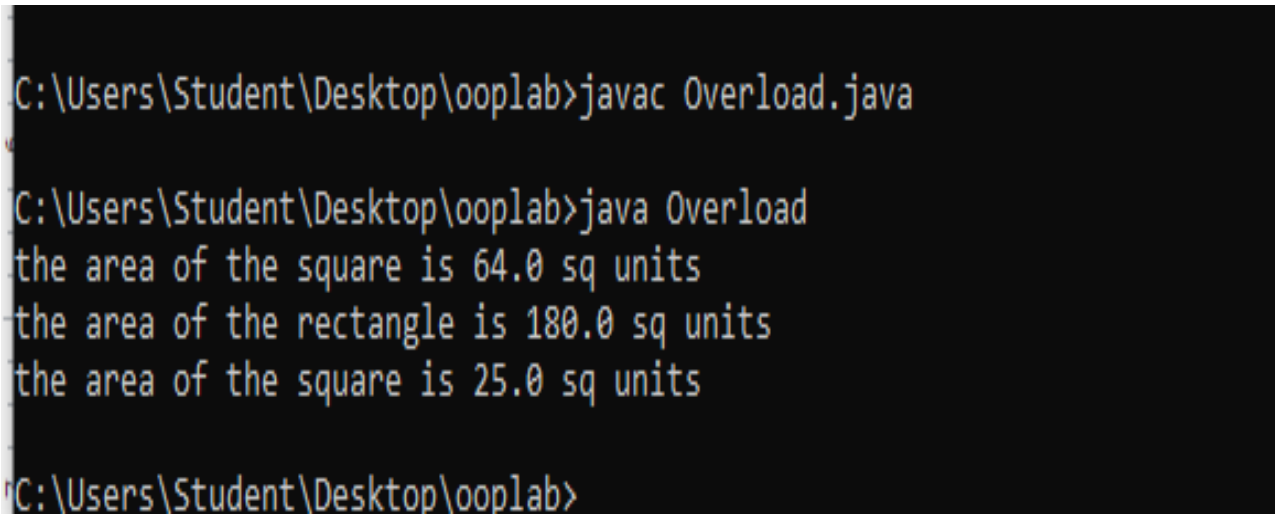
Aim

Area of different shapes using overloaded functions

Procedure

```
class OverloadDemo
{
   void area(float x)
   {
      System.out.println("the area of the square is "+Math.pow(x, 2)+" sq units");
   }
   void area(float x, float y)
   {
      System.out.println("the area of the rectangle is "+x*y+" sq units");
   }
   void area(double x)
   {
      double z = 3.14 * x * x;
      System.out.println("the area of the circle is "+z+" sq units");
   }
}
class Overload
{
    public static void main(String args[])
        {
          OverloadDemo obj = new OverloadDemo();
          ob.area(8);
          ob.area(12,15);
          ob.area(5);
       }
                }
```

Output Screenshot



Object oriented programming lab

Name: sandra pm
Roll No:34
Batch: mca-b
Date:17-5-2022

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 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.

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

```
C:\Users\mca\Desktop\javaprgm>javac EMPS.java

C:\Users\mca\Desktop\javaprgm>java EMPS
****EMPLOYEE DETAILS****
EmpID   : 101
Name    : Retha
Address : Rose villa
Salary  : 50000
Dept Name : MCA
Subject Name : DS
****EMPLOYEE DETAILS****
EmpID   : 102
Name    : Rina
Address : Deepalayam
Salary  : 110000
Dept Name : BBA
Subject Name : Commerce

C:\Users\mca\Desktop\javaprgm>
```

Object oriented programming lab

Name: sandra pm
Roll No:34
Batch: mca-b
Date:17-5-2022

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

```
Enter the name of the 1 teacher : saju
Enter the gender of the 1 teacher : male
Enter the address of the 1 teacher : Amrutham
Enter the age of the 1 teacher : 26
Enter the emp ID of the 1 teacher : 1001
Enter the company name of the 1 teacher : NS
Enter the qualification of the 1 teacher : Msc
Enter the salary of the 1 teacher : 35000
Enter the teacher ID of the 1 teacher : 1001
Enter the department of the 1 teacher : Cs
Enter the subject of the 1 teacher : Maths


Enter the name of the 2 teacher : Sandra
Enter the gender of the 2 teacher : female
Enter the address of the 2 teacher : puthuvalveli
Enter the age of the 2 teacher : 22
Enter the emp ID of the 2 teacher : 103
Enter the company name of the 2 teacher : Sanz
Enter the qualification of the 2 teacher : Mca
Enter the salary of the 2 teacher : 35000
Enter the teacher ID of the 2 teacher : 1002
Enter the department of the 2 teacher : cs
Enter the subject of the 2 teacher : computerscience


The name of the 1st is: saju
The gender of the 1st is: male
The address of the 1st is: Amrutham
The age of the 1st is: 26
The employee ID of the 1st is: 1001
The Company name of the 1st is: NS
The qualification of the 1st is: Msc
The salary of the 1st is: 35000.0
The teacher ID of the 1st is: 1001
The department of the 1st is: Cs
The subject of the 1st is: Maths


The name of the 2nd is: Sandra
The gender of the 2nd is: female
The address of the 2nd is: puthuvalveli
```

```
The age of the 2nd is: 22
The employee ID of the 2nd is: 103
The Company name of the 2nd is: Sanz
The qualification of the 2nd is: Mca
The salary of the 2nd is: 35000.0
The teacher ID of the 2nd is: 1002
The department of the 2nd is: cs
The subject of the 2nd is: computerscience


C:\Users\Student\Desktop\oops-34>
```

Object oriented programming lab

Name: sandra pm
Roll No:34
Batch: mca-b
Date:17-5-2022

Experiment No:13

Aim

4. 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 p_id;
 String p_name;
 Publisher(int p_id,String p_name)
{ this.p_id=p_id;
 this.p_name=p_name;
 }
}
class Book extends Publisher
{ int b_id;
 String b_name;
 Book(int p_id, String p_name, int b_id, String b_name)
{ super(p_id, p_name);
 this.b_id=b_id;
 this.b_name=b_name;
 }
}
class Literature extends Book
{ String cat;
 Literature(int p_id, String p_name, int b_id, String b_name)
{ super(p_id, p_name, b_id, b_name);
 this.cat="Literature";
 }
 void Display4()
{ System.out.println("\n\n.....Literature book details.....");
 System.out.println("Category name : "+this.cat);
System.out.println("Publisher id : "+this.p_id);
 System.out.println("Publisher name : "+ this.p_name);
 System.out.println("Book id : " +this.b_id);
```

```
 System.out.println("Book name : "+ this.b_name);
 }
}
class Fiction extends Book
{ String cat;
 Fiction(int p_id, String p_name, int b_id, String b_name)
{ super(p_id, p_name, b_id, b_name);
 this.cat="Fiction";
 }
 void Display4()
{ System.out.println("\n\n.....Fiction book details.....");
 System.out.println("Category name : "+this.cat);
System.out.println("Publisher id : "+this.p_id);
 System.out.println("Publisher name : "+ this.p_name);
 System.out.println("Book id : " +this.b_id);
System.out.println("Book name : "+ this.b_name);
 }
}
public class PublisherBooks
{ public static void main(String[] args)
 { int p_id, b_id;
 String p_name, b_name,t;
 Scanner s=new Scanner(System.in);
System.out.print("\nEnter the type of book(Type 'l/L' for Literature/'f/F' for Fiction)? ");
t=s.next();
if(t.equals("l")||t.equals("L"))
 { System.out.print("\nEnter the Publisher ID :");
 p_id=s.nextInt();
 System.out.print("Enter the Publisher Name :");
 p_name=s.next();
 System.out.print("Enter the Book ID :");
 b_id=s.nextInt();
 System.out.print("Enter the Book Name :");
 b_name=s.next();
 Literature lit=new Literature(p_id, p_name, b_id, b_name);
 lit.Display4();
}
else if(t.equals("f")||t.equals("F"))
 { System.out.print("\nEnter the Publisher ID :");
 p_id=s.nextInt();
 System.out.print("Enter the Publisher Name :");
p_name=s.next();
 System.out.print("Enter the Book ID :");
```

```
 b_id=s.nextInt();
 System.out.print("Enter the Book Name :");
 b_name=s.next();
 Fiction fic=new Fiction(p_id, p_name, b_id, b_name);
 fic.Display4();
 }
else
{ System.out.println("\n\n!!!!!!!!!!!!!!!Entry for type of book is  not valid!!!!!!!!!!!!!! ");
}
}
}
```

Output

```
C:\Users\Student\Desktop\oops-34>javac PublisherBooks.java

C:\Users\Student\Desktop\oops-34>java PublisherBooks

Enter the type of book(Type 'l/L' for Literature/'f/F' for Fiction)? l

Enter the Publisher ID :120
Enter the Publisher Name :sandra
Enter the Book ID :1206
Enter the Book Name :wings of fire


.............Literature book details...........
Category name : Literature
Publisher id : 120
Publisher name : sandra
Book id : 1206
Book name : wings

C:\Users\Student\Desktop\oops-34>
```

Object oriented programming lab

Name: sandra pm
Roll No:34
Batch: mca-b
Date:17-5-2022

Experiment No:14

Aim:

5. 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

```
C:\Users\mca\Desktop\javaprgm>javac result.java

C:\Users\mca\Desktop\javaprgm>java result

Enter the Sports Details of Student

 Sport:
football

 Sport Rating  out of 10:
78

Enter the Sports Details of Student

 Academic Grade:
1

 Overall percentage:
78

Sports Details of Student
Sport :football
Rating :78

Academic Details of Student
Academic Grade :1
Overall percentage :78.0

C:\Users\mca\Desktop\javaprgm>
```

OBJECT ORIENTED PROGRAMMING LAB

Experiment No.: 15

Name: sandra pm
Roll No:34
Batch :S2 MCA
Date:24/05/2022

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.

Procedure

```
import java.util.Scanner;
interface Circlerect{
  void area();
  void perimeter();

}
class Circle implements Circlerect{
                int r;
        double pi = 3.14, area,perimeter;
public void area(){

        Scanner s = new Scanner(System.in);
        System.out.println("Enter radius of circle:");
        r = s.nextInt();
        area = pi * r * r;
        System.out.println("Area of circle:"+area);
                }
public void perimeter(){
                perimeter = 2 * pi * r;
        System.out.println("Perimeter of circle:"+perimeter);
                }
          }
class Rectangle implements Circlerect{
                int l,b,area,perimeter;

public void area(){
                Scanner s = new Scanner(System.in);
                System.out.println("Enter length of rectangle:");
```

```
        l=s.nextInt();
        System.out.println("Enter breadth of rectangle:");
        b=s.nextInt();
                area=l*b;
                 System.out.println("Area of rectangle:"+area);
                 }
public void perimeter(){

                        perimeter=2 * (l + b);
                        System.out.println("Perimeter of rectangle:"+perimeter);
                }
                }
public class Interfacecirclerect{
public static void main(String args []){
        int n;
while(true){
        Circlerect c=new Circle();
        Circlerect r=new Rectangle();
 System.out.println("\n\nchoose the operations you can do:");
 System.out.println("1.circle\n\n 2.Rectangle\n\n3.exit");
 System.out.println("Enter your operations:");
 Scanner s = new Scanner(System.in);
 n=s.nextInt();
 switch(n)
 {
 case 1:System.out.println("circle");
                c.area();
                c.perimeter();
                break;
 case 2:System.out.println("Rectangle");
                r.area();
                r.perimeter();
                break;
case 3:System.exit(0);
                break;
}
}
}
}
```

Output

```
C:\Users\Sandra\OneDrive\Desktop\oops>java Interfacecirclerect


choose the operations you can do:
1.circle

 2.Rectangle

3.exit
Enter your operations:
1
circle
Enter radius of circle:
5
Area of circle:78.5
Perimeter of circle:31.400000000000002


choose the operations you can do:
1.circle

 2.Rectangle

3.exit
Enter your operations:
2
Rectangle
Enter length of rectangle:
5
Enter breadth of rectangle:
4
Area of rectangle:20
Perimeter of rectangle:18


choose the operations you can do:
1.circle

 2.Rectangle
```

Name: sandra pm
Roll No:34
Batch :S2 MCA
Date:24/05/2022

OBJECT ORIENTED PROGRAMMING LAB

Experiment No.: 16

Aim

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

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 BillCalc {
    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

```
Order no. #393758
Enter the date:22/03/2022
Enter how many products are there:
2

Enter product id:
202
Enter product name:
soap
Enter the Quantity:
10
Enter the unit price:
20

Enter product id:
204
Enter product name:
pen
Enter the Quantity:
15
Enter the unit price:
5
Date:22/03/2022
```

```
Product Id  Name   Quantity  unit price  Total
-------------------------------------------------------------
202             soap            10              20.0    200.0
204             pen             15              5.0     75.0
-------------------------------------------------------------
                                Net.Amount              275.0
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Sandra P M
Roll No:34
Batch: MCA B
Date:31-05-2022

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.

AreaCalculation.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();
   }
}
```

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);
   }
}
```

Rectangle.java

```
package graphics;

 interface  AreaInterface {
   public void Area();
 }

public class Rectangle implements AreaInterface{
   double length, breadth;
   public Rectangle(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 rectangle is : "+area);
   }
}
```

Circle.java

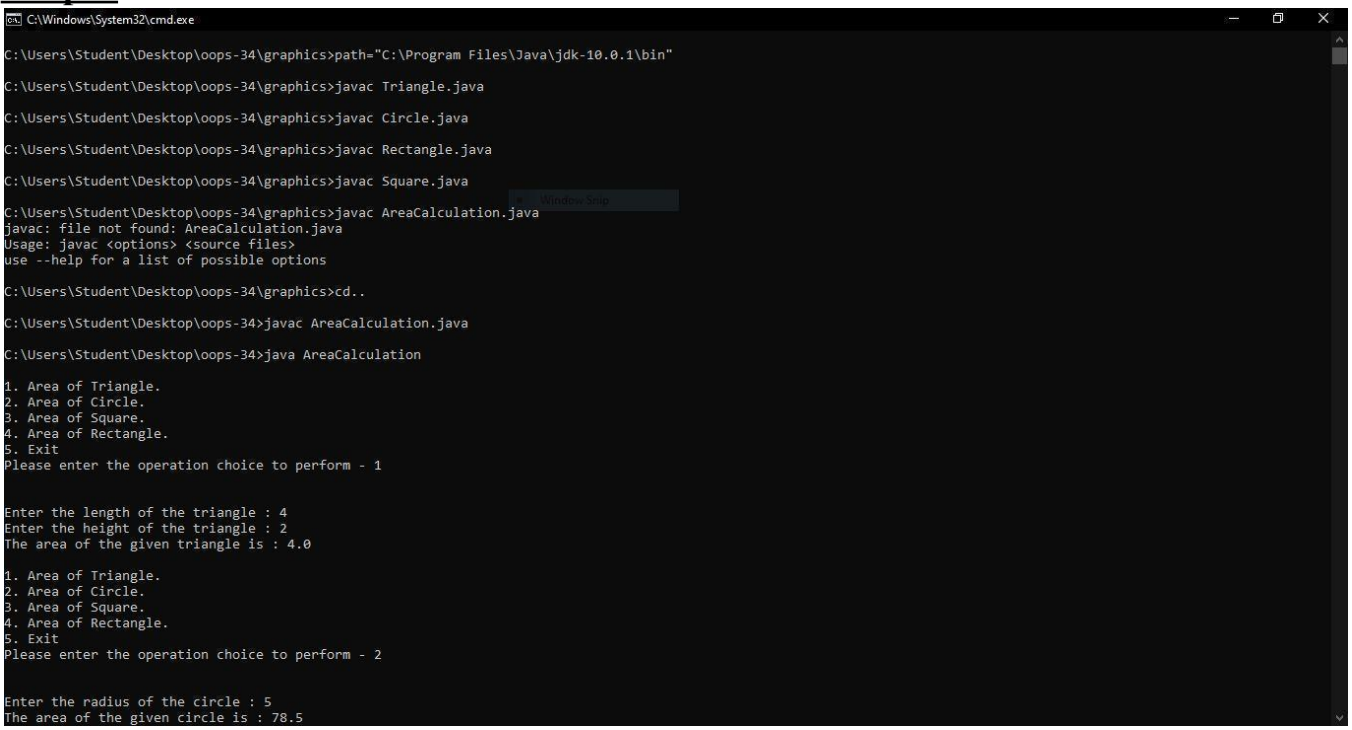
```
package graphics;

 interface AreaInterface {
   void Area();
}
```

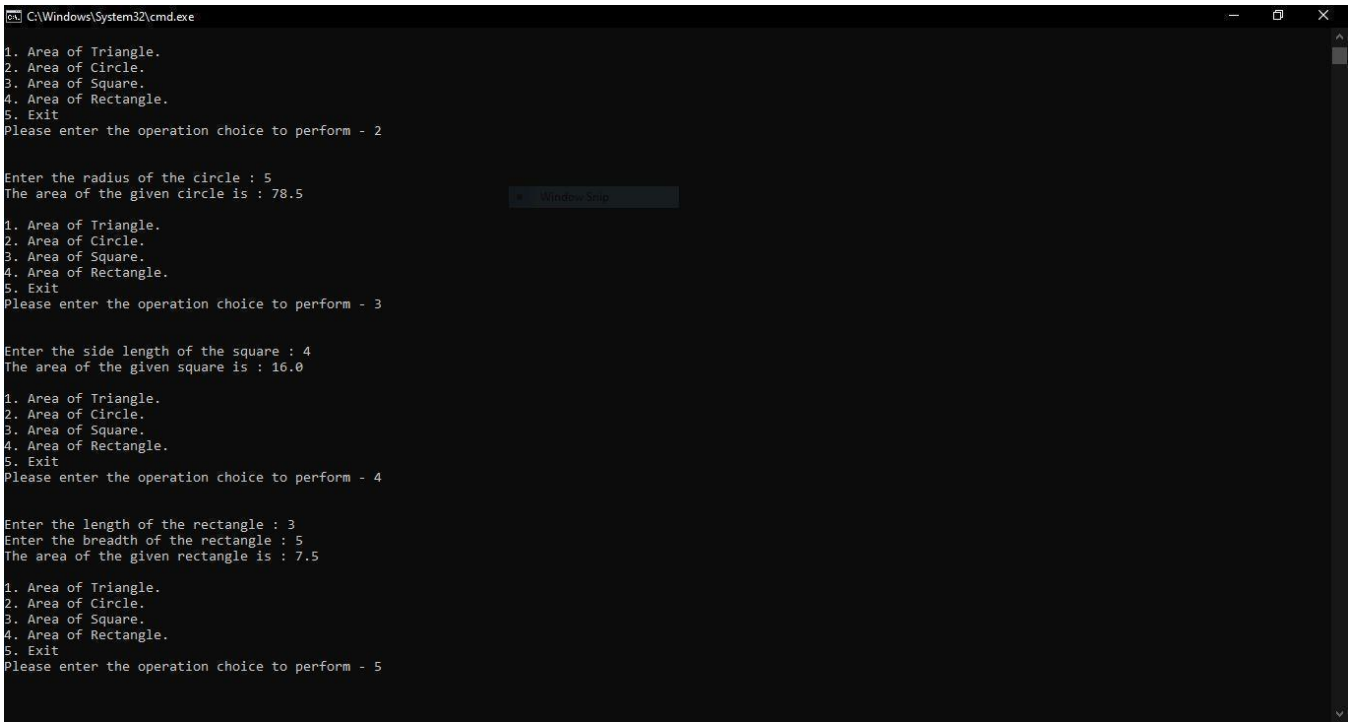
```
public class Circle implements AreaInterface{
    double radius;
    public Circle(double radius){
        this.radius= radius;
    }
    public void Area() {
        double area= 3.14 * this.radius * this.radius;
        System.out.println("The area of the given circle is : "+area);
    }
}
```

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);
    }
}
```

Output


C:\Windows\System32\cmd.exe
C:\Users\Student\Desktop\oops-34\graphics>path="C:\Program Files\Java\jdk-10.0.1\bin"
C:\Users\Student\Desktop\oops-34\graphics>javac Triangle.java
C:\Users\Student\Desktop\oops-34\graphics>javac Circle.java
C:\Users\Student\Desktop\oops-34\graphics>javac Rectangle.java
C:\Users\Student\Desktop\oops-34\graphics>javac Square.java
C:\Users\Student\Desktop\oops-34\graphics>javac AreaCalculation.java
javac: file not found: AreaCalculation.java
Usage: javac <options> <source files>
use --help for a list of possible options
C:\Users\Student\Desktop\oops-34\graphics>cd..
C:\Users\Student\Desktop\oops-34>javac AreaCalculation.java
C:\Users\Student\Desktop\oops-34>java AreaCalculation
1. Area of Triangle.
2. Area of Circle.
3. Area of Square.
4. Area of Rectangle.
5. Exit
Please enter the operation choice to perform - 1
Enter the length of the triangle : 4
Enter the height of the triangle : 2
The area of the given triangle is : 4.0
1. Area of Triangle.
2. Area of Circle.
3. Area of Square.
4. Area of Rectangle.
5. Exit
Please enter the operation choice to perform - 2
Enter the radius of the circle : 5
The area of the given circle is : 78.5


C:\Windows\System32\cmd.exe
1. Area of Triangle.
2. Area of Circle.
3. Area of Square.
4. Area of Rectangle.
5. Exit
Please enter the operation choice to perform - 2
Enter the radius of the circle : 5
The area of the given circle is : 78.5
1. Area of Triangle.
2. Area of Circle.
3. Area of Square.
4. Area of Rectangle.
5. Exit
Please enter the operation choice to perform - 3
Enter the side length of the square : 4
The area of the given square is : 16.0
1. Area of Triangle.
2. Area of Circle.
3. Area of Square.
4. Area of Rectangle.
5. Exit
Please enter the operation choice to perform - 4
Enter the length of the rectangle : 3
Enter the breadth of the rectangle : 5
The area of the given rectangle is : 7.5
1. Area of Triangle.
2. Area of Circle.
3. Area of Square.
4. Area of Rectangle.
5. Exit
Please enter the operation choice to perform - 5

OBJECT ORIENTED PROGRAMMING LAB

Experimentno:-18

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 = "san";

      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

```
C:\Users\Student\Desktop\oops-34>javac CustomExceptionExample1.java

C:\Users\Student\Desktop\oops-34>java CustomExceptionExample1
CustomExceptionExample1$InvalidUserException: Invalid username / password provided!
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Sandra P M
Roll No:34
Batch: MCA B
Date:31-05-2022

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

```
C:\Users\Student\Desktop\oops-34>javac Main.java

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>
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Sandra P M
Roll No:34
Batch: MCA B
Date:31-05-2022

Experiment no:20

AIM

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

```
C:\Users\Student\Desktop\oops-34>javac My.java

C:\Users\Student\Desktop\oops-34>java My
Enter Limit :
5
0  2  4
0 1 1 2 3 5 8 13
C:\Users\Student\Desktop\oops-34>
```

Name: Sandra P M
Roll No:34
Batch: MCA B
Date:31-05-2022

Experiment no:21

Aim

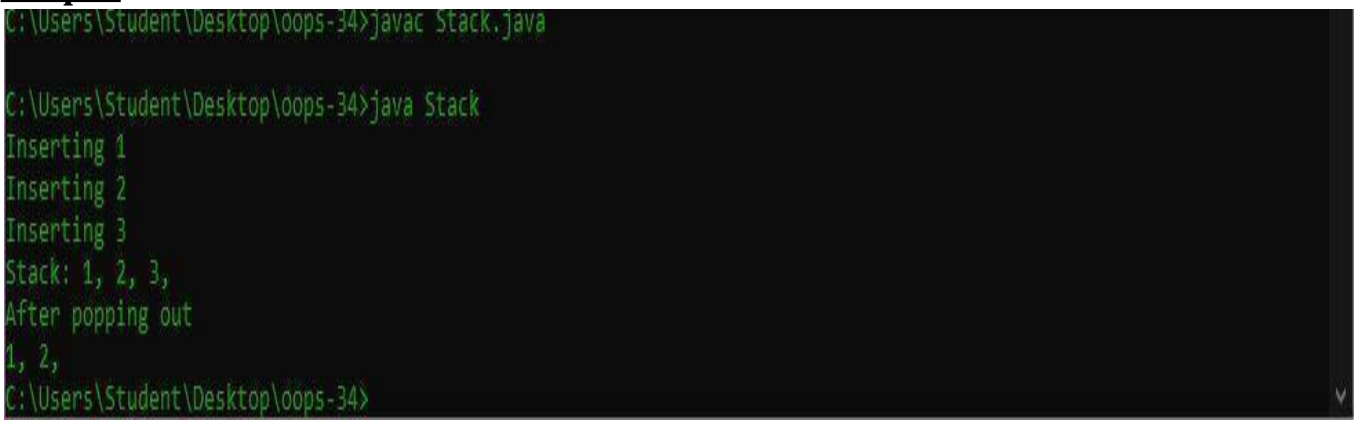
Program to create a generic stack and do the Push and Pop operations.

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


C:\Users\Student\Desktop\oops-34>javac Stack.java
C:\Users\Student\Desktop\oops-34>java Stack
Inserting 1
Inserting 2
Inserting 3
Stack: 1, 2, 3,
After popping out
1, 2,
C:\Users\Student\Desktop\oops-34>

Name: Sandra P M
Roll No:34
Batch: MCA B
Date:31-05-2022

Experiment no:22
AIM

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

Procedure

```
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

```
1 error

C:\Users\Student\Desktop\oops-34>javac Mycls.java

C:\Users\Student\Desktop\oops-34>java Mycls
[A, BB, C, D]
A
BB
[BB, C, D]
3
BB
C
D
[BB, C, D]
[]

C:\Users\Student\Desktop\oops-34>
```

Name: Sandra P M
Roll No:34
Batch: MCA B
Date:31-05-2022

Experiment no:23

Aim

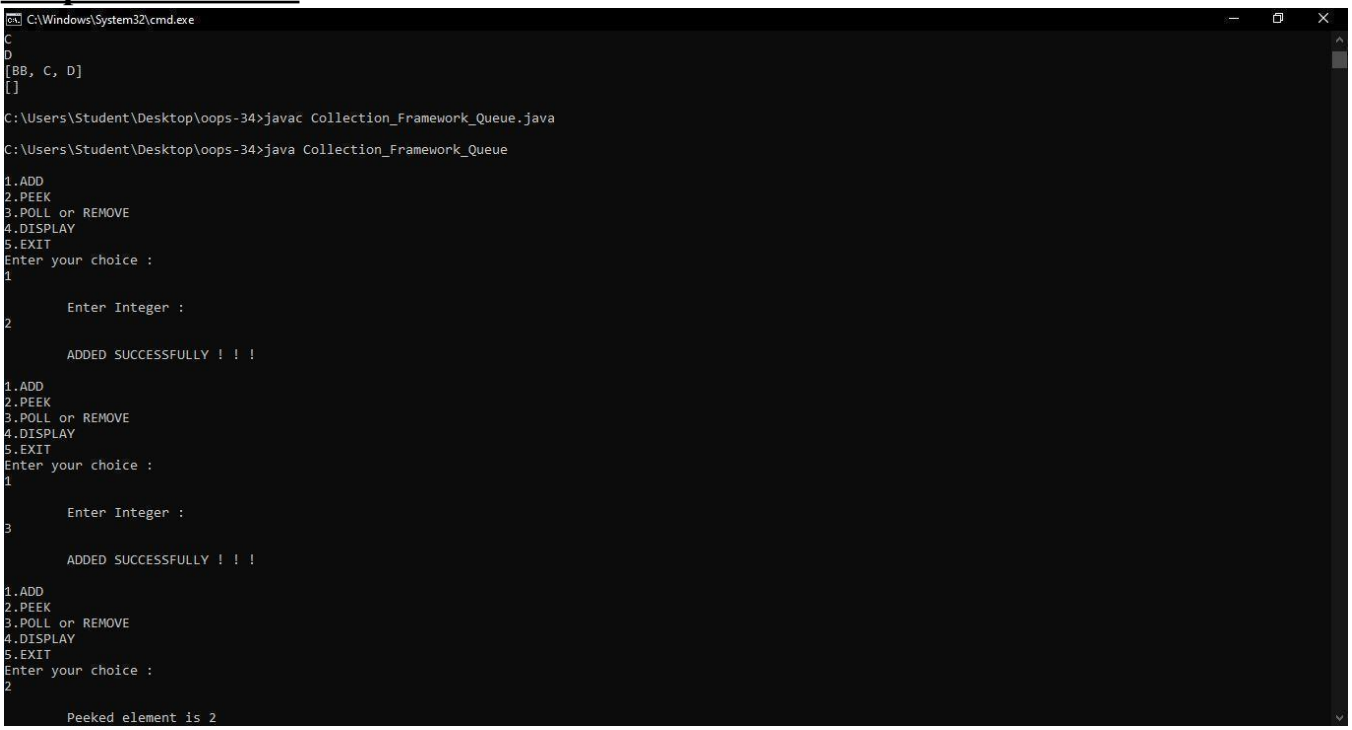
Program to demonstrate the creation of queue object using the Priority Queue class.

procedure

```
import java.util.*;

class Collection_Framework_Queue {
   public static void main(String args[]) {
      Queue<Integer> q = new PriorityQueue<Integer>(new Comp());
      int ch;
      Scanner sc = new Scanner(System.in);
      do {
         System.out.println("\n1.ADD\n2.PEEK\n3.POLL or
REMOVE\n4.DISPLAY\n5.EXIT");
         System.out.println("Enter your choice : ");
         ch = sc.nextInt();
         switch (ch) {
            case 1:
               System.out.println("\n\tEnter Integer : ");
               int n1 = sc.nextInt();
               q.add(n1);
               System.out.println("\n\tADDED SUCCESSFULLY ! ! ! ");
               break;
            case 2:
               if (q.isEmpty()) {
                  System.out.print("\n\tQueue Empty ! ! !");
               } else {
                  System.out.print("\n\tPeeked element is " + q.peek());
               }
               break;
            case 3:
               if (!q.isEmpty()) {
                  System.out.print("\n\tRemoved element is " + q.poll());
               } else {
```

```
                System.out.print("\n\tQueue Empty ! ! !");
            }
            break;
        case 4:
            if (!q.isEmpty()) {
                System.out.print("\nSize of queue : " + q.size());
                System.out.print("\nQueue elements  : " + q);
                System.out.println("\nQueue elements are");
                for (int i : q) {
                    System.out.println(i);
                }
            } else {
                System.out.print("\n\tQueue Empty ! ! !");
            }
            break;
        case 5:
            break;
        default:
            System.out.println("\n\tPlease enter valid choice ! ! ! ");
        }
    } while (ch != 5);

  }
}


class Comp implements Comparator<Integer> {
  public int compare(Integer a, Integer b) {
    return a % 10 > b % 10 ? 1 : -1;
  }
}
```

Output Screenshot



Name: Sandra P M
Roll No:34
Batch: MCA B
Date:31-05-2022

Experiment no:24

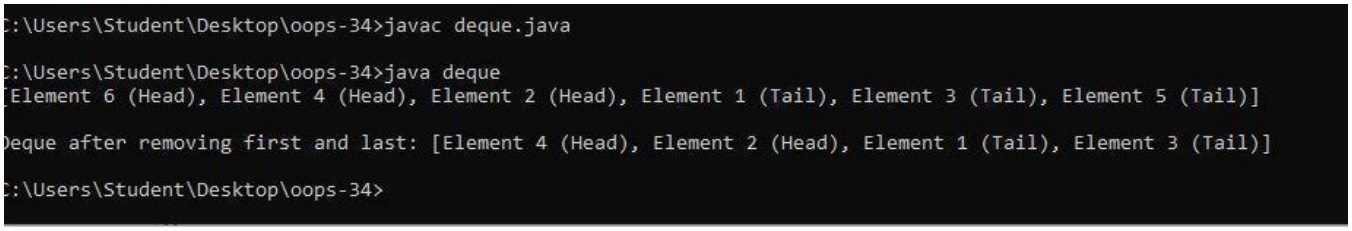
Aim

Program to demonstrate the addition and deletion of elements in deque.

```
import java.util.*;
 class deque
{
 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


C:\Users\Student\Desktop\oops-34>javac deque.java
C:\Users\Student\Desktop\oops-34>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\Desktop\oops-34>

Name: Sandra P M
Roll No:34
Batch: MCA B
Date:31-05-2022

Experiment no:25

Aim

Write a Java program to compare two hash set

```
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");
        }
    }
}
```

```
C:\Users\Student\Desktop\oops-34>javac Exercise10.java

C:\Users\Student\Desktop\oops-34>java Exercise10
Yes
No
Yes
No

C:\Users\Student\Desktop\oops-34>
```

Experiment no:26

Aim

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

```
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

```
C:\Users\Student\Desktop\oops-34>javac HashMapDemo.java
Note: HashMapDemo.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\Users\Student\Desktop\oops-34>java HashMapDemo
Jinu:3
sinu:2
Anu:1

C:\Users\Student\Desktop\oops-34>
```

OBJECT ORIENTED PROGRAMMING LAB

Name: sandra P M
Roll No:34
Batch:S2 MCA-B
Date:13/06/22

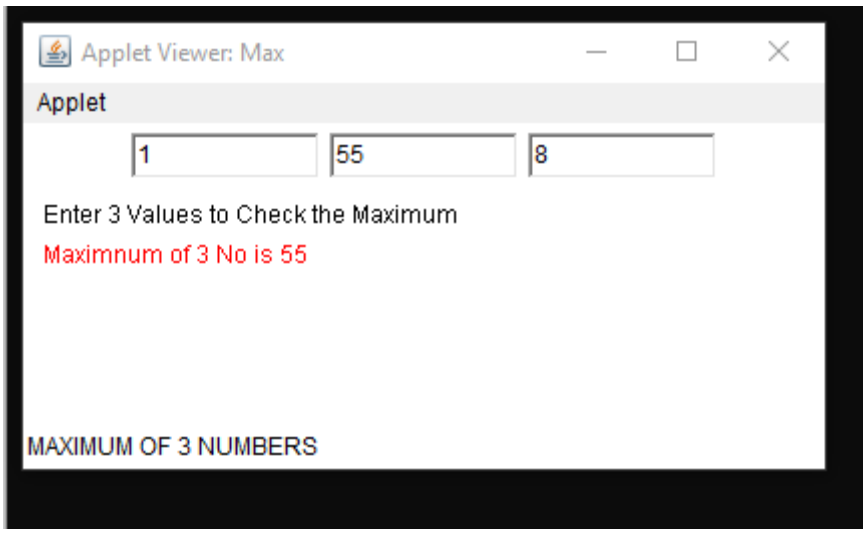
Experiment No.:27
Aim

Program to find maximum of three numbers using AWT
PROCEDURE

```
import java.awt.*;
import java.applet.*;

public class Max extends Applet
{
    TextField T1,T2,T3;

     public void init(){
        T1 = new TextField(10);
        T2 = new TextField(10);
        T3 = new TextField(10);

        add(T1);
        add(T2);
        add(T3);

        T1.setText("0");
        T2.setText("0");
        T3.setText("0");
      }

      public void paint(Graphics g){
          int m, a, x,result;
```

```
        String str;

        g.drawString("Enter 3 Values to Check the Maximum ",10,50);

        str=T1.getText();
        m=Integer.parseInt(str);
        str=T2.getText();
        a=Integer.parseInt(str);
        str=T3.getText();
        x=Integer.parseInt(str);

         g.setColor(Color.red);
         if (m>a) {
            if (m>x)
               result=m;
            else
               result=x;
           }
        else{
            if (a>x)
               result=a;
            else
               result=x;
          }
          g.drawString("Maximum of 3 No is "+result,10,70);
          showStatus("MAXIMUM OF 3 NUMBERS");
        }

        public boolean action(Event e, Object o){
            repaint();
            return true;
        }
}

      /* <applet code="Max" height=150 width=400>
        </applet> */
```

Output


Applet Viewer: Max
Applet
1
55
8
Enter 3 Values to Check the Maximum
Maximnum of 3 No is 55
MAXIMUM OF 3 NUMBERS

Name: Sandra P M
Roll No:34
Batch: MCA B
Date:09-06-2022

OBJECT ORIENTED PROGRAMMING LAB

Experiment No.: 28

Aim

Implement a simple calculator using AWT components

procedure

```
import java.awt.*;
import java.awt.event.*;
public class calculator implements ActionListener
{
   int c,n;
   String s1,s2,s3,s4,s5;
   Frame f;
   Button b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12,b13,b14,b15,b16,b17;
   Panel p;
   TextField tf;
   GridLayout g;
   calculator()
   {
      f = new Frame("My calculator");
      p = new Panel();
      f.setLayout(new FlowLayout());
      b1 = new Button("0");
      b1.addActionListener(this);
      b2 = new Button("1");
      b2.addActionListener(this);
      b3 = new Button("2");
      b3.addActionListener(this);
      b4 = new Button("3");
      b4.addActionListener(this);
      b5 = new Button("4");
      b5.addActionListener(this);
```

```
    b6 = new Button("5");
    b6.addActionListener(this);
    b7 = new Button("6");
    b7.addActionListener(this);
    b8 = new Button("7");
    b8.addActionListener(this);
    b9 = new Button("8");
    b9.addActionListener(this);
    b10 = new Button("9");
    b10.addActionListener(this);
    b11 = new Button("+");
    b11.addActionListener(this);
    b12 = new Button("-");
    b12.addActionListener(this);
    b13 = new Button("*");
    b13.addActionListener(this);
    b14 = new Button("/");
    b14.addActionListener(this);
    b15 = new Button("%");
    b15.addActionListener(this);
    b16 = new Button("=");
    b16.addActionListener(this);
    b17 = new Button("C");
    b17.addActionListener(this);
    tf = new TextField(20);
    f.add(tf);
    g = new GridLayout(4,4,10,20);
    p.setLayout(g);
    p.add(b1);p.add(b2);p.add(b3);p.add(b4);p.add(b5);p.add(b6);p.add(b7);p.add(b8);p.add(b9);
    p.add(b10);p.add(b11);p.add(b12);p.add(b13);p.add(b14);p.add(b15);p.add(b16);p.add(b17);
    f.add(p);
    f.setSize(300,300);
    f.setVisible(true);
  }
  public void actionPerformed(ActionEvent e)
  {
    if(e.getSource()==b1)
    {
      s3 = tf.getText();
      s4 = "0";
      s5 = s3+s4;
      tf.setText(s5);
    }
    if(e.getSource()==b2)
    {
      s3 = tf.getText();
      s4 = "1";
```

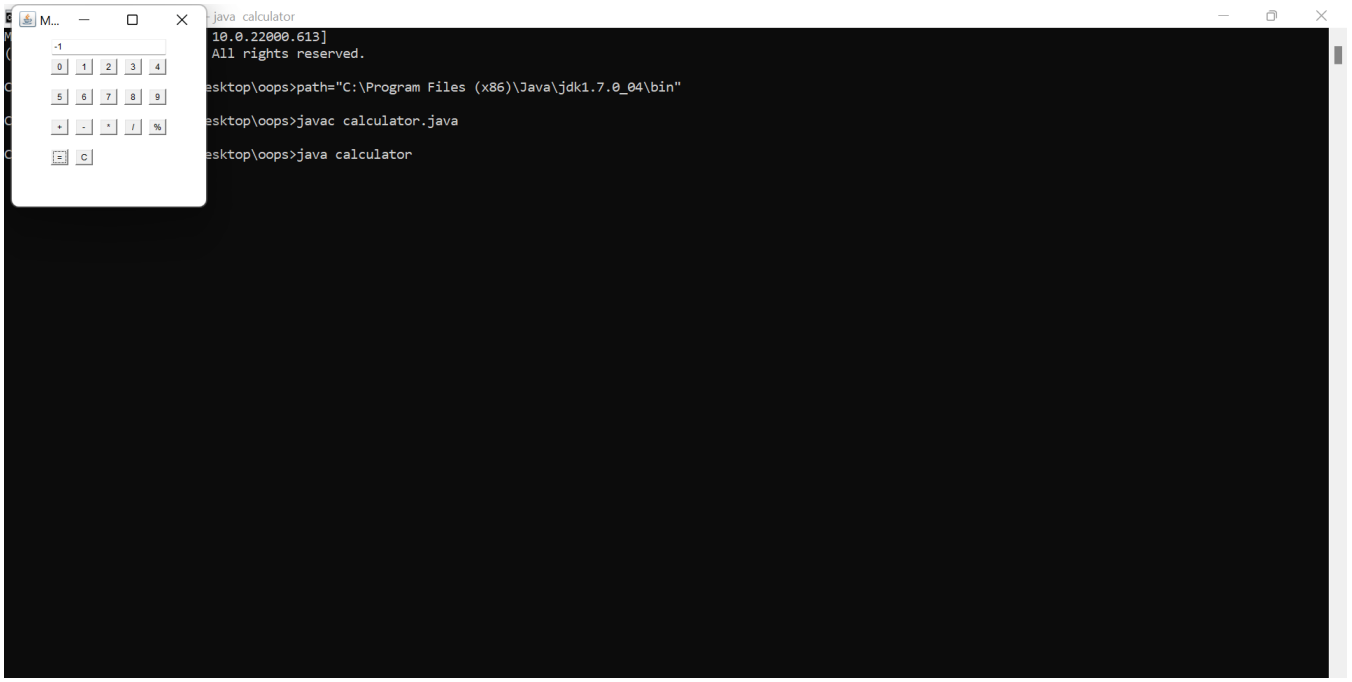
```
        s5 = s3+s4;
        tf.setText(s5);
    }
    if(e.getSource()==b3)
    {
        s3 = tf.getText();
        s4 = "2";
        s5 = s3+s4;
        tf.setText(s5);
    }if(e.getSource()==b4)
    {
        s3 = tf.getText();
        s4 = "3";
        s5 = s3+s4;
        tf.setText(s5);
    }
    if(e.getSource()==b5)
    {
        s3 = tf.getText();
        s4 = "4";
        s5 = s3+s4;
        tf.setText(s5);
    }
    if(e.getSource()==b6)
    {
        s3 = tf.getText();
        s4 = "5";
        s5 = s3+s4;
        tf.setText(s5);
    }
    if(e.getSource()==b7)
    {
        s3 = tf.getText();
        s4 = "6";
        s5 = s3+s4;
        tf.setText(s5);
    }
    if(e.getSource()==b8)
    {
        s3 = tf.getText();
        s4 = "7";
        s5 = s3+s4;
        tf.setText(s5);
    }
    if(e.getSource()==b9)
    {
        s3 = tf.getText();
```

```
        s4 = "8";
        s5 = s3+s4;
        tf.setText(s5);
    }
    if(e.getSource()==b10)
    {
        s3 = tf.getText();
        s4 = "9";
        s5 = s3+s4;
        tf.setText(s5);
    }
    if(e.getSource()==b11)
    {
        s1 = tf.getText();
        tf.setText("");
        c=1;

    }
    if(e.getSource()==b12)
    {
        s1 = tf.getText();
        tf.setText("");
        c=2;

    }
    if(e.getSource()==b13)
    {
        s1 = tf.getText();
        tf.setText("");
        c=3;

    }
    if(e.getSource()==b14)
    {
        s1 = tf.getText();
        tf.setText("");
        c=4;

    }
    if(e.getSource()==b15)
    {
        s1 = tf.getText();
        tf.setText("");
        c=5;

    }
    if(e.getSource()==b16)
```

```
    {
      s2 = tf.getText();
      if(c==1)
      {
        n = Integer.parseInt(s1)+Integer.parseInt(s2);
        tf.setText(String.valueOf(n));
      }
      else
      if(c==2)
      {
        n = Integer.parseInt(s1)-Integer.parseInt(s2);
        tf.setText(String.valueOf(n));
      }
      else
      if(c==3)
      {
        n = Integer.parseInt(s1)*Integer.parseInt(s2);
        tf.setText(String.valueOf(n));
      }
      if(c==4)
      {
        try
        {
          int p=Integer.parseInt(s2);
          if(p!=0)
          {
                    n = Integer.parseInt(s1)/Integer.parseInt(s2);
          tf.setText(String.valueOf(n));
           }
           else
             tf.setText("infinite");

        }
        catch(Exception i){}
      }
      if(c==5)
      {
        n = Integer.parseInt(s1)%Integer.parseInt(s2);
        tf.setText(String.valueOf(n));
      }
    }
    if(e.getSource()==b17)
    {
      tf.setText("");
    }
  }
```

```
    public static void main(String[] abc)
    {
        calculator v = new calculator();
    }
}
```

output



Name: Sandra P M
Roll No:34
Batch: MCA B
Date:09-06-2022

Experiment No.:29

Aim

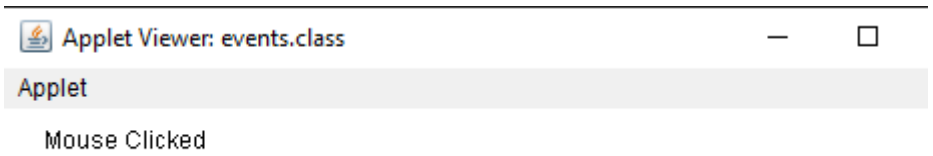
Develop a program to handle all mouse events and window events.

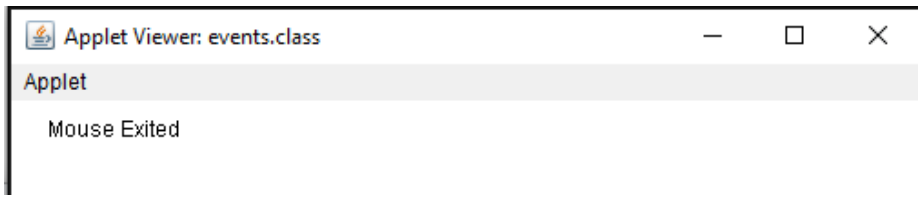
PROCEDURE

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
public class events extends Applet implements MouseListener
{
	String msg="Initial Message";
	public void init()
	{
		addMouseListener(this);
	}
	public void mouseClicked(MouseEvent me)
	{
		msg = "Mouse Clicked";
		repaint();
	}
	public void mousePressed(MouseEvent me)
	{
		msg = "Mouse Pressed";
		repaint();
	}
	public void mouseReleased(MouseEvent me)
	{
		msg = "Mouse Released";
		repaint();
	}
	public void mouseEntered(MouseEvent me)
```

```
        {
                msg = "Mouse Entered";
                repaint();
        }
        public void mouseExited(MouseEvent me)
        {
                msg = "Mouse Exited";
                repaint();
        }
        public void paint(Graphics g)
        {
                g.drawString(msg,20,20);
        }
}
/*
<applet code="events.class" height="300" width="500">
</applet>
*/
```

Output


Applet Viewer: events.class
Applet
Mouse Clicked


Applet Viewer: events.class
Applet
Mouse Exited

Name: Sandra P M
Roll No:34
Batch: MCA B
Date:09-06-2022

Experiment no:30

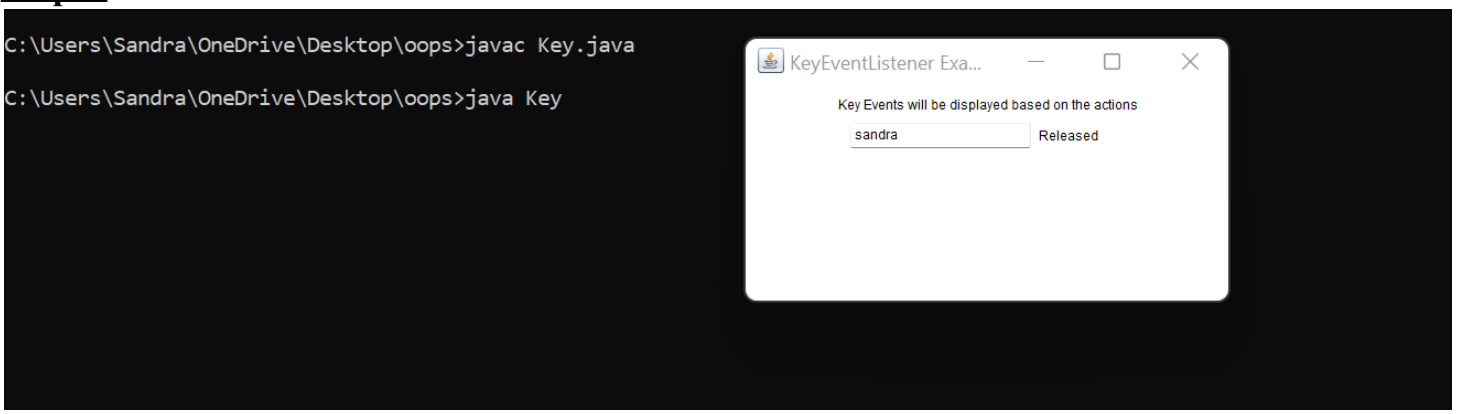
Aim

Develop a program to handle Key events.

```
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 Key implements KeyListener
{
   Label lb1, lbl2, lb;
   TextField tf1;
   Frame fr;
   String s;
   Key()
   {

      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 Key();
    }
}
```

output



Object oriented programming lab

Experiment No.: 31

Aim

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

```
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("file4.txt");

FileWriter fw = new FileWriter(file);
fw.write(var);
fw.close();
System.out.println("Reading File content");
FileReader fr = new FileReader("file4.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");
}
}
```

```
}
```

Output

```
C:\Users\Sandra\OneDrive\Desktop\oops>javac Reads.java

C:\Users\Sandra\OneDrive\Desktop\oops>java Reads
enter the text to create file:type Enter key 3 times to stop
sandra pm


reading file content
sandra pm
```

Experiment no:32

Name: Sandra PM
Roll No:34
Batch:MCA
Date:30/05/22

Aim

Write a program to copy one file to another.

```
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.*;
import java.util.*;
import java.io.File;
public class cp{
public static void main(String args[]){
Scanner sc=new Scanner(System.in);
System.out.println("enter the file name");
String source=sc.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=sc.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

```
1 error

C:\Users\mca\Desktop\oopslab>javac cp.java

C:\Users\mca\Desktop\oopslab>java cp
enter the file name
file4.txt
Reading from file file4.txt




 Enter the filename to copy
file5.txt
Copied from file4.txt to file5.txt Successfully..!

C:\Users\mca\Desktop\oopslab>
```

Experiment no:33

Aim

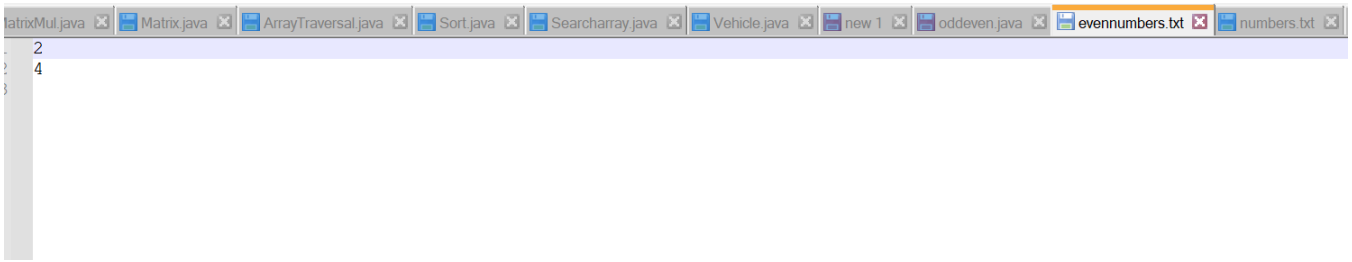
Write a program that reads from a file having integers. Copy even numbers and odd numbers to separate files.

```
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\Sandra\OneDrive\Desktop\oops>javac oddeven.java

C:\Users\Sandra\OneDrive\Desktop\oops>java oddeven

C:\Users\Sandra\OneDrive\Desktop\oops>
```

evennumbers.txt



oddnumbers.txt