

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING



JAVA PROGRAMMING

LAB MANUAL

Subject Code: KG21CS407

Regulation: KG21

Academic Year: 2022-23

II B. TECH II SEMESTER

COMPUTER SCIENCE AND ENGINEERING

KG REDDY COLLEGE OF ENGINEERING AND TECHNOLOGY

Affiliated To JNTUH, Chilkur,(V), Moinabad(M) R. R Dist, TS-501504

VISION AND MISSION OF THE INSTITUTION

VISION:

To become an institution which is internationally recognized for its holistic approach to engineering, innovative teaching and learning culture, research and entrepreneurial ecosystem, and sustainable social impact in the community.

MISSION:

- To offer undergraduate and post-graduate programs which are supported through industry relevant curriculum and innovative teaching and learning processes that would help students succeed in their professional careers.
- To provide faculty and students with an ecosystem that fosters innovation, research, entrepreneurship, and international exposure through strategic partnerships with government organizations and collaboration with industries.
- To provide holistic learning environment to students which will contribute to their personal and professional growth and enable them to become leaders in their respective fields.
- To contribute to the development of the region by using our technological expertise to work with nearby communities and support them in their social and economic development.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION:

To be recognized as a department of excellence by stimulating a learning environment in which students and faculty will thrive and grow to achieve their professional, institutional and societal goals.

MISSION:

- To provide high quality technical education to students that will enable life-long learning and build expertise in advanced technologies in Computer Science and Engineering.
- To promote research and development by providing opportunities to solve complex engineering problems in collaboration with industry and government agencies.
- To encourage professional development of students that will inculcate ethical values and leadership skills through entrepreneurship while working with the community to address societal issues.

PROGRAM EDUCATIONAL OBJECTIVES

PEO 1: Graduates will provide solutions to difficult and challenging issues in their profession by applying computer science and engineering theory and principles.

PEO 2: Graduates have successful careers in computer science and engineering fields or will be able to successfully pursue advanced degrees.

PEO 3: Graduates will communicate effectively, work collaboratively and exhibit high levels of professionalism, moral and ethical responsibility.

PEO 4: Graduates will develop the ability to understand and analyze engineering issues in a broader perspective with ethical responsibility towards sustainable development.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PROGRAM OUTCOMES

<ul style="list-style-type: none"> • PO I: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
<ul style="list-style-type: none"> • PO II: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
<ul style="list-style-type: none"> • PO III: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
<ul style="list-style-type: none"> • PO IV: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
<ul style="list-style-type: none"> • PO V: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
<ul style="list-style-type: none"> • PO VI: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
<ul style="list-style-type: none"> • PO VII: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of and need for sustainable development.
<ul style="list-style-type: none"> • PO VIII: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
<ul style="list-style-type: none"> • PO IX: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
<ul style="list-style-type: none"> • PO X: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
<ul style="list-style-type: none"> • PO XI: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in

a team, to manage projects and in multidisciplinary environments.

- **PO XII: Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
PROGRAM SPECIFIC OUTCOMES

PSO1: The Computer Science and Engineering graduates are able to analyze, design, develop, test and apply management principles, mathematical foundations in the development of intelligent systems with computational solutions, make them to expert in designing the secure application and hardware prototype

PSO2: The graduating student will be analyze the contemporary research issues in different areas of computer science & engineering and explore research gaps, analyze and carry out research in the specialized/emerging areas.

PSO3: Develop their skills to solve problems in the broad area of programming concepts and appraise environmental and social issues with ethics and manage different projects in multi-disciplinary field to conducive in cultivating skills for successful career, entrepreneurship and higher studies.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Java Programming Lab

Prerequisites: A Course on Programming for Problem solving in C

Course Objectives: The course will help students

1. To write programs using abstract classes for solving real world problems using java collection framework.
2. To write multithreaded programs in Java.
3. To write GUI programs using swing controls in Java.
4. To introduce java compiler and eclipse platform.
5. To impart hands on experience with java programming.

Course Outcomes: Upon completion of this course, the students will be able to

CO1: Write Java program using packages, inheritance, polymorphism and interface.

CO2: Write Java application programs using OOP principles and proper program structuring.

CO3: Write handling exceptions and multithreaded programs in Java.

CO4: Write java programs for solving real world problems using java collection framework.

CO5: Write GUI programs and Applets in Java.

Note:

1. Use LINUX and MySQL for the Lab Experiments. Though not mandatory, encourage the use of Eclipse platform.
2. The list suggests the minimum program set. Hence, the concerned staff is requested to add more problems to the list as needed.

List of Experiments:

1. Installation of Java software, study of any integrated development environment, Use Eclipse or Net-bean 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 to n.
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
3. Write a Java program to multiply two given matrices.
4. Write Java program on use of inheritance, preventing inheritance using final, abstract classes.
5. 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 given shape.
6. Write a java program to implement method overloading and constructors overloading.
7. Write a java program to implement method overriding.
8. Write a Java program of currency converter (Dollar to INR, EURO to INR, Yen) using Interfaces.
9. Write a Java program to implement user defined exception handling.
10. Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num 2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, 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 dialog box.

11. Write a Java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.
12. Write a Java program that correctly implements the producer – consumer problem using the concept of inter thread communication.
13. 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.
14. 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.
15. Write a java program that connects to a database using JDBC and does add, delete, modify and retrieve operations.
16. Write a java program 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 shown.
17. Write a java program 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.
18. Write a java program to demonstrate the key event handlers
19. Write a java program 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. Handle any possible exception like divided by zero.
20. Develop an applet in Java that displays a simple message.
21. Develop an applet in Java that receives an integer in one text field, and computes its factorial Value and returns it in another text field, when the button named “Compute” is clicked.

Reference Books:

1. P. J. Deitel and H. M. Deitel, “Java for Programmers”, Pearson Education, 10th Edition.
2. Bruce Eckel, “Thinking in Java”, Pearson Education.
3. D. S. Malik and P. S. Nair, “Java Programming”, Cengage Learning.
4. Cay S. Horstmann and G Cornell, “Core Java, Volume 1”, Pearson, 9th edition.

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 to n.

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

```
import static java.lang.Math.*;
public class QuadraticEquationExample2
{
//defining a static method that finds the roots of the quadratic equation
static void calculateRoots(int a, int b, int c)
{
//comparing the value of a with 0, if a is 0 then the equation is not quadratic

if (a == 0)
{
System.out.println("The value of a cannot be 0.");
return;
}
//calculating discriminant (d)
int d = b * b - 4 * a * c;
double sqrtval = sqrt(abs(d));
if (d > 0)
{
System.out.println("The roots of the equation are real and different. \n");
System.out.println((double)(-b + sqrtval) / (2 * a) + "\n" + (double)(-b -
sqrtval) / (2 * a));
}
else if (d == 0)
{
System.out.println("The roots of the equation are real and same. \n");
```

```
System.out.println(-(double)b / (2 * a) + "\n" + -(double)b / (2 * a));
}
// executes if d < 0
else
{
System.out.println("The roots of the equation are complex and different. \n");
;System.out.println(-(double)b / (2 * a) + " + i" + sqrtval + "\n" + -
(double)b / (2 * a)+ " i" + sqrtval);
} }
//main method
public static void main(String args[])
{
//Output 1
int a = 1, b = 5, c = 2;
//calling function
calculateRoots(a, b, c);
//Output 2: int a=1, b=1, c=1
//Output 3: int a=1, b=-2, c=1
}
}
```

Output 1:

```
The roots of the equation are real and different.
-0.4384471871911697
-4.561552812808831
```

Output 2:

```
The roots of the equation are complex and different.
-0.5 + i1.7320508075688772
-0.5 - i1.7320508075688772
```

Output 3:

```
The roots of the equation are real and same.
1.0
1.0
```

3. Write a Java program to multiply two given matrices

AIM: 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

4. 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();
}
}
```

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

5. 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 given shape.

SOURCE CODE:

```
import java.util.*;
abstract class shape
{
int x,y;
abstract void area(double x,double y);
}
class Rectangle extends shape
{
void area(double x,double y)
{
System.out.println("Area of rectangle is :"+(x*y));
}
}
class Circle extends shape
{
void area(double x,double y)
{
System.out.println("Area of circle is :"+(3.14*x*x));
}
}
class Triangle extends shape
```

```
{  
void area(double x,double y)  
{  
System.out.println("Area of triangle is :"+(0.5*x*y));  
}  
}  
public class AbstactDDemo  
{  
public static void main(String[] args)  
{  
Rectangle r=new Rectangle();  
r.area(2,5);  
Circle c=new Circle();  
c.area(5,5);  
Triangle t=new Triangle();  
t.area(2,5);  
}  
}
```

Output:

Area of rectangle is :10.0

Area of circle is :78.5

Area of triangle is :5.0

6. Write a java program to implement method overloading and constructors overloading.

AIM: To implement method overloading and constructors overloading.

SOURCE CODE:

```
public class Student {  
    //instance variables of the class  
    int id;  
    String name;  
    Student(){  
        System.out.println("this a default constructor");  
    }  
    Student(int i, String n){  
        id = i;  
        name = n;  
    }  
    public static void main(String[] args) {  
        //object creation  
        Student s = new Student();  
        System.out.println("\nDefault Constructor values: \n");  
        System.out.println("Student Id : "+s.id + "\nStudent Name : "+s.name);  
        System.out.println("\nParameterized Constructor values: \n");  
        Student student = new Student(10, "David");
```

```
System.out.println("Student Id : "+student.id + "\nStudent Name : "+student.name);  
} }
```

Output:

this a default constructor

Default Constructor values:

Student Id : 0

Student Name : null

Parameterized Constructor values:

Student Id : 10

Student Name : David

7. Write a java program to implement method overriding.

AIM: To implement method overriding.

SOURCE CODE:

//Java Program to demonstrate the real scenario of Java Method Overriding

//where three classes are overriding the method of a parent class.

//Creating a parent class.

```
class Bank{  
int getRateOfInterest(){return 0;}  
}
```

//Creating child classes.

```
class SBI extends Bank{  
int getRateOfInterest(){return 8;}  
}
```

```
class ICICI extends Bank{  
int getRateOfInterest(){return 7;}  
}
```

```
class AXIS extends Bank{  
int getRateOfInterest(){return 9;}  
}
```

//Test class to create objects and call the methods

```
class Test2{  
public static void main(String args[]){  
SBI s=new SBI();  
ICICI i=new ICICI();  
AXIS a=new AXIS();  
System.out.println("SBI Rate of Interest: "+s.getRateOfInterest());  
System.out.println("ICICI Rate of Interest: "+i.getRateOfInterest());  
System.out.println("AXIS Rate of Interest: "+a.getRateOfInterest());  
}  
}
```

Output:

SBI Rate of Interest: 8

ICICI Rate of Interest: 7

AXIS Rate of Interest: 9

8. Write a Java program of 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; returnEuros;
}
public double YenToINR(double Yens)
{
    double INR = 0;INR = Yens * ER; return INR;
}
public double INRToYen(double INR)
{
    double Yens =0; Yens = INR
    / ER; returnYens;
}
}
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 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)//outerSwitch
{
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.INRTToDollar(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.INRTToEuro(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.INRTToYen(inr)+" Yens");
break;
}
System.out.println("Do You want to go to Currency ConversionMenu?(y/n)
"); ans = input.next().charAt(0);
}while(ans=='y'); break;
case 2:System.out.println("\tDistanceConversion"); 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.areconverted miles"+dc.KMToMile(km)+" Miles");
break;
}
System.out.println("Do You want to go to Distance ConversionMenu?(y/n)
"); ans = input.next().charAt(0);
}while(ans=='y'); break;
case 3:System.out.println("\tTimeConversion"); 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 toMinutes:"); 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 tohours:");
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 ConversionMenu?(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 MainMenu?(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 ConversionMenu 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

9. Write a Java program to implement user defined exception handling.

AIM: Java program to implement user defined Exception handling. Source code:

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 exceptionsclass 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 blockOutput string=custom

10. Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num 2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, 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 dialog box.

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)
{
JOptionPane.showMessageDialog(null,"Divide by zero");
}
```

```
catch(NumberFormatException b)
{
OptionPane.showMessageDialog(null,"NumberFormateException");
}
}
}
```

Output: divide bye zero Number format exception

```
());
t3.start();
}
}
```

OUTPUT:
GOOD MORNING

11. Write a Java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number

AIM:Java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.

SOURCE CODE:

```
import java.util.*; // class for Even Number
class EvenNum implements Runnable {
    public int a;
    public EvenNum(int a) {
        this.a = a;
    }
    public void run() {
        System.out.println("The Thread "+ a +" is EVEN and Square of " + a + " is : " + a *
a);
    } } // class for Odd Number
class OddNum implements Runnable {
    public int a;
    public OddNum(int a) {
        this.a = a;
    }
    public void run() {
        System.out.println("The Thread "+ a +" is ODD and Cube of " + a + " is: " + a * a *
a);
    } } // class to generate random number
class RandomNumGenerator extends Thread {
```

```

public void run() {
    int n = 0;
    Random rand = new Random();
    try {
        for (int i = 0; i < 10; i++) {
            n = rand.nextInt(20);
            System.out.println("Generated Number is " + n);
            // check if random number is even or odd
            if (n % 2 == 0) {
                Thread thread1 = new Thread(new EvenNum(n));
                thread1.start();
            }
            else {
                Thread thread2 = new Thread(new OddNum(n));
                thread2.start();
            }
            // thread wait for 1 second
            Thread.sleep(1000);
            System.out.println("-----");
        }
    }
    catch (Exception ex) {
        System.out.println(ex.getMessage());
    }
}
} // Driver class
public class MultiThreadRandOddEven {
    public static void main(String[] args) {
        RandomNumGenerator rand_num = new RandomNumGenerator();
        rand_num.start();
    }
}

```

Output:



```
C:\StudyGlance\Java_Lab_Programs\W5>javac MultiThreadRandOddEven.java
C:\StudyGlance\Java_Lab_Programs\W5>java MultiThreadRandOddEven
Generated Number is 8
The Thread 8 is EVEN and Square of 8 is : 64
-----
Generated Number is 2
The Thread 2 is EVEN and Square of 2 is : 4
-----
Generated Number is 16
The Thread 16 is EVEN and Square of 16 is : 256
-----
Generated Number is 6
The Thread 6 is EVEN and Square of 6 is : 36
-----
Generated Number is 2
The Thread 2 is EVEN and Square of 2 is : 4
-----
C:\StudyGlance\Java_Lab_Programs\W5>javac MultiThreadRandOddEven.java
C:\StudyGlance\Java_Lab_Programs\W5>java MultiThreadRandOddEven
Generated Number is 1
The Thread 1 is ODD and Cube of 1 is: 1
-----
Generated Number is 4
The Thread 4 is EVEN and Square of 4 is : 16
-----
Generated Number is 1
The Thread 1 is ODD and Cube of 1 is: 1
-----
Generated Number is 4
The Thread 4 is EVEN and Square of 4 is : 16
-----
Generated Number is 16
The Thread 16 is EVEN and Square of 16 is : 256
-----
Generated Number is 8
The Thread 8 is EVEN and Square of 8 is : 64
-----
Generated Number is 0
The Thread 0 is EVEN and Square of 0 is : 0
-----
Generated Number is 12
The Thread 12 is EVEN and Square of 12 is : 144
-----
```

12. Write a Java program that correctly implements the producer – consumer problem using the concept of inter thread communication.

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

SOURCE CODE:

```
class Q
{
    int n; boolean valueSet=false; synchronized int get()
    {
        if(!valueSet)
        try
        {
            wait();
        }
        catch(InterruptedException e)
        {
            System.out.println("Interrupted Exception 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("Interrupted Exception 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

13. 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
spinachkale

14. 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());
}
}
}
```

15. Write a java program that connects to a database using JDBC and does add, delete, modify and retrieve operations

AIM: Java program that connects to a database using JDBC and does add, delete, modify and retrieve operations.

SOURCE CODE:

Connecting to the Database

```
import java.sql.*;
public class connect
{
public static void main(String args[])
{
try
{
Class.forName("oracle.jdbc.driver.OracleDriver");
Connection con = DriverManager.getConnection(
"jdbc:oracle:thin:@localhost:1521:orcl", "login1", "pwd1");
if (con != null)
System.out.println("Connected");
else
System.out.println("Not Connected");
con.close();
}
catch(Exception e)
{
System.out.println(e);
}
}
}
```

Output-

Connected

//Implementing Insert Statement

```
import java.sql.*;
public class insert1
{
public static void main(String args[])
{
String id = "id1";
String pwd = "pwd1";
String fullname = "geeks for geeks";
String email = "geeks@geeks.org";
try
{
Class.forName("oracle.jdbc.driver.OracleDriver");
Connection con = DriverManager.getConnection("
jdbc:oracle:thin:@localhost:1521:orcl", "login1", "pwd1");
Statement stmt = con.createStatement();
String q1 = "insert into userid values(" + id + ", " + pwd +
", " + fullname + ", " + email + ")";
int x = stmt.executeUpdate(q1);
if (x > 0)
System.out.println("Successfully Inserted");
else
System.out.println("Insert Failed");
con.close();
}
catch(Exception e)
{
System.out.println(e);
}
}
}
```

Output-

Successfully Registered

//Implementing Update Statement

```
import java.sql.*;
public class update1
{
public static void main(String args[])
{
String id = "id1";
String pwd = "pwd1";
String newPwd = "newpwd";
try
{
Class.forName("oracle.jdbc.driver.OracleDriver");
Connection con = DriverManager.getConnection("
jdbc:oracle:thin:@localhost:1521:orcl", "login1", "pwd1");
Statement stmt = con.createStatement();
// Updating database
String q1 = "UPDATE userid set pwd = " + newPwd +
" WHERE id = " + id+ " AND pwd = " + pwd + """;
int x = stmt.executeUpdate(q1);
if (x > 0)
System.out.println("Password Successfully Updated");
else
System.out.println("ERROR OCCURED :(");
con.close();
}
catch(Exception e)
{
System.out.println(e);
}
}
}
```

Output-

Password Successfully Updated

//Implementing Delete Statement

```
import java.sql.*;
public class delete
{
public static void main(String args[])
{
String id = "id2";
String pwd = "pwd2";
try
{
Class.forName("oracle.jdbc.driver.OracleDriver");
Connection con = DriverManager.getConnection("
jdbc:oracle:thin:@localhost:1521:orcl", "login1", "pwd1");
Statement stmt = con.createStatement();
// Deleting from database
String q1 = "DELETE from userid WHERE id = " + id +
" AND pwd = " + pwd + """;
int x = stmt.executeUpdate(q1);
if (x > 0)
System.out.println("One User Successfully Deleted");
else
System.out.println("ERROR OCCURED :(");
con.close();
}
catch(Exception e)
{
System.out.println(e);
}
}
}
```

Output-

One User Successfully Deleted

//Implementing Select Statement

```
import java.sql.*;
public class select
{
public static void main(String args[])
{
String id = "id1";
String pwd = "pwd1";
try
{
Class.forName("oracle.jdbc.driver.OracleDriver");
Connection con = DriverManager.getConnection("
jdbc:oracle:thin:@localhost:1521:orcl", "login1", "pwd1");
Statement stmt = con.createStatement();
String q1 = "select * from userid WHERE id = " + id +
" AND pwd = " + pwd + """;
ResultSet rs = stmt.executeQuery(q1);
if (rs.next())
{
System.out.println("User-Id : " + rs.getString(1));
System.out.println("Full Name : " + rs.getString(3));
System.out.println("E-mail : " + rs.getString(4));
}
else
{
System.out.println("No such user id is already registered");
}
con.close();
}
catch(Exception e)
{
System.out.println(e);
}
}
```

}

Output-

User-Id: id1

Full Name: Dataflair

16. Write a java program 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 shown.

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:



17. Write a java program 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.

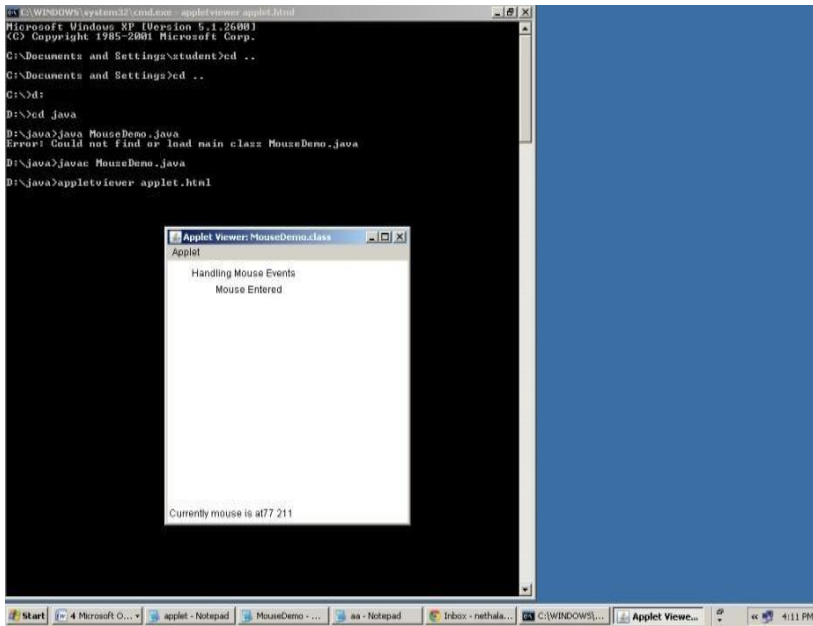
AIM: Develop a Java application that handles all mouse events and shows the event name at the center of the window when a mouse event is fired. Use adapterclasses.

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;
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:



18. Write a java program 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");
}
public void keyReleased(KeyEvent k)
{
    showStatus("KeyRealesed");
}
}
```

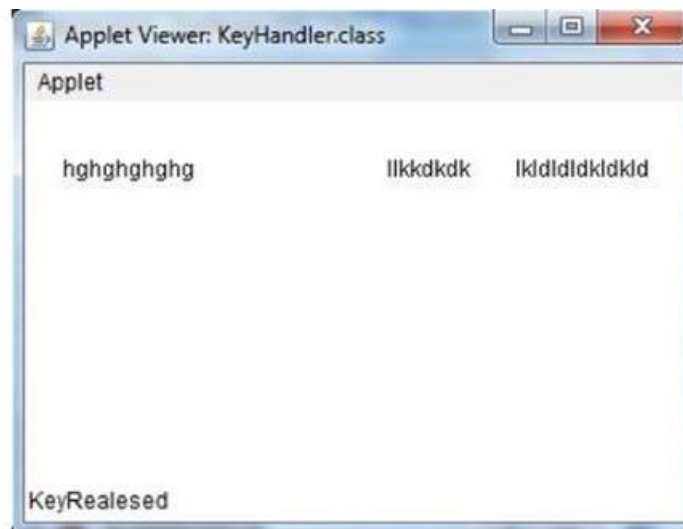
```
public void keyTyped(KeyEvent k)

{
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:



19. Write a java program 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. Handle any possible exception like divided by zero

AIM: Java program 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. Handle any possible exception like divided by zero

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.CENTER);
}
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");

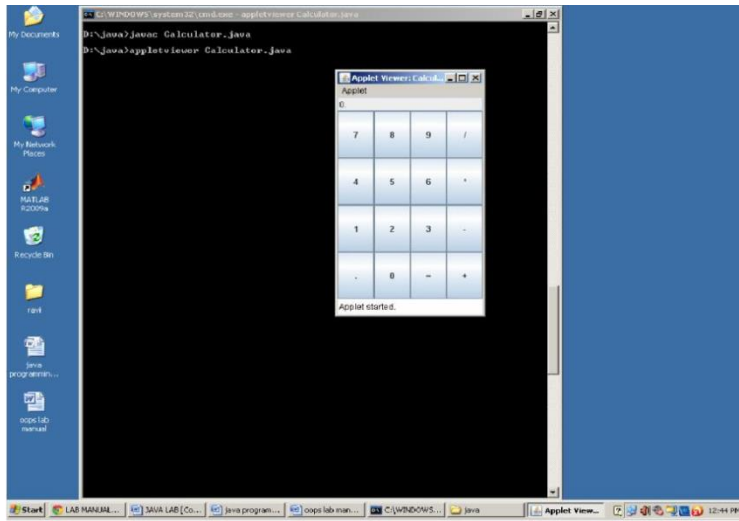
else
if(ae.getSource()==dot)
{
  if(((result.getText()).indexOf(".")==-1)
    result.setText(result.getText()+".");
}
else if(ae.getSource()==minus)
{
  preRes=Double.parseDouble(result.getText()); lastCommand="-";
  result.setText("0");
}
else if(ae.getSource()==div)
{
  preRes=Double.parseDouble(result.getText());lastCommand="/";
  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");
}

```

```
}  
}
```

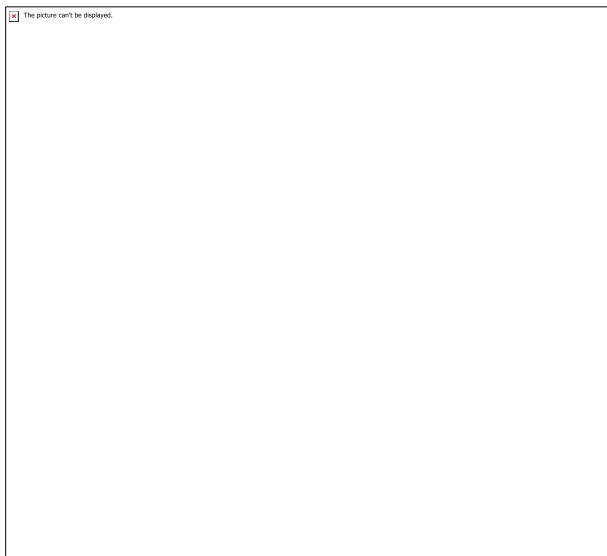
OUTPUT:



20. Develop an applet in Java that displays a simple message

```
import java.awt.*;  
import java.applet.*;  
  
/*  
  
<applet code="Demo" width=300 height=200>  
  
</applet>  
  
*/  
  
public class Demo extends Applet {  
    public void paint(Graphics g) {  
        g.drawString("Welcome to Goeduhub Technologies", 20, 20);  
    }  
}
```

Output



21. Develop an applet in Java that receives an integer in one text field, and computes its factorial Value and returns it in another text field, when the button named “Compute” is clicked.

AIM: An applet in Java that receives an integer in one text field, and computes its factorial Value and returns it in another text field, when the button named “Compute” is clicked.

SOURCE CODE:

```
/** Develop applet to find factorial of the given number */
import java.awt.*;
import java.applet.*;
import java.awt.event.*; /*<applet code="FactorialApplet" width=500 height=250>
</applet>*/
public class FactorialApplet extends Applet implements ActionListener {
    Label L1,L2;
    TextField T1,T2;
    Button B1;
    public void init() {
        L1=new Label("Enter any Number : ");
        add(L1);
        T1=new TextField(10);
        add(T1);
        L2=new Label("Factorial of Num : ");
        add(L2);
        T2=new TextField(10);
        add(T2);
        B1=new Button("Compute");
        add(B1);
        B1.addActionListener(this);
    }
    public void actionPerformed(ActionEvent e) {
        if(e.getSource()==B1)
        {
```

```
int value=Integer.parseInt(T1.getText());
int fact=factorial(value);
T2.setText(String.valueOf(fact));
}
}
int factorial(int n) {
    if(n==0)
        return 1;
    else
        return n*factorial(n-1);
}}
```

Output:

```
C:\StudyGlance\Java_Lab_Programs\W3>javac FactorialApplet.java
C:\StudyGlance\Java_Lab_Programs\W3>appletviewer FactorialApplet.java
```

