



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
B. M. S. COLLEGE OF ENGINEERING
(AUTONOMOUS COLLEGE UNDER VTU, BELGAUM)
BANGALORE – 560019

2024-25

OBJECT ORIENTED PROGRAMMING

Submitted by

NAME	USN
SAMRAAT DABOLAY	1BM22CS236

Department of Computer Science and Engineering
B.M.S College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560 019

M	T	W	T	F	S	S
Page No.:						
Date:						YOUVA

SI. No.

Contents

Date

Signature

Lab No. 1.	Lab 1 : Sample Programs	5 / 12 / 23	UV
① 2.	Lab 2 : Quadratic Equation	12 / 12 / 23	UV
② 3.	Lab 3 : Student Class	19 / 12 / 23	UV
③ 4.	Lab 4 : Book class	26 / 12 / 23	8 26/12/23
④ 5.	Lab 5 : Area class	02 / 01 / 24	UV 2/1/24
⑤ 6.	Lab 6 : Bark class	16 / 01 / 24	UV 23-1-24
7.	Lab 7 : Strings & Generics	23 / 01 / 24	UV 23
⑥ 8.	Lab 8 : CIE and SEE	23 / 01 / 24	UV 30
⑦ 9.	Lab 9 : Wrong Age	30 / 01 / 24	UV 30/1/24
⑧ 10.	Lab 10 : Threads	06 / 02 / 24	UV 6/2/24
⑨ 11.	Lab 11 : IPC and Deadlock	13 / 02 / 24	UV 13/2/24
⑩ 12.	Lab 12 : User Interface	20 / 02 / 24	UV 20/2/2024

Program List

- ① Demonstrate parseInt()
 - ② Demonstrate Scanner class
 - ③ 1D array & 2D array
 - ④ Factorial of a number
 - ⑤ Palindrome for 5 digit
 - ⑥ Sum of 5 digits
- ⑦ Conversions (widening, Narrowing, Promotions)

Lab 1

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 discriminant is -ve display a message that there are no real solutions.

```
import java.util.Scanner
class Quadratic
{
    int a, b, c;
    double r1, r2, d;
    void getd()
    {

```

```
        Scanner s = new Scanner (System.in);
        System.out.print ("Enter the coefficients
of a,b,c");

```

```
        a = s.nextInt ();
        b = s.nextInt ();
        c = s.nextInt ();
    }
}
```

```
void compute()
{
```

while ($a == 0$)

{

System.out.println ("Not a quadratic equation");

System.out.println ("Enter a non zero value for a : ");

Scanner s = new Scanner (System.in);
 $a = s.nextInt();$

}

$$d = b * b - 4 * a * c;$$

if ($d == 0$)

{

$$r_1 = (-b) / (2 * a);$$

System.out.println ("Roots are real & equal");

System.out.println ("Root1 = Root2 = " + r1);

}

else if ($d > 0$)

{

$$r_1 = ((-b) + (\text{Math.sqrt}(d))) / (2 * a);$$

$$r_2 = ((-b) - (\text{Math.sqrt}(d))) / (2 * a);$$

System.out.println ("Roots are real & distinct");

System.out.println ("Root1 = " + r1 + "Root2 = " + r2);

/

else if ($d < 0$)

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

$$r_1 = (-b) / (2 * a);$$

$$r_2 = \text{Math.sqrt}((-d) / (2 * a));$$

~~System.out.println ("Roots are imaginary");~~
~~(r1 = (-b) / (2 * a))~~

System.out.println ("Root1 = " + r1 + " + i " + r2);

System.out.println ("Root2 = " + r1 + " - i " + r2);

}

-

}

class QuadraticMain

{

public static void main (String args[])

{

Quadratic q = new Quadratic();
 q.getd();
 q.compute();

}

3

Output :

- ① Enter the coefficients of a, b, c :

3

4

5

Roots are imaginary

$$\text{Root 1} = 0.0 + i 1.1055$$

$$\text{Root 2} = 0.0 - i 1.1055$$

- ② Enter the coefficients of a, b, c :

1

2

1

Roots are real and equal

$$\text{Root 1} = \text{Root 2} = -1.0$$

- ③ Enter the coefficients of a, b, c :

1

5

6

Roots are real and distinct

$$\text{Root 1} = -2.0 \quad \text{Root 2} = -3.0$$

Some root Dabotay

1BM22CS236



Part
12/12 12/12 12/12

Lab 2: Develop a java program to create a class student with members usn, name , an array credits & an array marks . Include methods to display & accept details & a method to calculate SGPA of a student .

$$\text{SGPA} = \frac{\sum (\text{course credits} * \text{gradepoints})}{\sum (\text{course credits})}$$

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

```
class Subject
```

```
{
```

```
    int subMarks;
    int credits;
    int grade;
```

```
}
```

```
class Student
```

```
{
```

```
    Subject subject [ ];
    String name;
    String usn;
    double sgpa;
    Scanner s;
    int i;
```

```
Student()
```

```
{
```

```
    int i;
```

```
    subject = new Subject [8];
```

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

```
        subject [i] = new Subject();
```

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

```
}
```

void getStudentDetails()

{

System.out.println("Enter your name :");
 name = s.next();

~~System.out.println("Enter credits of")
 [Subject " + (i+1) + ":");~~

System.out.println("Enter your USN :");
 usn = s.next();

}

void getMarks()

{

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

System.out.println("Enter marks of
 subject " + (i+1) + ":");

subject[i].subMarks = s.nextInt();

System.out.println("Enter credits of
 subject " + (i+1) + ":");

subject[i].credits = s.nextInt();

subject[i].grade = (subject[i].
 subMarks / 10) + 1;

}

}

void computeSGPA()

{

int effscore = 0;

int totalredits = 0;

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

{

effscore += subject[i].grade
 * subject[i].credits;

totalredits += subject[i].credits;

}

M	T	W	T	F	S	S
Page No.						
Date:						

YOUVA

$SGPA = (\text{double}) \text{ average} / (\text{double}) \text{ totalmarks};$

class main

{

public static void main (String args[])

{

```
Student s1 = new Student ();
s1.getStudentDetails ();
s1.getMarks ();
s1.computeSGPA ();
```

```
System.out.println ("Name : " + s1.name);
System.out.println ("USN : " + s1.usn);
System.out.println ("SGPA : " + s1.SGPA);
```

}

}

Output :

Enter your name :

sasank

Enter your USN :

236

Enter marks of subject 1 :

90

Enter credits of subject 1 :

4

Enter marks of subject 2 :

91

Enter credits of subject 2 :

4

Enter marks of subject 3 :

92

Enter credits of subject 3:

3

Enter marks of subject 4:

87

Enter credits of subject 4:

3

Enter marks of subject 5:

97

Enter credits of subject 5:

3

Enter marks of subject 6:

95

Enter credits of subject 6:

1

Enter marks of subject 7:

86

Enter credits of subject 7:

1

Enter marks of subject 8:

94

Enter credits of subject 8:

1

Name: Sanerout

USN: 236

SGPA: 9.80

WAN
19-12-19

Lab 3: Create a class Book which contains four members : name, authors, price, numPages. 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 complete of the book. Develop a Java program to create n book objects.

import java.util.Scanner;

class Books

{

String name;

String authors;

int price;

int numPages;

Books (String name, String authors, int
price, int numPages)

{

this.name = name;

this.authors = authors;

this.price = price;

this.numPages = numPages

}

public String toString()

{

String name, authors, price, numPages;

name = "Book name: " + this.name + "\n";

authors = "Authors name: " + this.authors + "\n";

price = "Price: " + this.price + "\n";

numPages = "No. of pages: " + this.numPages
+ "\n";

```

        return name + author + price + numPages;
    }

    public void setBook (String name, String author,
                        int price, int numPages)
    {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }
}

```

class Main

{

```

public static void main (String args[])
{
    Scanner s = new Scanner (System.in);
    int n, i;
    String name;
    String author;
    int price;
    int numPages;
}

```

System.out.println ("Enter no. of books");
 n = s.nextInt();

Books b[];

b = new Books[n];

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

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

name = s.next();

System.out.println ("Enter author of book :");
 auth = s.nextInt();
 System.out.println ("Enter price of
book :");
 price = s.nextInt();
 System.out.println ("Enter number of
pages :");
 numPages = s.nextInt();
 b[i] = new Books (name, auth,
 price, numPages);
 }
}

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

System.out.println (b[i].toString());
 }
}

Output :

Enter number of books :

2

Enter name of book :

abc

Enter author of book :

Sam

Enter price of book :

126

~~Enter number of pages :~~

160

Enter name of book :

def

Enter author of book :

Sam

M	T	W	T	F	S	S
Page No.:						
Date:						

YOUVA

Enter price of book :

160

Enter number of pages :

700

Book name : abc

Author : Sam

Price : 126

Pages no. of pages : 160

Book name : def

Author : Sam

Price : 160

no. of pages : 700

26/12/23

Name : Sunerat D

USN : 1BM22CS236

Lab 4 : 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, Circle such that each one of the classes extends class Shape. Each one of the classes contain only the method printArea() that prints area of given shape.

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

```
class InputScanner {
    int d1, d2;
    Scanner s = new Scanner(System.in);
    InputScanner () {
        if (this.getClass () == Circle.class)
    }
    System.out.print ("Enter radius of
circle : ");
    d1 = s.nextInt ();
}
else {
    System.out.print ("Enter height and
width : ");
    d1 = s.nextInt ();
    d2 = s.nextInt ();
}
}
```

~~abstract class Shape extends InputScanner ()~~

```
{}
abstract void printArea();
```

```

class Triangle extends Shape {
    void printArea () {
        System.out.println ("Area of triangle is : "
            + (double)(d1 * d2) / 2 );
    }
}

```

```

class Rectangle extends Shape {
    void printArea () {
        System.out.println ("Area of rectangle is : "
            + (double)(d1 * d2));
    }
}

```

```

class Circle extends Shape {
    void printArea () {
        System.out.println ("Area of circle is : "
            + (double)(3.14 * d1 * d1));
    }
}

```

```

class AreaMain {
    public static void main (String args []) {
        Rectangle r = new Rectangle ();
        Triangle t = new Triangle ();
        Circle c = new Circle ();
        r.printArea ();
        t.printArea ();
        c.printArea ();
    }
}

```

Output:

Enter height and weight:

2

4

Enter height and weight:

6

7

Enter radius of circle:

5

Area of rectangle is: 8.0

Area of triangle is: 21.0

Area of circle: 78.5

~~Name: Samruddi Dasolay~~

~~USN: 1BM22CS236~~

26/1/2029
2/1/2029

10/10.

Lab 5

Develop a Java Program to create a class Bank that maintains two kinds of accounts for its customers, one called savings account (provides compound interest but no check book) and other called current account (has check book and minimum balance requirement which if not met, a service charge is imposed).

Create class Account which stores name, number and type. From this derive two classes of ever having specific requirements. Include methods for

- a) Accept deposit & update balance
- b) Display balance
- c) Compute & deposit interest
- d) Permit withdrawal
- e) Check minimum bank balance and impose penalty

```
import java.util.*;
import java.lang.Math;
```

```
abstract class Bank {
    void withdraw(double amt);
    void deposit(double amt);
    void display();
    void overdip();
}
```

```
class Account extends Bank {
    String name;
    int acc_num;
    String type;
```

double bal;

String menu = " ";

Account (String name, int acc_num, String type,
 double bal, String menu) {
 this.name = name;
 this.acc_num = acc_num;
 this.type = type;
 this.bal = bal;
 this.menu = menu;
}

public void withdraw (double amt) {

if (amt > bal) {

System.out.println (" Withdraw declined! ");

}

else {

bal = amt;

System.out.println (" Updated bal is: "+
 bal);

}

}

public void deposit (double amt) {

bal += amt;

System.out.println (" Updated bal is: "+
 amt);

}

public void display () {

System.out.println (" Acc.no.: " + acc_no);

System.out.println (" Acc. name = " + name);

System.out.println (" Acc. type = " + type);

System.out.println (" Acc. bal = " + bal);

}

public void menuDisp () {

menu = " ----- MENU ----- \n

1. deposit \n 2. withdraw \n 3. display "

}

}

class Savings extends Account {

double interest ;

Savings (String name, int acc_num, String type, double bal, String name, double interest) {
super (name, acc_num, type, bal, name);
this.interest = interest;

}

public double interest (int time) {

double comp ;

comp = bal + Math.pow ((bal *
(1 + (interest / 100))), time);
return comp;

}

public void menudisp() {

super. menudisp();

menu += "\n 4. Compute Interest
 \n 5. Exit";

}

.

class Current extends Account {

double minbal = 10000;

int overdraftlimit = -100;

Current (String name, int acc_num, String type, double bal, String name) {
super (name, acc_num, type, bal, name);

~~super (name, acc_num, type, bal, name);~~

public void menudisp() {

menu += " \n 4. Cheque book
 \n 5. Exit"

3

```

public void withdraw ( double amt ) {
    if ( amt > bal ) {
        System.out.println ("Withdraw
declined ! ");
    } else {
        bal -= amt ;
        if ( bal < minbal ) {
            bal -= 0.1 * bal ;
            System.out.println ("Balance
is lower than minimum balance.
10% service charge imposed ! New
balance is : " + bal );
        }
        System.out.println ("Updated balance
is : " + bal );
    }
}

```

```

public void chekbook ( double amt ) {
    if ( amt <= bal && ( bal - amt ) >=
overdraftlimit ) {
        bal -= amt ;
        System.out.println ("Cheque
issued successfully . Current balance :
+ bal );
    } else {
        System.out.println ("Insufficient funds ")
    }
}

```

class Bank {

```

    public static void main (String args[])
    {
        Scanner s = new Scanner (System.in);
        String name;
        String name = " ";
        int acc_num;
        String type;
        double bal = 0;
        double interest;
        int choice;
        int time;
        double money;
        System.out.println ("Enter customer name:");
        name = s.next();
        System.out.println ("Enter account number:");
        acc_num = s.nextInt();
        System.out.println ("----- Account Type -----");
        System.out.println ("1. Savings Account 2. Current Account");
        System.out.println ("Please select account:");
        type = s.next();
    }

```

if (type.equals ("savings")) {

```

        System.out.println ("Enter interest amount");
        interest = s.nextDouble();
        Savings accs = new Savings (name,
        acc_num, type, bal, menu, interest);
    do {

```

~~accs.menudisp();~~

System.out.println (accs.menu);

System.out.println ("Enter choice");

choice = s.nextInt();

switch (choice) {

case 1:

System.out.println ("Enter amount to be deposited");

money = s.nextDouble();
 acc.deposit(money);
 break;

case 2:

System.out.println("Enter amount
to be withdrawn : ");

money = s.nextDouble();
 acc.withdraw(money);
 break;

case 3:

acc.display();
 break;

case 4:

System.out.print("Enter
time : ");

time = s.nextInt();

money = acc.interest(time);

System.out.println("Compound
interest is : " + money);
 break;

case 5:

break;

}

} while (choice != 5);

}

~~else if (type.equals("current")) {~~

~~current acc = new current (name, accnum,~~

~~type, bal, memo);~~

~~do {~~

~~acc.menuDisp();~~

~~System.out.println(acc.menu);~~

~~System.out.println("Enter choice : ");~~

~~choice = s.nextInt();~~

~~switch (choice) {~~

case 1:

```
System.out.println("Enter  
amount to be deposited :");  
money = s.nextInt();  
acc.deposit(money);  
break;
```

case 2:

```
System.out.println("Enter  
amount to be withdrawn :");  
money = s.nextInt();  
acc.withdraw(money);  
break;
```

case 3:

```
acc.display();  
break;
```

case 4:

```
System.out.println("Enter  
amount to be deque :");  
money = s.nextInt();  
acc.checkbook(money);  
break;
```

case 5:

```
break;
```

}

} while (choice != 5);

else {

System.out.println("Wrong type choice!");

}

}

}

M	T	W	T	F	S	S
Page No.:						
Date:						

YOUVA

Output :

(1) Enter customer name : Sam
Enter acc. number : 1

---- Account Type -----

1. Savings account
2. Current account

Please select account : Savings

---- MENU -----

1. Deposit
2. Withdraw
3. Display
4. Compute Interest
5. Exit

Enter choice : 1

Enter amt. : 1000.0

Updated bal : 1000

Enter choice : 2

Enter amt. : 300

Updated bal : 700.0

Enter choice : 3

Acc. no. : 1

Acc. name : Sam

Acc. type : savings

Acc. bal : 700

Enter choice : 4

Enter time : 3

Compound Interest :

Enter choice : 5

C-2
Enter customer name : Som
Enter acc. no : 2

--- Account Type ---

1. Savings Account
2. Current Account

Please select accounts : Current

--- MENU ---

1. deposit
2. withdraw
3. display
4. Cheque book
5. Exit

Enter choice : 10000

Enter amount : 10000

Updated bal : 10000

Enter choice : 2

Enter amount : 1000

Balance is lower than minimum . 10/- service charge imposed !

Updated balance : 8100

Enter choice : 4

Enter amt : 100

Cheque successfully issued . Updated bal : 8000

Enter choice : 5

Name : Somnath Dabholkar

USN : 13M22CS236

16/11/24

Lab 6 : Strings and Classes

1. Output : Hello , world // String concatenation
2. Hello world , BMSCE // length , concat , concat
65
3. 20 // to string()
- 101 convert
- 4,5) // use getChars() & setBytes()
BMSCE
65
66
67
68
69
70
71
ABCDEFG // tochararray()

- 6,7) equals & region matches
Bmsce equals BmSce → true
BmSce equals College → false
BmSce equals BMSCe → false
BmSce equals Ignored BmSce → true

- 8,9) startWith() , endWith() & equals() vs ==
true
true
~~false~~
true

Display a.

Lab 6 : Display a package CIE which has 12 classes - students & internals . Student has name, usn, sem . The class internals derived from Student has an array that stores internal marks in 5 courses of current semester . Create another package SEE which has the class External which is derived of Student . This class also has an array to store marks of 5 subjects . Import 2 packages in a file and declare final marks in all 5 subjects of n students .

Student.java

```

package cie;
import java.util.*;
public class Student {
    protected String usn = new String();
    protected String name = new String();
    protected int sem;
    public void inputStudentDetails() {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter usn : ");
        usn = s.nextLine();
        System.out.print("Enter name : ");
        name = s.nextLine();
        System.out.print("Enter sem : ");
        sem = s.nextInt();
    }
}

```

```

public void displayStudentDetails() {
    System.out.print("\n USN : " + usn);
    System.out.print("\n Name : " + name);
    System.out.print("\n Sem : " + sem);
}

```

Intervals.java

```

package cie;
import java.util.Scanner;
public class Intervals extends Student {
    protected int marks[] = new int[5];
    public void input(IE marks) {
        int i;
        Scanner s = new Scanner(System.in);
        for (i = 0; i < 5; i++) {
            System.out.println("Enter marks in subject " + (i + 1));
            marks[i] = s.nextInt();
        }
    }
}

```

Externals.java

```

package cie;
import cie.*;
import java.util.Scanner;
public class Externals extends Intervals {
    protected int marks[];
    protected int finalMarks[];
    public Externals {
        marks = new int[5];
        finalMarks = new int[5];
    }
}

```

```

public void input SEE marks() {
    Scanner s = new Scanner(System.in);
}

```

```

for (int i = 0; i < 5; i++) {
    System.out.println ("Enter marks");
    subject = " " + (i+1));
    marks[i] = s.nextInt();
}

```

```

public void calculateFinalMarks() {
    for (int i = 0; i < 5; i++)
        finalMarks[i] = marks[i]/i
        + super.marks[i];
}

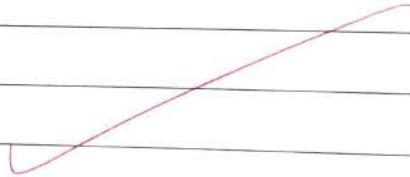
```

```

public void displayFinalMarks() {
    displayStudentDetails();
    for (int i = 0; i < 5; i++)
    {
        System.out.println ("Subject "
        +(i+1) + ":" + finalMarks[i]);
    }
}

```

}



Main.java

```

import see.Externals;
class Main {
    public static void main (String args[]) {
        int numofStudents = 2;
        Externals finalMarks[] = new Externals [numofStudents];
        for( int i = 0; i < numofStudents ; i++ ) {
            final Marks[i] = new Externals();
            final Marks[i].inputStudentDetails();
        }
    }
}

```

```

System.out.println ("Enter CIE marks : ");
final Marks[i].inputCIEmarks();
System.out.println ("Enter SEE marks : ");
final Marks[i].inputSEEmarks();
}

```

```

System.out.println ("Displaying Data");
for ( int i = 0; i < numofStudents ; i++ ) {
    final Marks[i].calculateFinalMarks();
    final Marks[i].displayFinalMarks();
}
}

```

}

}

}/

Output :

Enter usn : 136

Enter name : Sam

Enter sex : 2

Enter CIE marks:

Enter marks of subject 1 : 41

Enter marks of subject 2 : 42

Enter marks of subject 3 : 43

Enter marks of subject 4 : 44

Enter marks of subject 5 : 45

Enter SEE marks:

Enter marks of subject 1 : 91

Enter marks of subject 2 : 92

Enter marks of subject 3 : 93

Enter marks of subject 4 : 94

Enter marks of subject 5 : 95

Displaying data :

USN : 136

Name : Sam

sex : 2

Subject 1 : 86

Subject 2 : 88

Subject 3 : 90

Subject 4 : 42

Subject 5 : 43

None : Samrat Dasolay

USN : 1BM22CS 236

✓✓✓✓✓

25

Lab 7 : WAP that demonstrates handling of exceptions in inheritance tree. Create a base class "Father" and a derived class "Son". In father class, implement exception WrongAge() when the input age < 0. In son class, implement a constructor that copies both implemented a constructor that copies both father's age and throws an exception if son's age is >= father's age.

```
import java.util.*;  
class WrongAge extends Exception {  
    WrongAge (String s) {  
        super(s);  
    }  
}
```

```
class InputScanner {  
    Scanner s = new Scanner (System.in);  
}
```

```
class Father extends InputScanner {  
    int fatherAge;  
    Father () throws WrongAge {  
        System.out.print ("Enter father's age: ");  
        fatherAge = s.nextInt();  
        if (fatherAge < 0)  
            throw new WrongAge ("Age cannot be negative");  
    }  
}
```

```
void display () {  
    System.out.print ("In Father's age: ");  
    + fatherAge);  
}
```

}

```

class Son extends Father {
    int sonAge;
    Son() throws WrongAge {
        super();
        System.out.println("Enter son's age.");
        sonAge = s.nextInt();
        if (sonAge < 0)
            throw new WrongAge("Age cannot be negative.");
        else if (sonAge > fatherAge)
            throw new WrongAge("Son's age cannot be greater than father.");
        else {
            displayf();
            displayson();
            throw new WrongAge("Validation error");
        }
    }
}

```

3

```

void displayson() {
    System.out.println("In Son's age: " + sonAge);
}

```

class Main {

```

public static void main (String [] args) {
    try {
        Son son = new Son();
    } catch (WrongAge e) {
        System.out.println(e.getMessage());
    }
}

```

Output :

Enter father's age

43

Enter son's age

23

Father's age : 43

Son's age : 23

Valid age

Enter father's age

-2

Age can't be negative

Enter father's age

43

Enter son's age

46

Son's age cannot be greater than father's age

Name : Samruddi Dabholkar

USN : 1BM22CS236

1
2
3)

Lab 8 : WAP which creates two threads, one thread displaying "BMS college of Engineering" one every 10 seconds and another displaying "CSE" every 2 seconds.

import java.lang.*;

```
class DisplayMessageThread extends Thread {
    private final String message;
    private final long interval;
```

```
DisplayMessageThread (String message, long interval)
    this.message = message;
    this.interval = interval;
}
```

```
public void run () {
```

```
    try {
```

```
        while (true) {
```

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

```
            Thread.sleep (interval);
```

```
}
```

```
} catch (InterruptedException e) {
```

```
    System.out.println (Thread.currentThread().getname () + " interrupted.");
```

```
}
```

```
}
```

```
}
```

```

public class Main {
    public static void main (String args[]) {
        DisplayMessage thread t1 = new DisplayMessage (
            Thread ("BMS College of Engineering",
                    10000));
        DisplayMessage thread t2 = new DisplayMessage Thread
            ("CSE", 2000);

        t1.start ();
        t2.start ();
    }

    try {
        Thread.sleep (30000);
    } catch (InterruptedException e) {
        System.out.println ("Main thread interrupted");
    }

    t1.interrupt ();
    t2.interrupt ();

    System.out.println ("Main thread exiting!");
}

```

Output :

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

</div

Lab #9: Write a program that creates a user interface to perform integral divisions. The user enters 2 numbers in text fields num1 & num2. The division of it is displayed in result when divide button clicked. If num1 & num2 were not an integer, program would throw NumberFormatException. If num2 were zero, the program would throw an ArithmeticException. Displays the exception in a message dialog box.

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

```
public class DivisionMain1 extends Frame implements
ActionListener {
```

```
TextField num1, num2;
Button dResult;
Label outResult;
String out = "";
double resultNum;
int flag = 0;
```

```
public DivisionMain1() {
```

```
setLayout(new FlowLayout());
```

```
dResult = new Button("Result");
```

```
labelNumber1 = new Label("Number1:",  
Label.RIGHT);
```

```
labelNumber2 = new Label("Number2:",  
Label.RIGHT);
```

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

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

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

Date: YOUVA
add (number1);
add (num1);
add (number2);
add (num2);
add (dResult);
add (outResult);

num

num1.addActionListener (this);
num2.addActionListener (this);
dResult.addActionListener (this);
addWindowListener (new WindowAdapter()
{

public void windowClosing (WindowEvent e)

{

System.exit (0);

}

});

}

public void actionPerformed (ActionEvent ae) {

int n1, n2;

try {

if (ae.getSource () == dResult) {

n1 = Integer.parseInt (num1.getText());

n2 = Integer.parseInt (num2.getText());

if (n2 == 0)

throw new ArithmeticException();

out = n1 + " " + n2;

resultNum = n1 / n2;

out += String.valueOf (resultNum);

repaint();

}

}

catch (NumberFormatException e1)

{

ifFlag = 1;

out.println("Number format exception! "+e1);
repaint();

}

catch (ArithmaticException e2)

{

ifFlag = 1;

out.println("divide by 0 exception! "+e2);
repaint();

}

public void paint (Graphics g) {

if (ifFlag == 0)

g.drawString (out, outResult.getResultX()
+ outResult.getResultWidth(), outResult
.getResultY() + outResult.getResultHeight() - 2);

else

g.drawString (out, 100, 200);

ifFlag = 0;

}

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

{

DivisionPanel dm = new DivisionPanel();
dm.setSize (new Dimension (800, 400));
dm.setTitle ("Division of Integers");
dm.setVisible (true)

}

}

Output :

num 1 : 12 num 2 : 6 Result: 12 6 2.0

functions used

A button is basically a control component or label that generates an event when pushed.

setLayout() method allows you to set the layout of the container.

addActionListener() is a type of class in Java that receives a notification whenever any action is performed.

B

repaint() method is an asynchronous method of applet class.

setSize() sets the size of this Dimension object to the specified width & height.

setTitle() function defines the title to appear at top of sketch window.

setVisible() method makes the frame appear on screen.

Name: Sanasat Dasgupta

USN : 1BM22CS236

WEEK 20
20/2/2024

Lab 10: Demonstrate Inter process communication and deadlock.

class A {

int n; boolean valueset = false;

synchronized int get() {

while (!valueset) {

try {

System.out.println("In Consumer
Waiting " + n);

wait();

} catch (InterruptedException e) {

System.out.println("Interrupted Exception
caught");

}

}

System.out.println("got : " + n);

valueset = true;

System.out.println("In Intimate Producer");

notify();

return n;

}

synchronized void put (int n) {

while (valueset) {

try {

System.out.println("In Producer
Waiting " + n);

wait();

} catch (InterruptedException e) {

System.out.println("Interrupted Exception
caught");

}

}

this.n = n);

```
valueset = true;  
System.out.println ("Put : " + n);  
System.out.println ("Nr Intiate consumers : ");  
notify();
```

۴

3

class Producer implements Runnable {

Q 9)

Producers (Δq) {

$$\text{this} \cdot q = q^i$$

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

3

public void run () {

int i = 0;

while ($i < 15$) {

q.put(i++)

3

3

class Consumer implements Runnable {

Q q ;

Consumers (α & σ) {

this . q = q)

new Thread(this, "consumer").start()

1

```
public void run() {
```

int i = 0;

```
while (i < 15) {
```

int g1 = q.get();

i++ j

٣

3

3

class PC {

```
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 : 0

Get : 0

Put : 1

Get : 1

Put : 2

Get : 2

Name : Savanot Dadale

USN : 1BM22CS236

00/2/29
13/2/29

Deadlock :

```

class A {
    synchronized void foo (B b) {
        String name = Thread.currentThread();
        System.out.println(name + " entered A.foo");
        try {
            Thread.sleep(1000);
        } catch (Exception e) {
            System.out.println("A interrupted");
        }
        System.out.println(name + " trying to call B.last()");
        b.last();
    }
    void last() {
        System.out.println("Inside A.last");
    }
}

```

class B {

```

    synchronized void bar(A a) {
        String name = Thread.currentThread();
        getname();
        System.out.println(name + " entered B.bar");
        try {
            Thread.sleep(1000);
        } catch (Exception e) {
            System.out.println("B interrupted");
        }
        System.out.println(name + " trying to call A.last()");
        a.last();
    }
}

```

```
void last () {
```

```
    System.out.println ("Inside A.last");
```

```
}
```

}

```
class deadlock implements Runnable {
```

```
    A a = new A();
```

```
    B b = new B();
```

```
    deadlock () {
```

```
        Thread t = new Thread (this).start ();
```

```
        System.out.println ("Back in main thread");
```

```
    t.start ();
```

```
    a.foo (b);
```

```
    System.out.println ("Back in main thread");
```

```
}
```

```
public void run () {
```

```
    b.bar (a);
```

```
    System.out.println ("Back in other thread");
```

```
}
```

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

```
    new deadlock ();
```

```
}
```

}

Output :

MainThread entered A::foo

Rawg Thread entered B::bar

Rawg Thread trying to call A::last()

Inside A::last

MainThread trying to call B::last()

Inside A::last \Rightarrow

Back in main thread

But in other thread

Name : Sanvrat Dasolay

USN : 1BM22CS236

100
13.2.24

Source Code

Lab 1:

```
import java.util.Scanner;
class Quadratic{
int a,b,c;
double r1,r2,d;
void getd()
{
Scanner s = new Scanner(System.in);
System.out.println("Enter the coefficients of a,b,c: ");
a = s.nextInt();
b = s.nextInt();
c = s.nextInt();
}
void compute()
{
while(a==0)
{
System.out.println("Not a quadratic equation!");
System.out.println("Enter a non zero value of a: ");
Scanner s = new Scanner(System.in);
a = s.nextInt();
}
d= b*b - 4*a*c;
if(d==0)
{
r1 = (-b)/(2*a);
System.out.println("Roots are real and equal");
System.out.println("Root 1 = Root 2 = " + r1);
}
else if(d>0)
{
r1 = ((-b) + (Math.sqrt(d)))/(double)(2*a);
r2 = ((-b) - (Math.sqrt(d)))/(double)(2*a);
System.out.println("Roots are real and distinct");
System.out.println("Root 1 = " + r1 + " Root 2 = " + r2);
}
else if(d<0)
{
System.out.println("Roots are imaginary");
r1 = (-b)/(2*a);
r2 = Math.sqrt(-d)/(2*a);
System.out.println("Root 1 = " + r1 + " + i " + r2);
System.out.println("Root 2 = " + r1 + " - i " + r2);
}
}
}

class QuadraticMain
```

```

{
public static void main(String args[])
{
Quadratic q = new Quadratic();
q.getd();
q.compute();
System.out.println("Name: Samraat Dabolay");
System.out.println("USN: 1BM22CS236");
}
}

```

Lab 2:

```

import java.util.Scanner;

class Subject
{
    int subMarks;
    int credits;
    int grade;
}

class Student
{
    Subject subject[];
    String name;
    String usn;
    double SGPA;
    Scanner s;
    int i;

    Student()
    {
        int i;
        subject = new Subject[8];
        for(i=0;i<8;i++)
            subject[i] = new Subject();
        s = new Scanner(System.in);
    }

    void getStudentDetails()
    {
        System.out.println("Enter your name: ");
        name = s.next();
        System.out.println("Enter your USN: ");
        usn = s.next();
    }

    void getMarks()

```

```

    {
        for(i=0;i<8;i++)
        {
            System.out.println("Enter marks of subject " + (i+1) + ": ");
            subject[i].subMarks = s.nextInt();
            System.out.println("Enter credits of subject " + (i+1) + ": ");
            subject[i].credits = s.nextInt();
            subject[i].grade = (subject[i].subMarks/10) + 1;
        }
    }
    void computeSGPA()
    {
        int effscore = 0;
        int totalcreds = 0;
        for(i=0;i<8;i++)
        {
            effscore += subject[i].grade * subject[i].credits;
            totalcreds += subject[i].credits;
        }
        SGPA = (double)effscore/(double)totalcreds;
    }
}

class Main
{
    public static void main(String args[])
    {
        Student s1 = new Student();
        s1.getStudentDetails();
        s1.getMarks();
        s1.computeSGPA();
        System.out.println("Name: " + s1.name);
        System.out.println("USN: " + s1.usn);
        System.out.println("SGPA: " + s1.SGPA);
    }
}

```

Lab 3:

```

import java.util.Scanner;

class Books
{
    String name;
    String author;
    int price;
    int numPages;

    Books(String name, String author, int price, int numPages)
    {

```

```

        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }

    public String toString()
    {
        String name, author, price, numPages;
        name = "Book name: " + this.name + "\n";
        author = "Author name: " + this.author + "\n";
        price = "Price: " + this.price + "\n";
        numPages = "Number of Pages: " + this.numPages + "\n";
        return name + author + price + numPages;
    }
}

class books_main
{
    public static void main(String args[])
    {
        Scanner s = new Scanner(System.in);
        int n,i;
        String name;
        String author;
        int price;
        int numPages;

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

        Books b[];
        b = new Books[n];

        for(i = 0; i < n; i++)
        {
            System.out.println("Enter name of book: ");
            name = s.next();
            System.out.println("Enter author of book: ");
            author = s.next();
            System.out.println("Enter price of book: ");
            price = s.nextInt();
            System.out.println("Enter number of pages: ");
            numPages = s.nextInt();
            b[i] = new Books(name,author,price,numPages);
        }

        for(i = 0; i < n; i++)
        {
            System.out.println(b[i].toString());
        }
    }
}

```

```
    }  
}
```

Lab 4:

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

```
class InputScanner{  
int d1, d2;  
Scanner sc = new Scanner(System.in);  
InputScanner(){  
if(this.getClass() == Circle.class){  
System.out.println("Enter radius of circle: ");  
d1 = sc.nextInt();  
}  
else{  
System.out.println("Enter height and weight: ");  
d1 = sc.nextInt();  
d2 = sc.nextInt();  
}  
}  
}
```

```
abstract class Shape extends InputScanner{  
abstract void printArea();  
}
```

```
class Triangle extends Shape{  
void printArea(){  
System.out.println("Area of triangle is: " + (double)(d1*d2)/2);  
}  
}
```

```
class Rectangle extends Shape{  
void printArea(){  
System.out.println("Area of rectangle is: " + (double)(d1*d2));  
}  
}
```

```
class Circle extends Shape{  
void printArea(){  
System.out.println("Area of circle: " + (double)(3.14*d1*d1));  
}  
}
```

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

```

Rectangle r = new Rectangle();
Triangle tr = new Triangle();
Circle c = new Circle();
r.printArea();
tr.printArea();
c.printArea();
}
}

```

Lab 5:

```

import java.util.*;
import java.lang.Math;
abstract class Bank{
    abstract void withdraw(double amt);
    abstract void deposit(double amt);
    abstract void display();
    abstract void menudisp();
}

class Account extends Bank{
    String name;
    int acc_num;
    String type;
    double bal;
    String menu = " ";
    Account(String name, int acc_num, String type, double bal, String menu){
        this.name = name;
        this.acc_num = acc_num;
        this.type = type;
        this.bal = bal;
        this.menu = menu;
    }
    public void withdraw(double amt){
        if(amt>bal){
            System.out.println("Withdraw declined! Max amount you can withdraw is:
" + bal);
        }
        else{
            bal -= amt;
            System.out.println("Updated balance is: " + bal);
        }
    }

    public void deposit(double amt){
        bal += amt;
        System.out.println("Updated balance is: " + bal);
    }
}

```

```

public void display(){
    System.out.println("Account number: " + name);
    System.out.println("Account name: " + acc_num);
    System.out.println("Account type: " + type);
    System.out.println("Balance: " + bal);
}
public void menudisp(){
    menu = "-----MENU-----\n1. Deposit\n2. Withdraw\n3. Display";
}
}

class Savings extends Account{
    double interest;
    Savings(String name, int acc_num, String type, double bal, String menu, double interest){
        super(name, acc_num, type, bal, menu);
        this.interest = interest;
    }
    public double interest(int time){
        double comp;
        comp = bal + Math.pow((bal*(1+(interest/100))),time);
        return comp;
    }
    public void menudisp(){
        super.menudisp();
        menu += "\n4. Compute Interest\n5. Exit";
    }
}

class Current extends Account{
    double minbal = 10000;
    Current(String name, int acc_num, String type, double bal, String menu, double minbal){
        super(name, acc_num, type, bal, menu);
        this.minbal = minbal;
    }
    public void menudisp(){
        super.menudisp();
        menu += "\n4. Cheque Book\n5. Exit";
    }
    public void withdraw(double amt){
        if(amt>bal){
            System.out.println("Withdraw declined! Max amount you can withdraw is: "
" + bal);
        }
        else{
            bal -= amt;
            System.out.println("Updated balance is: " + bal);
        }
    }
}

```

```

}

class BankMain{
    public static void main(String args[]){
        Scanner s = new Scanner(System.in);
        String name;
        String menu = " ";
        int acc_num;
        String type;
        double bal = 0;
        double interest;
        int choice;
        int time;
        double money;
        System.out.println("Enter customer name: ");
        name = s.next();
        System.out.println("Enter account number: ");
        acc_num = s.nextInt();
        System.out.println("-----ACCOUNT TYPE-----\n1.Savings
Account\n2.Current Account\nPlease select account type: ");
        type = s.next();

        if(type.equals("savings")){
            System.out.println("Enter interest amount: ");
            interest = s.nextDouble();
            Savings accs = new Savings(name, acc_num, type, bal, menu, interest);
            do{
                accs.menudisp();
                System.out.println(accs.menu);
                System.out.println("Enter choice: ");
                choice = s.nextInt();
                switch(choice){
                    case 1:
                        System.out.println("Enter amount to be deposited:
");
                        money = s.nextDouble();
                        accs.deposit(money);
                        break;
                    case 2:
                        System.out.println("Enter amount to be withdrawn:
");
                        money = s.nextDouble();
                        accs.withdraw(money);
                        break;
                    case 3:
                        accs.display();
                        break;
                    case 4:
                        System.out.println("Enter time to calculate interest
in years: ");
                        time = s.nextInt();
                }
            }
        }
    }
}

```

```
        money = accs.interest(time);
        System.out.println("Compound interest is: " +
money);
    }
    break;
}
case 5:
    break;
}
}while(choice!=5);
}
```

Lab 6:

Student.java (/cie)

```
package cie;
import java.util.*;
public class Student {

protected String usn = new String();
protected String name = new String();
protected int sem;

public void inputStudentDetails() {

Scanner s = new Scanner(System.in);
System.out.println("Enter usn: ");
usn = s.nextLine();
System.out.println("Enter name: ");
name = s.nextLine();
System.out.println("Enter sem: ");
sem = s.nextInt();

}

public void displayStudentDetails() {
    System.out.println("\nUSN: " + usn);
    System.out.println("\nName: " + name);
    System.out.println("\nSem: " + sem);
}
```

Internals.java (/cie)

```
internals.java (31)  
package cie;  
import java.util.Scanner;  
public class Internals extends Student {  
    protected int marks[] = new int[5];
```

```

public void inputCIEmarks(){
    int i;
    Scanner s = new Scanner(System.in);
    for(i = 0;i<5;i++){
        System.out.println("Enter Marks in subject "+(i+1));
        marks[i] = s.nextInt();
    }
}

```

Externals.java (/see)

```

package see;
import cie.*;
import java.util.Scanner;
public class Externals extends Internals {

    protected int marks[];
    protected int finalMarks[];

    public Externals() {
        marks = new int[5];
        finalMarks = new int[5];
    }

    public void inputSEEmarks() {
        Scanner s = new Scanner(System.in);
        for(int i=0;i<5;i++) {
            System.out.print("Subject "+(i+1)+" marks: ");
            marks[i] = s.nextInt();
        }
    }

    public void calculateFinalMarks() {
        for(int i=0;i<5;i++)
            finalMarks[i] = marks[i]/2 + super.marks[i];
    }

    public void displayFinalMarks() {
        displayStudentDetails();
        for(int i=0;i<5;i++)
            System.out.println("Subject " + (i+1) + ": " + finalMarks[i]);
    }
}

```

```

Marksmain.java
import see.Externals;

class markmain {
    public static void main(String args[])
    {

        int numOfStudents = 2;
        Externals finalMarks[] = new Externals[numOfStudents];

        for(int i=0;i<numOfStudents;i++){
            finalMarks[i] = new Externals();
            finalMarks[i].inputStudentDetails();
            System.out.println("Enter CIE marks");
            finalMarks[i].inputCIEmarks();
            System.out.println("Enter SEE marks");
            finalMarks[i].inputSEEmarks();
        }
        System.out.println("Displaying data:\n");
        for(int i=0;i<numOfStudents;i++){
            finalMarks[i].calculateFinalMarks();
            finalMarks[i].displayFinalMarks();
        }
    }
}

```

Lab 7:

```

import java.util.*;
class WrongAge extends Exception{
    WrongAge(String s){
        super(s);
    }
}

class InputScanner{
    Scanner s = new Scanner(System.in);
}

class Father extends InputScanner{
    int fatherAge;
    Father() throws WrongAge {
        System.out.println("Enter father's age: ");
        fatherAge = s.nextInt();
        if (fatherAge < 0)
            throw new WrongAge("Age cannot be negative");
    }
}

```

```

void displayf(){
    System.out.println("\nFather's age: " + fatherAge);
}
}

class Son extends Father{
    int sonAge;
    Son() throws WrongAge{
        super();
        System.out.println("Enter son's age: ");
        sonAge = s.nextInt();
        if (sonAge < 0)
            throw new WrongAge("Age cannot be negative.");
        else if(sonAge > fatherAge)
            throw new WrongAge("Son's age cannot be greater than father's age.");
        else{
            displayf();
            displaySon();
            throw new WrongAge("Valid age.");
        }
    }
}

void displaySon(){
    System.out.println("\nSon's age : "+ sonAge);
}
}

class exceptionsmain {
    public static void main(String[] args) {
        try{
            Son son = new Son();
        } catch(WrongAge e) {
            System.err.println(e.getMessage());
        }
    }
}

```

Lab 8:

```

import java.lang.*;
class DisplayMessageThread extends Thread {
    private final String message;
    private final long interval; // in milliseconds

    DisplayMessageThread(String message, long interval) {
        this.message = message;
        this.interval = interval;
    }
}

```

```

public void run() {
    try {
        while (true) {
            System.out.println(message);
            Thread.sleep(interval);
        }
    } catch (InterruptedException e) {
        System.out.println(Thread.currentThread().getName() + " interrupted.");
    }
}

public class TwoThreadDemo {
    public static void main(String[] args) {
        DisplayMessageThread thread1 = new DisplayMessageThread("BMS College of
Engineering", 10000); // 10 seconds
        DisplayMessageThread thread2 = new DisplayMessageThread("CSE", 2000); // 2 seconds

        thread1.setName("Thread 1");
        thread2.setName("Thread 2");

        thread1.start();
        thread2.start();

        try {
            // Let the threads run for a while
            Thread.sleep(30000); // Let the program run for 30 seconds
        } catch (InterruptedException e) {
            System.out.println("Main thread interrupted.");
        }

        // Interrupt both threads to stop them
        thread1.interrupt();
        thread2.interrupt();

        System.out.println("Main thread exiting.");
    }
}

```

Lab 9:

IPC

```

import java.lang.*;
class Q {
    int n;
    boolean valueSet = false;
    synchronized int get() {
        while(!valueSet){
            try {

```

```

        System.out.println("\nConsumer waiting\n");
        wait();
    } catch(InterruptedException e) {
        System.out.println("InterruptedException caught");
    }
}
System.out.println("Got: " + n);
valueSet = false;
System.out.println("\nIntimate Producer\n");
notify();
return n;
}
synchronized void put(int n) {
    while(valueSet){
        try {
            System.out.println("\nProducer waiting\n");
            wait();
        } catch(InterruptedException e) {
            System.out.println("InterruptedException caught");
        }
    }
    this.n = n;
    valueSet = true;
    System.out.println("Put: " + n);
    System.out.println("\nIntimate Consumer\n");
    notify();
}
}

class Producer implements Runnable {
    Q q;
    Producer(Q q) {
        this.q = q;
        new Thread(this, "Producer").start();
    }
    public void run() {
        int i = 0;
        while(i<15) {
            q.put(i++);
        }
    }
}

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

```

```

        while(i<15) {
            int r=q.get();
            i++;
        }
    }

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

```

Deadlock

```

class A {

    synchronized void foo(B b) {

        String name =
        Thread.currentThread().getName();

        System.out.println(name + " entered A.foo");

        try {

            Thread.sleep(1000);

        } catch(Exception e) {

            System.out.println("A Interrupted");

        }

        System.out.println(name + " trying to call B.last()");

        b.last();

    }

    void last() {

        System.out.println("Inside A.last");

    }

}

class B {

```

```
synchronized void bar(A a) {  
  
    String name =  
    Thread.currentThread().getName();  
  
    System.out.println(name + " entered B.bar");  
  
    try {  
  
        Thread.sleep(1000);  
  
    } catch(Exception e) {  
  
        System.out.println("B Interrupted");  
  
    }  
  
    System.out.println(name + " trying to call A.last()");  
  
    a.last();  
  
}  
  
void last() {  
  
    System.out.println("Inside A.last");  
  
}  
  
}  
  
class Deadlock implements Runnable  
{  
  
    A a = new A();  
  
    B b = new B();  
  
    Deadlock() {  
  
        Thread.currentThread().setName("MainThread");  
  
        Thread t = new Thread(this,  
        "RacingThread");  
  
        t.start();  
  
        a.foo(b); // get lock on a in this thread.  
  
        System.out.println("Back in main thread");  
    }  
}
```

```

}

public void run() {
    b.bar(a); // get lock on b in other thread.

    System.out.println("Back in other thread");

}

public static void main(String args[]) {
    new Deadlock();
}

}

```

Lab 10:

```

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

public class DivisionMain1 extends Frame implements ActionListener
{
    TextField num1,num2;
    Button dResult;
    Label outResult;
    String out="";
    double resultNum;
    int flag=0;

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

        dResult = new Button("RESULT");
        Label number1 = new Label("Number 1:",Label.RIGHT);
        Label number2 = new Label("Number 2:",Label.RIGHT);
        num1=new TextField(5);
        num2=new TextField(5);
        outResult = new Label("Result:",Label.RIGHT);

        add(number1);
        add(num1);
        add(number2);
        add(num2);
        add(dResult);
        add(outResult);
    }
}
```

```

num1.addActionListener(this);
num2.addActionListener(this);
dResult.addActionListener(this);
addWindowListener(new WindowAdapter()
{
    public void windowClosing(WindowEvent we)
    {
        System.exit(0);
    }
});
}

public void actionPerformed(ActionEvent ae)
{
    int n1,n2;
    try
    {
        if (ae.getSource() == dResult)
        {
            n1=Integer.parseInt(num1.getText());
            n2=Integer.parseInt(num2.getText());

            /*if(n2==0)
                throw new ArithmeticException();*/
            out=n1+" "+n2;
            resultNum=n1/n2;
            out+=String.valueOf(resultNum);
            repaint();
        }
    }
    catch(NumberFormatException e1)
    {
        flag=1;
        out="Number Format Exception! "+e1;
        repaint();
    }
    catch(ArithmeticException e2)
    {
        flag=1;
        out="Divide by 0 Exception! "+e2;
        repaint();
    }
}

public void paint(Graphics g)
{
    if(flag==0)

```

```
g.drawString(out,outResult.getX()+outResult.getWidth(),outResult.getY()+outResult.getHeight()-8);
        else
            g.drawString(out,100,200);
            flag=0;
    }

public static void main(String[] args)
{
    DivisionMain1 dm=new DivisionMain1();
    dm.setSize(new Dimension(800,400));
    dm.setTitle("DivionOfIntegers");
    dm.setVisible(true);
}

}
```