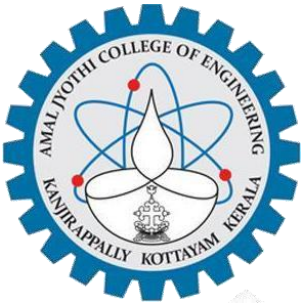
20MCA132 – OBJECT ORIENTED PROGRAMMING LAB

Lab Report Submitted By

JUSTIN V KALAPURA
Reg. No.: AJC21MCA-2068

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

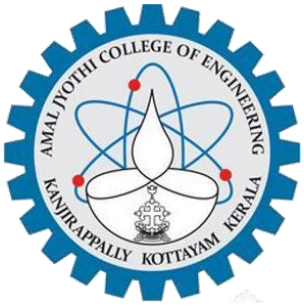
[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2021-2022

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 JUSTIN V KALAPPURA **(Reg.No:AJC21MCA-2068)** 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.

Ms.Sruthimol Kurian
Lab In-Charge

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

Internal Examiner

External Examiner

CONTENT

OBJECT ORIENTED PROGRAMMING LAB

Name: JUSTIN V KALAPPURA

Roll No:10

Batch:S2 MCA

Date:29/03/2022

Experiment No.: 1

Aim

Define a class "product" with data members pcode, pname, price .Create three objects in the class and find the product having lowest price.

Procedure

```
class Product
{
String   pcode;
String  pname;
double price;
        void details()
        {       System.out.println("Product code:"+pcode);
                System.out.println("Product name:"+pname);
                System.out.println("Price:"+price+"\n");
        }
}

public class Lowestproductprice
{
        public static void main(String[] args)
        {
        System.out.println("\t\nPRODUCT DETAILS:\n");
        Product pr=new Product();
        pr.pcode="p1";
        pr.pname="bread";
        pr.price=30;
        pr.details();
        Product pr1=new Product();
```

```
        pr1.pcode="p2";
        pr1.pname="jam";
        pr1.price=20;
        pr1.details();

        if(pr.price<pr1.price)
         {
                System.out.println("lowest price "+ pr.price);
         }
        else
         {
                 System.out.println("lowest price "+pr1.price);
         }
   }
}
```

Output Screenshot

```
D:\justin>javac Lowestproductprice.java

D:\justin>java Lowestproductprice

PRODUCT DETAILS:

Product code:p1
Product name:bread
Price:30.0

Product code:p2
Product name:jam
Price:20.0

lowest price 20.0
```

OBJECT ORIENTED PROGRAMMING LAB

Name: JUSTIN V KALAPPURA

Roll No:10

Batch: MCA

Date:06/04/22

Experiment No.: 2

Aim

Read 2 matrices from the console and perform matrix addition.

Procedure

Source Code

```
import java.util.Scanner;

public class Matrixaddition {
   public void Display(int [][] arr,int row,int col){
      for(int i=0;i<row;i++){
         for(int j=0;j<col;j++){
            System.out.print(arr[i][j]+"\t");
         }
         System.out.println();
      }
  }
   public static void main(String[] args) {
      int[][] mat1=new int[5][5];
      int[][] mat2=new int[5][5];
      int[][] mat3=new int[5][5];
      int rows1, cols1, rows2, cols2;

      Matrixaddition obj=new Matrixaddition();
      Scanner s=new Scanner(System.in);
      System.out.println("Enter the number of rows and columns of matrix1:");
      rows1=s.nextInt();
      cols1=s.nextInt();
      System.out.println("Enter the elements of matrix 1:");
      for(int i=0;i<rows1;i++)
      {
         for(int j=0;j<cols1;j++)
         {
            mat1[i][j]=s.nextInt();
         }
      }
      System.out.println("Enter the number of rows and columns of matrix2:");
      rows2=s.nextInt();
      cols2=s.nextInt();
      System.out.println("Enter the elements of matrix 2:");
```

```
    for(int i=0;i<rows2;i++)
    {    for(int j=0;j<cols2;j++)
         {
           mat2[i][j]=s.nextInt();
         }
    }
   if(rows1==rows2 && cols1==cols2)
    {
       for(int i=0;i<rows1;i++)
       {
          for(int j=0;j<cols1;j++)
          {
             mat3[i][j]=mat1[i][j]+mat2[i][j];
          }
       }

       System.out.println("First matrix:");
       obj.Display(mat1,rows1,cols1);
       System.out.println("Second matrix:");
       obj.Display(mat2,rows2,cols2);
       System.out.println("Addition of two matrix:");
       obj.Display(mat3,rows1,cols1);
    }
    else
    {
       System.out.println("The matrices cannot be added.");
    }

  }
}
```

Output Screenshot

```
C:\Users\Student\Desktop\Just-in Regular>javac Matrixaddition.java

C:\Users\Student\Desktop\Just-in Regular>java Matrixaddition
Enter the number of rows and columns of matrix 1:
2
2
Enter the elements of matrix 1:
1
2
3
4
Enter the number of rows and columns of matrix 2:
2
2
Enter the elements of matrix 2:
5
6
7
8
First matrix:
1       2
3       4
Second matrix:
5       6
7       8
Addition of matrix:
6       8
10      12
```

OBJECT ORIENTED PROGRAMMING LAB

Name: JUSTIN V KALAPPURA

Roll No:10

Batch: MCA

Date:06/04/22

Experiment No.: 3

Aim

Add complex numbers.

Procedure

Source Code

```
import java.util.Scanner;
class complex{
   int r,b;
   void display(){
      System.out.println("Complex Number is:"+r+"+"+b+"i");
   }
}
public class AddComplex {
   public static void main(String[] args) {

      complex c1=new complex();
      complex c2=new complex();
      complex c3=new complex();
      Scanner s=new Scanner(System.in);
      System.out.println("Enter the real and imaginary parts of complex number 1:");
      c1.r=s.nextInt();
      c1.b=s.nextInt();
      System.out.println("Enter the real and imaginary parts of complex number 2:");
      c2.r=s.nextInt();
      c2.b=s.nextInt();
      c3.r=c1.r+c2.r;
      c3.b=c1.b+c2.b;
      c3.display();
       }
   }
```

Output Screenshot

```
C:\Users\Student\Desktop\Just-in Regular>javac AddComplex.java

C:\Users\Student\Desktop\Just-in Regular>java AddComplex
Enter the real and imaginary parts of complex number 1:
12
2
Enter the real and imaginary parts of complex number 2:
13
3
Complex Number is:25+5i
```

OBJECT ORIENTED PROGRAMMING LAB

Name: JUSTIN V KALAPPURA

Roll No:10

Batch:MCA

Date:06/04/22

Experiment No.: 4

Aim

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

Procedure

Source Code

```
import java.util.Scanner;
public class SymmetricMatrix {
   public void Display(int [][] arr,int row,int col){
      for(int i=0;i<row;i++){
         for(int j=0;j<col;j++){
            System.out.print(arr[i][j]+"\t");
         }
         System.out.println();
      }
}
   public static void main(String[] args) {
      int [][] mat = new int[3][3];
      int [][] trans=new int[3][3];
      int row,col;

      SymmetricMatrix obj=new SymmetricMatrix();
      Scanner s=new Scanner(System.in);
      System.out.println("Enter the rows and columns of the matrix:");
      row=s.nextInt();
      col=s.nextInt();
      System.out.println("Enter the elements of the matrix:");
      for(int i=0;i<row;i++)
      {          for(int j=0;j<col;j++)
         {          mat[i][j]=s.nextInt();
```

```
        }
      }
      for(int i=0;i<row;i++)
      {         for(int j=0;j<col;j++)
        {          trans[j][i]=mat[i][j];
         }
       }
      System.out.println("Entered matrix:");
      obj.Display(mat,row,col);
      System.out.println("Transpose of the matrix:");
      obj.Display(trans,row,col);
      for(int i=0;i<row;i++){
        for(int j=0;j<col;j++){
          if(mat[i][j]!=trans[i][j]){
            System.out.println("Matrix is not symmetric.");
            System.exit(0);
          }
        }
      }
      System.out.println("The given matrix is symmetric.");
   }
}
```

Output Screenshot

```
C:\Users\Student\Desktop\Just-in Regular>javac SymmetricMatrix.java

C:\Users\Student\Desktop\Just-in Regular>java SymmetricMatrix
Enter the rows and columns of the matrix:
2
2
Enter the elements of the matrix:
1
2
3
4
Entered matrix:
1       2
3       4
Transpose of the matrix:
1       3
2       4
Matrix is not symmetric.
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Justin V Kalappura

Roll No: 10

Batch: MCA

Date: 17/05/2022

Experiment No.: 5

Aim:

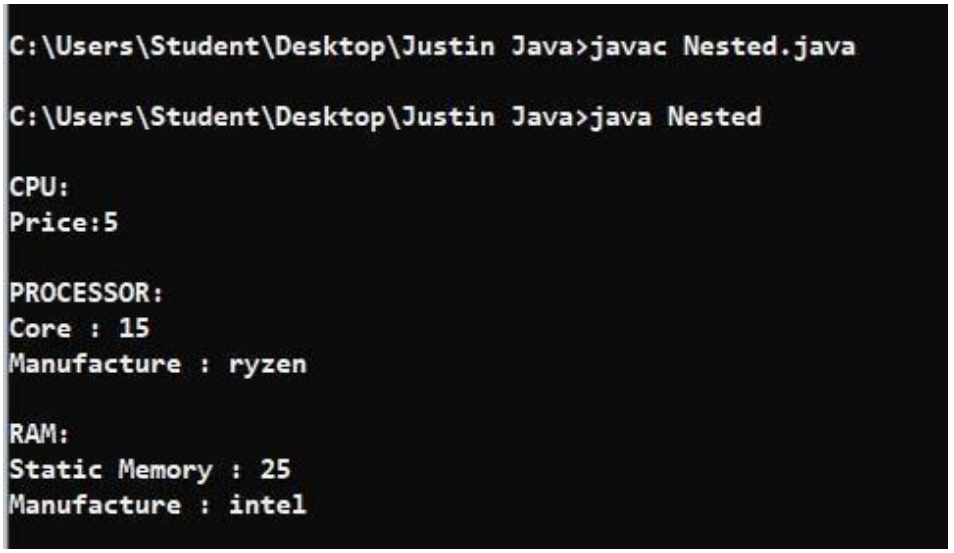
Create class Cpu with attributes price and create inner class processor (no of core, manufacture) and static nested class ram (static memory, manufacture). create an object of Cpu and print information at processor and ram.

Procedure:

```
public class Nested
{  public static void main(String args[])
  {  Cpu obj=new Cpu();
     Cpu.Processor obj1=obj.new Processor();
     Cpu.Ram obj2=new Cpu.Ram();
     System.out.println("\nCPU:");
          System.out.println("Price:"+obj.price);
     System.out.println("\nPROCESSOR:");
          System.out.println("Core : "+obj1.core);
          System.out.println("Manufacture : "+obj1.manufacture);
     System.out.println("\nRAM:");
          System.out.println("Static Memory : "+ Cpu.Ram.memory);
          System.out.println("Manufacture : "+obj2.manufacture);
   }
 } class Cpu { int
price   =5;   class
Processor
    { int core =15;
```

```
String manufacture="ryzen";
    }
        static class Ram
   { static int memory =25;
     String manufacture ="intel";
    }
}
```

Output Screenshot:



OBJECT ORIENTED PROGRAMMING LAB

Name: Justin V Kalappura

Roll No: 10

Batch: MCA

Date: 23/04/2022

Experiment No.: 6

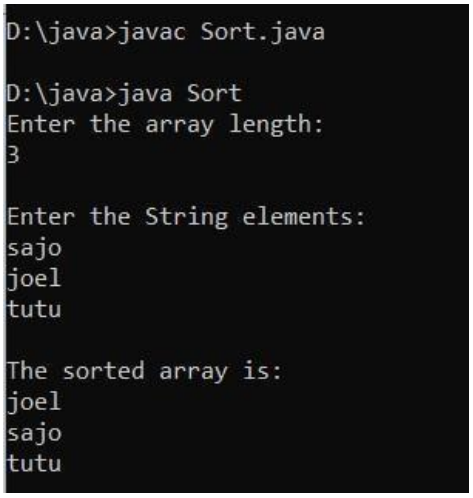
Aim:

Program to Sort strings.

Procedure:

```
import java.util.Scanner;
import java.util.Arrays;
public class Sort
{    public static void main(String [] args)
    { int a;
       Scanner sc = new Scanner(System.in);
       System.out.println("Enter the array length:");
       a = sc.nextInt();
                String [] b = new String[a];
                System.out.println("\nEnter the String elements:");
      for(int i=0;i<a;i++)
      {  b[i]= sc.next();
                }
       Arrays.sort(b);
                System.out.println("\nThe sorted array is:");
      for(int i=0;i<a;i++)
      {  System.out.println(b[i]);
                }
         }
 }
```

Output Screenshot:



OBJECT ORIENTED PROGRAMMING LAB

Name: Justin V Kalappura

Roll No: 10

Batch: MCA

Date: 25/04/2022

Experiment No.: 7

Aim:

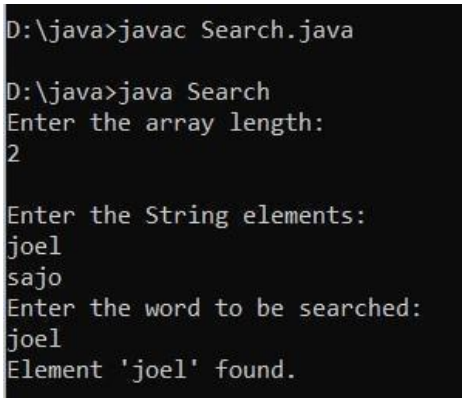
Search an element in an array

Procedure:

```
import java.util.Scanner;
public class Search
{  public static void main(String [] args)
   {    int a,j=0;
        String c;
             Scanner sc = new Scanner(System.in);
        System.out.println("Enter the array length:");
        a = sc.nextInt();
                  String [] b = new String[a];
                  System.out.println("\nEnter the String elements:");
        for(int i=0;i<a;i++)
        {  b[i]= sc.next();
                   }
                  System.out.println("Enter the word to be searched:");
                  c = sc.next();
        for(int i=0;i<a;i++)
        {  if(b[i].equals(c))
                          { j=0;
                       break;
                          }
                          else
```

```
            j=1;
                }
              if(j==0)
                    System.out.println("Element '"+c+"' found.");
      else
         System.out.println("Element not found.");
   }
}
```

Output Screenshot:



OBJECT ORIENTED PROGRAMMING LAB

Name: Justin V Kalappura

Roll No: 10

Batch: MCA

Date: 25/04/2022

Experiment No.: 8

Aim:

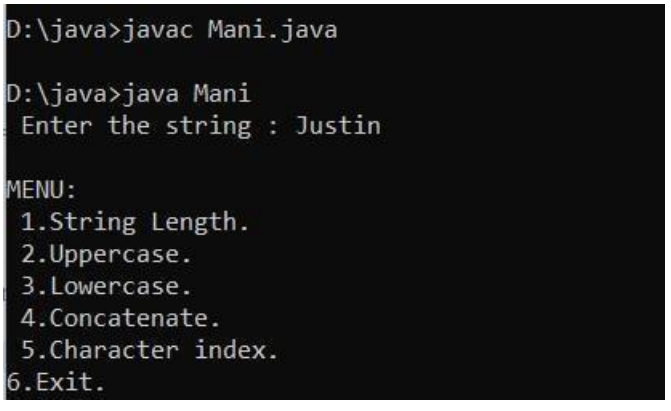
Perform string manipulations.

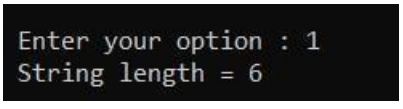
Procedure:

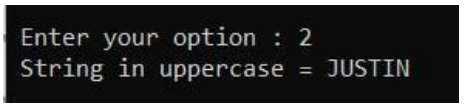
```
import java.util.Scanner;
import java.lang.*;
public class Mani
{  public static void main(String [] args)
   {    int a;
        String b,c;
             Scanner sc = new Scanner(System.in);
        System.out.print(" Enter the string : ");
        b = sc.nextLine();
                while(true)
                { 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:







```
Enter your option : 3
String in lowercase = justin
```

```
Enter your option : 4
Enter the string to be concatenate = George
Concatenated string = JustinGeorge
```

```
Enter your option : 5
Enter the Character to be searched in the given string = t
The character is found at 4.
```

```
Enter your option : 6

D:\java>
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Justin V Kalappura

Roll No: 10

Batch: MCA

Date: 25/04/2022

Experiment No.: 9

Aim:

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.Scanner;
import java.lang.*;
public class Arrayofobjects
{  public static void main(String [] args)
    {  int a,b,c,en,es;
          String ena;
          Scanner sc = new Scanner(System.in);
              Product [] Em = new Product[100];
              while(true)
              { System.out.print("Menu:\n1.Enter the details.\n2.Display the
                details.\n3.Exit.\nEnter the choice : ");
                b = sc.nextInt();
                switch(b)
               { case 1:{ System.out.print(" Enter the number of employee's:");
                          a = sc.nextInt();
                         for(int i=0;i<a;i++)
                        { System.out.print("\n Employee "+(i+1)+" : \n");
                          System.out.print(" Enter the employee number:");
                          en = sc.nextInt();
```

```
                    System.out.print(" Enter the employee name:");
                    ena  = sc.next();
                    System.out.print(" Enter the employee salary:");
                    es = sc.nextInt();
                    Em[i] = new Product(en,ena,es);
                    }
                    break;
                   }
            case 2:{ System.out.print("Enter the Employee number to be displayed : ");
                    c = sc.nextInt();
                    Em[c-1].display();
                    break;
                   }
            case 3: System.exit(0);
          }
        }
      }
}
class Product
{    int a,eno,esalary;
    String ename;
        Scanner sc = new Scanner(System.in);
        Product(int en,String ena,int es)
        { eno = en;
          ename = ena;
          esalary = es;
        }
        public void display()
        { System.out.println("\n Details of Employee ");
```

```
            System.out.println(" The employee no : "+eno);
            System.out.println(" The employee name : "+ename);
            System.out.println(" The employee salary : "+esalary);
          }
 }
```

Output Screenshot:

```
D:\java>javac Arrayofobjects.java

D:\java>java Arrayofobjects
Menu:
1.Enter the details.
2.Display the details.
3.Exit.
Enter the choice : 1
 Enter the number of employee's:3

 Employee 1 :
 Enter the employee number:1
 Enter the employee name:justin
 Enter the employee salary:1

 Employee 2 :
 Enter the employee number:2
 Enter the employee name:joel
 Enter the employee salary:2

 Employee 3 :
 Enter the employee number:3
 Enter the employee name:sajo
 Enter the employee salary:3
```

```
Menu:
1.Enter the details.
2.Display the details.
3.Exit.
Enter the choice : 2
Enter the Employee number to be displayed : 2

 Details of Employee
 The employee no : 2
 The employee name : joel
 The employee salary : 2
Menu:
1.Enter the details.
2.Display the details.
3.Exit.
Enter the choice : 3

D:\java>
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Justin V Kalappura

Roll No: 10

Batch: S2 RMCA-B

Date: 23/05/2022

Experiment No.: 10

Aim:

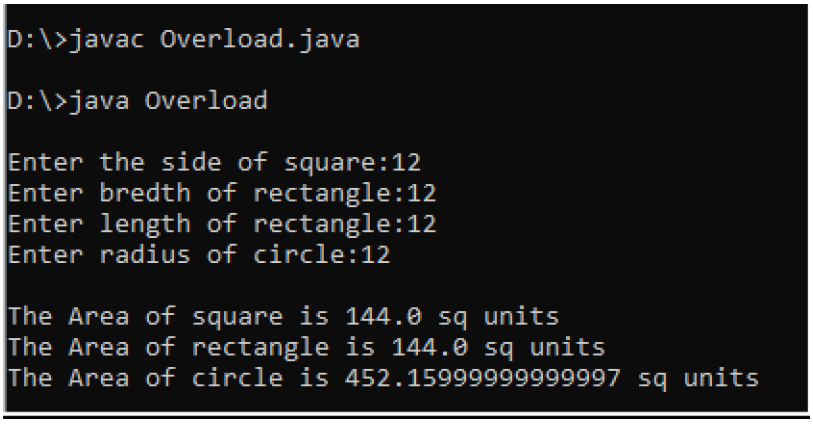
Area of different shapes using overloaded functions.

Procedure:

```
import java.util.*;
class OverloadDemo
{       void area(float x)
        {    System.out.println("\nThe Area of square is " +x*x+  " sq units");
        }
        void area(float x,float y)
        {    System.out.println("The Area of rectangle is " +x*y+  " sq units");
         }
         void area(double x)
        {    double z=3.14*x*x;
             System.out.println("The Area of circle is " +z+  " sq units");
          }
}
class Overload
{        public static void main(String args[])
         {    int square;
              int rect1,rect2;
              double circle;
              Scanner sc=new Scanner(System.in);
              System.out.print("\nEnter the side of square:");
```

```
            square=sc.nextInt();
            System.out.print("Enter bredth of rectangle:");
            rect1=sc.nextInt();
            System.out.print("Enter length of rectangle:");
            rect2=sc.nextInt();
            System.out.print("Enter radius of circle:");
            circle=sc.nextDouble();
           OverloadDemo obj=new OverloadDemo();
            obj.area(square);
            obj.area(rect1,rect2);
            obj.area(circle);
        }
    }
```

Output Screenshot:



OBJECT ORIENTED PROGRAMMING LAB

Name: Justin V Kalappura

Roll No: 10

Batch: S2 RMCA-B

Date: 23/05/2022

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:

```
import java.util.Scanner;
class Employeee
{   int e_id;
    String e_name,e_address;
    double e_sal;
    Employeee(int id, String name, double sal, String address)
   {   e_id=id;
        e_name=name;
        e_address=address;
        e_sal=sal;
    }
    void display()
   {   System.out.println("Employe Name = "+e_name);
        System.out.println("Employe Salary = "+e_sal);
        System.out.println("Employe Address = "+e_address);
     }
 }
class Teacher extends Employeee
```

```
{   String t_dept, t_sub;
    Teacher(String dept, String sub, String name,int id, String address, double sal)
   {   super(id, name, sal, address);
       t_dept=dept;
       t_sub=sub;
    }
    void display()
   {   System.out.println("\nDepartment = "+t_dept);
       System.out.println("Subject = "+t_sub);
       System.out.println("Employee Id = "+e_id);
       System.out.println("Employee Name = "+e_name);
       System.out.println("Employee Salary = "+e_sal);
       System.out.println("Employee Address = "+e_address);
    }
}
public class Arrofobj
{    public static void main(String[] args)
     {  int count,id;
        String name,dept,sub,address;
        double sal;
        Scanner s=new Scanner(System.in);
        System.out.print("Enter the number of Teachers :");
        count=s.nextInt();
        Teacher [] obj=new Teacher[count];
        for(int i=0;i<count;i++)
        {   System.out.print("Enter the Emp name :");
            name=s.next();
            System.out.print("Enter the Id :");
            id=s.nextInt();
```

```
            System.out.print("Enter the Address :");
            address=s.next();
            System.out.print("Enter the Salary :");
            sal=s.nextInt();
            System.out.print("Enter the Department :");
            dept=s.next();
            System.out.print("Enter the Subject :");
            sub=s.next();
            obj[i]=new Teacher(dept, sub, name, id, address, sal);
        }
        for(int i=0;i<count;i++)
       {   obj[i].display();
        }
    }
}
```

Output Screenshot:

```
D:\>javac Arrofobj.java

D:\>java Arrofobj
Enter the number of Teachers :1
Enter the Emp name :just
Enter the Id :1
Enter the Address :kala
Enter the Salary :1
Enter the Department :mca
Enter the Subject :eng

Department = mca
Subject = eng
Employee Id = 1
Employee Name = just
Employee Salary = 1.0
Employee Address = kala
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Justin V Kalappura

Roll No: 10

Batch: MCA

Date: 24/05/2022

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.*;
class Person
{   String Name;
   String Gender;
   String Address;
   int Age;
   Person(String name,String gender,String address,int age)
   {   this.Name=name;
      this.Gender=gender;
      this.Address=address;
      this.Age=age;
    }
}
class Employe extends Person
```

```
{    int Empid;
    String Company_name;
    String Qualificatiom;
    long Salary;
    Employe(String name,String gender,String address,int age,int empid,String company_name,String qualification,long salary)
    {     super(name,gender,address,age);
          this.Empid=empid;
          this.Company_name=company_name;
          this.Qualificatiom=qualification;
          this.Salary=salary;
     }
}
public class Teacher extends Employe
{     String Teacher_id;
     String Department,Subject;
     Teacher(String name,String gender,String address,int age,int empid,String company_name,String qualification,long salary,String 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 display()
    {System.out.println("Name: "+Name);
     System.out.println("   Gender: "+Gender);
```

```
System.out.println("   Address: "+Address);
System.out.println("   Age: "+Age);
System.out.println("   Employee Id: "+Empid);
System.out.println("   Company Name: "+Company_name);
System.out.println("   Qualificatiom: "+Qualificatiom);
System.out.println("   Salary: "+Salary);
System.out.println("   Teacher Id: "+Teacher_id);
System.out.println("   Department: "+Department);
System.out.println("   Subject: "+Subject);
}
public static void main(String args[])
{Scanner sc=new Scanner(System.in);
int n;
System.out.print("\nEnter number of teachers:");
n=sc.nextInt();
Teacher ob[]=new Teacher[n];
System.out.print("\n.....Enter The Teacher  Details. ..... ");
int x=0,j=0;
Scanner s=new Scanner(System.in);
for(int i=0;i<n;i++)
{    x=i+1;
    System.out.print("\n\n"+x+". ");
    System.out.print("Name:");
    String a=s.next();
    System.out.print("   Gender:");
    String b=s.next();
    System.out.print("   Address:");
```

```
        String c=s.next();
        System.out.print("   Age:");
        int d=s.nextInt();
        System.out.print("   Employee Id:");
        int e=s.nextInt();
        System.out.print("   Company name:");
        String f=s.next();
        System.out.print("   Qualificatiom:");
        String g=s.next();
        System.out.print("   Salary :");
        int h=s.nextInt();
        System.out.print("   Teacher Id:");
        String l=s.next();
        System.out.print("   Department:");
        String m=s.next();
        System.out.print("   Subject:");
        String k=s.next();
        ob[i]=new Teacher(a,b,c,d,e,f,g,h,l,m,k);
    }
    s.close();
    System.out.println(".....Teacher Details..... ");
    for(int i=0;i<n;i++)
    {    j=i+1;
         System.out.print("\n"+j+". ");
         ob[i].display();
     }
    sc.close();
```

```
    }
}
```

Output Screenshot:

```
C:\Users\Student\Desktop\Justin Java>javac Teacher.java

C:\Users\Student\Desktop\Justin Java>java Teacher

Enter number of teachers:1

.............Enter The Teacher  Details.............

1. Name:justin
   Gender:male
   Address:kala
   Age:2
   Employee Id:1
   Company name:just
   Qualificatiom:mca
   Salary :100
   Teacher Id:1
   Department:mca
   Subject:acn
..........Teacher Details.........

1. Name: justin
   Gender: male
   Address: kala
   Age: 2
   Employee Id: 1
   Company Name: just
   Qualificatiom: mca
   Salary: 100
   Teacher Id: 1
   Department: mca
   Subject: acn
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Justin V Kalappura

Roll No: 10

Batch: MCA

Date: 24/05/2022

Experiment No.: 13

Aim:

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

Procedure:

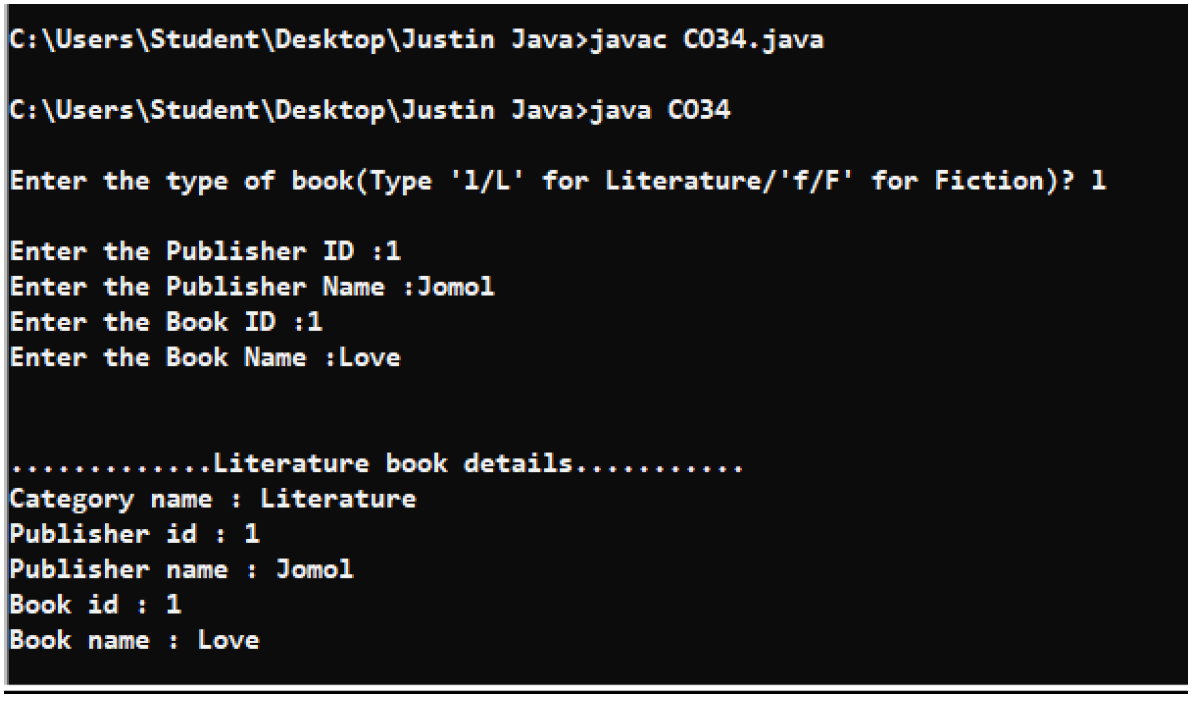
```
import java.util.Scanner;
class Publisher
{   int  p_id;
    String  p_name;
    Publisher(int p_id,String p_name)
        {   this.p_id=p_id;
        this.p_name=p_name;
    }
}
class Book extends Publisher
{   int  b_id;
    String  b_name;
    Book(int p_id, String p_name, int b_id, String b_name)
        {   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 CO34
{   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 Screenshot:



```
C:\Users\Student\Desktop\Justin Java>javac C034.java

C:\Users\Student\Desktop\Justin Java>java C034

Enter the type of book(Type '1/L' for Literature/'f/F' for Fiction)? 1

Enter the Publisher ID :1
Enter the Publisher Name :Jomol
Enter the Book ID :1
Enter the Book Name :Love


.............Literature book details...........
Category name : Literature
Publisher id : 1
Publisher name : Jomol
Book id : 1
Book name : Love
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Justin V Kalappura

Roll No: 10

Batch: MCA

Date: 24/05/2022

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("\n\n.....Sports Details of Student.....");
        System.out.println("Sport :"+sport);
        System.out.println("Rating :"+Rating);
        System.out.println("\n.....Academic 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("\n\n!!!!!!!!!!!!!!!Enter the Sports Details of
Student!!!!!!!!!!!!!");
        System.out.print(" Sport: ");
        String a =sc.next();
        System.out.print(" Sport Rating out of 10: ");
        int b =sc.nextInt();
        System.out.println("\n!!!!!!!!!!!!!!!Enter the Academic Details of
Student!!!!!!!!!!!!!");
        System.out.print(" Academic Grade: ");
        String c =sc.next();
        System.out.print(" Overall percentage: ");
        double d =sc.nextDouble();
        sc.close();
```

```
        Result obj= new Result(a,b,c,d);
        obj.display();
    }
}
```

Output Screenshot:

```
C:\Users\Student\Desktop\Justin Java>javac Result.java

C:\Users\Student\Desktop\Justin Java>java Result



!!!!!!!!!!!!!!Enter the Sports Details of Student!!!!!!!!!!!!!
 Sport: Basketball
 Sport Rating  out of 10: 9

!!!!!!!!!!!!!!Enter the Academic Details of Student!!!!!!!!!!!!!
 Academic Grade: A
 Overall percentage: 99
..........Sports Details of Student...........
Sport :Basketball
Rating :9

.........Academic Details of Student..........
Academic Grade :A
Overall percentage :99.0
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Justin V Kalappura

Roll No: 10

Batch: MCA

Date: 24/05/2022

Experiment No.: 15

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 Part
{   void  area();
    void perimeter();
}

class Rectangle implements Part
{ int l,b,ra,rp;
   public void area()
   { Scanner sc = new Scanner(System.in);
      System.out.print("\nEnter the length :");
      l = sc.nextInt();
      System.out.print("Enter the breadth :");
         b = sc.nextInt();
      ra = l*b;
         System.out.println("\nArea of the rectangle: "+ra);
```

```
    }
   public void perimeter()
   {  rp = 2*(l+b);
      System.out.print("Perimeter of the rectangle : "+rp);
          System.out.print("\n");
    }
 }

class Circle implements Part
{ double r,ca,cp;
   public void area()
   { Scanner sr = new Scanner(System.in);
      System.out.print("\nEnter the radius :");
          r = sr.nextInt();
      ca = 3.14*r*r;
          System.out.println("\nArea of the circle : "+ca);
    }
   public void perimeter()
   {  cp = 2*3.14*r;
      System.out.print("Perimeter of the circle : "+cp);
          System.out.print("\n");
    }
 }

public class Shape
```

```
{   public static void main(String args[])
   { Scanner s = new Scanner(System.in);
        Part ci = new Circle();
        Part r = new Rectangle();
        while(true)
        { int c;
       System.out.print("\nMENU : ");
          System.out.print("\n1.Rectangle\n2.Circle\n3.Exit\nEnter the choice :");
          c=s.nextInt();
             switch(c)
             {   case 1: r.area();
                       r.perimeter();
                              break;
                  case 2: ci.area();
                        ci.perimeter();
                              break;
                  case 3:System.exit(0);
                  default: System.out.println("!!!!!!!!!!!!Wrong Entry!!!!!!!!!!!!");
                }
          }
       }
}
```

Output Screenshot:

```
C:\Users\Student\Desktop\Justin Java>javac Shape.java

C:\Users\Student\Desktop\Justin Java>java Shape

MENU :
1.Rectangle
2.Circle
3.Exit
Enter the choice :1

Enter the length :1
Enter the breadth :1

Area of the rectangle: 1
Perimeter of the rectangle : 4

MENU :
1.Rectangle
2.Circle
3.Exit
Enter the choice :2

Enter the radius :1

Area of the circle : 3.14
Perimeter of the circle : 6.28

MENU :
1.Rectangle
2.Circle
3.Exit
Enter the choice :4
!!!!!!!!!!!Wrong Entry!!!!!!!!!!!

MENU :
1.Rectangle
2.Circle
3.Exit
Enter the choice :3
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Justin V Kalappura

Roll No:10

Batch: MCA B

Date:30-05-2022

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.print("\nEnter product id:");
      p_id = sc.nextLine();
      System.out.print("Enter product name:");
      name = sc.nextLine();
      System.out.print("Enter the Quantity:");
```

```
        quantity = sc.nextInt();
        System.out.print("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\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("\nOrder no: #"+ran);
        System.out.print("Enter the date:");
        date = sc.nextLine();
        System.out.print("Enter how many products are there for the bill:");
        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("\n\n\nDate:"+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\t\t        Net.Amount        \t"+ namount);
   }
}
```

Output Screenshot:

```
C:\Users\Student\Desktop\Just-in Regular>javac BillInterface.java

C:\Users\Student\Desktop\Just-in Regular>java BillInterface

Order no: #315063
Enter the date:12/2/22
Enter how many products are there for the bill:2

Enter product id:1
Enter product name:Steelsink
Enter the Quantity:1
Enter the unit price:1

Enter product id:2
Enter product name:Washbasin
Enter the Quantity:2
Enter the unit price:2



Date:12/2/22
Product_Id      Name            Quantity        Unit_price      Total
-------------------------------------------------------------------------
1               Steelsink               1               1.0             1.0
2               Washbasin               2               2.0             4.0
-------------------------------------------------------------------------
                                                Net.Amount              5.0
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Justin v kalappura

Roll No: 10

Batch: S2 MCA

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.

Procedure:

1. Graphics.java:

```
package  graphics;
import java.util.Scanner;
interface fig{
   public  double  recArea();
   public  double  cirArea();
   public double squArea();
   public  double triArea();
}
public class Graphics implements fig {
    Scanner s = new Scanner(System.in);
    int r,l,b,a;
      double pi = 3.14,area;
           public  double recArea(){
                  System.out.println("Enter length of rectangle:");
        l=s.nextInt();
        System.out.println("Enter breadth of rectangle:");
        b=s.nextInt();
                  area=l*b;
```

```
                return area;
                }
    public double cirArea(){
        System.out.println("Enter radius of circle:");
        r = s.nextInt();
        area = pi * r * r;
        return area;
                }
    public double squArea(){
        System.out.println("Enter the side of the square:");
        a = s.nextInt();
        area = a * a;
                        return area;
    }
    public double triArea(){
         System.out.println("Enter the width of the Triangle:");
      double base  = s.nextDouble();
      System.out.println("Enter the height of the Triangle:");
      double height = s.nextDouble();
      double area = (base* height)/2;
      return area;
    }
}
```

2. AreaGraphics.java:

```
import graphics.*;
public class AreaGraphics {
    public static void main(String []args){
        Graphics Ob = new Graphics();
```

```
        System.out.println(Ob.recArea());
        System.out.println(Ob.cirArea());
        System.out.println(Ob.squArea());
        System.out.println(Ob.triArea());
    }
```

Output Screenshot:

```
D:\jomol javalab\graphics>javac Graphics.java

D:\jomol javalab\graphics>cd ..

D:\jomol javalab>javac AreaGraphics.java

D:\jomol javalab>java AreaGraphics
Enter length of rectangle:
2
Enter breadth of rectangle:
1
2.0
Enter radius of circle:
1
3.14
Enter the side of the square:
2
4.0
Enter the width of the Triangle:
2
Enter the height of the Triangle:
3
3.0

D:\jomol javalab>
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Justin v kalappura

Roll No: 10

Batch: S2 MCA

Date: 31/05/2022

Experiment No.: 18

Aim:

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

Procedure:

```
import java.util.Scanner;
class UsernameException extends Exception {
 public UsernameException(String msg) {
 super(msg);
 }
}
class PasswordException extends Exception {
 public PasswordException(String msg) {
 super(msg);
 }
}
public class CheckExp {
 public static void main(String[] args) {
  Scanner s = new Scanner(System.in);
  String username, password;
  System.out.print("Enter username :: ");
  username = s.nextLine();
  System.out.print("Enter password :: ");
  password = s.nextLine();
  int length = username.length();
```

```
 try {
  if(length < 6)
   throw new UsernameException("Username must be greater than 6 characters ???");
  else if(!password.equals("hello"))
   throw new PasswordException("Incorrect password\nType correct password ???");
  else
   System.out.println("Login Successful !!!");
 }
 catch (UsernameException u) {
        u.printStackTrace();
 }
 catch (PasswordException p) {
 p.printStackTrace();
 }
 finally {
  System.out.println("The finally statement is executed");
 }
 }
}
```

Output Screenshot:

```
D:\jomol javalab>javac CheckExp.java

D:\jomol javalab>java CheckExp
Enter username :: jomol
Enter password :: jomol
UsernameException: Username must be greater than 6 characters ???
        at CheckExp.main(CheckExp.java:33)
The finally statement is executed

D:\jomol javalab>java CheckExp
Enter username :: Annsalomi
Enter password :: hello
Login Successful !!!
The finally statement is executed

D:\jomol javalab>
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Justin v kalappura

Roll No: 10

Batch: S2 MCA

Date: 31/05/2022

Experiment No.: 19

AIM:

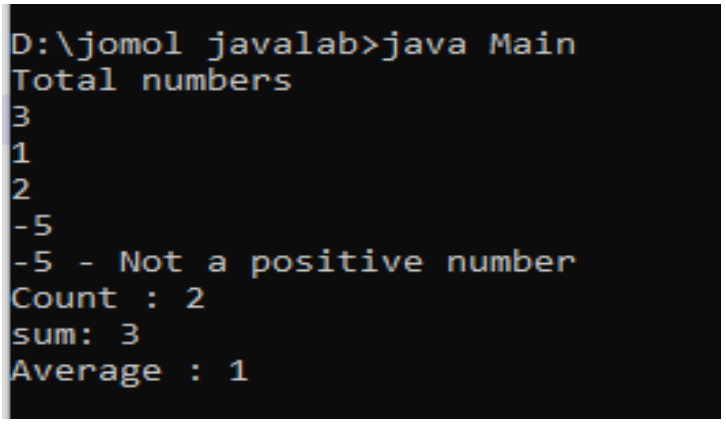
Find the average of N positive integers, raising a user defined exception for each negative input.

PROCEDURE:

```
import java.util.*;
class MyException extends Exception {
    public MyException(String value) {
        super(value);
    }
}
public 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:



OBJECT ORIENTED PROGRAMMING LAB

Name: Justin v kalappura

Roll No: 10

Batch: S2 MCA

Date: 31/05/2022

Experiment No.: 20

AIM:

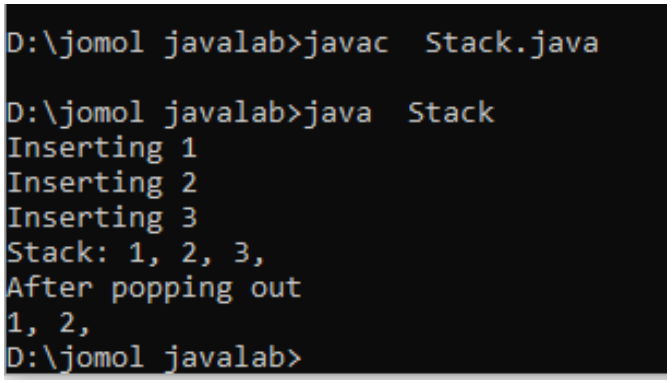
Program to create a generic stack and do the Push and Pop operations.

PROCEDURE:

```
public 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:



OBJECT ORIENTED PROGRAMMING LAB

Name: Justin v kalappura

Roll No: 10

Batch: S2 MCA

Date: 31/05/2022

Experiment No.: 21

AIM:

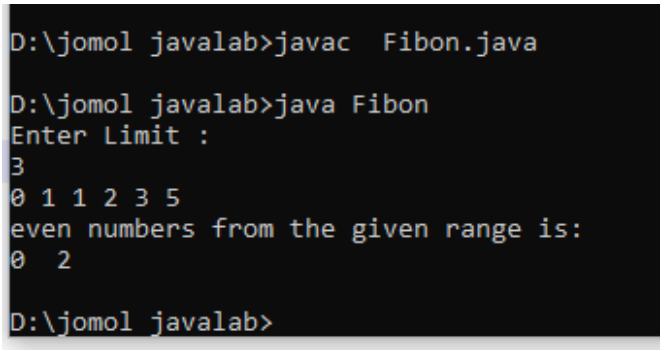
Define 2 classes; one for generating Fibonacci numbers and other for displaying even numbers in a given range. Implement using threads. (Runnable Interface).

PROCEDURE:

```
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() {
                System.out.println("");
                System.out.println("even numbers from the given range is:");
        for (int i = 0; i <= l; i++) {
            if (i % 2 == 0)
                System.out.print(i + "  ");
        }
        System.out.println("");
    }
}
public class Fibon {
    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:



OBJECT ORIENTED PROGRAMMING LAB

Name: Justin v kalappura

Roll No: 10

Batch: S2 MCA B

Date: 31-05-2022

Experiment No.: 22

Aim:

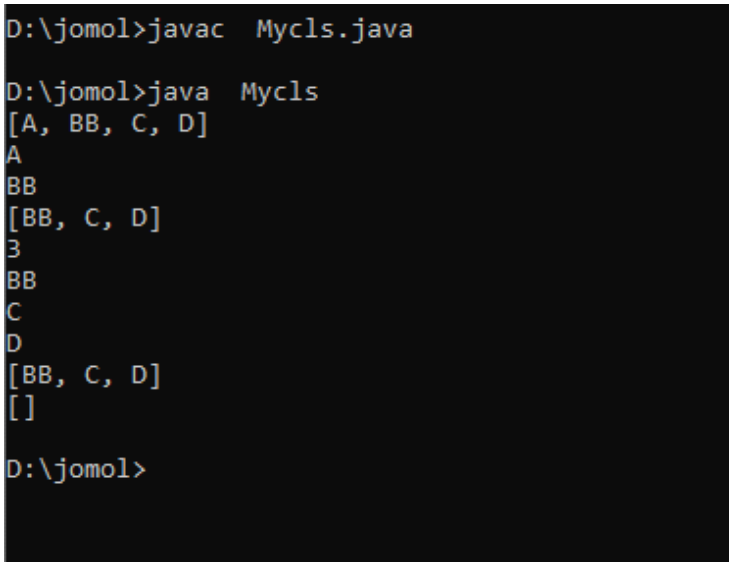
Maintain a list of Strings using ArrayList from collection framework, perform built-in operations.

PROCEDURE:

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



OBJECT ORIENTED PROGRAMMING LAB

Name: Justin v kalappura

Roll No: 10

Batch: S2 MCA B

Date: 31-05-2022

Experiment No.: 23

Aim:

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

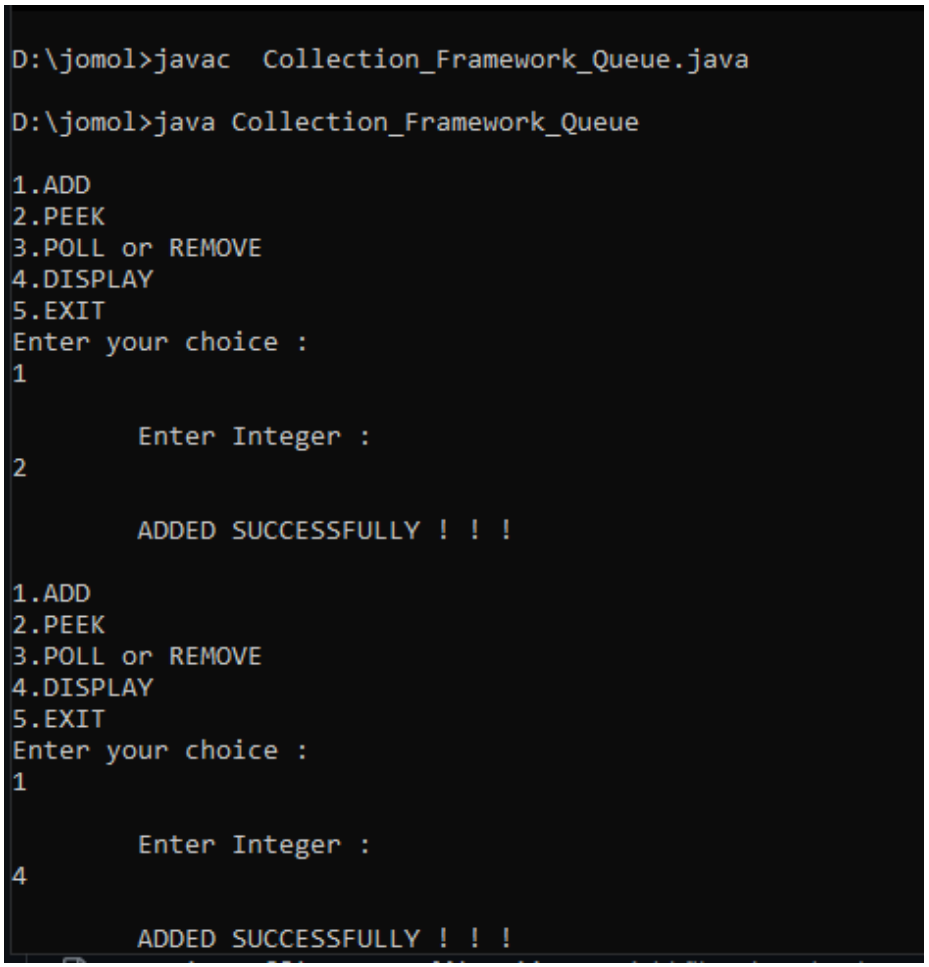
PROCEDURE:

```
import java.util.*;
public 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:



```
D:\jomol>javac  Collection_Framework_Queue.java

D:\jomol>java Collection_Framework_Queue

1.ADD
2.PEEK
3.POLL or REMOVE
4.DISPLAY
5.EXIT
Enter your choice :
1

        Enter Integer :
2

        ADDED SUCCESSFULLY ! ! !

1.ADD
2.PEEK
3.POLL or REMOVE
4.DISPLAY
5.EXIT
Enter your choice :
1

        Enter Integer :
4

        ADDED SUCCESSFULLY ! ! !
```

```
1.ADD
2.PEEK
3.POLL or REMOVE
4.DISPLAY
5.EXIT
Enter your choice :
1

        Enter Integer :
4

        ADDED SUCCESSFULLY ! ! !

1.ADD
2.PEEK
3.POLL or REMOVE
4.DISPLAY
5.EXIT
Enter your choice :
4

Size of queue : 2
Queue elements  : [2, 4]
Queue elements are
2
4

1.ADD
2.PEEK
3.POLL or REMOVE
```

```
3.POLL or REMOVE
4.DISPLAY
5.EXIT
Enter your choice :
4

Size of queue : 2
Queue elements  : [2, 4]
Queue elements are
2
4

1.ADD
2.PEEK
3.POLL or REMOVE
4.DISPLAY
5.EXIT
Enter your choice :
2

        Peeked element is 2
1.ADD
2.PEEK
3.POLL or REMOVE
4.DISPLAY
5.EXIT
Enter your choice :
3

        Removed element is 2
```

```
1.ADD
2.PEEK
3.POLL or REMOVE
4.DISPLAY
5.EXIT
Enter your choice :
3

        Removed element is 2
1.ADD
2.PEEK
3.POLL or REMOVE
4.DISPLAY
5.EXIT
Enter your choice :
5

D:\jomol>
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Justin v kalappura

Roll No: 10

Batch: S2 MCA B

Date: 07/06/2022

Experiment No.: 24

Aim:

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

Procedure:

```
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 Screenshot:

```
Microsoft Windows [Version 10.0.19044.1706]
(c) Microsoft Corporation. All rights reserved.

D:\jomol javalab>javac deque.java

D:\jomol javalab>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)]

D:\jomol javalab>
```

```
D:\jomol javalab>javac Interfacecirclerect.java

D:\jomol javalab>java Interfacecirclerect


choose the operations you can do:
1.circle

 2.Rectangle

3.exit
Enter your operations:
1
circle
Enter radius of circle:
4
Area of circle:50.24
Perimeter of circle:25.12


choose the operations you can do:
1.circle

 2.Rectangle

3.exit
Enter your operations:
2
Rectangle
```

```
Enter your operations:
2
Rectangle
Enter length of rectangle:
2
Enter breadth of rectangle:
4
Area of rectangle:8
Perimeter of rectangle:12


choose the operations you can do:
1.circle

 2.Rectangle

3.exit
Enter your operations:
3

D:\jomol javalab>
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Justin v kalappura

Roll No: 10

Batch: S2 MCA B

Date: 07/06/2022

Experiment No.: 25

Aim:

Write a Java program to compare two hash set.

Procedure:

```
import java.util.*;
 public class Hashset {
 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:

```
D:\jomol javalab>javac Hashset.java

D:\jomol javalab>java Hashset
Yes
No
Yes
No

D:\jomol javalab>
```

```
Enter your operations:
2
Rectangle
Enter length of rectangle:
2
Enter breadth of rectangle:
4
Area of rectangle:8
Perimeter of rectangle:12


choose the operations you can do:
1.circle

 2.Rectangle

3.exit
Enter your operations:
3

D:\jomol javalab>
```

```
D:\jomol javalab>javac Interfacecirclerect.java

D:\jomol javalab>java Interfacecirclerect


choose the operations you can do:
1.circle

 2.Rectangle

3.exit
Enter your operations:
1
circle
Enter radius of circle:
4
Area of circle:50.24
Perimeter of circle:25.12


choose the operations you can do:
1.circle

 2.Rectangle

3.exit
Enter your operations:
2
Rectangle
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Justin v kalappura

Roll No: 10

Batch: S2 MCA B

Date: 07/06/2022

Experiment No.: 26

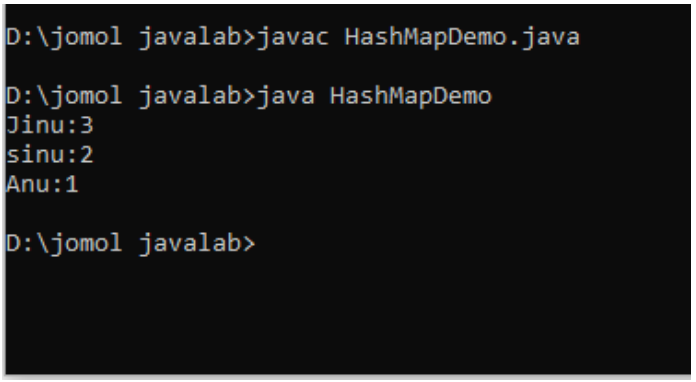
Aim:

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:



OBJECT ORIENTED PROGRAMMING LAB

Experiment No.: 27

Name: Justin v

kalappuraRoll No:10

Batch: S2 MCA

Date: 30/05/2022

Aim:

Write a program to write to a file, then read from the file and display the contentson 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");
}
}
}
```

Output Screenshot:

```
D:\jomol>javac Read.java

D:\jomol>java Read
Enter the text to create file : type ENTER key 3 times to stop
hello this is a java program for reading a file.
object oriented programming lab
it is robust and secure


Reading File content
hello this is a java program for reading a file.
object oriented programming lab
it is robust and secure
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Justin v k

Roll No:10

Batch: S2 MCA

Date: 30/05/2022

Experiment No.: 28

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 Screenshot:

```
D:\jomol>java Copy
Enter the source File Name
cop1.txt
Reading from file cop1.txt
hello sample program of java
it is the program for copying a file

 Enter the filename to copy
cop2.txt
Copied from cop1.txt to cop2.txt Successfully..!

D:\jomol>
```

OBJECT ORIENTED PROGRAMMING LAB

Experiment No.: 29

Name: Justin v k

Roll No:10

Batch: S2 MCA

Date: 30/05/2022

Aim:

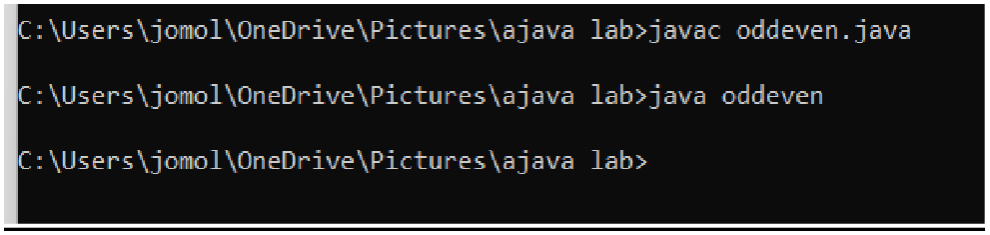
Write a program that reads from a file having integers. Copy even numbers andodd numbers to separate files.

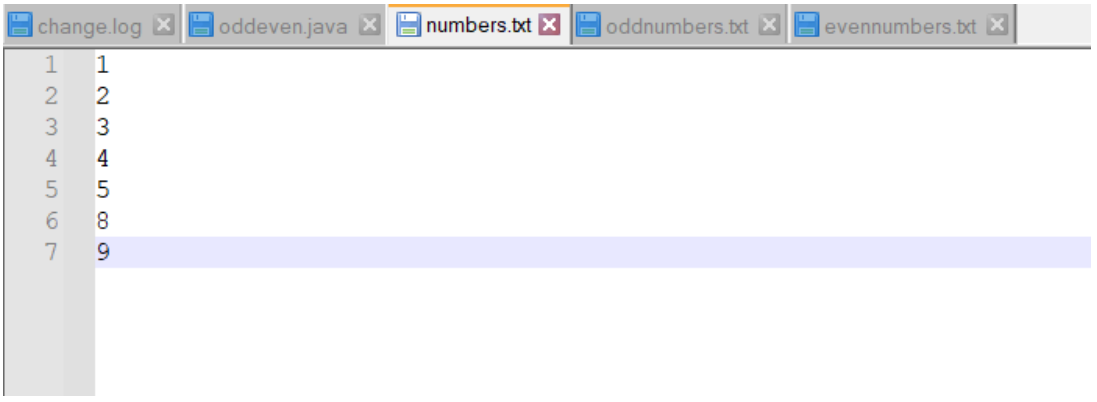
Procedure:

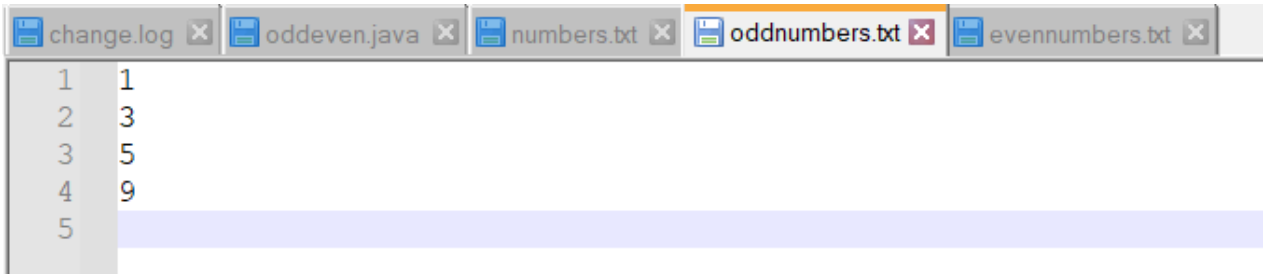
```
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.*;
import java.util.*;
import java.io.File;
public class oddeven {
public static void main(String[] args) {
try {
FileReader fr = new FileReader("numbers.txt");
BufferedReader br = new BufferedReader(fr);
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");
```

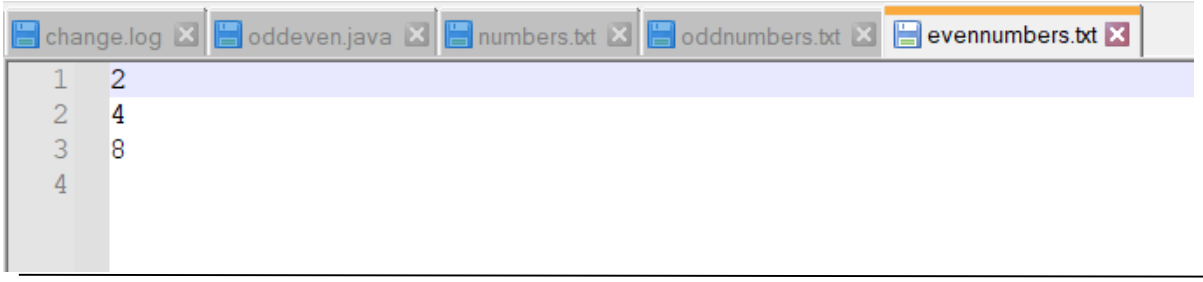
```
}
else {
fw1.write(num + "\n");
}}
fw1.close();
fw2.close();}
catch (Exception e)
{    System.out.println("Error");
}}}
```

Output Screenshot:









OBJECT ORIENTED PROGRAMMING LAB

Name: Justin V Kalappura

Roll No: 10

Batch: B

Date: 09-06-2022

Experiment No.:30

Aim:

Program to find maximum of three numbers using AWT.

Procedure:

```
import java.awt.*;
public class AWTexample extends Frame{
AWTexample(){
        Button b=new Button("click me");
        b.setBounds(50,100,80,40);
        add(b);
        setSize(400,400);
        setTitle("awt example");
        setLayout(null);
        setVisible(true);
}
public static void main(String args[]){
        AWTexample F=new AWTexample();
}}
```

Output Screenshot:

OBJECT ORIENTED PROGRAMMING LAB

Name: Justin V Kalappura

Roll No: 10

Batch: S2 MCA B

Date: 09-06-2022

Experiment No.: 31

Aim:

Implement a simple calculator using AWT components.

Procedure:

```
import java.awt.*;
import java.awt.event.*;
class MyCalc extends WindowAdapter implements ActionListener{
  Frame f;
Label l1;
Button b1,b2,b3,b4,b5,b6,b7,b8,b9,b0;
Button badd,bsub,bmult,bdiv,bmod,bcalc,bclr,bpts,bneg,bback;
double xd;
double num1,num2,check;

MyCalc(){
  f= new Frame("MY CALCULATOR");
// INSTANTIATING COMPONENETS
l1=new Label();
l1.setBackground(Color.LIGHT_GRAY);
l1.setBounds(50,50,260,60);

b1=new Button("1");
b1.setBounds(50,340,50,50);
b2=new Button("2");
b2.setBounds(120,340,50,50);
b3=new Button("3");
b3.setBounds(190,340,50,50);
b4=new Button("4");
b4.setBounds(50,270,50,50);

b5=new Button("5");
```

```
b5.setBounds(120,270,50,50);

b6=new Button("6");
b6.setBounds(190,270,50,50);
b7=new Button("7");
b7.setBounds(50,200,50,50); b8=new
Button("8");
b8.setBounds(120,200,50,50);
b9=new Button("9");
b9.setBounds(190,200,50,50);
b0=new Button("0");
b0.setBounds(120,410,50,50);
bneg=new Button("+/-");
bneg.setBounds(50,410,50,50);
bpts=new Button(".");
bpts.setBounds(190,410,50,50);
bback=new Button("back");
bback.setBounds(120,130,50,50);


 badd=new Button("+");
 badd.setBounds(260,340,50,50);
 bsub=new Button("-");
 bsub.setBounds(260,270,50,50);
 bmult=new Button("*");
 bmult.setBounds(260,200,50,50);
 bdiv=new Button("/");
 bdiv.setBounds(260,130,50,50);
 bmod=new Button("%");
 bmod.setBounds(190,130,50,50);
 bcalc=new Button("=");
 bcalc.setBounds(245,410,65,50);
 bclr=new Button("CE");
 bclr.setBounds(50,130,65,50);
```

```
b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);
b4.addActionListener(this);
b5.addActionListener(this);
b6.addActionListener(this);
b7.addActionListener(this);
b8.addActionListener(this);
b9.addActionListener(this);
b0.addActionListener(this);

bpts.addActionListener(this);
bneg.addActionListener(this);
bback.addActionListener(this);

badd.addActionListener(this);
bsub.addActionListener(this);
bmult.addActionListener(this);
bdiv.addActionListener(this);
bmod.addActionListener(this);
bcalc.addActionListener(this);
bclr.addActionListener(this);

f.addWindowListener(this);
//ADDING TO FRAME
f.add(l1);
f.add(b1); f.add(b2); f.add(b3); f.add(b4); f.add(b5);f.add(b6); f.add(b7); f.add(b8);f.add(b9);f.add(b0);

f.add(badd); f.add(bsub); f.add(bmod); f.add(bmult); f.add(bdiv); f.add(bmod);f.add(bcalc);

f.add(bclr); f.add(bpts);f.add(bneg); f.add(bback);

f.setSize(360,500);
```

```
f.setLayout(null);
f.setVisible(true);
  }
                    //FOR CLOSING THE WINDOW
  public void windowClosing(WindowEvent e) {
    f.dispose();
  }

  public void actionPerformed(ActionEvent e){
    String z,zt;
                      //NUMBER BUTTON
  if(e.getSource()==b1){
   zt=l1.getText();
   z=zt+"1";
   l1.setText(z);
  }
  if(e.getSource()==b2){
  zt=l1.getText();
  z=zt+"2";
  l1.setText(z);
  }
  if(e.getSource()==b3){
    zt=l1.getText();
    z=zt+"3";
    l1.setText(z);
  }
  if(e.getSource()==b4){
    zt=l1.getText();
    z=zt+"4";
    l1.setText(z);
  }
  if(e.getSource()==b5){
    zt=l1.getText();
```

```
  z=zt+"5";
  l1.setText(z);
 }
 if(e.getSource()==b6){
  zt=l1.getText();
  z=zt+"6";
  l1.setText(z);
 }
 if(e.getSource()==b7){
  zt=l1.getText();
  z=zt+"7";
  l1.setText(z);
 }
 if(e.getSource()==b8){
  zt=l1.getText();
  z=zt+"8";
  l1.setText(z);
 }
 if(e.getSource()==b9){
  zt=l1.getText();
  z=zt+"9";
  l1.setText(z);
 }
 if(e.getSource()==b0){
  zt=l1.getText();
  z=zt+"0";
  l1.setText(z);
 }

 if(e.getSource()==bpts){ //ADD DECIMAL PTS
  zt=l1.getText();
z=zt+"."; l1.setText(
```

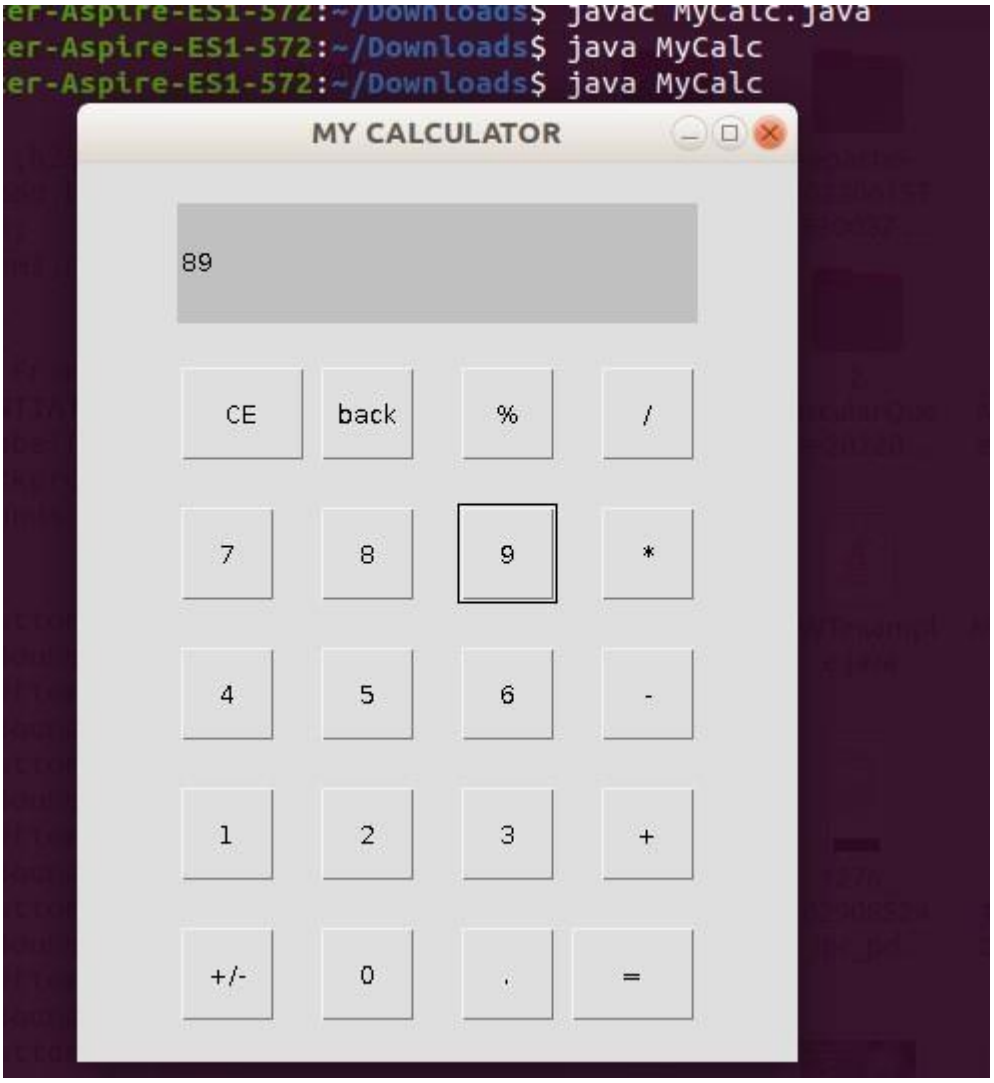
```
}
if(e.getSource()==bneg){ //FOR NEGATIVE
  zt=l1.getText();
  z="-"+zt;
  l1.setText(z);
}

if(e.getSource()==bback){ // FOR BACKSPACE
  zt=l1.getText();
  try{
    z=zt.substring(0, zt.length()-1);
    }catch(StringIndexOutOfBoundsException f){return;}
  l1.setText(z);
}
              //AIRTHMETIC BUTTON
if(e.getSource()==badd){                  //FOR ADDITION
  try{
    num1=Double.parseDouble(l1.getText());
    }catch(NumberFormatException f){
      l1.setText("Invalid Format");
      return;
    }
  z="";
  l1.setText(z);
  check=1;
}
if(e.getSource()==bsub){                  //FOR SUBTRACTION
  try{
    num1=Double.parseDouble(l1.getText());
    }catch(NumberFormatException f){
      l1.setText("Invalid Format");
      return;
    }
  z);
```

```
 z="";
 l1.setText(z);check=2;
}

 if(e.getSource()==bmult){                //FOR MULTIPLICATION
  try{
   num1=Double.parseDouble(l1.getText());
   }catch(NumberFormatException f){
    l1.setText("Invalid Format");
    return;
   }
  z="";
  l1.setText(z);
  check=3;
 }
 if(e.getSource()==bdiv){                 //FOR DIVISION
  try{
   num1=Double.parseDouble(l1.getText());
   }catch(NumberFormatException f){
    l1.setText("Invalid Format");
    return;
   }
  z="";
  l1.setText(z);
  check=4;
 }
 if(e.getSource()==bmod){                 //FOR MOD/REMAINDER
  try{
   num1=Double.parseDouble(l1.getText());
   }catch(NumberFormatException f){
    l1.setText("Invalid Format");
    return;
```

```
  z="";
  l1.setText(z);
  check=5;
}
                    //RESULT BUTTON
if(e.getSource()==bcalc){
  try{
    num2=Double.parseDouble(l1.getText());
    }catch(Exception f){
      l1.setText("ENTER NUMBER FIRST ");
      return;
    }
  if(check==1)
    xd =num1+num2;
  if(check==2)
    xd =num1-num2;
  if(check==3)
    xd =num1*num2;
  if(check==4)
    xd =num1/num2;
  if(check==5)
    xd =num1%num2;
  l1.setText(String.valueOf(xd));
}
                    //FOR CLEARING THE LABEL and Memory
if(e.getSource()==bclr){
  num1=0;
  num2=0;
  check=0;
  xd=0;
  z="";
   l1.setText(z);
```

```
}
//MAIN METHOD where objects of MyCalc is instantaiated
 public static void main(String args[]){
       new MyCalc();
   }
}
```

Output Screenshot:



OBJECT ORIENTED PROGRAMMING LAB

Name: Justin v Kalappura

Roll No: 10

Batch: S2 mca b

Date: 29-03-2022

Experiment No.: 32

Aim

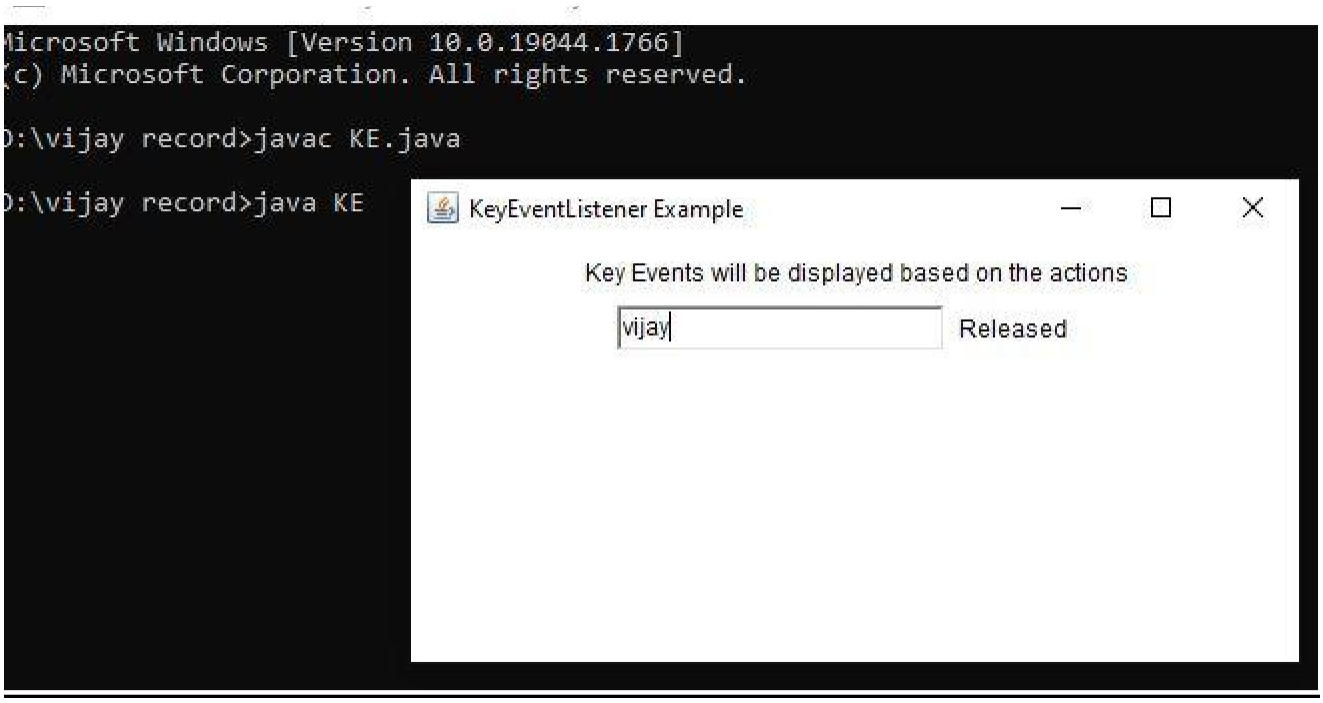
Develop a program to handle Key events.

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



OBJECT ORIENTED PROGRAMMING LAB

Name: Justin V Kalappura

Roll No: 10

Batch: S2 mca b

Date: 29-03-2022

Experiment No.: 33

Aim

Develop a program to handle all mouse events and window events

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