

# VEMU INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

## LAB MANUAL



19A05303P OBJECT ORIENTED PROGRAMMING THROUGH JAVA LAB

**Regulation R19**

**Year / Semester: II/I**

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech II-I Sem

L T P C  
0 0 3 1.5

19A05303P OBJECT ORIENTED PROGRAMMING THROUGH JAVA LAB

(Common to CSE & IT)

**Course Objectives**

- To introduce the concepts of Java.
  - To Practice object-oriented programs and build java applications.
  - To implement java programs for establishing interfaces.
  - To implement sample programs for developing reusable software components.
  - To establish database connectivity in java and implement GUIapplications.
1. Installation of Java software, study of any Integrated development environment, Use Eclipse or Netbean platform and acquaint with the various menus. Create a test project, add a test class and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods and classes. Try debug step by step with java program to find prime numbers between 1 ton.

**AIM:** Try debug step by step with java program to find prime numbers between 1 ton.

SOURCE CODE:

```
import java.util.Scanner;
class PrimeNumbers2
{
    public static void main (String[] args)
    {
        Scanner scanner = new Scanner(System. in);
        int i =0;
        int num =0;
        //Empty String
        String primeNumbers = "";
        System.out.println("Enter the value of n:");
        int n = scanner.nextInt();
        scanner.close();
        for (i = 1; i <= n; i++)
        {
            int counter=0;
```

```
for(num =i; num>=1; num--)  
{  
    if(i%num==0)  
    {  
        counter = counter + 1;  
    }  
}  
if (counter ==2)  
{  
    //Appended the Prime number to the String  
    primeNumbers = primeNumbers + i + " ";  
}  
}  
System.out.println("Prime numbers from 1 to n are :");  
System.out.println(primeNumbers);  
}  
}
```

### **OUTPUT:**

**Enter the value of n:**

**150**

**Prime numbers from 1 to n are :**

**2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89**

**97 101 103 107 109 113 127 131 137 139 149**

2). Write a Java program that prints all real solutions to the quadratic equation  $ax^2+bx+c=0$ . Read in a, b, c and use the quadratic formula.

**AIM:** Java program that prints all real solutions to the quadratic equation  $ax^2+bx+c=0$ . Read in a, b, c and use the quadratic formula.

### SOURCE CODE:

```
//Program to find roots of a quadratic equations
import java.util.*;
class Roots
{
    public static void main(String args[])
    {
        int a,b,c,d,f=0;
        Scanner scr=new Scanner(System.in);
        System.out.println("\nEnter the values of a ,b ,c : ");
        a=scr.nextInt();
        b=scr.nextInt();
        c=scr.nextInt();
        d=(b*b)-(4*a*c);
        if(d==0)
        {
            System.out.println("Roots are real and Equal");
            f=1;
        }
        else if(d>0)
        {
            System.out.println("Roots are real and UnEqual");
            f=1;
        }
        else
        System.out.println("Roots are imaginary");
        if(f==1)
        {
            float r1=(float)(-b+Math.sqrt(d))/(2*a);
            float r2=(float)(-b-Math.sqrt(d))/(2*a);
            System.out.println("Roots are : "+r1+" ,"+r2);
        }
    }
}
```

### OUTPUT:

```
Enter the values of a ,b ,c :
1
2
-3
Roots are real and UnEqual

Roots are : 1.0 ,-3.0
```

3). Develop a Java application to generate Electricity bill. Create a class with the following members: Consumer no., consumer name, previous month reading, current month reading, type of EBconnection

(i.e domestic or commercial). Commute the bill amount using the following tariff.

If the type of the EB connection is domestic, calculate the amount to be paid as follows:

- First 100 units - Rs. 1 perunit
- 101-200 units - Rs. 2.50 perunit
- 201 -500 units - Rs. 4 perunit
- >501units - Rs. 6 perunit

If the type of the EB connection is commercial, calculate the amount to be paid as follows:

- First 100 units - Rs. 2 perunit
- 101-200 units - Rs. 4.50 perunit
- 201 -500 units - Rs. 6 perunit
- >501units - Rs. 7 perunit

**AIM:** Develop a Java application to generate Electricity bill. Create a class with the following members: Consumer no., consumer name, previous month reading, current month reading, type of EBconnection

#### **SOURCE CODE:**

```
import java.util.Scanner;
class ElectBill
{
int ConsumerNo;
String ConsumerName;
int PrevReading;
int CurrReading;
String EBConn;
double Bill;
void input_data()
{
Scanner sc = new Scanner(System.in);
System.out.println("\n Enter Consumer Number: ");
ConsumerNo = sc.nextInt();
VEMUI
```

```

System.out.println("\n Enter Consumer Name: ");
ConsumerName = sc.next();
System.out.println("\n Enter Previous Units: ");
PrevReading = sc.nextInt();
System.out.println("Enter Current Units consumed:");
CurrReading = sc.nextInt();
System.out.println("Enter the types of EB Connection(domestic or commercial)");
EBConn = sc.next();
}
double calculate_bill()
{
int choice;
if(EBConn=="domenstic")
Object Oriented Programming L - 3 Object Oriented Programming Laboratory
choice=1;
else
choice=2;
switch(choice)
{
case 1:
if(CurrReading>=0 && CurrReading<=100)
Bill=CurrReading*1;
else if(CurrReading>100 && CurrReading <= 200)
Bill=(100*1)+((CurrReading-100)*2.50);
else if(CurrReading>200 && CurrReading <= 500)
Bill=(100*1)+(100*2.50)+((CurrReading-200)*4);
else
Bill=(100*1)+(100*2.50)+(300*4)+((CurrReading-500)*6);
break;
case 2:
if(CurrReading>=0 && CurrReading<=100)
Bill=CurrReading*2;

```

```

else if(CurrReading>100 && CurrReading <= 200)
Bill=(100*1)+((CurrReading-100)*4.50);
else if(CurrReading>200 && CurrReading <= 500)
Bill=(100*1)+(100*2.50)+((CurrReading-200)*6);
else
Bill=(100*1)+(100*2.50)+(300*4)+((CurrReading-500)*7);
break;
}
return Bill;
}
void display()
{
System.out.println("-----");
System.out.println("ELCTRICITY BILL");
System.out.println("-----");
System.out.println("Consumer Number: "+ConsumerNo);
System.out.println("Consumer Name: "+ConsumerName);
System.out.println("Consumer Previous Units: "+PrevReading);
System.out.println("Consumer Current Units: "+CurrReading);
System.out.println("Type of EBConnection: "+EBConn);
System.out.println("-----");
System.out.println("Total Amount(Rs.): "+Bill);
}
}
class ElectBillGen
{
public static void main (String[] args)
{
ElectBill b=new ElectBill();
b.input_data();
b.calculate_bill();
b.display();
}
}

```

}  
}

**OUTPUT:**

**Enter Consumer Number:**

**101**

**Enter Consumer Name:**

**AAA**

**Enter Previous Units:**

**310**

**Enter Current Units consumed:**

**480**

**Enter the types of EB Connection(domestic or commercial)**

**domestic**

-----  
**ELCTRICITY BILL**

-----  
**Consumer Number: 101**

**Consumer Name: AAA**

**Consumer Previous Units: 310**

**Consumer Current Units: 480**

**Type of EBConnection: domestic**

-----  
**Total Amount(Rs.): 2030.0**



4). Write a Java program to multiply two given matrices.

**AIM:** a Java program to multiply two given matrices.

**SOURCE CODE:**

```
public class MatrixMultiplicationExample{
public static void main(String args[]){
//creating two matrices
int a[][]={{1,1,1},{2,2,2},{3,3,3}};
int b[][]={{1,1,1},{2,2,2},{3,3,3}};

//creating another matrix to store the multiplication of two matrices
int c[][]=new int[3][3]; //3 rows and 3 columns

//multiplying and printing multiplication of 2 matrices
for(int i=0;i<3;i++){
for(int j=0;j<3;j++){
c[i][j]=0;
for(int k=0;k<3;k++)
{
c[i][j]+=a[i][k]*b[k][j];
} //end of k loop
System.out.print(c[i][j]+" "); //printing matrix element
} //end of j loop
System.out.println();//new line
}
}}
```

**OUTPUT:**

```
6 6 6
12 12 12
18 18 18
```

5). Write Java program on use of inheritance, preventing inheritance using final, abstract classes.

**AIM:** Java program on use of inheritance, preventing inheritance using final, abstract classes

**SOURCE CODE:**

**Use of Inheritance:**

```
class Employee{
    float salary=40000;
}
class Programmer extends Employee{
    int bonus=10000;
    public static void main(String args[]){
        Programmer p=new Programmer();
        System.out.println("Programmer salary is:"+p.salary);
        System.out.println("Bonus of Programmer is:"+p.bonus);
    }
}
```

**OUTPUT:** Programmer salary is: 40000.0  
Bonus of programmer is: 10000

**Preventing inheritance:**

```
final class Bike{}

class Honda1 extends Bike{
    void run(){
        System.out.println("running safely with 100kmph");}

    public static void main(String args[]){
        Honda1 honda= new Honda1();
        honda.run();
    }
}
VEMUI
```

```
}
```

**Output: running safely with 100kmph**

**Abstract Classes:**

```
abstract class Shape{
abstract void draw();
}
class Rectangle extends Shape{
void draw(){
System.out.println("drawing rectangle");}
}
class Circle1 extends Shape{
void draw(){
System.out.println("drawing circle");}
}
class TestAbstraction1{
public static void main(String args[]){
Shape s=new Circle1();//In a real scenario, object is provided through method, e.g., getShape()
method
s.draw();
}
```

**Output: drawing circle**

6. Write Java program on dynamic binding, differentiating method overloading and overriding.

**AIM:** Java program on dynamic binding, differentiating method overloading and overriding

**SOURCE CODE:**

**Dynamic Binding:**

```
class A{  
  
voidsamp(){System.out.println("hai...");}  
  
}  
  
class D extends A{  
  
voidsamp(){System.out.println("hello...");}  
  
  
public static void main(String args[]){  
  
    A a=new D();  
  
a.samp();  
  
}  
  
}
```

**Output:**

Hello

Overloading:

```
class Overload
```

```
{
```

```
void demo (int a)
```

```
{
```

```
System.out.println ("a: " + a);
```

```
}
```

```
void demo (int a, int b)
```

```
System.out.println ("a and b: " + a + ", " + b);
```

```
}
```

```
double demo(double a) {
```

```
System.out.println("double a: " + a);
```

```
return a*a;
```

```
}
```

```
}
```

```
class MethodOverloading
```

```
{
```

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

```
{
```

```
Overload Obj = new Overload();
```

```
double result;
```

```
Obj .demo(10);
```

```
Obj .demo(10, 20);
```

```
result = Obj .demo(5.5);
```

```
System.out.println("O/P : " + result);
```

```
}}
```

VEMUI

### **Overriding:**

```
public class BaseClass
{
    public void methodToOverride() //Base class method
    {
        System.out.println ("I'm the method of BaseClass");
    }
}

public class DerivedClass extends BaseClass
{
    public void methodToOverride() //Derived Class method
    {
        System.out.println ("I'm the method of DerivedClass");
    }
}

public class TestMethod
{
    public static void main (String args []) {
        // BaseClass reference
        and object BaseClass obj1 =
        new BaseClass();

        // BaseClass reference but
        DerivedClass object BaseClass obj2 =
```

```

new DerivedClass();

    // Calls the method from BaseClass class
obj1.methodToOverride();

    //Calls the method from DerivedClass class
obj2.methodToOverride();

    }

}

```

Output:

I'm the method of BaseClass

I'm the method of DerivedClass

**7).** Develop a java application to implement currency converter (Dollar to INR, EURO to INR, Yen) using Interfaces.

**AIM:** a java application to implement currency converter (Dollar to INR, EURO to INR, Yen) using Interfaces.

### **SOURCE CODE:**

**CurrencyConverter.java** package

ConversionDemo; public class

CurrencyConverter

```

{
    double ER = 0;
    public CurrencyConverter(double CurrentExchange)
    {
        ER = CurrentExchange;
    }
    public double DollarToINR(double Dollars)
    {
        double INR = 0;
        INR = Dollars *
        ER; return INR;
    }
}

```

```

}
public double INRToDollar(double INR)
{
    double Dollars =
    0; Dollars = INR
    / ER; return
    Dollars;
}
public double EuroToINR(double Euros)
{
    double INR = 0;
    INR = Euros *
    ER; return INR;
}
public double INRToEuro(double INR)
{
    double Euros =
    0; Euros = INR
    / ER; return
    Euros;
}
public double YenToINR(double Yens)
{
    double INR = 0;
    INR = Yens *
    ER; return INR;
}
public double INRToYen(double INR)
{
    double Yens =
    0; Yens = INR
    / ER; return
    Yens;
}

```



```
}
```

**DistanceConverter.java** package

ConversionDemo; public class

DistanceConverter

```
{  
    public double MeterToKM(double Meters)  
    {  
        double KM = 0;  
        KM = Meters /  
        1000; return KM;  
    }  
    public double KMTToMeter(double KM)  
    {  
        double Meters = 0;  
        Meters = KM *  
        1000; return  
        Meters;  
    }  
    public double MileToKM(double Miles)  
    {  
        double KM = 0;  
        KM = Miles / 0.621371;  
        return KM;  
    }  
    public double KMTToMile(double KM)  
    {  
        double Miles = 0;  
        Miles = KM * 0.621371;  
        return Miles;  
    }  
}
```

**TimeConverter.java** package

ConversionDemo; public

class TimeConverter

```
{  
    public double HrToMin(double Hours)  
    {  
        double Minutes = 0;  
        Minutes = Hours *  
        60; return Minutes;  
    }  
}
```

```
}  
public double MinToHour(double Minutes)  
{  
    double Hours = 0;  
    Hours = Minutes /  
    60;  
}
```

```

        return Hours;
    }
    public double HrToSec(double Hours)
    {
        double Seconds = 0;
        Seconds = Hours *
        3600; return Seconds;
    }
    public double SecToHour(double Seconds)
    {
        double Hours = 0;
        Hours = Seconds /
        3600; return Hours;
    }
}

```

**Step 2 :**

Now go to the parent directory of **ConversionDemo** directory and store following Java program under this parent directory

**Converter.java**

```

import
ConversionDemo.CurrencyConverter;
import
ConversionDemo.DistanceConverter;
import
ConversionDemo.TimeConverter;
import java.util.Scanner;
class Converter
{
    public static void main(String[] args) throws NoClassDefFoundError
    {
        double CurrentExchange;
        int
        choice,choice1,choice2,choice3;
        double inr;
        double km;
        double hr;
        char
        ans='y'; do
        {

```

```
System.out.println("\n Main Menu");
System.out.println("1.Currency Converter \n2.Distance
Converter \n3. Time Converter");
System.out.println("Enter your
choice: "); Scanner input = new
Scanner(System.in); choice =
input.nextInt(); switch(choice)//outer
Switch
{
    case 1: System.out.println("\tCurrency Conversion");
```

```

do
{
    System.out.println("Menu For Currency
    Conversion"); System.out.println("1. Dollar to
    INR"); System.out.println("2. INR to Dollar");
    System.out.println("3. Euro to INR");
    System.out.println("4. INR to Euro");
    System.out.println("5. Yen to INR");
    System.out.println("6. INR to Yen");

    System.out.println("Enter your
    choice: "); choice1 = input.nextInt();
    System.out.println("Please enter the current exchange
    rate: "); CurrentExchange = input.nextDouble();
    CurrencyConverter cc=new
    CurrencyConverter(CurrentExchange);
    switch(choice1)//inner switch
    {
        case 1:
            System.out.print("Enter Dollars
            :"); double
            dollar=input.nextDouble();
            System.out.println (dollar+" dollars are converted to
            "+cc.DollarToINR(dollar)+" Rs.");
            break;
        case 2:
            System.out.print("Enter INR
            :"); inr=input.nextDouble();
            System.out.println(inr+" Rs. are converted to
            "+cc.INRToDollar(inr)+" Dollars");
            break;
        case 3:
            System.out.print("Enter Euro
            :"); double
            euro=input.nextDouble();
            System.out.println(euro+" Euros are converted to
            "+cc.EuroToINR(euro)+" Rs.");
            break;
        case 4:

```

```
System.out.print("Enter INR
:"); inr=input.nextDouble();
System.out.println(inr+" Rs. are converted to
"+cc.INRToEuro(inr)+" Euros");
break;
case 5:
System.out.print("Enter Yen
:"); double
yen=input.nextDouble();
System.out.println(yen+" Yens are converted to
```

```

        "+cc.YenToINR(yen)+" Rs.");
    break;
    case 6:
        System.out.print("Enter INR
        :"); inr=input.nextDouble();
        System.out.println(inr+" Rs. are converted to
        "+cc.INRToYen(inr)+" Yens");
    break;

}
System.out.println("Do You want to go to Currency Conversion
Menu?(y/n)
"); ans =
input.next().charAt(0);
}while(ans=='
y'); break;
case 2: System.out.println("\tDistance
Conversion"); do
{
    System.out.println("Menu For Distance
Conversion"); System.out.println("1. Meter to
Km"); System.out.println("2. Km to Meter");
System.out.println("3. Miles to Km");
System.out.println("4. Km to Miles");
System.out.println("Enter your choice: ");
choice2 = input.nextInt();
DistanceConverter dc=new
DistanceConverter(); switch(choice2)
{
    case 1:
        System.out.print("Enter meters to convert to
Km:"); double meter=input.nextDouble();
System.out.println(meter+" Meters are converted to
"+dc.MeterToKM(meter)+" Km.");
        break;
    case 2:
        System.out.print("Enter Km to convert to
meters:"); km=input.nextDouble();
System.out.println(km+" Km. are converted meters

```

```
        "+dc.KMToMeter(km)+" Meters");  
break;  
case 3:  
    System.out.print("Enter Miles to convert to  
Km:"); double miles=input.nextDouble();  
    System.out.println(miles+" Miles are converted to  
        "+dc.MileToKM(miles)+" Km.");
```



```

        break;
    case 4:
        System.out.print("Enter Km to convert to
        miles:"); km=input.nextDouble();
        System.out.println(km+" Km. are converted miles
        "+dc.KMToMile(km)+" Miles");
        break;

    }
    System.out.println("Do You want to go to Distance Conversion
    Menu?(y/n)
    "); ans =
    input.next().charAt(0);
}while(ans=='
y'); break;
case 3:System.out.println("\tTime
Conversion"); do
{
    System.out.println("Menu For Time
    Conversion"); System.out.println("1. Hour to
    Minutes"); System.out.println("2. Minutes to
    Hour"); System.out.println("3. Hour to
    Seconds"); System.out.println("4. Seconds to
    Hour"); System.out.println("Enter your
    choice: ");
    choice3 = input.nextInt();
    TimeConverter tc=new
    TimeConverter(); switch(choice3)
    {
        case 1:
            System.out.print("Enter hours to convert to
            Minutes:"); hr=input.nextDouble();
            System.out.println(hr+" Hours are converted to
            "+tc.HrToMin(hr)+" min.");
            break;
        case 2:
            System.out.print("Enter Minutes to convert to
            hours:"); double minutes=input.nextDouble();
            System.out.println(minutes+" Minutes. are converted hours

```

```
        "+tc.MinToHour(minutes)+" Hours");  
break;  
case 3:  
    System.out.print("Enter Hours to convert to  
Seconds:"); hr=input.nextDouble();  
    System.out.println(hr+" Hours are converted to  
        "+tc.HrToSec(hr)+" Seconds.");  
break;
```

```

        case 4:
            System.out.print("Enter Seconds to convert to hours:");
            double seconds=input.nextDouble();
            System.out.println(seconds+" Seconds. are converted hours
                                "+tc.SecToHour(seconds)+" Hours");
            break;

    }
    System.out.println("Do You want to go to Time Conversion
    Menu?(y/n) "); ans = input.next().charAt(0);
}while(ans=='
y'); break;
} //end of outer switch
System.out.println("Do you want to go back to Main
Menu?(y/n)"); ans=input.next().charAt(0);
}while(ans=='y');
} //end of main
} //end of class

```

**Step 3 :**

Open the Command prompt and issue following commands

**Output**

```

D:\>cd ConversionDemo
D:\ConversionDemo>javac
CurrencyConverter.java
D:\ConversionDemo>javac
DistanceConverter.java
D:\ConversionDemo>javac
TimeConverter.java D:\ConversionDemo>cd..
D:\>javac
Converter.java
D:\>java Converter
Main Menu
1.Currency
Converter
2.Distance
Converter
3.      Time
Converter Enter
your choice: 1

```

## Currency Conversion

### Menu For Currency

#### Conversion

1. Dollar to INR
2. INR to Dollar
3. Euro to INR
4. INR to Euro
5. Yen to INR
6. INR to Yen

Enter your choice:

1

Please enter the current exchange rate:

66.21

Enter Dollars :25

25.0 dollars are converted to 1655.2499999999998 Rs. Do

You want to go to Currency Conversion Menu?(y/n) y

Menu For Currency Conversion

1. Dollar to INR
2. INR to Dollar
3. Euro to INR
4. INR to Euro
5. Yen to INR
6. INR to Yen Enter your

choice:

4

Please enter the current exchange rate:

81.1

Enter INR :5000

5000.0 Rs. are converted to 61.652281134401974 Euros Do

You want to go to Currency Conversion Menu?(y/n)

n

Do you want to go back to Main Menu?(y/n) y

Main Menu 1.Currency

Converter 2.Distance

Converter

3. Time Converter Enter

your choice: 2

Distance Conversion Menu For

Distance Conversion

1. Meter to Km
2. Km to Meter
3. Miles to Km
4. Km to Miles Enter your

choice:

1

Enter meters to convert to Km:4000 4000.0

Meters are converted to 4.0 Km.

Do You want to go to Distance Conversion Menu?(y/n) n

Do you want to go back to Main Menu?(y/n) N

---

8). Write Java program that inputs 5 numbers, each between 10 and 100 inclusive. As each number is read display it only if it's not a duplicate of any number already read display the complete set of unique values input after the user enters each new value.

AIM: Java program that inputs 5 numbers, each between 10 and 100 inclusive. As each number is read display it only if it's not a duplicate of any number already read display the complete set of unique values input after the user enters each new value.

SOURCE CODE:

```
import java.util.Scanner;
public class Duplicate{
    public static void main(String[ ] args){
        int a[]={0,0,0,0,0},t,i,j,s=0,r=0;
        Scanner z=new Scanner(System.in);
        System.out.println("Enter 5 unique values between 10 & 100 ");
        for(j=0;j<5;j++)
        {
            t=z.nextInt();
            if(t>=10&&t<=100)
            {
                for(i=0;i<r;i++)
                {
                    if(a[i]==t)
                    s++;
                }
                if(s>0)
                {
                    System.out.println("Duplicate value found retry");
                    s--;
                    j--;
                    continue;
                }
                else
                {
                    a[j]=t;
                    r++;
                }
            }
            else
            {
                System.out.println("value must be in between 10 & 100");
                j--;
            }
        }
        System.out.print("The five unique values are ");
        for(i=0;i<5;i++)
        {
            System.out.print(a[i]+" ");
        }
    }
}
```

Output:

Enter 5 unique vales between 10 and 100

12

25

36

56

98

Unique values are:12 25 36 98

9). Write a Java Program to create an abstract class named Shape that contains two integers and an empty method named print Area(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the givenshape.

AIM: a Java Program to create an abstract class named Shape that contains two integers and an empty method named print Area(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the givenshape.

Source code:

```
import java.io.*;
// Using abstract methods and classes.
abstract class Figure {
    double dim1;
    double dim2;
    Figure(double a, double b) {
        dim1 = a;
        dim2 = b;
    }
    // area is now an abstract method
    abstract double area();
}
class Rectangle extends Figure {
    Rectangle(double a, double b) {
        super(a, b);
    }
    // override area for rectangle
    double area() {
        System.out.println("Inside Area for Rectangle.");
        return dim1 * dim2;
    }
}
class Triangle extends Figure {
    Triangle(double a, double b) {
        super(a, b);
    }
    // override area for right triangle
    double area() {
        System.out.println("Inside Area for Triangle.");
        return dim1 * dim2 / 2;
    }
}
class AbstractAreas {
    public static void main(String args[]) throws IOException{
```

VEMUI

```

BufferedReader br=new BufferedReader(new InputStreamReader(System. in));

System.out.println("Enter legth  and breadth of Rectangle");

double len=Double.parseDouble(br.readLine());

double bdt=Double.parseDouble(br.readLine());

System.out.println("Enter height and side of Triangle");

double ht=Double.parseDouble(br.readLine());

double sd=Double.parseDouble(br.readLine());

Rectangle r = new Rectangle(len, bdt);

Triangle t = new Triangle(ht, sd);
Figure figref; // this is OK, no object is created
figref = r;
System.out.println("Area is " + figref.area());
figref = t;
System.out.println("Area is " + figref.area());
    }
}

```

Output:

Enter length and breadth of rectangle:10 20

Enter height and side of the triangle:

12 10

Inside area of rectangle : Area is 200

side area of triangle: area is:60



10).Write a Java program to read the time intervals (HH:MM) and to compare system time if the system Time between your time intervals print correct time and exit else try again to repute the same thing. By using String Toknizerclass.

AIM: Java program to read the time intervals (HH:MM) and to compare system time if the system Time between your time intervals print correct time and exit else try again to repute the same thing. By using String Toknizerclass.

```
SOURCE CODE: import java.util.*;
import java.text.*;
class Tokenizer
{
    static int[] cal(String y)
    {
        String a,b,x=":";
        int i[] = {0,0};
        StringTokenizer st=new StringTokenizer(y,x);
        a=(String) st.nextElement();
        b=(String) st.nextElement();
        i[0]=Integer.parseInt(a);
        i[1]=Integer.parseInt(b);
        return i;
    }
}
public class GetCurrentDateTime
{
    public static void main(String[] args) {

        DateFormat dateFormat = new SimpleDateFormat("HH:mm");
        Calendar cal = Calendar.getInstance();
        String y=dateFormat.format(cal.getTime());

        while(true)
        {

            String x,t1,a,b;int minute,hour;
            int HH[]={0,0},MM[]={0,0};
            t1=dateFormat.format(cal.getTime());
            int time1[]=Tokenizer.cal(t1);
            hour=time1[0];
            minute=time1[1];
            System.out.println("Enter the time intervals in HH MM fommat");
            Scanner z=new Scanner(System.in);
            String t2=z.next();
            String t3=z.next();
            int time2[]=Tokenizer.cal(t2);
            HH[0]=time2[0];
            MM[0]=time2[1];
            int time3[]=Tokenizer.cal(t3);
```

```

        HH[1]=time3[0];
        MM[1]=time3[1];
if(HH[0]>HH[1])
{
    int t=HH[0];
    HH[0]=HH[1];
    HH[1]=t;
}
if(HH[0]==HH[1]&&MM[0]>MM[1])
{
    int t=MM[0];
    MM[0]=MM[1];
    MM[1]=t;
}

if((hour>=HH[0]&&hour<HH[1])||(hour==HH[0]&&hour==HH[1])&&(minute>=MM[0]&&minute<=MM[
1]))
    {
        System.out.println("Current time is "+hour+" : "+minute);
        break;
    }
else
{
    System.out.println("Try again");
}
}
}
}

```

11).Write a Java program to implement user defined exception handling.

AIM: Java program to implement user defined Exception handling.

Source code:

**//Example of User defined exception in Java**

```
class MyException extends Exception  
{ String str1;  
MyException(String str2)  
{  
str1=str2;  
  
}
```

```
public String toString(){  
  
return ("Output String = "+str1) ;  
  
}  
  
}
```

```
class CustomException  
{
```

```
public static void main(String args[])  
{  
try  
{  
throw new MyException("Custom");
```

```
// I'm throwing user defined custom exception above
```

```
}
```

```
catch(MyException exp){ System.out.println("Hi this is my catch block") ;  
System.out.println(exp) ;  
}
```

```
}
```

```
}
```

```
//uses the predefined exceptions

class Excp
{
public static void main(String args[])
{
int a,b,c;
try
{
a=0; b=10;
c=b/a;

System.out.println("This line will not be executed");
}

catch(ArithmeticException e)
{
System.out.println("Divided by zero");
}

System.out.println("After exception is handled");
}
}
```

**Output: hi this is my catch block**

**Output string=custom**

12. Write a Java program that creates a user interface to perform integer division. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 and Num2 were not integers, the program would throw a Number Format Exception. If Num2 were zero, the program would throw an Arithmetic Exception Display the exception in a message dialogbox.

AIM: a Java program that creates a user interface to perform integer division. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 and Num2 were not integers, the program would throw a Number Format Exception. If Num2 were zero, the program would throw an Arithmetic Exception Display the exception in a message dialogbox.

```
SOURCE CODE: import java.awt.*;
import javax.swing.*;
import java.applet.Applet;
import java.awt.event.*;
public class Division extends Applet implements ActionListener{
    TextField t1,t2,t3;
    Button b;
    Label L1,L2,L3,L4;
    String s;
    Division e;
    public void init()
    {
        e=this;
        t1=new TextField(10);
        t2=new TextField(10);
        t3=new TextField(10);
        L1=new Label("enter num1");
        L2=new Label("enter num2");
        L3=new Label("Result is");
        L4=new Label("Division of 2 numbers");
        b=new Button("Divide");
        add(L4);
        add(L1);
        add(t1);
        add(L2);
        add(t2);
        add(L3);
        add(t3);
        add(b);
        b.addActionListener(this);
    }
    public void actionPerformed(ActionEvent ae)
    {
        try
        {
            int num1=Integer.parseInt(t1.getText());
            int num2=Integer.parseInt(t2.getText());
            s=""+(num1/num2);
            t3.setText(s);
        }
        catch(ArithmeticException a)
```

```

{
OptionPane.showMessageDialog(null,"Divide by zero");
}
catch(NumberFormatException b)
{
OptionPane.showMessageDialog(null,"NumberFormateException");
}
}
}

```

Output: divide bye zero  
Number format exception

**13. Write a Java program that creates three threads. First thread displays "Good Morning" every one second, the second thread displays "Hello" every two seconds and the third thread displays "Welcome" every three seconds.**

AIM: Java program that creates three threads .First thread displays —GoodMorning|every one second the second thread displays —Hello |every two seconds and the third thread displays—Welcome|every three seconds.

Source Code:

```

class A extends Thread
{
synchronized public void run()
{
try
{
while(true)
{
sleep(1000);
System.out.println("good morning");
}
}
catch(Exception e)
{
}
}
}
class B extends Thread
{
synchronized public void run()
{
try
{
while(true)
{
sleep(2000);
System.out.println("hello");
}
}
catch(Exception e)
{
}
}
}

```

```
class C extends Thread
{
synchronized public void run()
{
try
{
while(true)
{
sleep(3000);
System.out.println("welcome");
}
}
catch(Exception e)
{
}
}
}
class ThreadDemo
{
public static void main(String args[])
{
A t1=new A();
B t2=new B();
C t3=new C();
t1.start();
t2.start();
t3.start();
}
}
```

OUTPUT:  
GOOD MORNING

HELLO  
WELCOME

14) Write a java program to split a given text file into n parts. Name each part as the name of the original file followed by .part where n is the sequence number of the part file.

AIM: a java program to split a given text file into n parts. Name each part as the name of the original file followed by .part where n is the sequence number of the part file.

Source code:

```
File.txt  
carrot,squash,turnip  
potato,spinach,kale
```

Java program that reads file, splits lines

```
import java.io.BufferedReader;  
import java.io.FileReader;  
import java.io.IOException;  
  
public class Program {  
    public static void main(String[] args) throws IOException {  
  
        // Open this file.  
        BufferedReader reader = new BufferedReader(new FileReader(  
            "C:\\programs\\file.txt"));  
  
        // Read lines from file.  
        while (true) {  
            String line = reader.readLine();  
            if (line == null) {  
                break;  
            }  
            // Split line on comma.  
            String[] parts = line.split(",");  
            for (String part : parts) {  
                System.out.println(part);  
            }  
            System.out.println();  
        }  
  
        reader.close();  
    }  
}
```

Output

```
carrot  
squash  
turnip  
potato  
spinach  
kale
```



15). Write a Java program that reads a file name from the user, displays information about whether the file exists, whether the file is readable, or writable, the type of file and the length of the file in bytes.

AIM: a Java program that reads a file name from the user, displays information about whether the file exists, whether the file is readable, or writable, the type of file and the length of the file in bytes.

SOURCE CODE:

```
class AboutFile{

    public static void main(String[] args){

        Scanner input = new Scanner(System.in);

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

        String file_name = input.nextLine();

        File f = new File(file_name);

        if(f.exists())

            System.out.println("The file " +file_name+ " exists");

        else

            System.out.println("The file " +file_name+ " does not exist");

        if(f.exists()){

            if(f.canRead())

                System.out.println("The file " +file_name+ " is readable");

            else

                System.out.println("The file " +file_name+ " is not readable");

        }

    }

}
```

```

if(f.canWrite())

System.out.println("The file " +file_name+ " is writeable");

else

System.out.println("The file " +file_name+ " is not writeable");

System.out.println("The file type is: " +file_name.substring(file_name.indexOf('.')+1));

System.out.println("The Length of the file:" +f.length());

}

}

}

```

16). Write a java program that displays the number of characters, lines and words in a textfile.  
 AIM: a java program that displays the number of characters, lines and words in a text file.

Source code:

```

class FileDemo
{
    public static void main(String args[])
    {
        try
        {
            int lines=0,chars=0,words=0;
            int code=0;
            FileInputStream fis = new FileInputStream("sample.txt");
            while(fis.available()!=0)
            {
                code = fis.read();
                if(code!=10)
                chars++;
                if(code==32)
                words++;
                if(code==13)
                {
                    lines++;
                    words++;
                }
            }
        }
    }
}

```

```
        }
    }
    System.out.println("No.of characters = "+chars);
    System.out.println("No.of words = "+(words+1));
    System.out.println("No.of lines = "+(lines+1));
    fis.close();
}
catch(FileNotFoundException e)
{
    System.out.println("Cannot find the specified file...");
}
catch(IOException ie)
{
    System.out.println("Cannot read file...");
}
}
}
```

OUTPUT:

```
He is
a good
boy
```

Input and output for the above program is as follows:

```
No.of characters = 16
No.of words = 5
No.of lines =
```

**17)Write java program that inputs 5 numbers, each between 10 and 100 inclusive. As each number is read display it only if it's not a duplicate of any number already read. Display the complete set of unique values input after the user enters each new value.**

**SOURCE CODE:**

```
import java.lang.*;
import java.io.*;
import java.util.*;
class Number
{
public static void main(String args[])throws IOException
{
Scanner input =new Scanner(System.in);
int sid[]=new int[5];
int count=0;
int x=0;
int num=0;
while(x<sid.length)
{
System.out.println("enter number");
num=input.nextInt();
if((num>=10)&&(num<=100))
{
boolean digit=false;
x++;
for(int i=0;i<sid.length;i++)
{
if(sid[i]==num)
digit=true;
}
}
if(!digit)
{
```

```
sid[count]=num;
count++;
}
else
System.out.println("number was entered before\n");
}
else
{
System.out.println("the number must be between 10 &100");
}
for(inti=0;i<x;i++)
{
System.out.println(sid[i]+" ");
}
System.out.println();
}
}
}
```

**OUTPUT:**

Enter the numbers between 10 AND 100:12 15 34 67 85 67 98

The unique values:12 15 34 67 85 98

**18).** Write a java program that reads a file and displays the file on the screen with line number before each line.

Aim: a java program that reads a file and displays the file on the screen with line number before each line.

Source code: mport java.util.\*;

import java.io.\*;

class Rfile

```
{
    public static void main(String args[])throws IOException
    {
        int j=1;
        char ch;
        Scanner scr=new Scanner(System.in);
        System.out.print("\nEnter File name: ");
        String str=scr.next();
        FileInputStream f=new FileInputStream(str);
        System.out.println("\nContents of the file are");
        int n=f.available();
        System.out.print(j+": ");
        for(int i=0;i<n;i++)
        {
            ch=(char)f.read();
            System.out.print(ch);
            if(ch=='\n')
            {
                System.out.print(++j+": ");
            }
        }
    }
}
```

Output:

Enter File name: raj.txt

Contents of the file are :

1: kotireddy

2: Santhosh

3: Vamsi

**19).**Write a Java program that correctly implements producer consumer problem using the concept of inter thread communication.

AIM: a Java program that correctly implements producer consumer problem using the concept of inter thread communication.

SOURCE CODE: `class Q`

```
{  
  
    int n;  
  
    boolean valueSet=false;  
  
    synchronized int get()  
    {  
  
        if(!valueSet)    }  
}
```

```
    try
    {
        wait();
    }
    catch(InterruptedException e)
    {
        System.out.println("InterruptedException caught");
    }
    System.out.println("Got: "+n);
    valueSet=false;
    notify();
    return n;
}
synchronized void put(int n)
{
    if(valueSet)
    try
    {
        wait();
    }
    catch(InterruptedException e)
    {
        System.out.println("InterruptedException caught");
    }
    this.n=n;
    valueSet=true;
    System.out.println("Put: "+n);
}
```



```

        notify();
    }
}

class Producer implements Runnable
{
    Q q;

    Producer(Q q)
    {
        this.q=q;

        new Thread(this,"Producer").start();
    }

    public void run()
    {
        inti=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();
        }
    }
}
class ProdCons
{
    public static void main(String[] args)
    {
        Q q=new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-c to stop");
    }
}

```

Output:

```

Put:1
Got:1
Put:2
Got:2
Put:3
Got:3
Put:4
Got:4
Put:5
Got:5

```

Put:6  
Got:6  
Put:7  
Got:7  
Put:8  
Got:8  
Put:9  
Got:9  
Put:10  
Got:10  
Put:11  
Got:11  
Put:12  
Got:12  
Put:13  
Got:13  
Put:14  
Got:14

**20).** Develop a Java application for stack operation using Buttons and JOptionPane input and Message dialog box.

AIM: a Java application for stack operation using Buttons and JOptionPane input and Message dialog box.

Source code:

```
import javax.swing.*;
```

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

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

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

```
public class VisStackApplet extends JApplet implements ActionListener, Runnable {
```

```
    ScrolledPanel visPanel; //Where to paint graphics
```

```
    MyScrollPane msp;
```

```
    Button executeButton;
```

```
    Button historyButton;
```

```
    TextArea userInputText;
```

```
    TextArea history;
```

```
    JFrame historyFrame;
```

```
    JTextField statusLine;
```

```
    MyStack theStack; // The data structure being demonstrated
```

```
    Font stackFont;
```

```
    int cellHeight = 20; // For drawing the stack.
```

```
    int cellWidth = 200; // How wide to plot pink rectangles
```

```
    int cellGap = 4; // vertical space between successive cells
```

```
    int topMargin = 25; // Space above top of stack.
```

```
    int fontSize = 16; // Height of font for displaying stack elemens.
```

```
    int leftMargin = 20; // x value for left side of cells
```

```
    int bottomMargin = 10; // Minimum space betw. bot. of visPanel and bot. of lowest cell.
```

```
    int leftOffset = 5; // space between left side of cell and contents string.
```

```
    int delay = 300; // default is to wait 300 ms between updates.
```

```
    Thread displayThread = null;
```

```
    public void init() {
```

```
        setSize(300,300); // default size of applet.
```

```
        visPanel = new ScrolledPanel();
```

```
        visPanel.setPreferredSize(new Dimension(400,400));
```

```
        msp = new MyScrollPane(visPanel);
```

```
        msp.setPreferredSize(new Dimension(400,200));
```

```
    VEMUI
```

```

Container c = getContentPane();
c.setLayout(new BorderLayout());
c.add(msp, BorderLayout.CENTER);
JPanel buttons = new JPanel();
buttons.setLayout(new FlowLayout());
JPanel controls = new JPanel();
controls.setLayout(new BorderLayout());
executeButton = new Button("Execute");
executeButton.addActionListener(this);
buttons.add(executeButton);
historyButton = new Button("History");
historyButton.addActionListener(this);
buttons.add(historyButton);
userInputText = new TextArea(";Enter commands here.");
statusLine = new JTextField();
statusLine.setBackground(Color.lightGray);
controls.add(buttons, BorderLayout.WEST);
controls.add(userInputText, BorderLayout.CENTER);
controls.add(statusLine, BorderLayout.SOUTH);
controls.setPreferredSize(new Dimension(400,100));
c.add(controls, BorderLayout.SOUTH);
c.validate();

```

```

theStack = new MyStack();
stackFont = new Font("Helvetica", Font.PLAIN, 20);
history = new TextArea("VisStackApplet history:\n", 20, 40);
}

```

```

class ScrolledPanel extends JPanel {
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        paintStack(g);
    }
}

class MyScrollPane extends JScrollPane {
    MyScrollPane(JPanel p) {
        super(p,
            JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,
            JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS);
    }
}
VEMUI

```

```
}
```

```
class MyStack extends Vector {
```

```
    int n; // number of elements in the stack  
    int npushes; // number of PUSH operations so far.  
    int npops; // number of POP operations so far.
```

```
    void init() {  
        n = 0; npushes = 0; npops = 0;  
    }
```

```
    void push(Object elt) {  
        add(n, elt);  
        n++;  
        npushes++;  
    }
```

```
    Object pop() {  
        if (n == 0) { return null; }  
        Object o = lastElement();  
        n--;  
        npops++;  
        remove(n);  
        return o;  
    }  
}
```

```
public void actionPerformed(ActionEvent e) {  
    if (e.getActionCommand().equals("Execute")) {  
        displayThread = new Thread(this);  
        displayThread.start();  
        return;  
    }  
    if (e.getActionCommand().equals("History")) {  
        if (historyFrame == null) {  
            historyFrame = new JFrame("History of the VisStackApplet");  
            historyFrame.getContentPane().add(history);  
            historyFrame.setSize(new Dimension(300,300));  
        }  
        historyFrame.show();  
    }  
}
```

VEMUI

```
    System.out.println("Should have displayed the history window");
}
}
```

// The following is executed by a separate thread for the display.

```
public void run() {
    String commands = userInputText.getText();
    String line = "";
    StringTokenizer lines;
    for (lines = new StringTokenizer(commands, "\n\r\f");
        lines.hasMoreTokens();) {
        line = lines.nextToken();
        process(line);
    }
    userInputText.setText(""); // Erase all the processed input.
}
```

// Helper function called by the run method above:

```
void process(String command) {
    String arg = "";
    StringTokenizer st = new StringTokenizer(command);
    if (! st.hasMoreTokens()) { return; }
    String firstToken = st.nextToken();
    if (firstToken.startsWith(";")) { return; }
    history.appendText(command + "\n");
    statusLine.setText(command);
    if (firstToken.equals("RESET")) {
        theStack = new MyStack();
        updateDisplay(); return;
    }
    if (firstToken.equals("SIZE")) {
        String stats = "Current number of elements: " + theStack.n;
        statusLine.setText(stats);
        history.appendText("; " + stats + "\n");
        return;
    }
    if (firstToken.equals("STATS")) {
        String stats = "npushes: " + theStack.npushes +
            "; npops: " + theStack.npops;
        statusLine.setText(stats);
        history.appendText("; " + stats + "\n");
    }
}
```

VEMUI

```

return;
}
if (firstToken.equals("DELAY")) {
    if (st.hasMoreTokens()) {
        arg = st.nextToken();
        try { delay =(new Integer(arg)).intValue(); }
        catch(NumberFormatException e) {
            delay = 0;
        }
        statusLine.setText("delay = " + delay);
    }
    history.appendText("; delay is now " + delay + "\n");
    return;
}
if (firstToken.equals("PUSH")) {
    arg = "UNDEFINED ELEMENT";
    if (st.hasMoreTokens()) { arg = st.nextToken(); }
    theStack.push(arg);
    checkScrolledPanelSize();
    updateDisplay(); return;
}
if (firstToken.equals("POP")) {
    theStack.pop();
    updateDisplay(); return;
}
history.appendText("[Unknown Stack command]\n");
statusLine.setText("Unknown Stack command: " + command);
}

// Here is a "middleman" method that updates the display waiting with
// the current time delay after each repaint request.
void updateDisplay() {
    visPanel.repaint();
    if (delay > 0) {
        try {
            Thread.sleep(delay);
        }
        catch(InterruptedException e) {}
    }
}
}

```



```

// Here is the graphics method to actually draw the stack.
// It's called by the ScrolledPanel paintComponent method.
void paintStack(Graphics g) {
    g.setFont(stackFont);
    g.drawString("Top of Stack", 10,20);
    int ystart = theStack.n * (cellHeight + cellGap) + topMargin;
    int ycentering = (cellHeight - fontSize) / 2;
    int ypos = ystart;
    for (Enumeration e = theStack.elements(); e.hasMoreElements();) {
        String elt = (String) e.nextElement();
        g.setColor(Color.pink);
        g.fillRect(leftMargin, ypos, cellWidth, cellHeight);
        g.setColor(Color.black);
        g.drawString(elt, leftMargin + leftOffset, ypos+cellHeight - ycentering);
        ypos -= (cellHeight + cellGap);
    }
}

// The following computes the height of the display area needed by the current
// stack, and if it won't fit in the scrolled panel, it enlarges the scrolled panel.
// In the current implementation, the panel never gets smaller, even if the stack
// becomes empty. This could easily be changed.
void checkScrolledPanelSize() {
    int heightNeeded = topMargin + theStack.n * (cellHeight + cellGap) + cellHeight+ bottomMargin;
    Dimension d = visPanel.getPreferredSize();
    int currentHeight = (int) d.getHeight();
    int currentWidth = (int) d.getWidth();
    if (heightNeeded > currentHeight) {
        visPanel.setPreferredSize(new Dimension(currentWidth, heightNeeded));
        visPanel.revalidate(); // Adjust the vertical scroll bar.
    }
}
}
}

```

21) ).Develop a Java application to perform Addition, Division, Multiplication and subtraction using JOptionPane dialog Box and Textfields.

AIM: Develop a Java application to perform Addition, Division, Multiplication and subtraction using JOptionPane dialog Box and Textfields.

SOURCE CODE:

```
import java.util.*;
import javax.swing.*;

public class Calculator
{
public static void main(String[] arg)
{
String input1,input2,operator;
Double operand1,operand2;

    input1 = JOptionPane.showInputDialog("Enter 1st Value");
operator = JOptionPane.showInputDialog("Enter Operator(+,-,*,%./)");
    input2 = JOptionPane.showInputDialog("Enter 2nd Value");

    operand1=Double.parseDouble(input1);
    operand2=Double.parseDouble(input2);

double result=0.0;
if (operator.equals("+"))
result = operand1+operand2;
else if(operator.equals("-"))
result = operand1-operand2;
else if(operator.equals("*"))
result = operand1*operand2;
else if(operator.equals("/"))
result = operand1/operand2;
else if(operator.equals("%"))
result = operand1%operand2;
else
JOptionPane.showMessageDialog(null, "Inappropriate Input");

JOptionPane.showMessageDialog(null,"Your results are " + result);
}

// TODO Auto-generated method stub
```

```
private static double operand2(double result) {  
// TODO Auto-generated method stub  
return 0;  
}  
}
```

22. Develop a Java application for the blinking eyes and mouth should open while blinking.

AIM: Java application for the blinking eyes and mouth should open while blinking

SOURCE CODE:

```
import java.applet.Applet;  
//<applet code="A.class" width=200 height=200></applet>
```

```
import java.awt.BorderLayout;
```

```
import java.awt.Canvas;
```

```
import java.awt.Color;
```

```
import java.awt.Graphics;
```

```
public class A extends Applet {
```

```
    private static final long serialVersionUID = -1152278362796573663L;
```

```
    public class MyCanvas extends Canvas {
```

```
        private static final long serialVersionUID = -4372759074220420333L;
```

```
        private int flag = 0;
```

```
        public void paint(Graphics g) {
```

```
    draw();  
}
```

```
public void draw() {  
    Graphics g = this.getGraphics();  
    g.setColor(Color.BLACK);  
    super.paint(g);  
  
    if (flag == 0) {  
  
        System.out.println(flag);  
  
        g.drawOval(40, 40, 120, 150); // face  
  
        g.drawRect(57, 75, 30, 5); // left eye shut  
  
        g.drawRect(110, 75, 30, 20); // right eye  
  
        g.drawOval(85, 100, 30, 30); // nose  
  
        g.fillArc(60, 125, 80, 40, 180, 180); // mouth  
  
        g.drawOval(25, 92, 15, 30); // left ear  
  
        g.drawOval(160, 92, 15, 30); // right ear  
  
        flag = 1;  
  
    } else {  
  
        System.out.println(flag);  
        g.drawOval(40, 40, 120, 150); // face  
  
        g.drawOval(57, 75, 30, 20); // left eye  
  
        g.drawOval(110, 75, 30, 20); // right eye  
  
        g.fillOval(68, 81, 10, 10); // left pupil  
  
        g.fillOval(121, 81, 10, 10); // right pupil  
  
        g.drawOval(85, 100, 30, 30); // nose  
  
    }  
}
```

```
        g.fillArc(60, 125, 80, 40, 180, 180);// mouth

        g.drawOval(25, 92, 15, 30);// left ear

        g.drawOval(160, 92, 15, 30);// right ear

        flag = 0;

    }

    try {
        Thread.sleep(900);
    } catch (Exception e) {
        System.out.println("killed while sleeping");
    }

    this.repaint(100);
}
}
```

```
public void init() {

    this.C = new MyCanvas();

    this.setLayout(new BorderLayout());

    this.add(C, BorderLayout.CENTER);

    C.setBackground(Color.GRAY);

}
```

```
private MyCanvas C;

}
```

23).Develop a Java application that simulates a traffic light. The program lets the user select one of three lights: Red, Yellow or Green with radio buttons. On selecting a button an appropriate message with—STOP |or—READY |or|GO |should appear above the buttons in selected color. Initially, there is no message show.

**AIM:** Develop a Java application that simulates a traffic light. The program lets the user select one of three lights: Red, Yellow or Green with radio buttons. On selecting a button an appropriate message with—STOP |or—READY |or|GO |should appear above the buttons in selected color. Initially, there is no message show.

#### **SOURCE CODE:**

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

/*
 * <applet code = "TrafficLightsExample" width = 1000 height = 500>
 * </applet>
 * */

public class TrafficLightsExample extends Applet implements ItemListener{

    CheckboxGroup grp = new CheckboxGroup();
    Checkbox redLight, yellowLight, greenLight;
    Label msg;
    public void init(){
        redLight = new Checkbox("Red", grp, false);
        yellowLight = new Checkbox("Yellow", grp, false);
        greenLight = new Checkbox("Green", grp, false);
        msg = new Label("");

        redLight.addItemListener(this);
        yellowLight.addItemListener(this);
        greenLight.addItemListener(this);

        add(redLight);
        add(yellowLight);
        add(greenLight);
        add(msg);
        msg.setFont(new Font("Serif", Font.BOLD, 20));
    }
    public void itemStateChanged(ItemEvent ie) {
        redLight.setForeground(Color.BLACK);
        yellowLight.setForeground(Color.BLACK);
        greenLight.setForeground(Color.BLACK);

        if(redLight.getState() == true) {
```

```
        redLight.setForeground(Color.RED);
        msg.setForeground(Color.RED);
        msg.setText("STOP");
    }
    else if(yellowLight.getState() == true) {
        yellowLight.setForeground(Color.YELLOW);
        msg.setForeground(Color.YELLOW);
        msg.setText("READY");
    }
    else{
        greenLight.setForeground(Color.GREEN);
        msg.setForeground(Color.GREEN);
        msg.setText("GO");
    }
}
}
```

**OUTPUT:**



24). Develop a Java application to implement the opening of a door while opening man should present before hut and closing man should disappear.

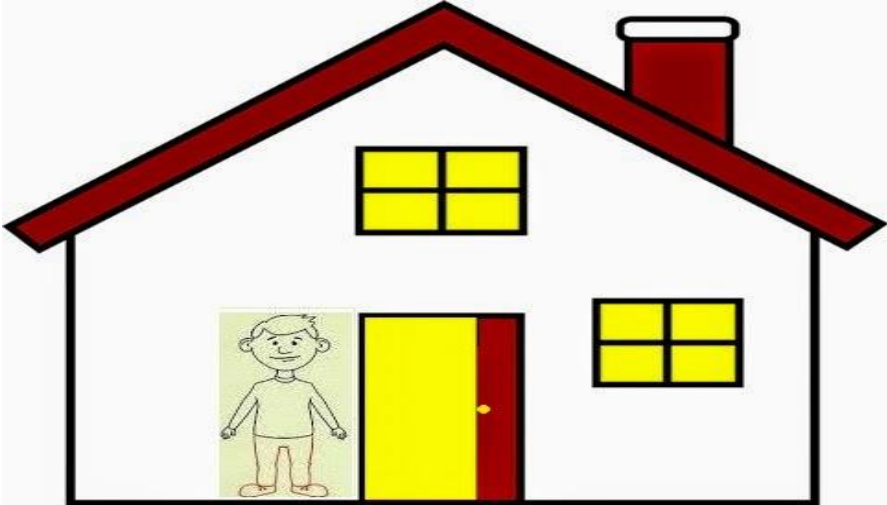
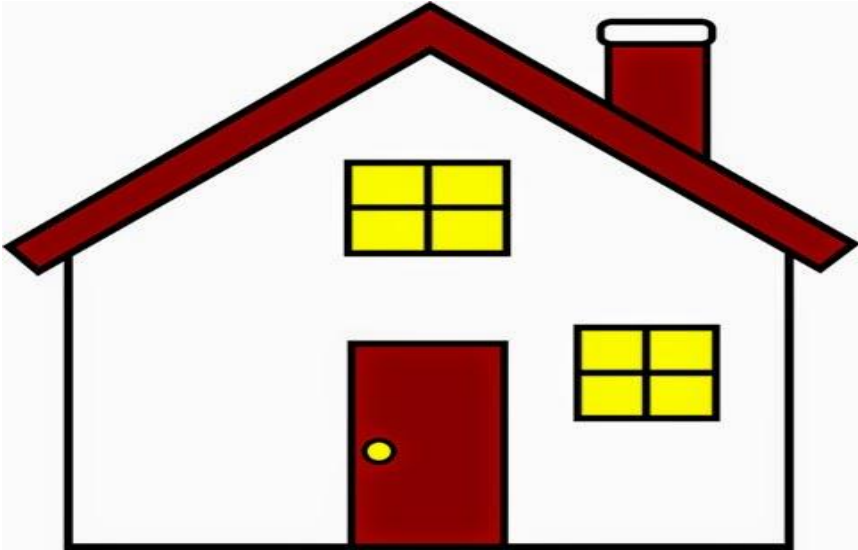
AIM: Develop a Java application to implement the opening of a door while opening man should present before hut and closing man should disappear.

Source code:

```
import java.awt.*;
import java.awt.event.*;
class Animation extends JFrame implements ActionListener
{
    ImageIcon ii1, ii2;
    Container c;
    JButton b1,b2;
    JLabel lb1;
    Animation()
    {
        c = getContentPane();
        c.setLayout(null);
        ii1 = new ImageIcon("house0.jpg");
        ii2 = new ImageIcon("house1.jpg");
        lb1 = new JLabel(ii1);
        lb1.setBounds(50,10,500,500);
        b1 = new JButton("Open");
        b2 = new JButton("Close");
        b1.addActionListener(this);
        b2.addActionListener(this);
        b1.setBounds(650,240,70,40);
        b2.setBounds(650,320,70,40);
        c.add(lb1);
        c.add(b1);
        c.add(b2);
    }
    public void actionPerformed(ActionEvent ae)
    {
        String str = ae.getActionCommand();
        if( str.equals("Open") )
            lb1.setIcon(ii2);
        else
            lb1.setIcon(ii1);
    }
    public static void main(String args[])
    {
        Animation ob = new Animation();
        ob.setTitle(" Animation");
        ob.setSize(800,600);
        ob.setVisible(true);
        ob.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```



Output:



25). Develop a Java application by using J text Field to read decimal value and converting a decimal number into binary number then print the binary value in another Jtext Field.

AIM: a Java application by using J text Field to read decimal value and converting a decimal number into binary number then print the binary value in another Jtext Field.

SOURCE CODE:

```
<applet code="Rottenapplet.class" height=300 width=300></applet>
```

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

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

```
import java.applet.*;  
import javax.swing.*;
```

```
public class rottenapplet extends JApplet implements ActionListener
```

```
{
```

```
JPanel mainpanel=new JPanel(new GridLayout (3,1));
```

```
JPanel p1=new JPanel(new FlowLayout(0));
```

```
JPanel p2=new JPanel(new FlowLayout (0));
```

```
JPanel p3=new JPanel(new FlowLayout ());
```

```
JTextField q1=new JTextField (10);
```

```
JTextField q2=new JTextField (10);
```

```
JButton clickbutton = new JButton("convert");
```

```
public void init()
```

```
{
```

```
getContentPane().add(mainpanel);
```

```
mainpanel.add(p1);
```

```
mainpanel.add(p2);  
mainpanel.add(p3);
```

```
p1.add(new JLabel("Insert Decimal:"));
```

VEMUI

```
p1.add(q1);

p2.add(clickbutton);

p3.add(new JLabel("Decimal to Binary:"));

p3.add(q2);
clickbutton.addActionListener(this);
}

public void actionPerformed(ActionEvent x)

{

if(x.getSource()==clickbutton)

{

int counter,dec,user;

user=Integer.valueOf(q1.getText()).intValue();

String[]conversion=new String[8];

String[]complete=new String[4];

counter=0;

complete[0]="";
```

```
do

{

dec=user%2;

conversion[counter]=String.valueOf(dec);

complete[0]=conversion[counter]+complete[0];

user=user/2;

counter+=1;

}

while(user !=0);

q2.setText(String.valueOf(complete[user]));

}

}

}
```

26).Develop a Java application that handles all mouse events and shows the event name at the center of the window when a mouse event is fired. Use adapter classes.

AIM: Develop a Java application that handles all mouse events and shows the event name at the center of the window when a mouse event is fired. Use adapter classes.

Source code:

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
/*<applet code="MouseDemo" width=300 height=300>

</applet>*/

public class Mouse Demo extends Applet implements Mouse Listener, MouseMotionListener

{

int mx=0; int my=0;
String msg=""; public void init()
{

addMouseListener(this); addMouseMotionListener(this);
}

public void mouseClicked(MouseEvent me)

{

mx=20; my=40;
msg="Mouse Clicked"; repaint();
}

public void mousePressed(MouseEvent me)

{

mx=30; my=60;
msg="Mouse Pressed";

repaint();
}

public void mouseReleased(MouseEvent me)

{

mx=30; my=60;
VEMUI
```

```

msg="Mouse Released"; repaint();
}

public void mouseEntered(MouseEvent me)

{

mx=40; my=80;
msg="Mouse Entered"; repaint();
}

public void mouseExited(MouseEvent me)

{

mx=40; my=80;
msg="Mouse Exited"; repaint();
}

public void mouseDragged(MouseEvent me)

{

mx=me.getX();

my=me.getY();

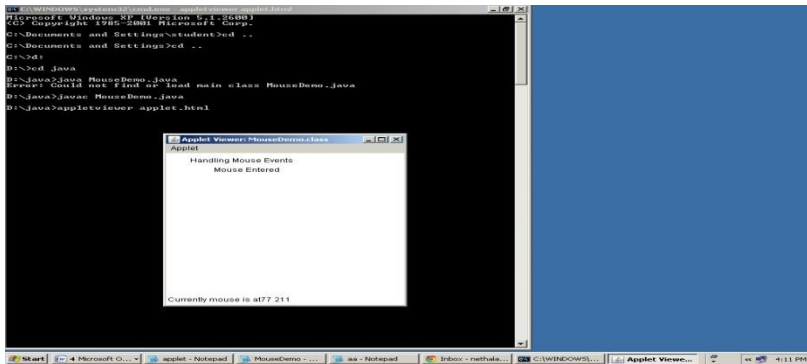
setStatus("Currently mouse dragged"+mx+" "+my);

repaint(); }

public void mouseMoved(MouseEvent me)
{
mx=me.getX();
my=me.getY();
setStatus("Currently mouse is at"+mx+" "+my); repaint();
}
public void paint(Graphics g)
{
g.drawString("Handling Mouse Events",30,20);
g.drawString(msg,60,40);
}
}

```

## OUTPUT:



27. Develop a Java application to demonstrate the key event handlers.

AIM: Develop a Java application to demonstrate the key event handlers.

Source Code:

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

public class Test extends Applet implements KeyListener
{
    String msg="";

    public void init()
    {
        addKeyListener(this);
    }

    public void keyPressed(KeyEvent k)
    {
        showStatus("KeyPressed");
    }
}
```

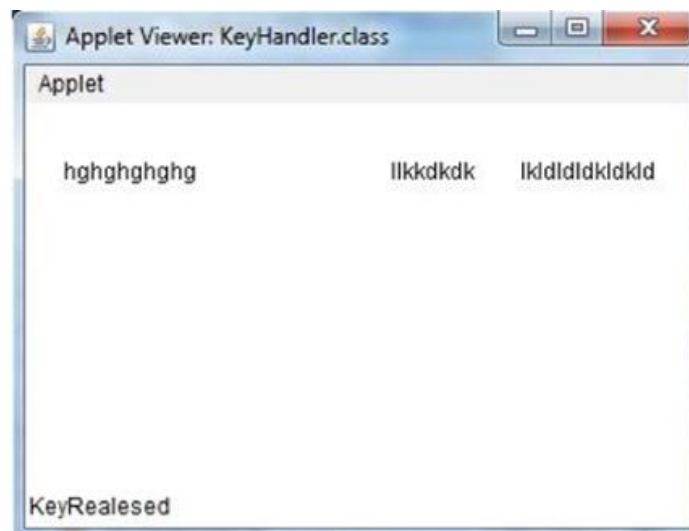
VEMUI

```
public void keyReleased(KeyEvent k)
{
    showStatus("KeyRealesed");
}
public void keyTyped(KeyEvent k)
{
    msg = msg+k.getKeyChar();
    repaint();
}
public void paint(Graphics g)
{
    g.drawString(msg, 20, 40);
}
}
```

HTML code:

```
<applet code="Test" width=300, height=100>
</applet>
```

Output:





28). Develop a Java application to find the maximum value from the given type of elements using a generic function.

Source code:

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import java.applet.*;
import java.awt.event.*;
import java.awt.*;
public class Test extends Applet implements KeyListener
{
String msg="";
public void init()
{
    addKeyListener(this);
}
public void keyPressed(KeyEvent k)
{
    showStatus("KeyPressed");
}
public void keyReleased(KeyEvent k)
{
    showStatus("KeyRealesed");
}
public void keyTyped(KeyEvent k)
{
    msg = msg+k.getKeyChar();
    repaint();
}
public void paint(Graphics g)
{
    g.drawString(msg, 20, 40);
}
}
```

HTML code:

```
<applet code="Test" width=300, height=100>
</applet>
```

OUTPUT

```
D:\>Java Prgs>javac genericdemo.java
```

```
D:\>Java Prgs>java genericdemo
```

```
Max value in inums: 10
```

```
Max value in inums: 1
```

```
Max value in chs: v
```

```
Max value in chs: a
```

```
Max value in chs: 88.3
```

```
Max value in chs: 10.4
```

29). Develop a Java application that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -, \*, % operations. Add a text field to display the result.

Source code:

```
import javax.swing.*;
import java.awt.*;
import
java.awt.event.*;
//<applet code=Calculator height=300
width=200></applet> public class Calculator extends
JApplet
{
    public void init()
    {
        CalculatorPanel calc=new CalculatorPanel();
        getContentPane().add(calc);
    }
}
class CalculatorPanel extends JPanel implements ActionListener
{
    JButton
    n1,n2,n3,n4,n5,n6,n7,n8,n9,n0,plus,minus,mul,div,dot,equal;
    static JTextField result=new JTextField("0",45);
    static String lastCommand=null;
    JOptionPane p=new
    JOptionPane(); double
    preRes=0,secVal=0,res; private
    static void assign(String no)
    {
        if((result.getText().equals("0")
        ) result.setText(no);
        else if(lastCommand=="")
        {
            result.setText(no);
            lastCommand=null;
        }
    }
    else
        result.setText(result.getText()+no);
}

public CalculatorPanel()
{
    setLayout(new BorderLayout());
    result.setEditable(false);

    result.setSize(300,200);
```

```
add(result, BorderLayout.NORTH);
```

```
JPanel panel=new JPanel();
```

```
panel.setLayout(new
```

```
GridLayout(4,4));
```

```
n7=new JButton("7");
```

```
panel.add(n7);
```

```
n7.addActionListener(this);
```

```
n8=new JButton("8");
```

```
panel.add(n8);
```

```
n8.addActionListener(this);
```

```
n9=new JButton("9");
```

```
panel.add(n9);
```

```
n9.addActionListener(this);
```

```
div=new JButton("/");
```

```
panel.add(div);
```

```
div.addActionListener(this);
```

```
n4=new JButton("4");
```

```
panel.add(n4);
```

```
n4.addActionListener(this);
```

```
n5=new JButton("5");
```

```
panel.add(n5);
```

```
n5.addActionListener(this);
```

```
n6=new JButton("6");
```

```
panel.add(n6);
```

```
n6.addActionListener(this);
```

```
mul=new JButton("*");
```

```
panel.add(mul);
```

```
mul.addActionListener(this);
```

```
n1=new JButton("1");
```

```
panel.add(n1);
```

```
n1.addActionListener(this);
```

```
n2=new JButton("2");
```

```
panel.add(n2);
```

```
n2.addActionListener(this);
```

```
n3=new JButton("3");
```

```
panel.add(n3);
```

```
n3.addActionListener(this);
```

```
minus=new JButton("-");
```

```
panel.add(minus);
```

```
minus.addActionListener(this);
```

```
dot=new JButton(".");
```

```
panel.add(dot);
```

```
dot.addActionListener(this);
```

```
n0=new JButton("0");
```

```
panel.add(n0);
```

```

n0.addActionListener(this);
equal=new JButton("=");
panel.add(equal);
equal.addActionListener(this);
plus=new JButton("+");
panel.add(plus);
plus.addActionListener(this);
add(panel, BorderLayout.CENTE
R);
}
public void actionPerformed(ActionEvent ae)

{
if(ae.getSource()==n1) assign("1");
else if(ae.getSource()==n2)
assign("2"); else
if(ae.getSource()==n3) assign("3");
else if(ae.getSource()==n4)
assign("4"); else
if(ae.getSource()==n5) assign("5");
else if(ae.getSource()==n6)
assign("6"); else
if(ae.getSource()==n7) assign("7");
else if(ae.getSource()==n8)
assign("8"); else
if(ae.getSource()==n9) assign("9");
else if(ae.getSource()==n0)
assign("0"); else
if(ae.getSource()==dot)
{
if(((result.getText()).indexOf(".")==-1)
result.setText(result.getText()+".");
}
}
else if(ae.getSource()==minus)
{
preRes=Double.parseDouble(result.getText
()); lastCommand="-";
result.setText("0");
}
else if(ae.getSource()==div)
{
preRes=Double.parseDouble(result.getText());
lastCommand="/";
result.setText("0");
}
else if(ae.getSource()==equal)
{
secVal=Double.parseDouble(result.getText());
if(lastCommand.equals("/"))

```

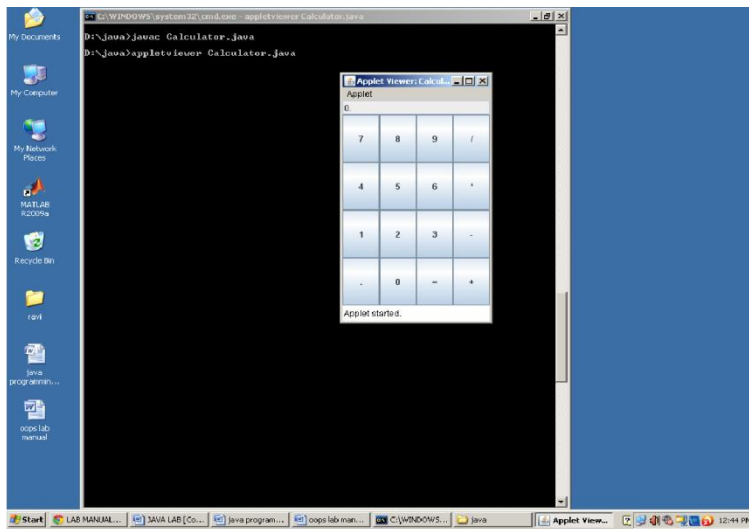
```

        res=preRes/secVal;
    else if(lastCommand.equals("*"))
        res=preRes*secVal;
    else if(lastCommand.equals("-"))
        res=preRes-secVal;
    else if(lastCommand.equals("+"))
        res=preRes+secVal;
    result.setText(" "+res);
    lastCommand="=";
    }
    else if(ae.getSource()==mul)
    {
        preRes=Double.parseDouble(result.getText());
        lastCommand="*";
        result.setText("0");
    }
    else if(ae.getSource()==plus)
    {
        preRes=Double.parseDouble(result.getText());
        lastCommand="+";

        result.setText("0");
    }
}
}
}

```

OUTPUT:



30).Develop a Java application for handling mouse events  
Source code:

```
import java.awt.*;

import java.awt.event.*;

import java.applet.*;
/*

<applet code="Mouse" width=500 height=500>

</applet>

*/

public class Mouse extends Applet

implements MouseListener,MouseMotionListener

{

int X=0,Y=20;

String msg="MouseEvents";

public void init()

{

addMouseListener(this);

addMouseMotionListener(this);
```

```
        setBackground(Color.black);

        setForeground(Color.red);

    }

    public void
        mouseEntered(MouseEvent m)

    {

        setBackground(Color.magenta);

        showStatus("Mouse Entered");

        repaint();

    }

    public void mouseExited(MouseEvent m)

    {

        setBackground(Color.black);

        showStatus("Mouse Exited");

        repaint();

    }

}
```

```
}
```

```
public void mousePressed(MouseEvent m)
```

```
{
```

```
    X=10;
```

```
    Y=20;
```

```
    msg="NEC";
```

```
    setBackground(Color.green);
```

```
        repaint();
```

```
}
```

```
public void mouseReleased(MouseEvent m)
```

```
{
```

```
    X=10;
```

```
    Y=20;
```

```
    msg="Engineering";
```

```
    setBackground(Color.blue);
```



```
        repaint();

    }

    public void mouseMoved(MouseEvent m)

    {

        X=m.getX();

        Y=m.getY();

        msg="College";

        setBackground(Color.white);

        showStatus("Mouse Moved");

        repaint();

    }

    public void mouseDragged(MouseEvent m)

    {

        msg="CSE";
```

```
        setBackground(Color.yellow);  
showStatus("MouseMoved"+m.getX()+" "+m.getY());
```

```
        repaint();
```

```
    }
```

```
public void mouseClicked(MouseEvent m)
```

```
{
```

```
    msg="Students";
```

```
    setBackground(Color.pink);
```

```
    showStatus("MouseClicked");
```

```
    repaint();
```

```
}
```

```
public void paint(Graphics g)
```

```
{
```

```
    g.drawString(msg,X,Y);
```

}

}

31. Develop a Java application to establish a JDBC connection, create a table student with properties name, register number, mark1, mark2, mark3. Insert the values into the table by using the java and display the information of the students at front end.

Source code:

```
Import javax.swing.*;
```

```
Import java.sql.*;
```

```
Class StudentForm extends JFrame
```

```
{
```

```
JLabel l1,l2,l3,l4,l5,l6,l7;
```

```
JTextField t1,t2,t3,t4,t5,t6,t7;
```

```
JButton b1,b2;
```

```
Connection con;
```

```
PreparedStatement insert;
```

```
PreparedStatement update;
```

```
PreparedStatement delet;
```

```
PreparedStatement select;
```

```
StudentForm()
```

```
{
```

```
setSize(355,300);
```

```
setLocation(100,100);
```

```
Container c=getContentPane();
```

```
Title=new JLabel("Student Details");
```

```
Title.setFont(new Font("Dialog",Font.BOLD,15));
```

```
l1=new JLabel("Register No");
```

```
l2=new JLabel("Student Name");
```

```
l3=new JLabel("Marks1");
```

```
l4=new JLabel("Marks2");
```

```
l5=new JLabel("Marks3");
```

```
t1=new JTextField(10);
```

```
t2=new JTextField(10);
```

```
t3=new JTextField(10);
```

```
t4=new JTextField(10);
```

```
t5=new JTextField(10);
```

```
b1=new JButton("Insert");
```

```
b2=new JButton("Display");
```

```
c.setLayout(null);
```

```
title.setBounds(60,10,160,20);
```

```
c.add(title);
```

```
l1.setBounds(40,40,50,20);
```

```
c.add(l1);
```

```
t1.setBounds(95,40,108,20);
```

```
c.add(t1);
```

```
l2.setBounds(40,70,50,20);
```

```
c.add(l2);
```

```
t2.setBounds(95,70,108,20);
```

```
c.add(t2);
```



```
l3.setBounds(40,100,50,20);
```

```
c.add(l3);
```

```
t3.setBounds(95,100,108,20);
```

```
c.add(t3);
```

```
b1.setBounds(10,140,65,40);
```

```
c.add(b1);
```

```
b2.setBounds(77,140,65,40);
```

```
c.add(b2);
```

```
//b3.setBounds(144,140,65,40);
```

```
//c.add(b3);
```

```
//b4.setBounds(211,140,65,40);
```

```
//c.add(b4);
```

```
Info=new Label("Getting connected to the database");
```

```
Info.setFont(new Font("Dialog",Font.BOLD,15));
```

```
Info.setBounds(20,190,330,30);
```

```
c.add(info);
```

```
b1.addActionListener(new InsertListener());
```

```
b2.addActionListener(new DisplayListener());
```

```
setVisible(true);
```

```
getConnection();
```

```
}
```

```
Void getConnection()
```

```
{
```

```
try
```

```
{
```

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
```

```
String url="jdbc:odbc:student";
```

```
Con=DriverManager.getConnection(url,"scott","tiger");
```

```
Info.setText("Connection is established with the database");
```

```
insertps=con.prepareStatement("Insert into student values(?,?,?,?)");
```

```
selectps=con.prepareStatement("select * from student where studentno=?");
```

```
}
```

```
Catch(ClassNotFoundException e)
```

```
{
```

```
System.out.println("Driver class not found....");
```

```
System.out.println(e);
```

```
}
```

```
Catch(SQLException e)
```

```
{
```

```
Info.setText("Unable to get connected to the database");
```

```
}
```

```
}
```

Class insertListener implements ActionListener

```
{
```

```
Public void actionPerformed(ActionEvent e)
```

```
{
```

```
Try
```

```
{
```

```
Int sno=Integer.parseInt(t1.getText());
```

```
String name=t2.getText();
```

```
Int m1= Integer.parseInt(t3.getText());
```

```
Int m2=Integer.parseInt(t1.getText());
```

```
Int m3=Integer.parseInt(t1.getText());
```

```
Insertps.setInt(1,sno);
```

```
Insertps.setString(2,name);
```

```
Insertps.setInt(3,m1);
```

```
Insertps.setInt(4,m2);
```

```
Insertps.setInt(5,m3);
```

```
Insertps.executeUpdate();
```

```
Info.setText("One row inserted successfully");
```

```
Insertps.clearParameters();
```

```
T1.setText("");
```

```
T2.setText("");
```

```
T3.setText("");
```

```
T4.setText("");
```

```
T5.setText("");
```

```
}
```

```
Catch(SQLException se)
```

```
{
```

```
Info.setText("Failed to insert a record...");
```

```
}
```

```
Catch(Exception de)
```

```
{
```

```
Info.setText("enter proper data before insertion...");
```



```
}
```

```
}
```

```
}
```

Class DisplayListener implements ActionListener

```
{
```

Public void actionPerformed(ActionEvent e)

```
{
```

```
    try
```

```
    {
```

```
        int  
        sno=Integer.parseInt(t1.getText());
```

```
Selectps.setInt(1,sno);
```

```
Selectps.execute();
```

```
ResultSet
```

```
rs=selectps.getResultSet();
```

```
rs.next();
```

```
t2.setText(rs.getString(2));
```

```
t3.setText(""+rs.getFloat(3));
```

```
info.setText("One  
row displayed successfully");
```

```
selectps.clearPameters();
```

```
}
```

```
Catch(SQLException se)
```

```
{
```

```
    Info.setText("Failed to  
show the record...");
```

```
}
```

```
Catch(Exception de)
```

```
{
```

```
    Info.setText("enter proper student no before selecting  
show..");
```

```
}
```

```
}
```

```
}
```

```
Public static void main(String args[])
```

```
{
```

```
    New StudentForm();
```

```
}
```

}



```
        result.setText("0");  
    }  
}
```

Output: