

**B.M.S COLLEGE OF ENGINEERING BENGALURU**  
Autonomous Institute, Affiliated to VTU



## **JAVA LAB REPORT**

### **OBJECT ORIENTED JAVA PROGRAMMING**

Bachelor of Engineering  
in  
Computer Science and Engineering

*Submitted by:*

**Manjari Joshi**  
**1BM22CS146**

*Submitted to:*

**Vikranth BM**  
**Assistant Professor**  
**BMS College of Engineering**

Develop 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. If the discriminant  $b^2 - 4ac$  is negative, display a message stating that there are no real solutions.

CODE:

```
import java.util.Scanner;
class quadratic
{
    public static void main (String args[])
    {
        int a;
        int b;
        int c;
        double d;
        Scanner s1=new Scanner(System.in);
        System.out.println("Enter coefficients of quadratic equation");
        a=s1.nextInt();
        b=s1.nextInt();
        c=s1.nextInt();
        d=(b*b)-(4*a*c);
        if(d>0.0)
        {
            double r1=(-b) + Math.pow(d,0.5)/(2.0*a);
            double r2=(-b) - Math.pow(d,0.5)/(2.0*a);
            System.out.println("Roots are real and distinct");
            System.out.println("Root1="+r1+"Root2="+r2);
        }
        else if(d==0.0)
        {
            double r3=(-b)/(2.0*a);
            System.out.println("Roots are real and equal");
            System.out.println("Roots="+r3);
        }
        else
        {
            System.out.println("Roots are imaginary");
        }
    }
}
```

OUTPUT:

```
PS C:\Users\joshi> cd OneDrive
PS C:\Users\joshi\OneDrive> cd Desktop
PS C:\Users\joshi\OneDrive\Desktop> javac quadratic.java
PS C:\Users\joshi\OneDrive\Desktop> java quadratic
Enter coefficients of quadratic equation
1 -2 1
Roots are real and equal
Roots=1.0
PS C:\Users\joshi\OneDrive\Desktop> java quadratic
Enter coefficients of quadratic equation
2 5 7
Roots are imaginary
PS C:\Users\joshi\OneDrive\Desktop> java quadratic
Enter coefficients of quadratic equation
1 3 7
Roots are imaginary
PS C:\Users\joshi\OneDrive\Desktop> java quadratic 1 1 1
Enter coefficients of quadratic equation
1 1 1
Roots are imaginary
PS C:\Users\joshi\OneDrive\Desktop> java quadratic
Enter coefficients of quadratic equation
-2 2 1
Roots are real and distinct
Root1=-0.3660254037844386Root2=1.3660254037844386
```

Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student

CODE:

```
import java.util.Scanner;

class Details {
    int usn;
    String name;
    int[] marks = new int[8];
    int[] credit = new int[8];
    int[] credit_points = new int[8];
    Scanner s1 = new Scanner(System.in);

    void acceptDetails() {
        System.out.println("Enter student usn:");
        usn = s1.nextInt();
        System.out.println("Enter student name:");
        name = s1.next();
        System.out.println("Enter marks in order of credits");
        for (int i = 0; i < 8; i++) {
            marks[i] = s1.nextInt();
        }
        System.out.println("Enter order of credits ");
        for (int i = 0; i < 8; i++) {
            credit[i] = s1.nextInt();
        }
    }

    void calculate() {
        for (int i = 0; i < 8; i++) {
            if (marks[i] >= 90) {
                credit_points[i] = 10 * credit[i];
            } else if (marks[i] >= 80) {
                credit_points[i] = 9 * credit[i];
            } else if (marks[i] >= 70) {
                credit_points[i] = 8 * credit[i];
            } else if (marks[i] >= 60) {
                credit_points[i] = 7 * credit[i];
            } else if (marks[i] >= 50) {
                credit_points[i] = 6 * credit[i];
            }
        }
    }
}
```

```

        } else if (marks[i] >= 40) {
            credit_points[i] = 5 * credit[i];
        }
    }
    int sum = 0;
    int count = 0;
    double SGPA;
    for (int j = 0; j < 8; j++) {
        sum = sum + credit_points[j];
        count=count+credit[j];
    }
    SGPA = sum/count;
    System.out.println("SGPA is : " + SGPA);
}
}

class SGPA {
    public static void main(String[] args) {
        Details d = new Details();
        d.acceptDetails();
        d.calculate();
    }
}

```

OUTPUT:

```

PS C:\Users\joshi\OneDrive\Desktop> javac SGPA.java
PS C:\Users\joshi\OneDrive\Desktop> java SGPA
Enter student usn:
146
Enter student name:
Manjari
Enter marks in order of credits
97 97 95 89 87 90 92 98
Enter order of credits
4 4 3 3 3 1 1 1
SGPA is : 9.0

```

Create a class Book which contains four members: name, author, price, num\_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a `toString()` method that could display the complete details of the book. Develop a Java program to create n book objects.

CODE:

```
import java.util.Scanner;

class book {
    String name;
    String author;
    double price;
    int pages;

    book(String name, String author, double price, int pages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.pages = pages;
    }

    void setDetails() {
        Scanner S = new Scanner(System.in);
        System.out.println("Enter name of books:");
        this.name = S.nextLine();
        System.out.println("Author:");
        this.author = S.nextLine();
        System.out.println("Enter price");
        this.price = S.nextDouble();
        System.out.println("Enter no.of.pages");
        this.pages = S.nextInt();
    }

    void getDetails() {
        System.out.println("Book name:" + this.name);
        System.out.println("Author:" + this.author);
        System.out.println("Price:$" + this.price);
        System.out.println("Number of pages" + this.pages);
    }

    public String toString() {
```

```
        return "Book Details:\n" + "Name:" + name + "\n" + "Author:" + author + "\n" + "Prices : $" +
price + "\n"
        + "Number of pages:" + pages;
    }
}

class BookDemo2 {
    public static void main(String args[]) {
        Scanner S1 = new Scanner(System.in);
        System.out.println("Enter number of books");
        int n = S1.nextInt();
        book[] b = new book[n];
        for (int i = 0; i < n; i++) {
            System.out.println("\n Enter details for book" + (i + 1) + ":");

            b[i] = new book(" ", " ", 0.0, 0);
            b[i].setDetails();
        }
        System.out.println("\n Details of books");
        for (int i = 0; i < n; i++) {
            System.out.println("\n Book" + (i + 1) + ":");

            b[i].getDetails();
        }
        System.out.println("\n Complete details of all books:");
        for (int i = 0; i < n; i++) {
            System.out.println("\n Book" + (i + 1) + ":\n" + b[i]);
        }
    }
}
```

## OUTPUT:

```
PS C:\Users\joshi\OneDrive\Desktop> javac BookDemo2.java
PS C:\Users\joshi\OneDrive\Desktop> java BookDemo2
Enter number of books
2

    Enter details for book1:
Enter name of books:
rich dad poor dad
Author:
robert
Enter price
45
Enter no.of.pages
453

    Enter details for book2:
Enter name of books:
xyz
Author:
abc
Enter price
67
Enter no.of.pages
890

    Details of books

    Book1:
Book name:rich dad poor dad
Author:robert
Price:$45.0
Number of pages453

    Book2:
Book name:xyz
Author:abc
Price:$67.0
Number of pages890

    Complete details of all books:

    Book1:
Book Details:
Name:rich dad poor dad
Author:robert
Prices : $45.0
Number of pages:453

    Book2:
Book Details:
Name:xyz
Author:abc
Prices : $67.0
Number of pages:890
PS C:\Users\joshi\OneDrive\Desktop>
```

Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea( ) that prints the area of the given shape.

CODE:

```
import java.util.Scanner;
abstract class shape
{
    int a;
    int b;
    abstract void printArea();
}
class rect extends shape
{
    void printArea()
    {
        System.out.println("Area of rectangle is:"+ (a*b));
    }
}
class tri extends shape
{
    void printArea()
    {
        System.out.println("Area of triangle is:"+(0.5*a*b));
    }
}
class cir extends shape
{
    void printArea()
    {
        System.out.println("Area of circle is:"+(314*a*a));
    }
}
class AbstractDemo
{
    public static void main (String args[])
    {
        Scanner s1=new Scanner (System.in);
        System.out.println("Press:\n 1.Rectangle \n 2.Triangle \n 3.Circle");
        int choice;
        choice=s1.nextInt();
        switch(choice)
        {
```

```

case 1: System.out.println("Enter l and b of Rectangle");
int l=s1.nextInt();
int br=s1.nextInt();
rect r=new rect();
r.a=l;
r.b=br;
r.printArea();
break;
case 2: System.out.println("Enter l and b of Triangle");
int h=s1.nextInt();
int bre=s1.nextInt();
tri t=new tri();
t.a=h;
t.b=bre;
t.printArea();
break;
case 3: System.out.println("Enter r of Circle");
int rad=s1.nextInt();
cir c =new cir();
c.a=rad;
c.printArea();
break;
default:System.out.println("Enter valid choice");
}
}
}
}

```

OUTPUT:

```

PS C:\Users\joshi\OneDrive\Desktop> javac AbstractDemo.java
PS C:\Users\joshi\OneDrive\Desktop> java AbstractDemo
Press:
1.Rectangle
2.Triangle
3.Circle
1
Enter l and b of Rectangle
4 4
Area of rectangle is:16

```

### Program 5:

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks: a) Accept deposit from customer and update the balance. b) Display the balance. c) Compute and deposit interest d) Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance.

### CODE:

```
import java.util.Scanner;

class Account {

    String customerName;
    int accountNumber;
    String accountType;
    double balance;

    public Account(String customerName, int accountNumber, String accountType, double balance) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = balance;
    }
}
```

```
public void deposit(double amount) {  
    balance += amount;  
    System.out.println("Deposit of $" + amount + " successful.");  
}  
  
public void displayBalance() {  
    System.out.println("Balance: $" + balance);  
}  
}  
  
class CurAcct extends Account {  
    double minBalance;  
    double penaltyCharge;  
  
    public CurAcct(String customerName, int accountNumber, double balance) {  
        super(customerName, accountNumber, "Current", balance);  
        minBalance = 1000;  
        penaltyCharge = 50;  
    }  
  
    public void withdraw(double amount) {  
        if (balance - amount >= minBalance) {  
            balance -= amount;  
            System.out.println("Withdrawal of $" + amount + " successful.");  
        } else {  
            System.out.println("Insufficient balance. Withdrawal failed.");  
        }  
    }  
}
```

```
    }

}

class SavAcct extends Account {

    double interestRate;

    public SavAcct(String customerName, int accountNumber, double balance) {

        super(customerName, accountNumber, "Savings", balance);

        interestRate = 0.05; // 5% interest rate for savings account
    }

    public void depositInterest() {

        double interest = balance * interestRate;

        balance += interest;

        System.out.println("Interest of $" + interest + " deposited.");
    }

    public void withdraw(double amount) {

        if (balance - amount >= 0) {

            balance -= amount;

            System.out.println("Withdrawal of $" + amount + " successful.");
        } else {

            System.out.println("Insufficient balance. Withdrawal failed.");
        }
    }
}

public class BankDemo {

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

```
Scanner scanner = new Scanner(System.in);

System.out.println("Enter details for current account:");

System.out.print("Customer Name: ");

String currentCustomerName = scanner.nextLine();

System.out.print("Account Number: ");

int currentAccountNumber = Integer.parseInt(scanner.nextLine());

System.out.print("Initial Balance: $");

double currentInitialBalance = Double.parseDouble(scanner.nextLine());

CurAcct currentAccount = new CurAcct(currentCustomerName, currentAccountNumber,
currentInitialBalance);

System.out.println("\nEnter details for savings account:");

System.out.print("Customer Name: ");

String savingsCustomerName = scanner.nextLine();

System.out.print("Account Number: ");

int savingsAccountNumber = Integer.parseInt(scanner.nextLine());

System.out.print("Initial Balance: $");

double savingsInitialBalance = Double.parseDouble(scanner.nextLine());

SavAcct savingsAccount = new SavAcct(savingsCustomerName, savingsAccountNumber,
savingsInitialBalance);

System.out.print("\nEnter deposit amount for current account: $");

double depositAmountCurrent = Double.parseDouble(scanner.nextLine());

currentAccount.deposit(depositAmountCurrent);

System.out.print("Enter withdrawal amount for current account: $");

double withdrawAmountCurrent = Double.parseDouble(scanner.nextLine());

currentAccount.withdraw(withdrawAmountCurrent);
```

```
currentAccount.displayBalance();

System.out.print("\nEnter deposit amount for savings account: $");

double depositAmountSavings = Double.parseDouble(scanner.nextLine());

savingsAccount.deposit(depositAmountSavings);

savingsAccount.depositInterest();

savingsAccount.displayBalance();

System.out.print("Enter withdrawal amount for savings account: $");

double withdrawAmountSavings = Double.parseDouble(scanner.nextLine());

savingsAccount.withdraw(withdrawAmountSavings);

savingsAccount.displayBalance();

}

}
```

OUTPUT:

```
C:\Users\Admin\Desktop>javac BankDemo.java

C:\Users\Admin\Desktop>java BankDemo
Enter details for current account:
Customer Name: manj
Account Number: 123
Initial Balance: $500

Enter details for savings account:
Customer Name: manj
Account Number: 124
Initial Balance: $300

Enter deposit amount for current account: $900
Deposit of $900.0 successful.
Enter withdrawal amount for current account: $1500
Insufficient balance. Withdrawal failed.
Balance: $1400.0

Enter deposit amount for savings account: $300
Deposit of $300.0 successful.
Interest of $30.0 deposited.
Balance: $630.0
Enter withdrawal amount for savings account: $500
Withdrawal of $500.0 successful.
Balance: $130.0
```

Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

CODE:

**Pack/CIE/Internal.Java**

```
package CIE;
```

```
import java.util.Scanner;
```

```
public class Internal extends CIE.Student{
```

```
    public int m[] = new int[5];
```

```
    CIE.Student student = new CIE.Student();
```

```
    public void accept(){
```

```
        student.accept();
```

```
        Scanner s1 = new Scanner(System.in);
```

```
        System.out.println("Enter Internal Marks:");
```

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

```
            m[i] = s1.nextInt();
```

```
        }
```

```
}
```

```
    public void display(){
```

```
student.display();

for(int i=0;i<5;i++){
    System.out.println("Marks of sub" + (i+1) + " = " + m[i]);
}

}
```

#### Pack/CIE/Student.Java

```
package CIE;

import java.util.Scanner;

public class Student{

    public String usn;
    public String name;
    public int sem;

    public void accept(){

        Scanner s = new Scanner(System.in);
        System.out.println("Enter Name:");
        this.name = s.nextLine();
        System.out.println("Enter usn:");
        this.usn = s.nextLine();
    }
}
```

```
System.out.println("Enter sem");
this.sem = s.nextInt();

}

public void display(){
System.out.println("Name: " + this.name + "\nUSN: " + this.usn + "\nSem: " + this.sem);
}

}
```

#### Pack/SEE/External.Java

```
package SEE;

import java.util.Scanner;

import CIE.Internal;
import CIE.Student;

public class External extends CIE.Student{
    public int x[] = new int[5];
    public void accept(){
        Scanner s2 = new Scanner(System.in);
        System.out.println("Enter External Marks.");
        for(int i=0;i<5;i++){

```

```
x[i] = s2.nextInt();

}

}

public void display(){

super.display();

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

System.out.println("Marks of sub" + (i+1) + " = " + x[i]);

}

}

}
```

### Pack/Final.Java

```
import java.util.Scanner;

import CIE.Student;
import CIE.Internal;
import SEE.External;

public class Final{

    public static void main(String[] args) {

        Scanner n = new Scanner(System.in);
```

```
System.out.println("Enter n:");
int y = n.nextInt();

CIE.Internal[] c1 = new CIE.Internal[y];
SEE.External[] c2 = new SEE.External[y];

for(int i=0;i<y;i++){
    c1[i] = new CIE.Internal();
    c2[i] = new SEE.External();

    c1[i].accept();
    c2[i].accept();

    // c1[i].accept();c2[i].accept();
    c1[i].display();c2[i].display();

    for(int j=0;j<5;j++){
        double calc = c1[i].m[j]+((c2[i].x[j])/2);
        System.out.println("Final marks of sub["++(i+1)+"]= "+calc);
    }
}
}
```

OUTPUT:

```
Enter n:  
5  
Enter Name:  
Aish  
Enter usn:  
123  
Enter sem  
3  
Enter Internal Marks:  
50 49 47 39 35  
Enter External Marks:  
100 100 100 100 100  
Name: Aish  
USN: 123  
Sem: 3  
Marks of sub1 = 50  
Marks of sub2 = 49  
Marks of sub3 = 47  
Marks of sub4 = 39  
Marks of sub5 = 35  
Name: null  
USN: null  
Sem: 0  
Marks of sub1 = 100  
Marks of sub2 = 100  
Marks of sub3 = 100  
Marks of sub4 = 100  
Marks of sub5 = 100  
Final marks of sub[1]= 100.0  
Final marks of sub[1]= 99.0  
Final marks of sub[1]= 97.0  
Final marks of sub[1]= 89.0  
Final marks of sub[1]= 85.0
```

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age<0. In Son class, implement a constructor that cases both father and son’s age and throws an exception if son’s age is >=father’s age.

CODE:

```
import java.util.Scanner;
class WrongAgeException extends Exception
{
    public WrongAgeException(String message)
    {
        super(message);
    }
}

class father
{
    private int FatherAge;
    public father(int age) throws WrongAgeException
    {
        if(age<0)
        {
            throw new WrongAgeException("Age cannot be Negative");
        }
        this.FatherAge=age;
    }
}
class son extends father
{
    private int sonAge;
    public son(int FatherAge, int sonAge) throws WrongAgeException
    {
        super(FatherAge);
        if(sonAge >= FatherAge)
        {
            throw new WrongAgeException("son's age should be less than Father's age");
        }
        this.sonAge=sonAge;
        System.out.println("Father's Age:"+FatherAge);
        System.out.println("son's Age:"+sonAge);
    }
}
```

```
public class ExceptionDemo
{
    public static void main(String[] args)
    {
        Scanner Scanner= new Scanner(System.in);
        try
        {
            System.out.println("Enter Father's age:");
            int FatherAge=Scanner.nextInt();
            father father=new father(FatherAge);
            System.out.println("Enter son's Age:");
            int sonAge= Scanner.nextInt();
            son son= new son(FatherAge,sonAge);
        }
        catch(WrongAgeException e)
        {
            System.out.println("Exception:"+e.getMessage());
        }
    }
}
```

OUTPUT:

```
PS C:\Users\joshi\OneDrive\Desktop> javac ExceptionDemo.java
PS C:\Users\joshi\OneDrive\Desktop> java ExceptionDemo
Enter Father's age:
-1
Exception:Age cannot be Negative
PS C:\Users\joshi\OneDrive\Desktop> java ExceptionDemo
Enter Father's age:
35
Enter son's Age:
45
Exception:son's age should be less than Father's age
```

Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds.

CODE:

```
class NewThread1 implements Runnable
{
    Thread t1;
    NewThread1()
    {
        t1 = new Thread(this,"Thread1");
        System.out.println("CT:"+t1);
        t1.start();
    }
    public void run()
    {
        try
        {
            for (int n=5; n>0;n--)
            {

                System.out.println("BMS College of Engineering");

                Thread.sleep (10000);
            }
        }
        catch( InterruptedException ie)
        {
            System.out.println("Thread1 interrupted");
        }

        System.out.println ("Thread 1 quitting");
    }
}
class NewThread2 implements Runnable
{
    Thread t2;
    NewThread2()
    {
        t2=new Thread (this,"Thread2");
        System.out.println("CT:"+t2);
        t2.start();
    }
    public void run()
```

```

{
try
{
    for (int n=5; n>0;n--)
    {
        System.out.println("cse");
        Thread.sleep(2000);
    }
}
catch(InterruptedException ie)
{
    System.out.println("Thread 2 Interrupted");
}
System.out.println ("Thread 2 quitting");
}

}

class MainThread {
public static void main(String args[])
{
new NewThread1();
new NewThread2();

try {
    Thread.sleep(40000);
    System.out.println("MainThread is awake\n");
} catch (InterruptedException ie) {
    System.out.println("MainThread Interrupted");
}
System.out.println("MainThread exiting");
}
}

```

OUTPUT:

```
PS C:\Users\joshi\OneDrive\Desktop> javac MainThread.java
PS C:\Users\joshi\OneDrive\Desktop> java MainThread
CT:Thread[#21,Thread1,5,main]
BMS College of Engineering
CT:Thread[#22,Thread2,5,main]
cse
cse
cse
cse
cse
BMS College of Engineering
Thread 2 quitting
BMS College of Engineering
BMS College of Engineering
MainThread is awake

MainThread exiting
BMS College of Engineering
Thread I quitting
```

Write a 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 Num2 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 NumberFormatException. If Num2 were Zero, the program would throw an ArithmeticException. Display the exception in a message dialog box.

CODE:

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

public class DivisionMain1 extends Frame implements ActionListener {
    TextField num1, num2;
    Button divideButton;
    Label resultLabel;

    public DivisionMain1() {
        setLayout(new FlowLayout());

        divideButton = new Button("Divide");
        Label number1 = new Label("Number 1:", Label.RIGHT);
        Label number2 = new Label("Number 2:", Label.RIGHT);
        num1 = new TextField(5);
        num2 = new TextField(5);
        resultLabel = new Label("Result:", Label.RIGHT);

        add(number1);
        add(num1);
        add(number2);
        add(num2);
        add(divideButton);
        add(resultLabel);

        divideButton.addActionListener(this);

        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent we) {
                System.exit(0);
            }
        });
    }

    public void actionPerformed(ActionEvent ae) {
        try {
```

```

int n1 = Integer.parseInt(num1.getText());
int n2 = Integer.parseInt(num2.getText());
if (n2 == 0) {
    throw new ArithmeticException("Cannot divide by zero");
}
int result = n1 / n2;
resultLabel.setText("Result: " + result);
} catch (NumberFormatException e1) {
    showErrorDialog("Number Format Exception: Please enter integers only");
} catch (ArithmeticException e2) {
    showErrorDialog("Arithmetic Exception: " + e2.getMessage());
}
}

private void showErrorDialog(String message) {
    Dialog dialog = new Dialog(this, "Error", true);
    dialog.setLayout(new FlowLayout());
    Label label = new Label(message);
    Button okButton = new Button("OK");
    okButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            dialog.dispose();
        }
    });
    dialog.add(label);
    dialog.add(okButton);
    dialog.setSize(300, 100);
    dialog.setVisible(true);
}

public static void main(String[] args) {
    DivisionMain1 divisionMain = new DivisionMain1();
    divisionMain.setSize(new Dimension(400, 200));
    divisionMain.setTitle("Integer Division");
    divisionMain.setVisible(true);
}
}

```

## OUTPUT:



Q1. Develop 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.

$\Rightarrow$  import java.util.Scanner;

```
class quadratic
```

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

```
    int a;
```

```
    int b;
```

```
    int c;
```

```
    double d;
```

```
    Scanner s1 = new Scanner(System.in);
```

```
    System.out.println("Enter coefficients of  
quadratic equation");
```

```
    a = s1.nextInt();
```

```
    b = s1.nextInt();
```

```
    c = s1.nextInt();
```

```
    d = (b * b) - (4 * a * c);
```

```
    if (d > 0.0)
```

```
{
```

```
    double r1 = ((-b) + Math.sqrt(d)) /  
        (2.0 * a);
```

```
    double r2 = ((-b) - Math.sqrt(d)) /  
        (2.0 * a);
```

```
    System.out.println("Roots are real and distinct");
    System.out.println("Root 1 = " + r1 + " Root 2  
= " + r2);
```

```
    " else if (d == 0.0)
    {
```

```
        double r3 = (-b) / (2.0 * a);
        System.out.println("Roots are real and equal");
```

```
    }
    System.out.println("Roots = " + r3);
```

```
else
```

```
{
```

```
    System.out.println("Roots are imaginary");
```

```
}
```

~~OUTPUT 1:~~

~~Enter coefficients of quadratic equation:~~

~~1 1 1~~

~~Roots are imaginary~~

~~OUTPUT 2:~~

~~Enter coefficients of quadratic equation:~~

~~1 5 2~~

~~Roots are real and distinct~~

~~Root 1 = -0.4384471871911697 Root 2 = -4.56155281280883~~

~~OUTPUT 3:~~

~~Enter coefficients of quadratic equation:~~

~~1 -2 1~~

~~Roots are real and equal~~

~~Root 1 = 1.0~~

## Q. JAVA program for SGPA calculation.

```
import java.util.Scanner;
```

```
class student
```

```
{
```

```
    int id;
```

```
    String name;
```

```
    int[] marks = new int[8];
```

```
    int[] credit = new int[8];
```

```
    int[] credit_points = new int[8];
```

```
    Scanner si = new Scanner(System.in);
```

```
    void accept_details()
```

```
{
```

```
    System.out.println("Enter student id :");
```

```
    id = si.nextInt();
```

```
    System.out.println("Enter student name :");
```

```
    name = si.next();
```

```
    System.out.println("Enter marks in order");
```

```
    of credits");
```

```
    for (int i = 0; i < 8; i++)
```

```
{
```

```
        marks[i] = si.nextInt();
```

```
}
```

~~```
    System.out.println("Enter order of credits");
```~~~~```
    for (int i = 0; i < 8; i++)
```~~~~```
{
```~~~~```
        credit[i] = si.nextInt();
```~~~~```
}
```~~

```
    void calculate()
```

```
{
```

```
for (int i=0; i< 8; i++)
```

```
    if (marks[i] >= 90)
```

{

```
        credit_points[i] = 10 * credit[i];
```

}

```
    else if (marks[i] >= 80)
```

{

```
        credit_points[i] = 9 * credit[i];
```

{

```
    else if (marks[i] >= 70)
```

{

```
        credit_points[i] = 8 * credit[i];
```

{

```
    else if (marks[i] >= 60)
```

{

```
        credit_points[i] = 7 * credit[i];
```

{

```
    else if (marks[i] >= 50)
```

{

```
        credit_points[i] = 6 * credit[i];
```

{

```
    else if (marks[i] >= 40)
```

{

```
        credit_points[i] = 5 * credit[i];
```

{

```
int sum = 0;
```

```
int count = 0;
```

```
double SGPA;
```

```
for (int j = 0; j < 8; j++)
```

```
    sum = sum + credit_points[j];
```

```
    count = count + credit[j];
```

(operator overloading)

$$\text{SGPA} = \text{sum}/\text{count};$$

$$\text{System.out.println("SGPA is: " + SGPA);}$$

{

class Main

{

(operator overloading)

```
public static void main(String args[])
    {
        System.out.println("Enter number of students");
        Scanner s = new Scanner(System.in);
        int n = s.nextInt();
        student st[] = new student[n];
        for (int i=0; i<n; i++)
            st[i] = new student();
        for (int i=0; i<n; i++)
            st[i].acceptDetails();
        for (int i=0; i<n; i++)
            st[i].calculate();
    }
}
```

{

OUTPUT:

Enter number of students:

2

Enter student id:

146

Enter student name:

Mayuri

Enter marks in order of credit:

98 96 97 89 96 96 94 97

Enter credit in order:

4 4 3 1 1 3 3 1

SGPA is: 9.5

```
a. import java.util.Scanner;  
class book  
{  
    String name;  
    String author;  
    double price;  
    int pages;  
    book(String name, String author, double price, int  
          pages)  
    {  
        this.name = name;  
        this.author = author;  
        this.price = price;  
        this.pages = pages;  
    }  
    void setDetails()  
    {  
        Scanner s = new Scanner(System.in);  
        System.out.println("Enter name of book:");  
        this.name = s.next();  
        System.out.println("Author:");  
        this.author = s.next();  
        System.out.println("Enter price:");  
        this.price = s.nextDouble();  
        System.out.println("Enter no. of pages:");  
        this.pages = s.nextInt();  
    }  
    void getDetails()  
    {  
    }
```

```
System.out.println("Book name: " + this.name);
System.out.println("Author: " + this.author);
System.out.println("Price: $" + this.price);
System.out.println("Number of pages: " +
    this.pages);
```

```
§ public String toString()
```

```
§     return "Book details:\n" + "Name: " +
        + name + "\n" + "Author: " +
        author + "\n" + "Price: $" +
        price + "\n" + "Num of pages: " +
        + pages;
```

```
§ class Main
```

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

```

```
Scanners s1 = new Scanner(System.in);
```

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

```
int n = s1.nextInt();
```

```
book[] b = new book[n];
```

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

```
§
```

```
System.out.println("\nEnter details for book " + (i + 1) + ": ");

```

```
b[i] = new Book(" ", " ", 0.0,
    0);
```

```
§     b[i] = setDetails();
```

```
System.out.println ("In Details of books:");
for (int i = 0; i < n; i++)
}
```

```
System.out.println ("InBook " + (i + 1) + ":");
b[i].get_details();
```

```
System.out.println ("Incomplete details of all
books :");
```

```
for (int i = 0; i < n; i++)
}
```

```
System.out.println ("In Book " + (i + 1) +
":\n" + b[i].get_details(););
```

```
}
```

```
}
```

### OUTPUT :

Enter number of book:

~~2  
Enter details for book 1~~

~~Enter name of book.~~

~~Rich dad poor dad~~

~~Enter Author:~~

~~abcde~~

Enter price: 45

Enter num. of pages: 380

Enter details for book 2 .

Enter name of book: abc

~~Enter Author: xyz~~

Enter price: 50

Enter no. of pages: 20

Details of all books:

Book 1:

Book Name: Rich dad poor dad

Author: abcde

Price: \$50.0

Number of pages: 20

Book 2:

Book name: abcde

Name Author: xyz

Price: \$ 45.0

Number of pages: 20

a.

```
import java.util.Scanner;  
  
class book  
{  
    String name;  
    String author;  
    double price;  
    int pages;  
  
    book(String name, String author, double price, int  
          pages)  
    {  
        this.name = name;  
        this.author = author;  
        this.price = price;  
        this.pages = pages;  
    }  
  
    void set_details()  
    {  
        Scanner s = new Scanner(System.in);  
        System.out.println("Enter name of book:");  
        this.name = s.next();  
        System.out.println("Author:");  
        this.author = s.next();  
        System.out.println("Enter price");  
        this.price = s.nextDouble();  
        System.out.println("Enter no. of pages:");  
        this.pages = s.nextInt();  
    }  
  
    void get_details()  
    {  
        System.out.println("Name: " + this.name);  
        System.out.println("Author: " + this.author);  
        System.out.println("Price: " + this.price);  
        System.out.println("Pages: " + this.pages);  
    }  
}
```

```
System.out.println("Book name: " + this.name);
System.out.println("Author: " + this.author);
System.out.println("Price: $" + this.price);
System.out.println("Number of pages: " +
    this.pages);
```

{

```
public String toString()
```

{

```
return "Book Details: \n" + "Name: "
    + name + "\n" + "Author: "
    + author + "\n" + "Price: $" +
    price + "\n" + "Num of pages: "
    + pages;
```

{

```
class Main
```

{

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

```
Scanner s1 = new Scanner(System.in);
System.out.println("Enter number of
books: ");
int n = s1.nextInt();
book[] b = new book[n];
for (int i = 0; i < n; i++)
```

{

```
System.out.println("Enter details
for book " + (i + 1) + ": ");
book[i] = new Book(" ", " ", 0.0,
```

```
0);
```

```
book[i] = setDetails();
```

{

```
System.out.println("In Details of books:");
for (int i = 0; i < n; i++)
{}
```

```
System.out.println("In Book " + (i + 1) + ":");
} b[i].getDetails();
```

```
System.out.println("Incomplete details of all
books");
```

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

```
System.out.println("In Book " + (i + 1) +
":\n" + b[i].getDetails());
}
```

{}

**OUTPUT:**

Enter number of book:

~~Enter details for book 1~~

~~Enter name of book:~~

~~Rich dad poor dad~~

~~Enter Author:~~

~~abcde~~

Enter price: 45

Enter num. of pages: 380

Enter details for book 2.

Enter name of book: abc

Enter Author: xyz

Enter price: 50

Enter no. of pages: 20

Details of all books:

Book 1:

Book Name: Rich dad poor dad

Author: abcde

Price: \$50.0

Number of pages: 20

Book 2:

Book name: abcde

Name Author: xyz

Price: \$45.0

Number of pages: 20

LAB PROGRAM 1: Abstract

```
import java.util.Scanner;  
abstract class Shape  
{  
    int a;  
    int b;  
    abstract void printArea();  
}
```

```
class rect extends Shape
```

```
{  
    void printArea()  
    {  
        System.out.println("Area of rectangle is : "  
            + (a * b));  
    }  
}
```

```
class tri extends Shape
```

```
{  
    void printArea()  
    {  
        System.out.println("Area of triangle is : "  
            + (0.5 * a * b));  
    }  
}
```

```
class cir extends Shape
```

```
{  
    void printArea()  
    {  
        System.out.println("Area of circle is : " + (3.14  
            * a * a));  
    }  
}
```

## class Main

{

public static void main(String args[])

{

Scanner s1 = new Scanner(System.in);

System.out.println("Press 1. Rectangle 2.  
Triangle 3. Circle");

int choice;

choice = s1.nextInt();

switch (choice)

{

case 1: System.out.println("Enter l and b  
of rectangle");

int l = s1.nextInt();

int br = s1.nextInt();

rect r = new rect();

r.a = l;

r.b = br;

r.printArea();

break;

case 2: System.out.println("Enter h and b  
of triangle");

int h = s1.nextInt();

int bre = s1.nextInt();

tri t = new tri();

t.a = h;

t.b = bre;

t.printArea();

break;

case 3: System.out.println("Enter r of  
circle");

int rad = s1.nextInt();

cir c = new cir();

c.a = rad;

c.printArea();

break;

default : System.out.println ("Enter valid choice");

3

3

OUTPUT :

1. Press

1. Rectangle
2. Triangle
3. Circle

1

Enter l and b of rectangle

3 2

Area of rectangle is: 6

2. Press

1. Rectangle
2. Triangle
3. Circle

2

Enter h and b of triangle

2 2

Area of triangle is: 2.0

3. Press

1. Rectangle
2. Triangle
3. Circle

3

Enter r of circle

2. Area of circle is 12.56

27/7/24

Student.java

```
package CIE;
import java.util.Scanner;
```

```
public class Student {
```

```
    public String usn;
```

```
    public String name;
```

```
    public int sem;
```

```
    public void accept()
```

```
{
```

```
        Scanner s = new Scanner(System.in);
```

```
        System.out.println("Enter name : ");
```

```
        this.name = s.nextLine();
```

```
        System.out.println("Enter usn : ");
```

```
        this.usn = s.nextLine();
```

```
        System.out.println("Enter sem ");
```

```
        this.sem = s.nextInt();
```

```
}
```

```
    public void display()
```

```
{
```

```
        System.out.println("Name : " + this.name +
```

```
        "\n USN : " + this.usn + "\n Sem : " + this.sem);
```

```
}
```

```
{
```

## Internal.java

```
package abc;
import java.util.Scanner;

public class Internal extends Student {
    public int m[] = new int[5];
    public void accept() {
        super.accept();
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter internal marks");
        for (int i = 0; i < 5; i++) {
            m[i] = sc.nextInt();
        }
    }
    public void display() {
        super.display();
        for (int i = 0; i < 5; i++) {
            System.out.print("Mark of sub " + (i + 1) + " = " + m[i]);
        }
    }
}
```

## External.java

package SEE;

import java.util.Scanner;  
import CIE.Student;

public class External extends Student

{  
 public int x[] = new int [5];

public void accept()  
{

Scanner s2 = new Scanner(System.in);  
 System.out.println("Enter external marks");  
 for (int i=0; i<5; i++)

x[i] = s2.nextInt();

public void display()

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

System.out.print("Marks of sub " +  
 +(i+1) + " = " + x[i]);

}

{ (i+1) > 0 & i < 5 }

}

Final.java

```
import java.util.Scanner;  
import CIE.Student;  
import CIE.Internal;  
import CIE.External;  
  
public class Final {  
    public static void main(String args[]) {  
        Scanner n = new Scanner(System.in);  
        System.out.println("Enter n:");  
        int y = n.nextInt();  
        CIE.Internal[] c1 = new CIE.Internal[y];  
        CIE.External[] c2 = new CIE.External[y];  
  
        for (int i = 0; i < y; i++) {  
            c1[i] = new CIE.Internal();  
            c2[i] = new CIE.External();  
  
            c1[i].accept();  
            c2[i].accept();  
  
            c1[i].display();  
            c2[i].display();  
  
        }  
  
        for (int j = 0; j < y; j++) {  
            double calc = c1[j].m1[j]  
                + ((c2[j].x[j]) / 2);  
            System.out.println(  
                "Final marks of sub C " +
```

{j+1} + "J = " + calc);  
3  
3  
3  
OUTPUT:

Enter n:  
1  
Enter Name:  
Manjari

Enter USN:  
146  
Enter sem:  
4

Enter Internal Marks:  
41 42 43 44 45

Enter External Marks  
91 92 93 94 95

Name: Manjari

USN: 146

sem: 2

Marks of sub 1 = 41

Marks of sub 2 = 42

Marks of sub 3 = 43

SEE Marks of sub 1 = 91

SEE Marks of sub 2 = 92

SEE Marks of sub 3 = 93

Final marks of sub [1] = 86.0

Final marks of sub [2] = 88.0

Final marks of sub [3] = 89.0

:

## EXCEPTION HANDLING:

```
import java.util.Scanner;
```

```
class WrongAgeException extends Exception
```

```
    public WrongAgeException(String message)  
        super(message);
```

```
class Father
```

```
private int fatherAge;
```

```
public Father(int age) throws WrongAgeException
```

```
if (age < 0)
```

```
    throw
```

```
new WrongAgeException("Age
```

```
");
```

```
this.fatherAge = age;
```

```
class Son extends Father
```

```
private int sonAge;
```

```
public Son(int fatherAge, int sonAge)
```

```
throws WrongAgeException
```

```
super(fatherAge);
```

```
if (sonAge >= fatherAge)
```

```
throw new WrongAgeException ("")
```

3      Soni age should be less than father's age");  
 this.sonAge = sonAge;  
 System.out.println("father's Age : " + fatherAge);  
 System.out.println("son's Age : " + sonAge);

3  
 public class ExceptionInheritanceDemo

public static void main (String [] args)

Scanner scanner = new Scanner(System.in);  
 try {

System.out.println ("Enter father's age : ");  
 int fatherAge = scanner.nextInt();  
 Father father = new Father (fatherAge);  
 System.out.println ("Enter son's Age : ");  
 int sonAge = scanner.nextInt();  
 Son son = new Son (fatherAge, sonAge);

3      Age  
 catch (WrongException e)

System.out.println ("Exception : " + e.getMessage());

}

3

OUTPUT 1:

Enter father's Age : -1

Exception: Age cannot be negative

OUTPUT 2:

Enter father's Age : 45

Enter son's Age : 70      son's age should be less than father's age.

Exception: son's age should be less than father's age.

OUTPUT.3:

Enter father's Age : 45

Enter son's Age : 9

Father's Age: 45

Son's Age: 9

X 21/21

## 8<sup>th</sup> LAB PROGRAM

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

class NewThread1 implements Runnable

Thread t1;

NewThread1()

{

t1 = new Thread(this, "Thread1");  
System.out.println("CT : " + t1);  
t1.start();

}

public void run()

{

try

{

for (int n=5; n>0; n--)  
Thread.sleep(10000);

System.out.println("BMS College of Engineering")  
Thread.sleep(10000);

}

catch(InterruptedException e)

{

System.out.println("Thread 1 interrupted");

}

System.out.println("Thread 1 quitting");

}

(Program execution starts after 5 sec)

class NewThread implements Runnable

{

    Thread t2;

    NewThread2()

{

        t2 = new Thread(this, "Thread 2");

        System.out.println("C1 : " + ++c);

        t2.start();

{

    public void run()

{

        try

            for (int n=5; n>0; n--)

{

                System.out.println("C2 : ");

                Thread.sleep(2000);

{(00001) probably

    catch (InterruptedException e)

{

        System.out.println("Thread 2 interrupted");

{

        System.out.println("Thread 2 exiting");

{

{

class MainThread

{

    public static void main(String args[])

{

        new NewThread1();

        new NewThread2();

{

```
try
{ Thread.sleep(40000);
  System.out.println("Main Thread is awake\n");
}
catch(InterruptedException ie){}
  System.out.println("Main Thread interrupted");}
```

OUTPUT:

CT: Thread[#,Thread1,5,main]

BMS College of Engineering

CT: Thread[#,Thread2,5,main]

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

Thread 2 exiting

BMS College of Engineering

BMS College of Engineering → Main Thread is awake

BMS College of Engineering → Main Thread exiting

Thread 1 exiting.

✓  
5/2/2024

PROGRAM 5:-

```
import java.util.Scanner;  
class Account  
{  
    String customerName;  
    int accountNumber;  
    String accountType;  
    double balance;
```

```
    public Account(String customerName, int accountNumber,  
                   String accountType, double balance)  
{
```

```
        this.customerName = customerName;  
        this.accountNumber = accountNumber;  
        this.accountType = accountType;  
        this.balance = balance;
```

```
    }  
    public void deposit(double amount)  
{
```

```
        balance += amount;  
        System.out.println("Deposit of $" +  
                           amount + " successful");
```

```
    }  
    public void displayBalance()  
{
```

```
        System.out.println("Balance: $" + balance);
```

```
    }  
    class subAcct extends Account  
{
```

```
        double minBalance;  
        double penaltyCharge;
```

```
public Current (String customerName, int accountNumber, double balance)
```

```
super (customerName, accountNumber,
```

```
"Current", balance);
```

```
minBalance = 1000;
```

```
penaltyCharge = 50;
```

```
public void withdraw (double amount)
```

```
if (balance - amount >= minBalance)
```

```
balance -= amount;
```

```
System.out.println ("Withdrawal of  
+ amount + " successful.");
```

```
else
```

```
System.out.println ("Inufficient balance.  
Withdrawal failed.");
```

```
} class Savings extends Account
```

```
double interestRate;
```

```
public Savings (String customerName, int accountNumber,  
double balance)
```

```
super (customerName, accountNumber, "Savings");  
balance);
```

```
interestRate = 0.05;
```

```
public void deposit(double interest)
```

```
{ balance = balance + interest;
```

```
double interest = balance * interestRate;
```

```
balance += interest;
```

```
System.out.println("Interest of $" +  
interest + " deposited.");
```

```
}
```

```
public void withdraw(double amount)
```

```
{ balance = balance - amount;
```

```
if (balance - amount >= 0)
```

```
{ balance -= amount;
```

```
System.out.println("Withdrawal of $" +  
amount + " successful.");
```

```
else
```

```
{ balance += amount;
```

```
System.out.println("Insufficient balance.  
withdrawal failed.");
```

```
}
```

```
public class BankDemo
```

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

```
{
```

```
Scanner s = new Scanner(System.in);
```

```
System.out.println("Enter details for  
current account.");
```

```
System.out.println("Customer Name");
```

```
String currentCustomerName = s.nextLine();
```

```

System.out.println ("Account Number . . .");
int currentAccountNumber = Integer.parseInt (s.nextLine());
System.out.println ("Initial Balance: $ . . .");
double currentInitialBalance = Double.parseDouble (s.nextLine ());
Customer currentAccount = new Customer (current
    .getCustomerName (), currentAccountNumber, current
    .getInitialBalance ());
System.out.println ("\nEnter details for savings
    account . . .");
System.out.println ("Customer Name: . . .");
System.out.print ("Account Number: . . .");
int SavingsAccountNumber = Integer.parseInt
    (s.nextLine ());
SavAcct savingsAccount = new SavAcct (
    savingsCustomerName, savingsAccountNumber,
    savingsInitialBalance);
System.out.println ("\nEnter deposit amount
    for current account: $ . . .");
double depositAmountCurrent = Double.parseDouble
    (s.nextLine ());
currentAccount.deposit (depositAmountCurrent);
savingsAccount.depositInterest ();
savingsAccount.displayBalance ();
System.out.println ("\nEnter withdrawal amount
    for savings account: $ . . .");
double withdrawAmountSavings = Double.parseDouble
    (scanner.nextLine ());
SavingsAccount.withdraw (withdrawAmountSavings);
savingsAccount.displayBalance ();
}

```

OUTPUT:

Enter details for current account;

Customer Name : manjari

Account Number : 123

Initial Balance : \$500.

Enter details for savings account:

Customer Name : manj

Account Number : 124

Initial Balance : \$300

Enter deposit amount for current account : \$900.

Deposit of \$100.0 successful.

Enter withdrawal amount for current account:  
\$1500.

Insufficient balance. Withdrawal failed.

Balance : \$1400.0

Enter deposit amount for savings account : \$300.

Deposit of \$300.0 successful.

Interest of \$30.0 deposited.

Balance : \$630.0

Enter withdrawal amount for savings account : \$800.

Withdrawal of \$500.0 successful.

Balance : \$130.0

## PROGRAM - 9

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

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

```
public class DivisionMain1 extends frame implements  
ActionListener
```

{

```
    TextField num1, num2;
```

```
    Button divideButton;
```

```
    Label resultLabel;
```

```
    public DivisionMain1()
```

```
        setLayout(new flowLayout());
```

```
        divideButton = new Button("Divide");
```

```
        label number1 = new Label("Number 1:", label.  
RIGHT);
```

```
        label number2 = new Label("Number 2:", label.RIGHT);
```

```
        num1 = new TextField(5);
```

```
        num2 = new TextField(5);
```

```
        resultLabel = new Label("Result:", label.RIGHT);
```

```
        add(number1);
```

```
        add(num1);
```

```
        add(number2);
```

```
        add(num2);
```

```
        add(divideButton);
```

```
        add(resultLabel);
```

```
        divideButton.addActionListener(this);
```

```
        addWindowListener(new WindowAdapter()
```

{}

```
public void windowClosing(WindowEvent we){  
    system.exit(0);  
}
```

```
public void actionPerformed(ActionEvent ae){
```

{

try

{

```
int n1=Integer.parseInt(text1.getText());  
int n2=Integer.parseInt(text2.getText());  
if(n2==0) throw new DivideByZeroException();
```

```
else result=(n1/n2);
```

```
else throw new ArithmeticException("cannot divide  
by zero");
```

```
resultLabel.setText("Result: " + result);
```

```
catch(NumberFormatException e1){  
    resultLabel.setText("Result: " + e1.getMessage());  
}  
catch(ArithmeticException e2){  
    resultLabel.setText("Result: " + e2.getMessage());  
}
```

```
showErrorDialog("Number Format Exception");  
}  
catch(ArithmeticException e2){  
    resultLabel.setText("Result: " + e2.getMessage());  
}  
showErrorDialog("Result: " + e2.getMessage());  
}  
private void showErrorDialog(String message){
```

{

```
Dialog dialog = new Dialog (this, "Error", true);
```

```
dialog.setLayout (new FlowLayout ());
```

```
Label label = new Label (message);
```

```
Button okButton = new Button ("OK");
```

~~```
okButton okButton = new B x
```~~

```
okButton.addActionListener (new ActionListener ()
```

{

```
public void actionPerformed (ActionEvent)
```

{

```
dialog.dispose ();
```

}

});

```
dialog.add (label);
```

```
dialog.add (okButton);
```

```
dialog.setSize (300, 100);
```

```
dialog.setVisible (true);
```

}

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

{

```
DivisionMain divisionMain = new DivisionMain ()
```

};

```
divisionMain .setSize (new Dimension (400, 200));
```

```
divisionMain .setTitle ("Integer Division");
```

```
divisionMain .setVisible (true);
```

}

{