20MCA132-OBJECT ORIENTED PROGRAMMING LAB

RECORD

JOMEESH JOSE

ROLL NO : 03

S2-REG-MCA-B

TABLE OF CONTENTS

EXP NO.	EXPERIMENT	PAGE NO.
1	Create 3 objects of the class	5
2	matrix addition	8
3	Add complex numbers	11
4	Read a matrix and check whether it is symmetric or not	13
5	Program to Sort strings	16
6	Search an element in an array	18
7	Perform string manipulations	21
8	Program using Array of Objects	22
9	Area of different shapes using overloaded functions	25
10	Use array of objects to display details of N teachers	27
11	Use array of objects to display details of N teachers	31
12	Program to read and print book information using inheritance	36
13	Multiple inheritance using interface	40
14	Create a menu driven program to find area and perimeter of objects using interface	42
15	Program to calculate method from interface	47
16	Create a graphic package and test it	50
17	Create an arithmetic package and test it	54

18	**Write a user defined exception class to authenticate the user name and password**	**57**
19	**Find average of n positive integer and raise exception for each negative integer**	**60**
20	**Arithmetic operation using thread**	**63**
21	**Fibonacci series and even numbers using thread**	**66**
22	**Program to draw Circle, Rectangle, Line in Applet**	**68**
23	**Program to find maximum of three numbers using AWT.**	**71**
24	**Display happy face and sad face using applet based on marks secured**	**74**
25	**Program using Mouse event**	**80**
26	**Implement a simple calculator using AWT components.**	**84**
27	**Draw shapes for given parameters as per user's choice**	**88**
28	**Develop a program to handle all window events**	**92**
29	**Develop a program to handle all mouse events**	**94**
30	**Develop a program to handle Key events**	**96**
31	**Producer/Consumer using ITC**	**98**
32	**Program to create a generic stack and do the Push and Pop operations.**	**103**
33	**Using generic method perform Bubble sort**	**107**
34	**Program to demonstrate the creation of queue object using the PriorityQueue class**	**109**
35	**Program to remove all the elements from a linked list**	**111**
36	**program to demonstrate the addition and deletion of elements in dequeue**	**113**

37	**list of Strings using ArrayList from collection framework, perform built-in operations.**	**115**
38	**Program to demonstrate the working of map interface by adding ,removing,changing.**	**116**
39	**program to convert hash map to tree map.**	**117**
40	**Program list subdirectory and files , perform search operation**	**119**
41	**Program to write content to a file and display it on the console**	**121**

1. 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.

```
import java.util.*;
public class Product {
    int pcode;
    String pname;
    int price;
    public static void main(String[] args) {
        int smallest;
      Product p1 = new Product();
      Product p2 = new Product();
      Product p3 = new Product();
       p1.pcode=1001;
       p1.pname="RAM";
       p1.price=7000;
       p2.pcode=1002;
       p2.pname="Processor";
       p2.price=37000;
       p3.pcode=1003;
       p3.pname="SSD";
       p3.price=16700;
       if(p1.price<p2.price) {
```

```
    if(p3.price<p1.price) {
        smallest = p3.price;
        System.out.println(p3.pname+ " is the cheapest.");
    } else {
        smallest = p1.price;
        System.out.println(p1.pname+ " is the cheapest.");
    }
  } else {
    if(p2.price<p3.price) {
        smallest = p2.price;
        System.out.println(p2.pname+ " is the cheapest.");
    } else {
        smallest = p3.price;
        System.out.println(p3.pname+ " is the cheapest.");
    }
  }
  }
}
```

OUTPUT

```
C:\Users\Jomesh\Desktop\JAVA TRYING>javac Product.java

C:\Users\Jomesh\Desktop\JAVA TRYING>java Product
RAM is the cheapest.

C:\Users\Jomesh\Desktop\JAVA TRYING>
```

RESULT : The program has been executed and the output was verified.

2. Read 2 matrices from the console and perform matrix addition.

```
import java.util.*;
class matrixadd{
public static void main(String[] args)
{
int row,col,i,j;
Scanner sc=new Scanner(System.in);
System .out.print("enter the no of rows:");
row=sc.nextInt();
System .out.print("enter the no of columns:");
col=sc.nextInt();
int mat1[][]=new int[row][col];
int mat2[][]=new int[row][col];
int mat3[][]=new int[row][col];
System.out.print("enter the elements of matrix1 :");
for(i=0;i<row;i++)
{
for(j=0;j<col;j++)
{
mat1[i][j]=sc.nextInt();
}
```

```
System.out.println();
}
System.out.print("enter the elements of matrix2 :");
for(i=0;i<row;i++)
{
for(j=0;j<col;j++)
{
mat2[i][j]=sc.nextInt();
}
System.out.println();
}

for(i=0;i<row;i++)
{
for(j=0;j<col;j++)
{
mat3[i][j]=mat1[i][j]+mat2[i][j];
}
}
System.out.print("sum of matrix :");
for(i=0;i<row;i++)
{
```

```
for(j=0;j<col;j++)
{
System.out.print(mat3[i][j]+"\t");
}
System.out.println();
}
}
}
```

OUTPUT

```
C:\Users\Jomesh\Desktop\JAVA TRYING>javac matrixadd.java

C:\Users\Jomesh\Desktop\JAVA TRYING>java matrixadd
enter the no of rows:2
enter the no of columns:2
enter the elements of matrix1 :1
2

3
4

enter the elements of matrix2 :1
5

6
9

sum of matrix :2        7
9       13

C:\Users\Jomesh\Desktop\JAVA TRYING>
```

RESULT : The program has been executed and the output was verified.

3. Add complex numbers

```
public class Complex{
   double a, b;
   Complex(double r, double i){
    this.a = r;
    this.b = i;
   }
   public static Complex sum(Complex c1, Complex c2)
   {
       Complex temp = new Complex(0, 0);


       temp.a = c1.a + c2.a;
       temp.b = c1.b+ c2.b;
       return temp;
    }
    public static void main(String args[]) {
    Complex c1 = new Complex(5, 4);
    Complex c2 = new Complex(6, 3.5);
       Complex temp = sum(c1, c2);
       System.out.printf("Sum is: "+ temp.a+" + "+ temp.b +"i");
    }
```

```
}
```

OUTPUT

```
C:\Users\Jomesh\Desktop\JAVA TRYING>javac Complex.java

C:\Users\Jomesh\Desktop\JAVA TRYING>java Complex
Sum is: 11.0 + 7.5i
C:\Users\Jomesh\Desktop\JAVA TRYING>
```

RESULT : The program has been executed and the output was verified.

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

```
import java.util.Scanner;
public class Symmetric
{
   public static void main(String[] args)
   {
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter the no. of rows : ");
      int rows = sc.nextInt();
      System.out.println("Enter the no. of columns : ");
      int cols = sc.nextInt();
      int matrix[][] = new int[rows][cols];
      System.out.println("Enter the elements :");
      for (int i = 0; i < rows; i++)
      {
         for (int j = 0; j < cols; j++)
         {
            matrix[i][j] = sc.nextInt();
         }
      }
```

```
System.out.println("Printing the input matrix :");
for (int i = 0; i < rows; i++)
{
   for (int j = 0; j < cols; j++)
   {
      System.out.print(matrix[i][j]+"\t");
   }
   System.out.println();
}
if(rows != cols)
{
   System.out.println("The given matrix is not a square matrix, so it can't be symmetric.");
}
else
{
   boolean symmetric = true;
   for (int i = 0; i < rows; i++)
   {   for (int j = 0; j < cols; j++)   {
         if(matrix[i][j] != matrix[j][i])   {
            symmetric = false;
            break;         }
```

```
            }    }
        if(symmetric)
        {
            System.out.println("The given matrix is symmetric...");
        }
        else
        {
            System.out.println("The given matrix is not symmetric...");
        }   }   sc.close();
    }   }
```

OUTPUT

```
C:\Users\Jomesh\Desktop\JAVA TRYING>javac Symmetric.java

C:\Users\Jomesh\Desktop\JAVA TRYING>java Symmetric
Enter the no. of rows :
2
Enter the no. of columns :
3
Enter the elements :
1
2
3
4
5
5
Printing the input matrix :
1       2       3
4       5       5
The given matrix is not a square matrix, so it can't be symmetric.

C:\Users\Jomesh\Desktop\JAVA TRYING>
```

RESULT : The program has been executed and the output was verified.

5.Program to Sort strings

```
public class sortstring{
public static void main(String[] args)
{
String names[]={"amal","jyothi","college","of","engineering"};
String temp;
int n= names.length;
int i;
int j;
for(i=0;i<n;i++)
{
for(j=i+1;j<n;j++)
{
if(names[i].compareTo(names[j])>0)
{
        temp=names[i];
        names[i]=names[j];
        names[j]=temp;
}
}
}
System.out.println("the sorted array of string is :");
```

```
for(i=0;i<n;i++)
{
        System.out.println(names[i]);

}
}
}
```

OUTPUT

```
C:\Users\Jomesh\Desktop\JAVA TRYING>javac sortstring.java

C:\Users\Jomesh\Desktop\JAVA TRYING>java sortstring
the sorted array of string is :
amal
college
engineering
jyothi
of

C:\Users\Jomesh\Desktop\JAVA TRYING>
```

RESULT : The program has been executed and the output was verified.

6.Search an element in an array.

```
import java.util.*;
public class searchele{
public static void main(String[] args)
{
int n,i,b,flag=0;
Scanner s=new Scanner(System.in);
System.out.println("enter the number of elements for the array :");
n=s.nextInt();
int a[]=new int[n];
System.out.println("enter the elements of the array :");

for(i=0;i<n;i++)
{
a[i]=s.nextInt();
}
System.out.println("enter the element u want to search :");
b=s.nextInt();

for(i=0;i<n;i++)
{
```

```
if(a[i]==b)
{
flag=1;
break;
}
else
{
flag=0;
}
}
if(flag==1)
{
System.out.println("element found at position :"+(i+1));
}
else
{
System.out.println("element not found");
}
}
}
```

OUTPUT

```
C:\Users\Jomesh\Desktop\JAVA TRYING>java searchele.java
enter the number of elements for the array :
3
enter the elements of the array :
1
2
6
enter the element u want to search :
2
element found at position :2

C:\Users\Jomesh\Desktop\JAVA TRYING>
```

RESULT : The program has been executed and the output was verified.

7. Perform string manipulations.

```
public class Sample_String
{
  public static void main(String[] args)
{
  String str_Sample = "RockStar";
  System.out.println("Length of String: " + str_Sample.length());
  System.out.println("Character at position 5: " + str_Sample.charAt(5));
  System.out.println("EndsWith character 'r': " + str_Sample.endsWith("r"));
  System.out.println("Replace 'Rock' with 'Duke': " + str_Sample.replace("Rock", "Duke"));
}       }
```

OUTPUT

```
C:\Users\Jomesh\Desktop\JAVA TRYING>javac Sample_String.java

C:\Users\Jomesh\Desktop\JAVA TRYING>java Sample_String
Length of String: 8
Character at position 5: t
EndsWith character 'r': true
Replace 'Rock' with 'Duke': DukeStar

C:\Users\Jomesh\Desktop\JAVA TRYING>
```

RESULT : The program has been executed and the output was verified.

8. 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.

```
import java.util.Scanner;

public class Employee {

 int empid;
 String name;
 float salary;

 public void getInput() {

  Scanner in = new Scanner(System.in);
  System.out.print("Enter the empid :: ");
  empid = in.nextInt();
  System.out.print("Enter the name :: ");
  name = in.next();
  System.out.print("Enter the salary :: ");
  salary = in.nextFloat();
 }
```

```
public void display() {

 System.out.println("Employee id = " + empid);
 System.out.println("Employee name = " + name);
 System.out.println("Employee salary = " + salary);
}

public static void main(String[] args) {

 Employee e[] = new Employee[5];

 for(int i=0; i<5; i++) {

  e[i] = new Employee();
  e[i].getInput();
 }

 System.out.println("**** Data Entered as below ****");

 for(int i=0; i<5; i++) {

  e[i].display();
```

```
 }
 }
}
```

OUTPUT

```
C:\Users\Jomesh\Desktop\JAVA TRYING>java Employee
Enter the empid :: 101
Enter the name :: jomeesh
Enter the salary :: 25000
Enter the empid :: 102
Enter the name :: rajan
Enter the salary :: 30000
Enter the empid :: 103
Enter the name :: raju
Enter the salary :: 32000
Enter the empid :: 104
Enter the name :: babu
Enter the salary :: 33000
Enter the empid :: 105
Enter the name :: vikram
Enter the salary :: 26000
**** Data Entered as below ****
Employee id = 101
Employee name = jomeesh
Employee salary = 25000.0
Employee id = 102
Employee name = rajan
Employee salary = 30000.0
Employee id = 103
Employee name = raju
Employee salary = 32000.0
Employee id = 104
Employee name = babu
Employee salary = 33000.0
Employee id = 105
Employee name = vikram
Employee salary = 26000.0

C:\Users\Jomesh\Desktop\JAVA TRYING>
```

RESULT : The program has been executed and the output was verified.

9.Area of different shapes using overloaded functions

```
public class shape
{  int side,as,ar;
public void area(int a)//area of square
{  side=a;
as=a*a;
System.out.println("area of square is"+as);
}
public void area(double r)//area of circle
{
double radi=r;
double ac=(22/7)*radi*radi;
System.out.println("area of circle is"+ac);
}
public void area(int l,int w)//area of rectangle
{   int len=l;
int wid=w;
ar=len*wid;
System.out.println("area of rectangle"+ar);
}
public void area(int h,double r)//area of cylinder
```

```
{ int he=h;
double rad=r;
double acy=(2*(22/7)*rad*he)+((22/7)*rad*rad);
System.out.println("area of cylinder"+acy);    }
public static void main(String[] args)
{   shape s=new shape();
s.area(4);//area of square
s.area(5.52);//area of circle
s.area(5,4);//area of rectangle
s.area(5,4.5);      //area of cylinder  }
}
```

OUTPUT

```
C:\Users\Jomesh\Desktop\JAVA TRYING>javac shape.java

C:\Users\Jomesh\Desktop\JAVA TRYING>java shape
area of square is16
area of circle is91.41119999999998
area of rectangle20
area of cylinder195.75

C:\Users\Jomesh\Desktop\JAVA TRYING>
```

RESULT : The program has been executed and the output was verified.

10 .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.

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

int empid;
String name,address;
double salary;
public Employee(int empid, String name, String address, double salary) {
this.empid = empid;
this.name = name;
this.address = address;
this.salary = salary;
}
}
public class Teacher extends Employee
{
```

```
String subject,department;

public Teacher(int empid, String name, String address, double salary,String department,String subject ) {

super(empid, name, address, salary);

this.subject = subject;

this.department = department;

}


void display()

{

System.out.println("Empid : "+this.empid+" Name : "+this.name+" Salary : "+this.salary+" Address : "+this.address+" department : "+this.department+" Subjects : "+this.subject);

}

public static void main(String[] args) {

// TODO Auto-generated method stub

Scanner sc=new Scanner(System.in);

int n;

System.out.println("Enter number of Teachers : ");

n=sc.nextInt();

Teacher obj[]=new Teacher[n];

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

int j = i+1;
```

```
System.out.print("Enter Empid of teacher "+j+" : ");
int Empid = sc.nextInt();
System.out.print("Enter Name of teacher "+j+" : ");
String Name = sc.next();
System.out.print("Enter Salary of teacher "+j+" : ");
double Salary = sc.nextDouble();
System.out.print("Enter Address of teacher "+j+" : ");
String Address = sc.next();
System.out.print("Enter department of teacher "+j+" : ");
String department =sc.next();
System.out.print("Enter Subjects of teacher "+j+" : ");
String Subjects =sc.next();
obj[i] = new Teacher(Empid, Name, Address, Salary, department, Subjects);
}
System.out.println("\n-----------------------------------------------------------------\n");
System.out.println("Teacher's List \n");
for(int i=0;i<n;i++) {
obj[i].display();
}
}
}
```

OUTPUT

```
D:\MCA NOTES\Sem 2\Object Oriented Programming Lab\08-06-2021>java Teacher
Enter number of Teachers :
2
Enter Empid of teacher 1 : 1
Enter Name of teacher 1 : Rini
Enter Salary of teacher 1 : 40000
Enter Address of teacher 1 : Alappuzha
Enter department of teacher 1 : Mca
Enter Subjects of teacher 1 : Network
Enter Empid of teacher 2 : 2
Enter Name of teacher 2 : Vivin
Enter Salary of teacher 2 : 50000
Enter Address of teacher 2 : Trivandrum
Enter department of teacher 2 : Mca
Enter Subjects of teacher 2 : Java

------------------------------------------------------------------------

Teacher's List

Empid : 1 Name : Rini Salary : 40000.0 Address : Alappuzha department : Mca Subjects : Network
Empid : 2 Name : Vivin Salary : 50000.0 Address : Trivandrum department : Mca Subjects : Java
```

RESULT : The program has been executed and the output was verified.

11.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.

```
import java.util.Scanner;
class Person
{ String name,gender,address;
int age;
public Person(String name, String gender, String address, int age) {
super();
this.name = name;
this.gender = gender;
this.address = address;
this.age = age; } }
class Employee extends Person {
int empid;
String company_name,qualification;
double salary;
```

```
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 {
String subject,department;
int teacherid;
public Teacher(String name, String gender, String address, int age, int empid, String company_name,
String qualification, double salary, String subject, String department, int teacherid) {
super(name, gender, address, age, empid, company_name, qualification, salary);
this.subject = subject;
this.department = department;
this.teacherid = teacherid;  }
void display() {
System.out.println("....Personal details...");
System.out.println(" Name : "+this.name+" Gender : "+this.gender+" Age :"+this.age);
System.out.println("...Employee details....");
```

```
System.out.println("Empid : "+this.empid +" company_name : "+this.company_name+" Salary : "+this.salary+" Address : "+this.address+" qualification : "+this.qualification);

System.out.println("...Teacher's details...");

System.out.println(" teacherid : "+this.teacherid+ " department : "+this.department+" Subjects : "+this.subject);    }   }

public class Main {

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

int n;

System.out.println("Enter number of Teachers : "); n=s.nextInt();

Teacher obj[]=new Teacher[n];

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

System.out.println("Enter the person name:"); String nam1=s.next();

System.out.println("Enter the Gender: "); String gen1=s.next();

System.out.println("Enter the Address: "); String adr1=s.next();

System.out.println("Enter the Age:"); int age1=s.nextInt();

System.out.println("Enter the Employee id: ");

int id1=s.nextInt();

System.out.println("Enter the Company name: ");

String cname1=s.next();

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

double sal1=s.nextDouble();
```

```
System.out.println("Enter the Qualification:");
String qu1=s.next();
System.out.println("Enter the Teacher id: ");
int tid1=s.nextInt();
System.out.println("Enter the Department:");
String dept1=s.next();
System.out.println("Enter the Subject:");
String sub1=s.next();
obj[i]=new Teacher(nam1,gen1,adr1,age1,id1,cname1,qu1,sal1,sub1,dept1,tid1);   }
System.out.println("\n-----------------------------------------------------------------\n");
for(int i=0;i<n;i++) {
obj[i].display();  }  }   }
```

OUTPUT

```
D:\MCA NOTES\Sem 2\Object Oriented Programming Lab\08-06-2021>java Main
Enter number of Teachers :
1
Enter the person name:
Rini
Enter the Gender:
Female
Enter the Address:
Alappuzha
Enter the Age:
26
Enter the Employee id:
5
Enter the Company name:
AJCE
Enter the Salary:
40000
Enter the Qualification:
MCA
Enter the Teacher id:
2
Enter the Department:
MCA
Enter the Subject:
Network

------------------------------------------------------------------------

....Personal details...
 Name : Rini Gender : Female Age :26
...Employee details....
Empid : 5 company_name : AJCE Salary : 40000.0 Address : Alappuzha qualification : MCA
...Teacher's details...
 teacherid : 2 department : MCA Subjects : Network
```

RESULT : The program has been executed and the output was verified.

12 . 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.

```
import java.util.Scanner;
class Publisher {
String Pubname;
Publisher()
{
Scanner s=new Scanner(System.in);
System.out.println("Enter publisher name");
Pubname=s.next();
}
}
class Book extends Publisher
{
String title, author;
int price;
Book()
{
Scanner s=new Scanner(System.in);
System.out.println("Enter Title of the book");
title=s.next();
```

```
System.out.println("Enter Author's name");
author=s.next();
System.out.println("Enter price");
price=s.nextInt();
} }
class Literature extends Book
{ Literature()
{ System.out.println("Literature Books"); }
void display()
{
System.out.println("Publisher name: "+Pubname);
System.out.println("Title of the book: "+title);
System.out.println("Author's name: "+author);
System.out.println("Price: "+price);
} }
class Fiction extends Literature
{ Fiction()
{ System.out.println("Friction Books");  }
void display()
{ super.display(); }
public static void main(String args[])
{  int n;
```

```
Scanner s=new Scanner(System.in);
System.out.println("Enter the No of literature book: ");
int a=s.nextInt();
Literature L[]=new Literature[a];
for(int i=0;i<a;i++)
{ L[i]=new Literature(); }
System.out.println("Enter the No of Fiction book: ");
int b=s.nextInt();
Fiction F[]=new Fiction[b];
for(int i=0;i<b;i++)
{ F[i]=new Fiction(); }
int no;
System.out.println("Enter your choice of book");
no=s.nextInt();
int type =no;
switch (no) {
case 1:
System.out.println(".....Details of literature books");
for(int i=0;i<a;i++)
L[i].display();
break;
case 2:
```

```
System.out.println(".....Details of fiction books");
for(int i=0;i<b;i++)
F[i].display();
break;
default:
System.out.println("Wrong input"); } }
}
```

OUTPUT

```
C:\Users\Jomesh\Desktop\JAVA TRYING>java Fiction
Enter the No of literature book:
1
Enter publisher name
ram
Enter Title of the book
vettam
Enter Author's name
jhonson
Enter price
125
Literature Books
Enter the No of Fiction book:
1
Enter publisher name
ram
Enter Title of the book
manoharam
Enter Author's name
charli
Enter price
450
Literature Books
Friction Books
Enter your choice of book
1
.....Details of literature books
Publisher name: ram
Title of the book: vettam
Author's name: jhonson
Price: 125

C:\Users\Jomesh\Desktop\JAVA TRYING>
```

RESULT : The program has been executed and the output was verified.

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

```
interface student
{ void stresullt(); }
interface sports
{       void spresult();   }
class result implements student,sports{
        public void spresult()  {
        String hundred="First";
        String twohundred="Second";
        String fivehundred="First";
        String relay="Second";
        System.out.println("Sports Result");
       System.out.println("Hundred Meter:"+hundred);
        System.out.println("Two Hundred Meter:"+twohundred);
       System.out.println("Five Hundred Meter:"+fivehundred);
       System.out.println("Relay:"+relay);   }
        public void stresullt()   {
        int physics=30;
        int chemistry=40;
        int maths=45;
        int english=50;
```

```
 int computer=50;
System.out.println("Marks");
System.out.println("Physics:"+physics);
System.out.println("Chemistry:"+chemistry);
System.out.println("Mathematics:"+maths);
System.out.println("English:"+english);
System.out.println("Computer:"+computer);    }
 public static void main(String[] args)
 {    result r = new result();
        r.stresullt();
        r.spresult();    }  }
```

OUTPUT

```
D:\MCA NOTES\Sem 2\Object Oriented Programming Lab\15-6-2021>java result
Marks
Physics:30
Chemistry:40
Mathematics:45
English:50
Computer:50
Sports Result
Hundred Meter:First
Two Hundred Meter:Second
Five Hundred Meter:First
Relay:Second
```

RESULT : The program has been executed and the output was verified.

14 .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.

```
import java.util.Scanner;
interface Shape
{
   void input();
   void area();
   void perimeter();
}
class Circle implements Shape
{

   int r = 0;
   double pi = 3.14, ar = 0,per=0;
   public void input()
   {       Scanner s = new Scanner(System.in);
           System.out.print("Enter radius of circle:");
          r= s.nextInt();
   }
   public void area()
   {
```

```
        ar = pi * r * r;
        System.out.println("Area of circle:"+ar);
    }
    public void  perimeter()
    {
            per = 2 * pi * r;
            System.out.println("Perimeter of circle:"+per);
    }

}
class Rectangle implements Shape
{
    int l = 0, b = 0;
    double ar,per;
    public void input()
    {    Scanner s = new Scanner(System.in);
         System.out.print("Enter length of rectangle:");
        l = s.nextInt();
         System.out.print("Enter breadth of rectangle:");
        b = s.nextInt();     }
    public void area()
    {    ar = l * b;
```

```
        System.out.println("Area of rectangle:"+ar);    }
    public void perimeter()
    {      per = 2 * (l + b);
             System.out.println("Perimeter of rectangle:"+per);   }    }
public class shapes
{
    public static void main(String[] args)
    { int n;
        Scanner s = new Scanner(System.in);
       Rectangle obj1 = new Rectangle();
        Circle obj2 = new Circle();


        System.out.println("1.Area of circle");
        System.out.println("2.Perimeter of circle");
        System.out.println("3.Area of rectangle");
        System.out.println("4.Perimeter of rectangle");
        System.out.println("Enter your option:");
            n= s.nextInt();
        switch(n) {
        case 1:
           obj2.input();
            obj2.area();
```

```
        break;
        case 2:
           obj2.input();
            obj2.perimeter();
        break;
          case 3:
           obj2.input();
            obj2.area();
        break;
        case 4:
           obj2.input();
            obj2.perimeter();
        break;
        default:
         System.out.println("Invalid option");
}
  }
   }
```

OUTPUT

```
D:\MCA NOTES\Sem 2\Object Oriented Programming Lab\Java>javac shapes.java

D:\MCA NOTES\Sem 2\Object Oriented Programming Lab\Java>java shapes
1.Area of circle
2.Perimeter of circle
3.Area of rectangle
4.Perimeter of rectangle
Enter your option:
1
Enter radius of circle:3
Area of circle:28.259999999999998
```

RESULT : The program has been executed and the output was verified.

15.Prepare bill with the given format using calculate method from interface.
Order No.

```
interface bill
{
	int productdetails();
}
class product1 implements bill{
		int id = 101,quantity= 2,unit=25,total=0;
		String name="A";

	public int productdetails()
	{
	total = quantity * unit;
		System.out.println("Product Id :"+id);
		System.out.println("Name :"+name);
		System.out.println("Quantity :"+quantity);
		System.out.println("Unit price :"+unit);
		System.out.println("Total :"+total);
		return(total);

	}
```

```
}
class product2 implements bill{
        int id = 102,quantity= 1,unit=100,total=0;
                String name="B";


        public int productdetails()
        {
                total = quantity * unit;
                System.out.println("Product Id :"+id);
                System.out.println("Name :"+name);
                System.out.println("Quantity :"+quantity);
                System.out.println("Unit price :"+unit);
                System.out.println("Total :"+total);
                return(total);      }    }
public class productbill
{
         public static void main(String[] args)
         {
                 product1 p1 = new product1();
                 product2 p2 = new product2();
                int t1= p1.productdetails();
                int t2= p2.productdetails();
```

```
            int t3=t1+t2;


        System.out.println("Net. Amount :"+t3);


         }
}
```

OUTPUT

```
D:\MCA NOTES\Sem 2\Object Oriented Programming Lab\15-6-2021>javac productbill.java

D:\MCA NOTES\Sem 2\Object Oriented Programming Lab\15-6-2021>java productbill
Product Id :101
Name :A
Quantity :2
Unit price :25
Total :50
Product Id :102
Name :B
Quantity :1
Unit price :100
Total :100
Net. Amount :150
```

RESULT : The program has been executed and the output was verified.

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

```
package Graphiccs;
interface Area1
{
        public void Rectangle();public
        void Triangle(); public void
        Square(); public void Circle();
        public void getRect(); public
        void getTri(); public void
        getSqr(); public void getCrl();
}

//shapes.java

package Graphiccs;import
java.util.*;

public class shapess implements Area1
{
        double lr,lb,ra,th,tb,ta,saa,sa,cr,cc;
        public void getrect()
        {
                Scanner ab= new Scanner(System.in);
                System.out.println("Enter the length of the
                rectangle");lr=ab.nextInt();
                System.out.println("Enter the breadth of the
                rectangle");lb=ab.nextInt();
```

```
}
public void rectangle()
{
        ra=lr*lb;
        System.out.println("Area of Rectangle is "+ra);
}
public void getTri()
{
        Scanner cb= new Scanner(System.in);
        System.out.println("Enter the height of the
        Triangle");th=cb.nextInt();
        System.out.println("Enter the base of the
        Triangle");tb=cb.nextInt();
}
public void Triangle()
{
        ta=0.5*th*tb;
        System.out.println("Area of Triangle angle is "+ta);
}
public void getSqr()
{
        Scanner sq= new Scanner(System.in);
        System.out.println("Enter the Side of the
        Square");sa=sq.nextInt();
}
```

```
public void Square()
{
        saa=sa*sa;
        System.out.println("Area of Square is "+saa);
}

public void getCrl()
{
        Scanner sc= new Scanner(System.in);
        System.out.println("Enter the radius of the
        Circle");cc=sc.nextInt();
}
public void Circle()
{
        cr=3.14*cc*cc;
        System.out.println("Area of Square is "+cr);
}
public static void main(String[] args)
{
        shapess o= new shapess();
        o.getrect();
        o.rectangle();o.getTri();
        o.Triangle();
        o.getSqr();
        o.Square();
```

```
            o.getCrl();
            o.Circle();
        }
}
```

OUTPUT

RESULT : The program has been executed and the output was verified.

17. Create an Arithmetic package that has classes and interfaces for the 4 basicarithmetic operations. Test the package by implementing all operations ontwo given numbers

```
package Aarithmetic;

interface operations

{

        public void input(); public
        void add(); public void
        substract();public void
        multiply(); public void
        division();


}

package Aarithmetic;import
java.util.*;



public class basic implements operations

{

        double a,b,ad,dif,mult,div;public
        void input()
        {

                Scanner ab=new Scanner(System.in);
                System.out.println("Enter two numbers");
                a=ab.nextInt();
                b=ab.nextInt();
```

```
}
public void add()
{
        ad=a+b;
        System.out.println("Sum is "+ad);
}
public void substract()
{

        dif=a-b;
        System.out.println("Difference is "+dif);
}
public void multiply()
{
        mult=a*b;
        System.out.println("Product is "+mult);
        }
public void division()
{
        div=a/b;
        System.out.println("Quotient is "+div);
}
public static void main(String[] args)
```

```
        {
                basic o=new basic();o.input();
                o.add(); o.substract();
                o.multiply();
                o.division();
        }
}
```

OUTPUT

RESULT : The program has been executed and the output was verified.

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

```
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 CheckLoginCredential {

 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

```
D:\MCA NOTES\Sem 2\Object Oriented Programming Lab\14-08-2021(Bijimol)>java CheckLoginCredential
Enter username :: Antony
Enter password :: 12345
PasswordException: Incorrect password
Type correct password ???
        at CheckLoginCredential.main(CheckLoginCredential.java:35)
The finally statement is executed

D:\MCA NOTES\Sem 2\Object Oriented Programming Lab\14-08-2021(Bijimol)>
```

RESULT : The program has been executed and the output was verified.

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

```
import java.util.Scanner;
import java.util.InputMismatchException;
public class TestDemo
{
	public static void main(String args[])
	{
		double total = 0, N, userInput;
		Scanner input = new Scanner(System.in);
		while (true)
		{
		 System.out.print("Enter how many numbers(N) to calculate average:");
		 userInput = input.nextDouble();
		  if (userInput > 0)
		  {
			N = userInput;
			break;
		  }
		  else
				System.out.println("N must be positive.");
		}
```

```
        for (int i = 0; i < N; i++)
        {
            while (true)
            {
                System.out.print("Enter number:");
                try
                {
                    userInput = input.nextDouble();
                    total += userInput;
                    break;

                }
                catch (InputMismatchException e)
                {
                    input.nextLine();
                    System.out.println("Input must bea number. Try again");
                }
            }
        }
        System.out.println("Average: "+ total / N);
    }
}
```

OUTPUT

```
D:\MCA NOTES\Sem 2\Object Oriented Programming Lab\14-08-2021(Bijimol)>javac TestDemo.java

D:\MCA NOTES\Sem 2\Object Oriented Programming Lab\14-08-2021(Bijimol)>java TestDemo
Enter how many numbers(N) to calculate average:5
Enter number:2
Enter number:5
Enter number:7
Enter number:14
Enter number:12
Average: 8.0
```

RESULT : The program has been executed and the output was verified.

20. Define 2 classes; one for generating multiplication table of 5 and other for displaying first N prime numbers. Implement using threads. (Thread class)

```
import java.util.*;
class ThreadA extends Thread{
    public void run( ) {
       int n = 5;
       for (int i = 1; i <= 10; ++i)
          System.out.println(n + " * " + i +
                             " = " + n * i);
       System.out.println("Exiting from Thread A ...");
    }
}

class ThreadB extends Thread
{
   public void run( )
   {
       Scanner sc = new Scanner(System.in);
 int  i,n,p,count,flag;
 System.out.println("Enter the number of prime terms you want!");
    n=sc.nextInt();
    System.out.println("First "+n+" prime numbers are :-");
```

```
 p=2;
   i=1;
      while(i<=n)
   {
      flag=1;
      for(count=2;count<=p-1;count++)
      {
         if(p%count==0)
         {
          flag=0;
          break;
         }
      }
         if(flag==1)
         {
           System.out.print(p+" ") ;
          i++;
         }
      p++;
   }
}
```

```
//System.out.println("Exiting from Thread B ...");
}
public class Demonstration_111
{
   public static void main(String args[]) {
        ThreadA a = new ThreadA();
        ThreadB b = new ThreadB();
        a.start();
        b.start();
        System.out.println("... Multithreading is over ");
   }
}
```

OUTPUT

```
D:\MCA NOTES\Sem 2\Object Oriented Programming Lab\14-08-2021(Bijimol)>java Demonstratio
... Multithreading is over
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
Exiting from Thread A ...
Enter the number of prime terms you want!
4
First 4 prime numbers are :-
2 3 5 7
```

RESULT : The program has been executed and the output was verified.

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

```
public class Mythread {


   public static void main(String[] args) {
      Runnable r = new Runnable1();
      Thread t = new Thread(r);
      t.start();
      Runnable r2 = new Runnable2();
      Thread t2 = new Thread(r2);
      t2.start();
   }
}


class Runnable2 implements Runnable{
   public void run(){
      for(int i=0;i<11;i++){
         if(i%2 == 1)
            System.out.println(i);
      }
   }
}
```

```
class Runnable1 implements Runnable{
   public void run(){
        int n1=0,n2=1,n3,i,count=10;
 System.out.print(n1+" "+n2);//printing 0 and 1

 for(i=2;i<count;++i)//loop starts from 2 because 0 and 1 are already printed
 {
  n3=n1+n2;
  System.out.print(" "+n3);
  n1=n2;
  n2=n3;
 }
    }
}
```

OUTPUT

```
D:\MCA NOTES\Sem 2\Object Oriented Programming Lab\14-08-2021(Bijimol)>javac Mythread.java

D:\MCA NOTES\Sem 2\Object Oriented Programming Lab\14-08-2021(Bijimol)>java Mythread
0 1 1 2 3 5 8 13 211
3
5
7
9
 34
```

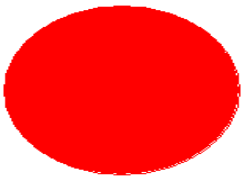
RESULT : The program has been executed and the output was verified.

22. Program to draw Circle, Rectangle, Line in Applet.

```
import java.awt.*;
import java.applet.*;
public class circle extends Applet
{
public void paint(Graphics g)
{   g.setColor(Color.red);
   g.fillOval(80,70,150,150);
        g.drawOval(80,70,150,150);
        g.setColor(Color.BLACK);
}
}
```

```
<html>
<head>
</head>
<body>
<div align="center">
<applet code="circle.class"width="800"height="500">
</applet>
</div>
</body>
```

```
</html>
```



```
import java.awt.*;
import java.applet.*;
public class rectapplet extends Applet
{
public void paint(Graphics g)
{   g.setColor(Color.YELLOW);
    g.fillRect(50,100,180,80);
        g.setColor(Color.BLACK);
        g.drawRect(50,100,180,80);

}
}
```

```
<html>
<head>
</head>
<body>
<div align="center">
<applet code="rectapplet.class"width="800"height="500">
</applet>
</div>
</body>
</html>
```

RESULT : The program has been executed and the output was verified.

23. Program to find maximum of three numbers using AWT.

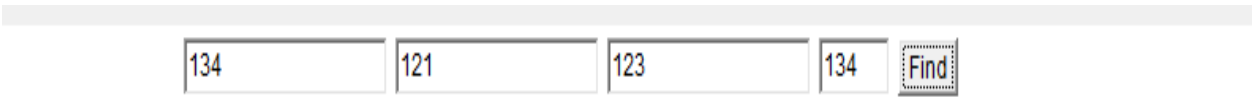
```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
public class findlarge extends Applet implements ActionListener
{
 TextField t1,t2,t3,t4;
 Button b1;
 public void init()
 {
    t1=new TextField(15);
    t1.setBounds(100,25,50,20);
    t2=new TextField(15);
    t2.setBounds(100,25,50,20);
    t3=new TextField(15);
    t3.setBounds(100,25,50,20);
    t4=new TextField("Ans");
    t4.setBounds(175,50,50,20);
    b1= new Button("Find");
    b1.setBounds(175,65,50,40);
    add(t1);
```

```
    add(t2);
    add(t3);
    add(t4);
    add(b1);
    b1.addActionListener(this);

}
public void actionPerformed(ActionEvent e)
{
    int i,j,k;
    i=Integer.parseInt(t1.getText());
    j=Integer.parseInt(t2.getText());
    k=Integer.parseInt(t3.getText());
   if(i<j)
   {
       if(j<k)
       t4.setText(""+k);
       else
       t4.setText(""+j);
   }
   else
    t4.setText(""+i);
```

```
 }
}
<html>
<head>
</head>
<body>
<div align="center">
<applet code="findlarge.class" width="800" height="500">
</applet>
</div>
</body>
</html>
```

OUTPUT



RESULT : The program has been executed and the output was verified.

24. Find the percentage of marks obtained by a student in 5 subjects. Display a happy face if he secures above 50% or a sad face if otherwise.

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
public class marks extends Applet implements ActionListener {
public int per =0;
Label l1 = new Label("enter Marks of Subject 1: ");
 Label l2 = new Label("enter Marks of Subject 2: ");
 Label l3 = new Label("enter Marks of Subject 3: ");
 Label l4 = new Label("enter Marks of Subject 4: ");
 Label l5 = new Label("enter Marks of Subject 5: ");
 Label l6 = new Label("Total Percentage: ");

 TextField t1 = new TextField(10);
 TextField t2 = new TextField(10);
 TextField t3 = new TextField(10);
 TextField t4 = new TextField(10);
 TextField t5 = new TextField(10);
 TextField t6 = new TextField(10);
```

```
Button b1 = new Button("CALCULATE PERCENTAGE");


public marks()
{
l1.setBounds(50, 100, 280, 20);
l2.setBounds(50, 150, 280, 20);
l3.setBounds(50, 200, 280, 20);
l4.setBounds(50, 250, 280, 20);
l5.setBounds(50, 300, 280, 20);
l6.setBounds(50, 350, 280, 20);

t1.setBounds(200, 100, 300, 20);
t2.setBounds(200, 150, 300, 20);
t3.setBounds(200, 200, 300, 20);
t4.setBounds(200, 250, 300, 20);
t5.setBounds(200, 300, 300, 20);
t6.setBounds(200, 350, 300, 20);

b1.setBounds(200,400, 200, 20);
GridLayout g1 = new GridLayout(20, 2, 5, 5);
setLayout(g1);
add(l1);
```

```
add(t1);
add(l2);
add(t2);
add(l3);
add(t3);
add(l4);
add(t4);
add(l5);
add(t5);
add(l6);
add(t6);
add(b1);
b1.addActionListener(this);
}
@Override
public void actionPerformed(ActionEvent e) {
// TODO Auto-generated method stub
int m1 = Integer.parseInt(t1.getText());
int m2= Integer.parseInt(t2.getText());
int m3= Integer.parseInt(t3.getText());
int m4= Integer.parseInt(t4.getText());
int m5= Integer.parseInt(t5.getText());
```

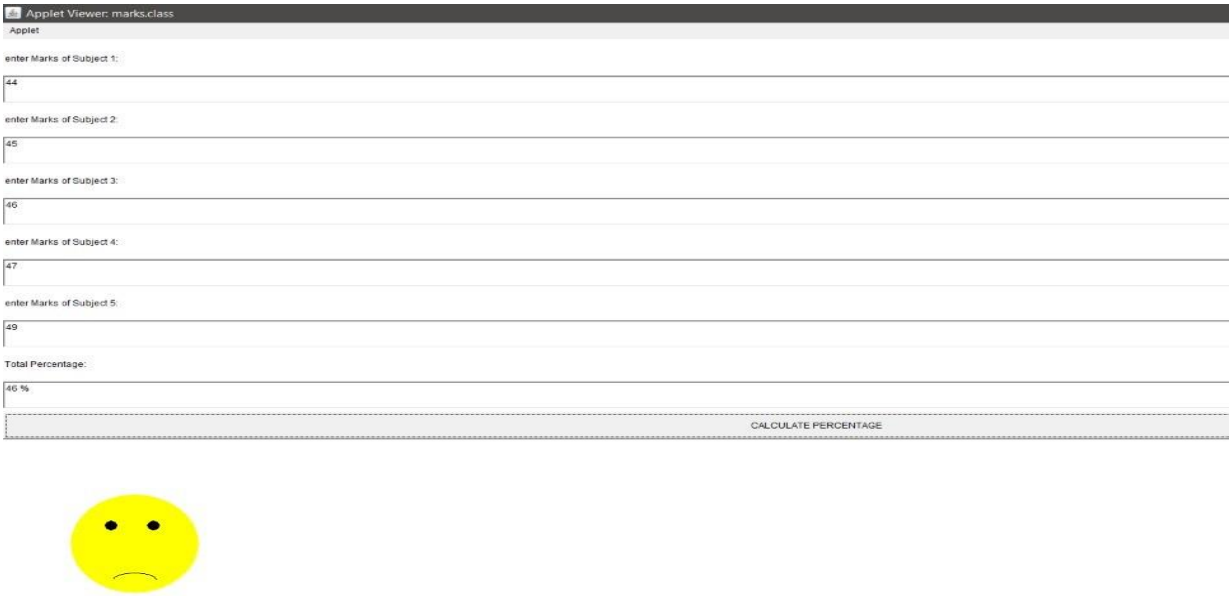
```
 if(e.getSource()==b1)
 {
 int add=m1+m2+m3+m4+m5;
 per=add/5;
 t6.setText(String.valueOf(per)+" %");

 repaint();
 }

}
public void paint(Graphics g)
 {
if(per>=50)
{
g.setColor(Color.yellow);
g.drawOval(100, 700, 150, 150);
g.fillOval(100, 700, 150, 150);
g.setColor(Color.BLACK);
g.fillOval(120, 740, 15, 15);
g.fillOval(170, 740, 15, 15);
g.drawArc(130, 800, 50, 20, 180, 180);
```

```
}
else if(per>0 && per<50)
{
g.setColor(Color.yellow);
g.drawOval(100, 700, 150, 150);
g.fillOval(100, 700, 150, 150);
 g.setColor(Color.BLACK);
 g.fillOval(120, 740, 15, 15);
 g.fillOval(170, 740, 15, 15);
g.drawArc(130,820,50,20,0,180);
}
 }
public static void main(String args[]) {
 new marks();
 }   }
<html><head>
</head>
<body><div align="center">
<applet code="marks.class"width="1000"height="1000">
</applet></div>
</body></html>
```

OUTPUT



RESULT : The program has been executed and the output was verified.

25. Using 2D graphics commands in an Applet, construct a house. On mouse click event, change the color of the door from blue to red.

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

public class house extends  Applet implements MouseListener,Runnable
{
        private Color textColor = Color.BLUE;
public void paint(Graphics g)
{ int [] x = {150, 300, 225};
int [] y = {150, 150, 25};
g.drawRect(150, 150, 150, 200); //House
g.drawRect(200, 200, 50, 150);
g.setColor(Color.blue);
g.setColor(textColor);
g.fillRect(200, 200, 50, 150); // Door
g.setColor(Color.black);
g.fillPolygon(x, y, 3); // Roof
}
```

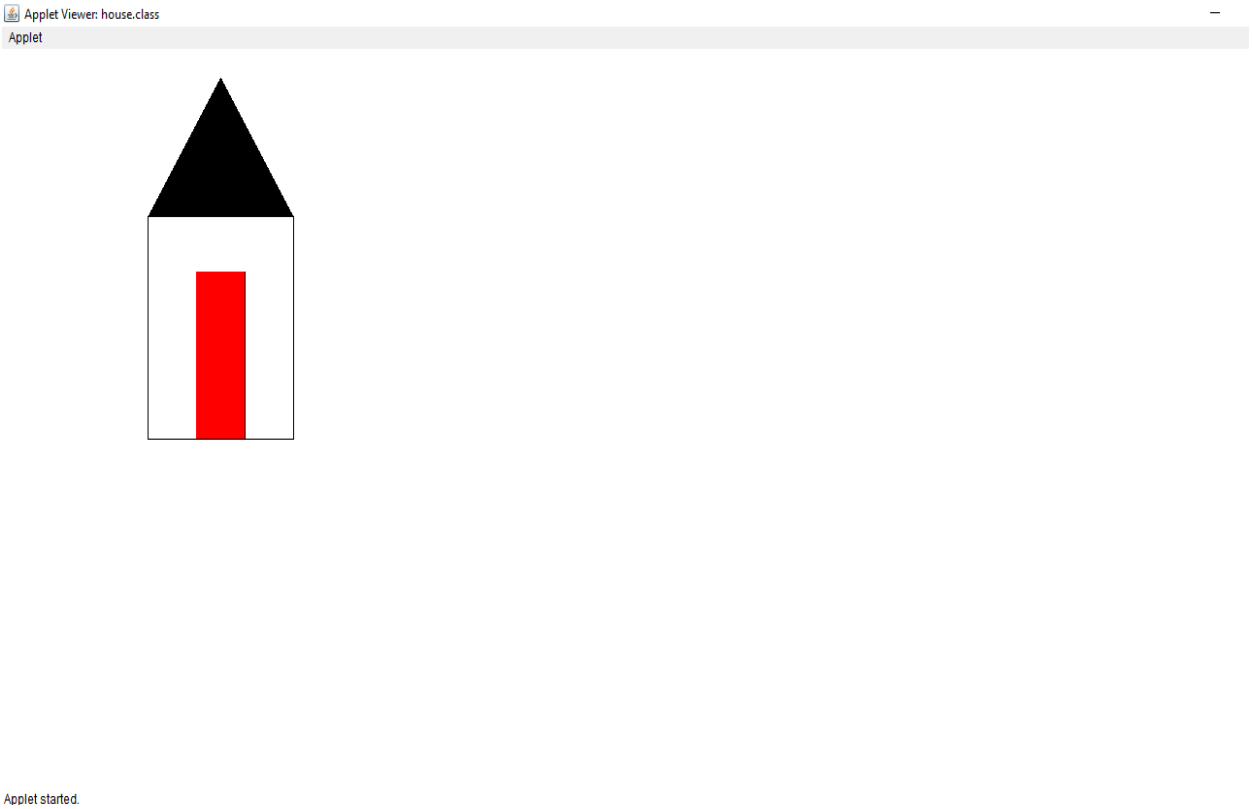
```
public void init()
   {
      this.setSize(200,200);
      addMouseListener(this);
   }
   public void run()
   {
      while(true)
      {
         repaint();
         try
         {
           Thread.sleep(17);
         }
         catch (InterruptedException e)
         {
            e.printStackTrace();
         }

      }
   }
   public void mouseClicked(MouseEvent e)
```

```
    {
        int x=e.getX(),y=e.getY();
        if(x>=60 && x<=120 && y>=80 && y<=95)
            textColor=Color.BLUE;
        else
            textColor=Color.RED;
            repaint();
            System.out.println("Mouse Position: X= "+x+"Y"+y);
    }
    public void mousePressed(MouseEvent e){}
    public void mouseReleased(MouseEvent e){}
    public void mouseEntered(MouseEvent e){}
    public void mouseExited(MouseEvent e){}
}


<html><head></head>
<body><div align="center">
<applet code="house.class"width="800"height="500">
</applet></div>
</body></html>
```

OUTPUT



RESULT : The program has been executed and the output was verified.

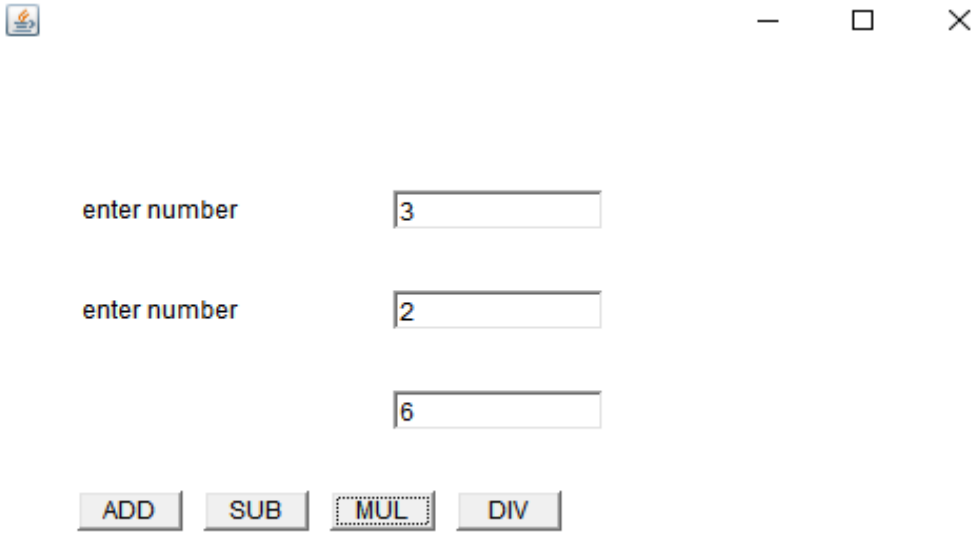
26. Implement a simple calculator using AWT components.

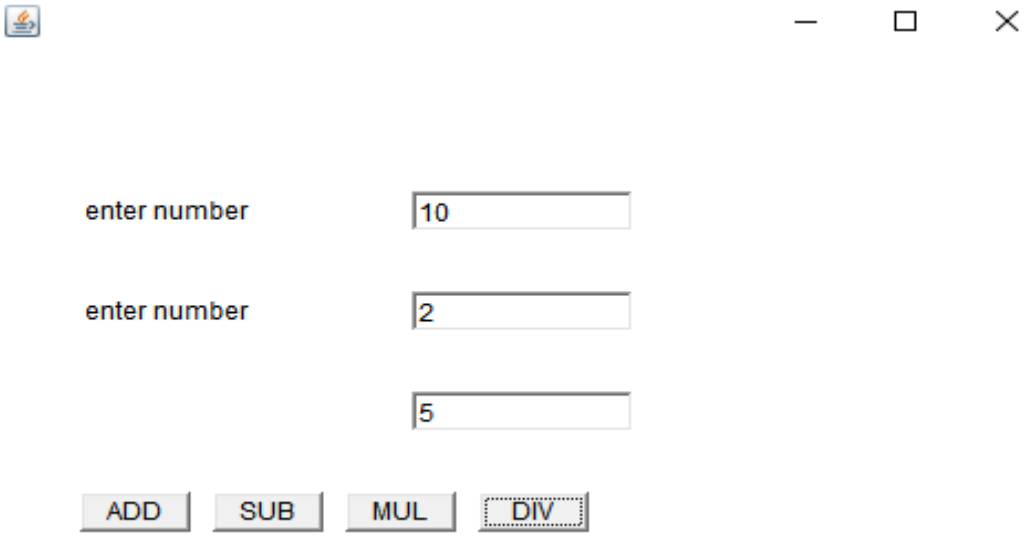
```
import java.awt.*;
import java.awt.event.*;
class calc implements ActionListener
{
Frame f=new Frame();
Label l1=new Label("enter number ");
Label l2=new Label("enter number ");
Label l3=new Label("result");
TextField t1=new TextField();
TextField t2=new TextField();
TextField t3=new TextField();
Button b1=new Button("ADD");
Button b2=new Button("SUB");
Button b3=new Button("MUL");
Button b4=new Button("DIV");
calc()  {
l1.setBounds(50,100,100,20);
l2.setBounds(50,150,100,20);
l3.setBounds(50,200,100,20);
t1.setBounds(200,100,100,20);
t2.setBounds(200,150,100,20);
```

```
t3.setBounds(200,200,100,20);
b1.setBounds(50,250,50,20);
b2.setBounds(110,250,50,20);
b3.setBounds(170,250,50,20);
b4.setBounds(230,250,50,20);
f.add(l1);
f.add(l2);
f.add(t1);
f.add(t2);
f.add(t3);
f.add(b1);
f.add(b2);
f.add(b3);
f.add(b4);
b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);
b4.addActionListener(this);
f.setLayout(null);
f.setVisible(true);
f.setSize(500,500);    }
public void actionPerformed(ActionEvent e){
```

```
int i=Integer.parseInt(t1.getText());
int j=Integer.parseInt(t2.getText());
if(e.getSource()==b1) {
t3.setText(String.valueOf(i+j));   }
if(e.getSource()==b2)  {
t3.setText(String.valueOf(i-j));   }
if(e.getSource()==b3)
{
t3.setText(String.valueOf(i*j));
}
if(e.getSource()==b4)
{
t3.setText(String.valueOf(i/j));     }
}
public static void main(String args[]) {
new calc();
}
}
```

OUTPUT





RESULT : The program has been executed and the output was verified.

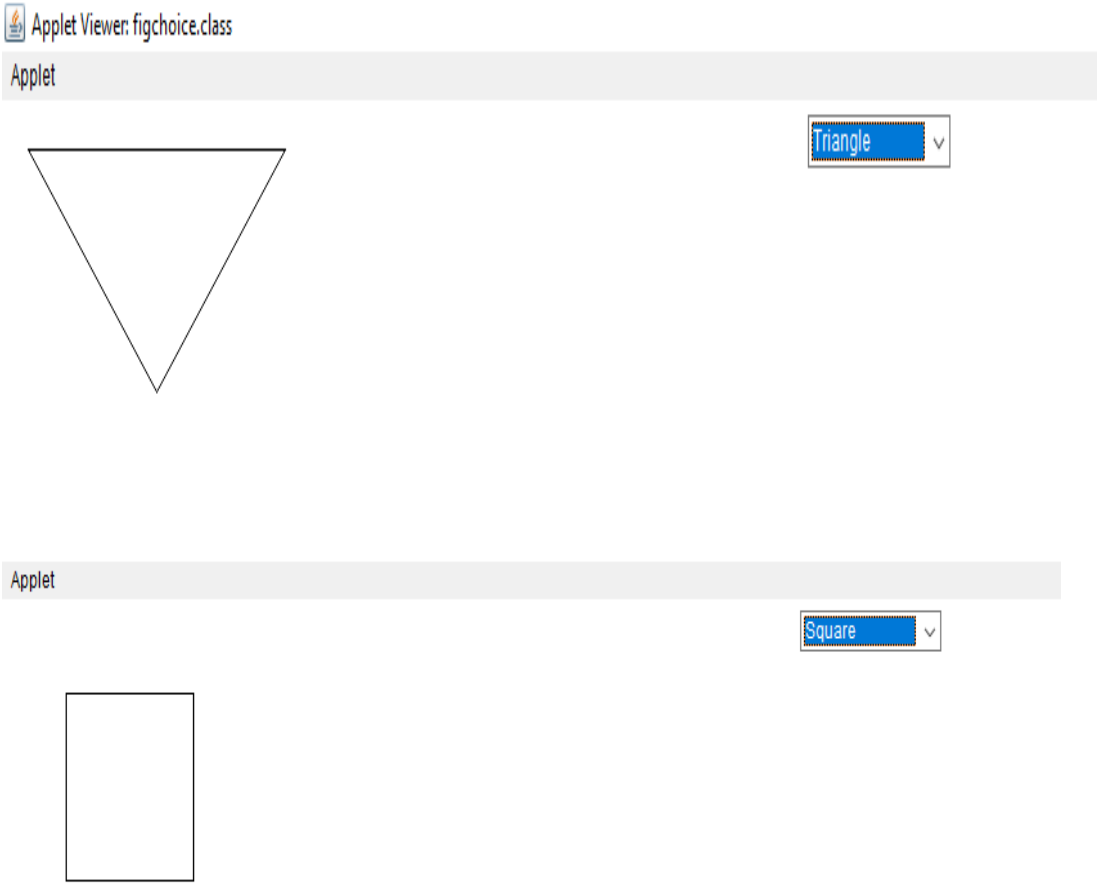
27. Develop a program that has a Choice component which contains the names of shapes such as rectangle, triangle, square and circle. Draw the corresponding shapes for given parameters as per user's choice.

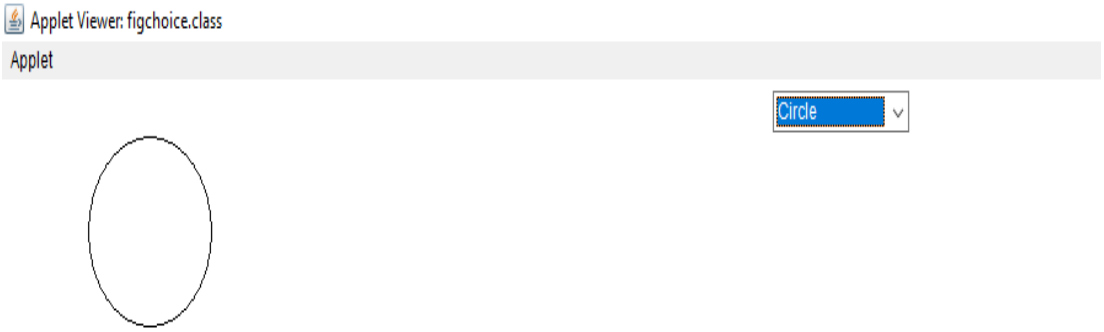
```
import java.applet.*;
import java.awt.*;
import java.awt.Graphics;
import java.awt.event.*;
public class figchoice extends Applet implements ItemListener {
 Choice ch;
 int x1[]= {50,120,220,20};
int y1[]= {50,120,20,20};
int n=4;
 int Selection;
 public void init()
 {
 ch = new Choice();
 ch.addItem("Select a Shape");
 ch.addItem("Rectangle");
 ch.addItem("Triangle");
 ch.addItem("Square");
 ch.addItem("Circle");
 add(ch);
 ch.addItemListener(this);
```

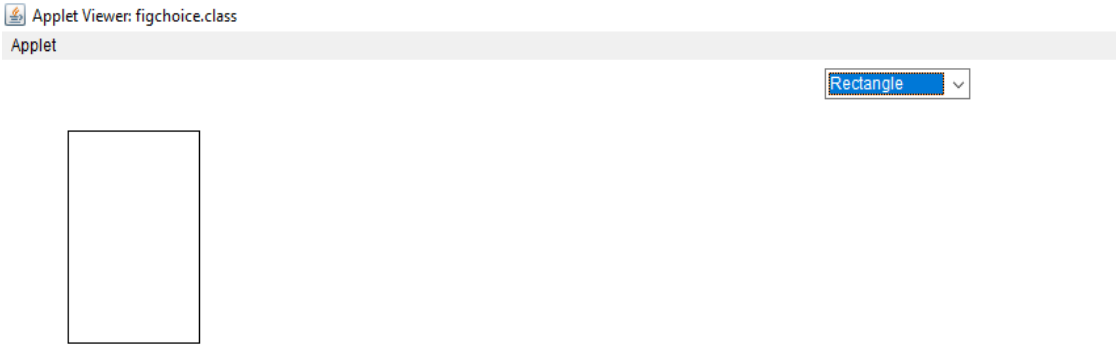
```
 }
public void itemStateChanged (ItemEvent e)
 {
 Selection = ch.getSelectedIndex();
 repaint();
 }
public void paint(Graphics g)
{
 super.paint(g);
 if (Selection == 1)
 {    g.drawRect(50,50,100,150);     }
 if (Selection == 2)
 {    g.drawPolygon(x1,y1,n);    }
 if (Selection == 3)
 {    g.drawRect(50,50,100,100);          }
 if (Selection == 4)
 {
 g.drawOval(70,30,100,100);
 } } }
<html><head>
</head>
<body>
```

```
<div align="center">
<applet code="figchoice.class"width="800"height="500">
</applet>
</div>
</body>
</html>
```

OUTPUT







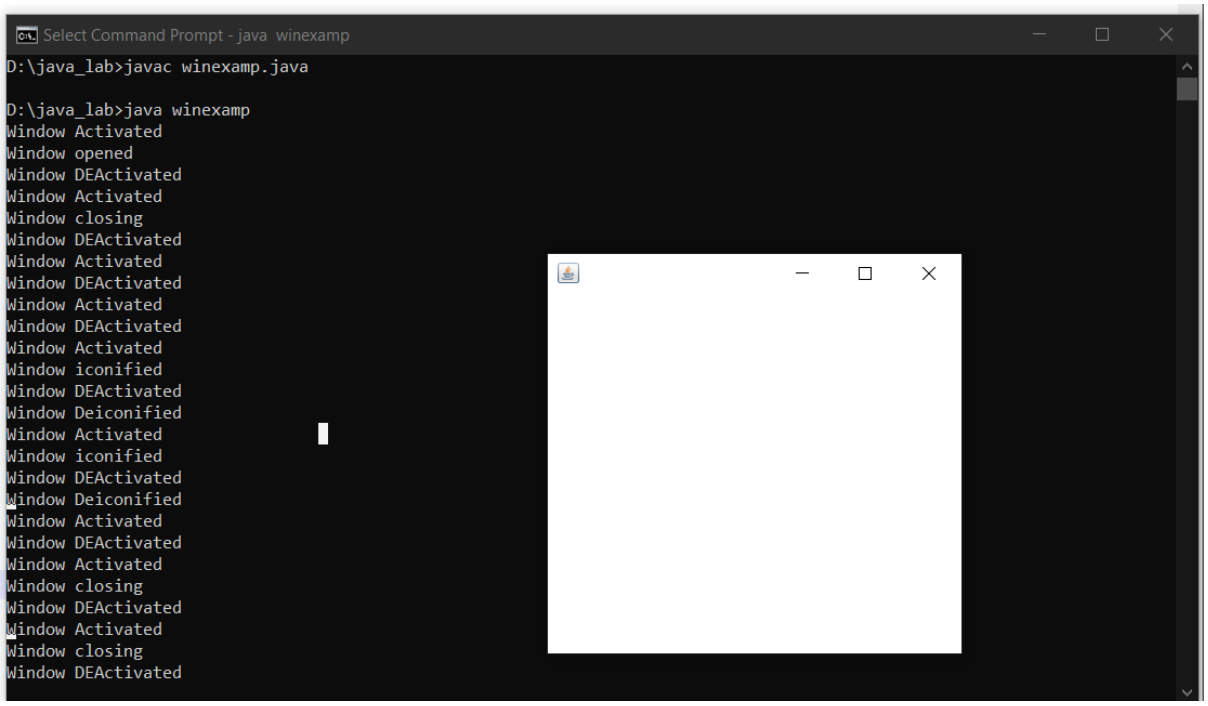
RESULT : The program has been executed and the output was verified.

28. Develop a program to handle all window events

```
import java.awt.*;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;
public class winexamp extends Frame implements WindowListener
{
winexamp()
{
addWindowListener(this);
setSize(400,400);
setLayout(null);
setVisible(true);
}
public static void main(String args[])
{
new winexamp();
}
public void windowActivated(WindowEvent arg0)
{
System.out.println("Window Activated");
}
public void windowClosed(WindowEvent args0)
{
System.out.println("Window closed");
}
public void windowClosing(WindowEvent arg0)
{
System.out.println("Window closing");
}
public void windowDeactivated(WindowEvent arg0)
{
System.out.println("Window DEActivated");
```

```
}
public void windowDeiconified(WindowEvent arg0)
{
System.out.println("Window Deiconified");
}
public void windowIconified(WindowEvent arg0)
{
System.out.println("Window iconified");
}
public void windowOpened(WindowEvent arg0)
{
System.out.println("Window opened");
}
}
```

OUTPUT



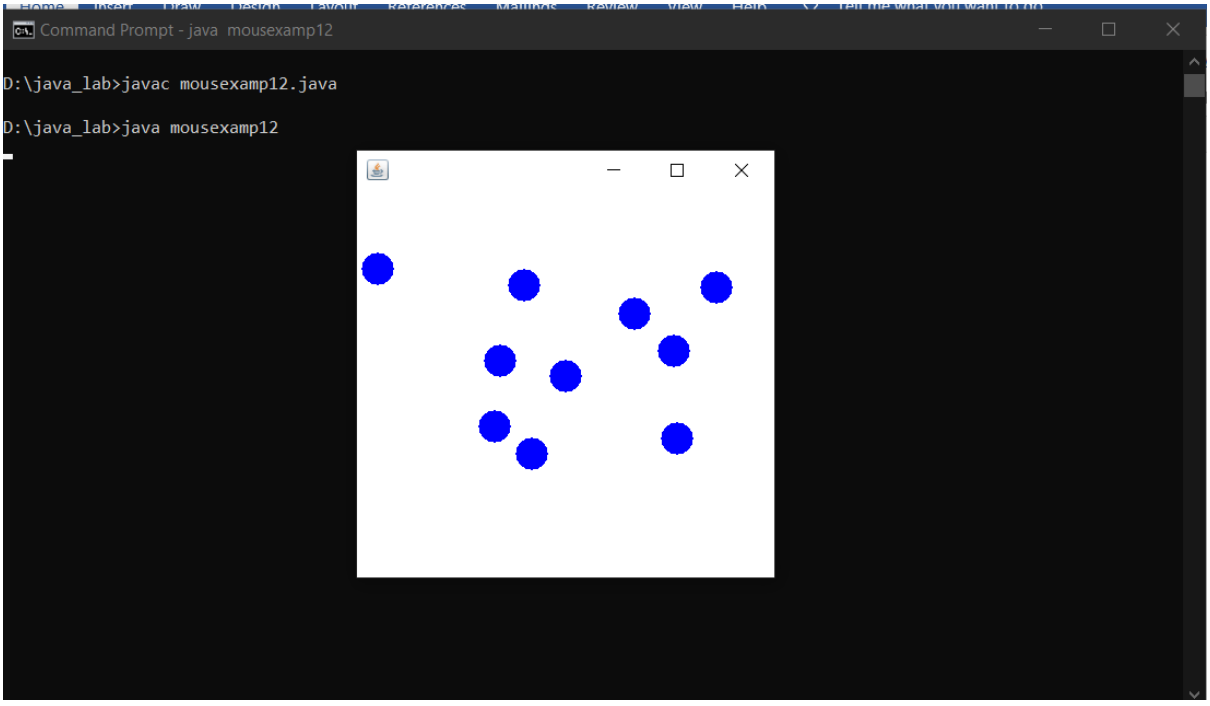
RESULT : The program has been executed and the output was verified.

29.Develop a program to handle all mouse events

```
import java.awt.*;
import java.awt.event.*;
public class mousexamp12 extends Frame implements MouseListener
{
        mousexamp12()
        {
        addMouseListener(this);
        setSize(400,400);
        setLayout(null);
        setVisible(true);
        }
        public void mouseClicked(MouseEvent e)
        {
        Graphics g=getGraphics();
        g.setColor(Color.blue);
        g.fillOval(e.getX(),e.getY(),30,30);
        }
public void mouseEntered(MouseEvent e)
{
}
public void mouseExited(MouseEvent e)
{
```

```
}
public void mousePressed(MouseEvent e)
{
}
public void mouseReleased(MouseEvent e){
}
public static void main(String args[])
{
new mousexamp12();
}
}
```

OUTPUT



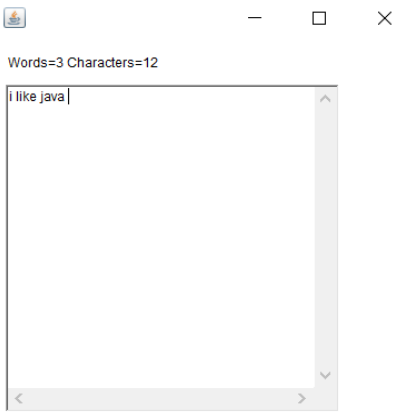
RESULT : The program has been executed and the output was verified.

30.Develop a program to handle Key events.

```
import java.awt.*;
import java.awt.event.*;
public class keyexamp extends Frame implements KeyListener
{
Label l;
TextArea a;
keyexamp()
{
l=new Label();
l.setBounds(20,50,200,20);
a=new TextArea();
a.setBounds(20,80,300,300);
a.addKeyListener(this);
add(l);
add(a);
setSize(400,400);
setLayout(null);
setVisible(true);
}
public void keyPressed(KeyEvent e)
{
}
public void keyReleased(KeyEvent e)
```

```
{
String t=a.getText();
String w[]=t.split("\\s");
l.setText("Words="+w.length+" Characters="+t.length());
}
public void keyTyped(KeyEvent e)
{}
public static void main(String args[])
{
new keyexamp();
}
}
```

OUTPUT



RESULT : The program has been executed and the output was verified.

31.Producer/Consumer using ITC

```
import java.util.*;
class Q
{
int n;
boolean statusFlag=false;
synchronized void put(int n)
{
try
{
while(statusFlag)
{
wait();
}
}
catch(InterruptedException e){}
this.n=n;
System.out.println("Put :"+n);
statusFlag=true;
notify();
}
synchronized int get()
```

```
{
try{
while(!statusFlag)
{
wait();
}
}
catch(InterruptedException e){}
statusFlag=false;
System.out.println("Got :"+n);
notify();
return n;
}
}
class Producer implements Runnable
{
Q q;
Producer(Q q)
{
this.q=q;
new Thread(this, "Producer").start();
}
```

```
public void run()
{
int i=0;
while(true)
{
q.put(i++);
}
}
}
class Consumer implements Runnable
{
Q q;
Consumer(Q q)
{
this.q=q;
new Thread(this,"Consumer").start();
}
public void run()
{
while(true)
{
q.get();
```

```
}
}
}
public class D
{
public static void main(String[] args)
{
Q q=new Q();
Producer p=new Producer(q);
Consumer c=new Consumer(q);
}
}
```

OUTPUT

```
Got: 699
Put: 700
Got: 700
Put: 701
Got: 701
Put: 702
Got: 702
Put: 703
Got: 703
Put: 704
Got: 704
Put: 705
Got: 705
Put: 706
Got: 706
Put: 707
Got: 707
Put: 708
Got: 708
Put: 709
Got: 709
Put: 710
Got: 710
Put: 711
Got: 711
Put: 712
Got: 712
Put: 713
Got: 713
Put: 714
Got: 714
Put: 715
Got: 715
Put: 716
Got: 716
Put: 717
Got: 717
Put: 718
Got: 718
Put: 719
```

RESULT : The program has been executed and the output was verified.

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

```
public class StackAsLinkedList {

    StackNode root;

    static class StackNode {
        int data;
        StackNode next;

        StackNode(int data) { this.data = data; }
    }

    public boolean isEmpty()
    {
        if (root == null) {
            return true;
        }
        else
            return false;
    }
```

```
public void push(int data)
{
    StackNode newNode = new StackNode(data);

    if (root == null) {
        root = newNode;
    }
    else {
        StackNode temp = root;
        root = newNode;
        newNode.next = temp;
    }
    System.out.println(data + " pushed to stack");
}

public int pop()
{
    int popped = Integer.MIN_VALUE;
    if (root == null) {
        System.out.println("Stack is Empty");
    }
    else {
```

```
        popped = root.data;
        root = root.next;
    }
    return popped;
}

public int peek()
{
    if (root == null) {
        System.out.println("Stack is empty");
        return Integer.MIN_VALUE;
    }
    else {
        return root.data;
    }
}

// Driver code
public static void main(String[] args)
{

    StackAsLinkedList sll = new StackAsLinkedList();
```

```
        sll.push(10);
        sll.push(20);
        sll.push(30);

        System.out.println(sll.pop()
                           + " popped from stack");

        System.out.println("Top element is " + sll.peek());
    }
}
```

OUTPUT

```
D:\MCA NOTES\Sem 2\Object Oriented Programming Lab\20-09-2021(Sr. Elsin)>java StackAsLinkedList
10 pushed to stack
20 pushed to stack
30 pushed to stack
30 popped from stack
Top element is 20
```

RESULT : The program has been executed and the output was verified.

33.Using generic method perform Bubble sort.

```
public class BubbleSort {
  static void bubbleSort(int[] arr) {
    int n = arr.length;
    int temp = 0;

    for(int i = 0; i < n; i++) {
      for(int j=1; j < (n-i); j++) {
        if(arr[j-1] > arr[j]) {
          temp = arr[j-1];
          arr[j-1] = arr[j];
          arr[j] = temp;
        }
      }
    }
  }
  public static void main(String[] args) {
    int arr[] = { 1, 6, -2, 6, -4, 8, 5, -7, -9, 4 };
    System.out.println("Array Before Bubble Sort");

    for(int i = 0; i < arr.length; i++) {
      System.out.print(arr[i] + " ");
```

```
    }
    System.out.println();
    bubbleSort(arr);
    System.out.println("Array After Bubble Sort");

    for(int i = 0; i < arr.length; i++) {
      System.out.print(arr[i] + " ");
    }
  }
}
```

OUTPUT

```
D:\MCA NOTES\Sem 2\Object Oriented Programming Lab\20-09-2021(Sr. Elsin)>javac BubbleSort.java

D:\MCA NOTES\Sem 2\Object Oriented Programming Lab\20-09-2021(Sr. Elsin)>java BubbleSort
Sorted array
11 12 22 25 34 64 90
```

RESULT : The program has been executed and the output was verified.

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

```
import java.util.*;
class PriorityQueue1{
public static void main(String args[]){
PriorityQueue<String> queue=new PriorityQueue<String>();
queue.add("Amit");
queue.add("Vijay");
queue.add("Karan");
queue.add("Jai");
queue.add("Rahul");
System.out.println("head:"+queue.element());
System.out.println("head:"+queue.peek());
System.out.println("iterating the queue elements:");
Iterator itr=queue.iterator();
while(itr.hasNext()){
System.out.println(itr.next());
}
queue.remove();
queue.poll();
System.out.println("after removing two elements:");
Iterator<String> itr2=queue.iterator();
```

```
while(itr2.hasNext()){
System.out.println(itr2.next());
}
}
}
```

OUTPUT

```
D:\MCA NOTES\Sem 2\Object Oriented Programming Lab\20-09-2021(Sr. Elsin)>java PriorityQueue1
head:Amit
head:Amit
iterating the queue elements:
Amit
Jai
Karan
Vijay
Rahul
after removing two elements:
Karan
Rahul
Vijay
```

RESULT : The program has been executed and the output was verified.

35.Program to remove all the elements from a linked list

```
import java.util.*;
 public class removelink
 {
 public static void main(String[] args)
{
   // create an empty linked list
    LinkedList<String> l_list = new LinkedList<String>();
  // use add() method to add values in the linked list
        l_list.add("violet");
        l_list.add("Green");
        l_list.add("Black");
        l_list.add("Pink");
        l_list.add("blue");
    // print the list
  System.out.println("The Original linked list: " + l_list);
 // Removing all the elements from the linked list
   l_list.clear();
   System.out.println("The New linked list: " + l_list);
 }}
```

OUTPUT

```
D:\MCA NOTES\Sem 2\Object Oriented Programming Lab\20-09-2021(Sr. Elsin)>javac removelink.java

D:\MCA NOTES\Sem 2\Object Oriented Programming Lab\20-09-2021(Sr. Elsin)>java removelink
The Original linked list: [violet, Green, Black, Pink, blue]
The New linked list: []
```

RESULT : The program has been executed and the output was verified.

36.program to demonstrate the addition and deletion of elements in dequeue

```
import java.util.*;
 public class deque
{
public static void main(String[] args)
 {
 Deque<String> deque = new LinkedList<String>();
 // We can add elements to the queue
// in various ways
// Add at the last
deque.add("Element 1 (Tail)");
// Add at the first
deque.addFirst("Element 2 (Head)");
// Add at the last
 deque.addLast("Element 3 (Tail)");
// Add at the first
deque.push("Element 4 (Head)");
 // Add at the last
deque.offer("Element 5 (Tail)");
// Add at the first
 deque.offerFirst("Element 6 (Head)");
```

```
System.out.println(deque + "\n");
// We can remove the first element
// or the lastelement.
deque.removeFirst();
deque.removeLast();
System.out.println("Deque after removing " + "first and last: " + deque);
}
}
```

OUTPUT

```
D:\java_lab>javac deque.java

D:\java_lab>java deque
[Element 6 (Head), Element 4 (Head), Element 2 (Head), Element 1 (Tail), Element 3 (Tail), Element 5 (Tail)]
```

RESULT : The program has been executed and the output was verified.

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

```
import java.util.*;

class arrayjava{

public static void main(String args[]){

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

alist.add("appu");

alist.add("ammu");

alist.add("minnu");      alist.add("thomu");

alist.add("pinky");    alist.add("Tom");

//displaying elements

 System.out.println(alist);

//Adding "appu" at the fourth position alist.add(3, "appu");

 //displaying elements

 System.out.println(alist);

}    }
```

OUTPUT

```
D:\MCA NOTES\Sem 2\Object Oriented Programming Lab\20-09-2021(Sr. Elsin)>javac arrayjava.java

D:\MCA NOTES\Sem 2\Object Oriented Programming Lab\20-09-2021(Sr. Elsin)>java arrayjava
[appu, ammu, minnu, thomu, pinky, Tom]
[appu, ammu, minnu, thomu, pinky, Tom]
```

RESULT : The program has been executed and the output was verified.

38.Program to demonstrate the working of map interface by adding ,removing,changing.

```
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));
       // Traversing through the map
       for (Map.Entry<String, Integer> me : hm.entrySet()) {
           System.out.print(me.getKey() + ":");
           System.out.println(me.getValue());       }
   }     }
```

OUTPUT

```
D:\java_lab>javac hashmap.java

D:\java_lab>java hashmap
Jinu : 3
Anu  : 1
sinu : 2
```

RESULT : The program has been executed and the output was verified.

39.program to convert hash map to tree map.

```
import java.util.*;
import java.util.stream.*;
public class HT
{
  public static void main(String args[])
 {
    Map<String, String> map = new HashMap<>();
    map.put("1", "One");
    map.put("2", "Two");
    map.put("3", "Three");
    map.put("4", "Four");
    map.put("5", "Five");
    map.put("6", "Six");
    map.put("7", "Seven");
    map.put("8", "Eight");
    map.put("9", "Nine");
    System.out.println("HashMap = " + map);
    Map<String, String> treeMap = new TreeMap<>();
    treeMap.putAll(map);
    System.out.println("TreeMap (HashMap to TreeMap) " + treeMap);
  }
```

```
}
```

OUTPUT

```
D:\Java\14-08-2021(Sister Elsin)>java HT
HashMap = {1=One, 2=Two, 3=Three, 4=Four, 5=Five, 6=Six, 7=Seven, 8=Eight, 9=Nine}
TreeMap (HashMap to TreeMap) {1=One, 2=Two, 3=Three, 4=Four, 5=Five, 6=Six, 7=Seven, 8=Eight, 9=Nine}
```

RESULT : The program has been executed and the output was verified.

40.Program to list the sub directories and files in a given directory and also search for a file name.

```
import java.io.File;
import java.util.*;
import java.io.*;
public class p1 {
 public static final String RED="\033[0;31m";
 public static final String RESET="\033[0m";
 static void RecursivePrint(File[] arr, int index, int level, String search
for) {
 // exit condition
 if (index == arr.length)
 return;
 // space for internbal level
 for (int i = 0; i < level; i++)
 System.out.print("\t");
 if(arr[index].getName().toLowerCase().contains(searchfor))
 System.out.print(RED);
 else
 System.out.print(RESET);
 // for files
 if (arr[index].isFile())
 System.out.println(arr[index].getName());
 else if (arr[index].isDirectory()) {
 System.out.println("[" + arr[index].getName() + "]");
 RecursivePrint(arr[index].listFiles(), 0, level + 1, searchfor);
 }
 RecursivePrint(arr, ++index, level, searchfor);
 }
 public static void main(String[] args) {
 Scanner scan = new Scanner(System.in);
```

```
        System.out.println("Enter the directory path");
        String maindirpath = scan.nextLine();
        System.out.println("Enter the file/directory name to search");
        String searchfor = scan.nextLine();
        File maindir = new File(maindirpath);
        if (maindir.exists() && maindir.isDirectory()) {
        File arr[] = maindir.listFiles();
        System.out.println("################################################");
        System.out.println("Files from main directory" + maindir);
        System.out.println("################################################");
       RecursivePrint(arr, 0, 0, searchfor.toLowerCase()); // array,index
       ,level,search
        }
        }
}
```

OUTPUT

```
D:\java_lab>javac p1.java

D:\java_lab>java p1
Enter the directory path
D:\java_lab\neww
Enter the file/directory name to search
ArrayListex.java
################################################
Files from main directoryD:\java_lab\neww
################################################
←[0mArrayListex.class
←[0;31mArrayListex.java

D:\java_lab>
```

RESULT : The program has been executed and the output was verified.

41. 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 exit to stop"
);
 while (!var.endsWith("exit\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) {
 // Storing every character in the string
 str += (char) i;
 }
 System.out.println(str);
 fr.close();
```

```
        } catch (IOException e) {
        System.out.println("There are some exception");
        }
        }
}
```

Output

```
D:\java_lab>javac read.java

D:\java_lab>java read
Enter the text to create file : type exit to stop
hai friends
exit
Reading File content
hai friends
exit


D:\java_lab>
```

RESULT : The program has been executed and the output was verified.