

# **B.M.S COLLEGE OF ENGINEERING BENGALURU**

Autonomous Institute, Affiliated to VTU



## **LAB REPORT**

### **23CS3PCOOJ**

Submitted in partial fulfilment of the requirements for Lab

Bachelor of Engineering

in

Computer Science and Engineering

Submitted by:

**NAVEEN RAMKUMAR**

**1BM22CS173**

Department of Computer Science and Engineering,

B.M.S College of Engineering,

Bull Temple Road, Basavanagudi, Bangalore, 560 019

2023-2024.

# Lab Program 1

Date \_\_\_\_\_  
Page \_\_\_\_\_

Lab program 1: Quadratic equation

```
import java.util.Scanner;
class Quadratic{
    int a, b, c;
    double r1, r2, d;
    void getdata(){
        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 eqn");
            System.out.println("Initial value: " + a);
            System.out.println("Value of b: " + b);
            Scanner s = new Scanner(System.in);
            d = s.nextInt();
            if(d == 0)
                r1 = ((-b) + Math.sqrt(d)) / (2 * a);
                r2 = ((-b) - Math.sqrt(d)) / (2 * a);
                System.out.println("Roots are real and distinct");
                System.out.println("Roots are: " + r1 + " & " + r2);
            }
        }
}
```

`else if (d < 0)`

{  
 System.out.println("Roots are imaginary");  
 r1 = -b / (2 \* a);  
 r2 = Math.sqrt(-d) / (2 \* a);  
 System.out.println("Root1 = " + r1 + "  
 " + r2 + " + j" + r2);  
 System.out.println("Root2 = " + r1 + "  
 " + r2 + "j");  
 }  
 }  
 }

Class Quadratic Main

{  
 public static void main (String args [ ]) {  
 Quadratic q = new Quadratic ();  
 q . solve ();  
 q . compute ();  
 }  
 }  
 }

OUTPUT:

Navleen Ramburaw  
 USN: 1BM22CS173

java QuadraticMain . java

java QuadraticMain

Enter the coefficients of a, b, c  
 1 -7 10

Roots are real and distinct  
 Root 1 = 5.0 Root 2 = 2.0

java QuadraticMain

Enter the coefficients of a, b, c  
0 6 7

Not a quadratic equation.

Enter a non zero value for a.  
3

Roots are imaginary

Root1 = -1.0 + i 1.1547005383792515

Root2 = -1.0 - i 1.1547005383792515

java QuadraticMain

Enter the coefficients of a, b, c  
25 -20 4

Roots are real and equal

Root1 = Root2 = 0.0

java QuadraticMain

Enter the coefficients of a, b, c  
1 -6 9

Roots are real and equal

Root1 = Root2 = 3.0

By  
12/12/2023

## Source Code:

```
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 for 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("Root1 = Root2 = " + 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("Root1 = " + r1 + " Root2 = " + r2);
    }
    else if(d<0)
    {
        System.out.println("Roots are imaginary");
        r1 = (-b)/(2*a);
        r2 = Math.sqrt(-d)/(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();
    }
}
```

## Lab Program 2

Lab(2) - 19/12/2023

Lab Program 2: Student Marks and Scores

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

```
class Subject
```

```
{
```

```
    int subjectMarks;
```

```
    int Credits;
```

```
    int grade;
```

```
}
```

```
class Student
```

```
{
```

```
    int i, j;
```

```
    Subject[] subjects = new Subject[9];
```

```
    String name;
```

```
    String usn;
```

```
    float avg = 0;
```

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

```
    Student();
```

```
{
```

```
        for (i = 0; i < 9; i++)
```

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

```
    }
```

```
    void getStudentDetails()
```

```
{
```

```
    System.out.print("Enter the name of the student:
```

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

```
    System.out.print("Enter the usn of the student:
```

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

```
};
```

void getmarks()

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

System.out.println ("Enter the marks for  
subject " + (i+1) + ")")

this.subject[i].subjectmarks = input.nextInt();

System.out.println ("Enter the credits  
for subject " + (i+1) + ")")

this.subject[i].credits = input.nextInt();

while (this.subject[i].subjectmarks > 100)

System.out.println ("Marks cannot be  
greater than 100. Enter valid marks.")  
(this.subject[i].subjectmarks = 0);

} if (this.subject[i].subjectmarks == 100)

this.subject[i].grade = 'D';

else if (this.subject[i].subjectmarks < 40)

this.subject[i].grade = 'O';

else

this.subject[i].grade = 'A';

(first)(this.subject[i].subjectmarks / 10) + 1

}

}

~~int totalredits = 0;~~  
~~for (i=0; i<8; i++)~~

this.SGPA[i] = this.subject[i].grade

\* this.subject[i].credits;

totalredits = totalredits + this.subject[i].credits;

}

② ② Enr.SGPA = this.SGPA / totalredits ??

class StudentMain

{ public static void main(String args[])

student s = new Student();

s.getStudentDetails();

s.getMarks();

s.computeSGPA();

System.out.println("Name : " + student.name + " " + s.name);

System.out.println("Name : " + s.name + " " + student.name + " " + s.SGPA);

System.out.println("Name : " + s.name + " " + student.name + " " + s.SGPA);

}

}

### OUTPUT

Naveen Ramkumar

VSN: IBM22CS173

Enter the name of the student.

Praveen

Enter the USN of the student.

IBM22CS173

Enter the marks for project 1

89

Enter the marks for project 1

4

Enter the marks for project 2

90

Enter the marks for project 2

4

Enter the marks for project 3

67

Enter the marks for project 3

3

Enter the marks for subject A

60

Enter the marks for subject A

3

Enter the marks for subject B

76

Enter the marks for subject C

3

Enter the marks for subject D

87

" " English " " 6

1

" " marks " " 7

97

" " English " " 7

1

" " marks " " 8

78

" " English " " 8

1

Student name: Parmen

Student ID: 13M22L5522

Q.SGPA of the student: 8.05

Q.SGPA  
19/2/23

# Source Code:

```
import java.util.Scanner;
class Subject
{
    int subjectMarks;
    int credits;
    int grade;
}
class Student
{
    int i;
    Subject[] subject=new Subject[9];
    String name;
    String usn;
    float SGPA=0;
    Student()
    {
        for(i=0;i<9;i++)
            subject[i]=new Subject();
    }
    Scanner s=new Scanner(System.in);
    void getStudentDetails()
    {
        System.out.println("Enter the name of the student.");
        this.name=s.nextLine();
        System.out.println("Enter the usn of the student.");
        this.usn=s.nextLine();
    }
    void getMarks()
    {
        for(i=0; i<8 ;i++)
        {
            System.out.println("Enter the marks for subject "+(i+1));
            this.subject[i].subjectMarks=s.nextInt();
            System.out.println("Enter the credits for subject "+(i+1));
            this.subject[i].credits=s.nextInt();
            while(this.subject[i].subjectMarks>100)
            {

```

```

        System.out.println("Marks cannot be greater than 100.
Enter valid marks.");
        this.subject[i].subjectMarks=s.nextInt();
    }
    if(this.subject[i].subjectMarks==100)
        this.subject[i].grade=10;
    else if(this.subject[i].subjectMarks<40)
        this.subject[i].grade=0;
    else

        this.subject[i].grade=(int)(this.subject[i].subjectMarks/10)+1;
    }
}
void computeSGPA()
{
    int total_credits=0;
    for(i=0;i<8;i++)
    {
        this.SGPA+=this.subject[i].grade*this.subject[i].credits;
        total_credits=total_credits+this.subject[i].credits;
    }
    this.SGPA=this.SGPA/total_credits;
}
class Student_Main
{
    public static void main(String args[])
    {
        Student s1=new Student();
        s1.getStudentDetails();
        s1.getMarks();
        s1.computeSGPA();
        System.out.println("\n\n"+ "Student name : "+s1.name);
        System.out.println("\n"+ "Student usn : "+s1.usn);
        System.out.println("\n"+ "SGPA of the student : "+s1.SGPA);
    }
}

```

# Lab Program 3

Lab - (3) - 26/12/2023

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

Solution:

```
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 = "Book Price: " + this.price + "\n";  
}
```

numPages" Number of Pages: " + the numPages "m";  
returns (name + author + price + numPages))

}

Class Book - Main

{ public static void main (String args [])

Scanner s = new Scanner (System.in);

int n;

String name;

String author;

int price;

int numPages;

int i;

System.out.println ("Enter the number of books  
to be entered");

n = s.nextInt();

s.nextLine();

Books

Books [] b = new Books (n);

System.out.println ("m");

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

System.out.print ("Enter the name  
of the book " + (i + 1));

name = s.nextLine();

System.out.print ("Enter the name  
of the author of the book " + (i + 1));

author = s.nextLine();

System.out.print ("Enter the price  
of the book " + (i + 1));

price = s.nextInt();

s.nextLine();

```
System.out.println("Enter the number  
of pages in the book " + (i + 1));  
numPages = s.nextInt();  
s.nextLine();  
b[i] = new Book(s.nextLine(), author, price,  
numPages);  
System.out.println("The book  
details are: " + b[i]);  
for (i = 0; i < n; i++)
```

```
}  
System.out.println(  
(b[i].toString() + "\n")  
)
```

3

### OUTPUT

Enter the number of books to be entered.

3

Enter the name of the book 1

Martins Station

Enter the name of the author of the book 1

Elizabeth Duley

Enter the price of book 1

300

Enter the number of pages in the book 1

150

Enter the name of the book 2

Pony Johnson

Enter the name of the author of the book 2  
Rich Rordan

Enter the price of the book 2  
500

Enter the number of pages in the book 2  
300

Enter the name of the books

Harry Potter

Enter the name of the author of the book 3  
J K Roring

Enter the price of the book 3  
500

Enter the number of pages in the book 3  
500

The book details are:

Book name: Geronimo Stilton

Author name: Elizabeth Davis

Book price: 300

Number of pages: 150

~~Price~~  
~~200~~

~~Page 300~~

Book name: Harry Potter

Author name: J K Roring

Book price: 500

Number of pages: 300

CLASSMATE

Date

Page

Book Name: Harry Potter

Author Name: J K Rowling

Book Price: 500

Number of Pages: 500

## Source Code:

```
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="Book 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;
        String name;
        String author;
        int price;
        int numPages;
        int i;
        System.out.println("Enter the number of books to be
entered.");
```

```

n=s.nextInt();
s.nextLine();
Books[] b=new Books[n];
System.out.println("\n");
for(i=0;i<n;i++)
{
    System.out.println("Enter the name of the book
"+(i+1));
    name=s.nextLine();
    System.out.println("Enter the name of the
author of the book "+(i+1));
    author=s.nextLine();
    System.out.println("Enter the price of the book
"+(i+1));
    price=s.nextInt();
    System.out.println("Enter the number of pages
in the book "+(i+1));
    numPages=s.nextInt();
    s.nextLine();
    b[i]=new Books(name,author,price,numPages);
    System.out.println("\n");
}
System.out.println("\n"+"The Book details
are:"+ "\n\n");
for(i=0;i<n;i++)
{
    System.out.println(b[i].toString()+"\n\n");
}
}

```

# Lab Program 4

Lab-② - 2/1/2024

Date \_\_\_\_\_  
Page \_\_\_\_\_

Lab Program A:

```
import java.util.Scanner;  
class InputScanner  
{  
    Scanner scanner = new Scanner(System.in);  
}  
abstract class Shape extends InputScanner  
{  
    int base;  
    int height;  
    abstract void printArea();  
}  
class Rectangle extends Shape  
{  
    void printArea()  
    {  
        System.out.println("Area of Rectangle = " +  
            (base * height));  
    }  
}  
class Triangle extends Shape  
{  
    void printArea()  
    {  
        System.out.println("Area of Triangle = " +  
            ((base * height) / 2));  
    }  
}  
class Circle extends Shape
```

```
class Circle extends Shape
```

```
{ void printArea()
```

```
{ System.out.println("Area of Circle = " + (3.14 * base * base))  
}
```

```
}
```

```
class MainClass
```

```
{ static void input( Rectangle r, Triangle t, Circle c )
```

```
{ System.out.print("Enter the dimensions of the  
rectangle : ");  
r.length = s.nextInt();  
r.height = r.nextInt();
```

```
System.out.print("Enter the dimensions of  
the triangle : ");  
t.base = t.nextInt();  
t.height = t.nextInt();
```

```
System.out.print("Enter the dimension  
of the circle : radius.");  
c.radius = c.nextInt();
```

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

~~Rectangle r = new Rectangle();~~

~~Triangle t = new Triangle();~~

~~Circle c = new Circle();~~

~~input( r, t, c );~~

~~r.printArea();~~

~~t.printArea();~~

~~c.printArea();~~

```
}
```

output:

Enter the dimensions of the rectangle

2 3

Enter the dimensions of the triangle

2 4

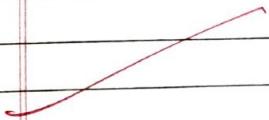
Enter the dimensions of the circle

3

Area of the rectangle = 6

Area of Triangle = 4

Area of Circle =  $28.250000000000002$



Ans  
21/2/2024

## Source Code:

```
import java.util.Scanner;

class InputScanner

{

    Scanner s=new Scanner(System.in);

}

abstract class shape extends InputScanner

{

    int base;
    int height;

    abstract void printArea();

}

class Rectangle extends shape

{

    void printArea()

    {

        System.out.println("Area of Rectangle="+ (base*height));

    }

}

class Triangle extends shape

{
```

```
void printArea()

{

    System.out.println("Area of Triangle="+(base*height)/2));
}

}

class Circle extends shape

{

void printArea()

{

    System.out.println("Area of Circle="+(3.14*base*base));
}

}

class MainClass

{

    static void input(Rectangle r, Triangle t, Circle c)

    {

        System.out.println("Enter the dimensions of the
rectangle ");

        r.base=r.s.nextInt();
        r.height=r.s.nextInt();
        System.out.println("Enter the dimensions of the triangle ");
        t.base=t.s.nextInt();
        t.height=t.s.nextInt();
        System.out.println("Enter the dimensions of the circle ");
        c.base=c.s.nextInt();
    }

    public static void main(String args[])

    {
        Rectangle r=new Rectangle();
```

```
Triangle t=new Triangle();
Circle c=new Circle();
input(r, t, c);
r.printArea();
t.printArea();
c.printArea();

}
```

# Lab Program 5

Date \_\_\_\_\_  
Page \_\_\_\_\_

Lab (5) - 9/1/2024

Lab Program 5

Sonika Lotlikar

```
import java.util.Scanner  
import java.lang.Math  
class Account
```

```
{  
    String accountname;  
    int accountnumber;  
    float balance = 0;  
    void deposit (float amount)
```

balance = balance + amount;

}

```
} class has -> it extends Account
```

```
{  
    int penaltyimposed = 0;  
    void chekMinBalance()
```

}

```
{ if (balance < 300)
```

System.out.println ("Your current account balance is less than Rs. 300. A fine of Rs.

50/- is imposed. ");

balance = balance - 50;

penaltyimposed++;

}

else

{

penaltyimposed=0;

}

}

work with drawl ( float amount )

{ if ( (balance - amount) > 0 )

{ if ( penaltyAmount == 0 )

    balance = balance - amount

    theMin - minimumBalance() ;

}

else

{

    System.out.println("The penalty  
is already imposed due to lack of  
minimum balance (limit withdrawal)");

}

} do

    System.out.println("Balance is too low  
to withdraw");

}

} class SavingsAccount extends Account

{

    double interestRate = 0.06;

    double calculateInterest ( int years )

    { return (Math.pow(1.06, years) \* balance); }

} work with drawl ( float amount )

{ if ( balance < amount )

    System.out.println("Balance is too less to  
withdraw the required amount"));

}

```
else  
{
```

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

```
}
```

```
}
```

```
class Bank
```

```
{
```

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

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

```
String accountno;
```

```
Customer current = new Customer();
```

```
Save Acc savings = new Save_Acc();
```

```
int choice;
```

```
int amount;ope;
```

```
System.out.println("Enter the account no");
```

```
current.setCustomerNo(sc.nextLine());
```

```
=> next line");
```

```
System.out.println("Enter the account no");
```

```
current.setAccNo(sc.nextLine());
```

```
=> next line");
```

```
System.out.println("Deposit a minimum balance
```

```
of Rs 500 for the current account");
```

```
current.bal=(int)500;
```

```
while(true)
```

```
{
```

~~System.out.println("In ENTRY---");~~~~System.out.println("Enter 1 to deposit~~~~money) 2 to withdraw money) 3 to be~~~~Customer interest for savings account~~~~4 to display the details and balance~~~~and 5 to exist");~~~~System.out.println("Enter your choice");~~

~~(char i < nextInt())  
if (have == 2)~~

{(int amount)}

System.out.println ("Enter 1 to deposit into  
account and enter 2 to deposit in the  
savings account")  
amount = nextInt();  
if (amount == 1)

System.out.println ("Enter the amount to  
deposit into the account")  
amount = nextInt();  
savings.deposit (amount);

}  
else if (amount == 2)

System.out.println ("Enter the amount to  
deposit into the savings account")  
amount = nextInt();  
savings.deposit (amount);

}  
else

System.out.println ("Invalid number  
entered")

}  
else if (have == 2)

{(int amount)}

System.out.println ("Enter 1 to withdraw from  
the account and enter 2 to withdraw from  
the savings account")  
amount = nextInt();

if (allowType == 1)

System.out.println("Enter the amount to withdraw from the account")  
amount = sc.nextInt()  
amount = amount - 1000  
amount = withdraw(amount)

}  
else if (allowType == 2)

System.out.println("Enter the amount to withdraw from the savings account")  
amount = sc.nextInt()  
amount = withdraw(amount)  
amount = amount - 1000  
amount = withdraw(amount)

else

System.out.println("Invalid number entered")

}  
else if (allowType == 3)

int years;  
System.out.println("Enter the number of years to compound")  
years = sc.nextInt()

System.out.println("The balance in your savings account at the end of " + years + " years after compounding annually at the rate of 6% is " + savingsBalanceWithInterest(years))

}  
else if (allowType == 4)

System.out.println("Account Details")  
System.out.println("Customer Name: " + name + " " + address + " " + phoneNo)

```
System.out.println("Account Number:");  
+ current account number)  
System.out.println("Saving Account Number:");  
+ savings balance)  
System.out.println("Current Account Number:");  
+ current balance)
```

```
}  
else if (choice == 3)  
break;
```

```
else
```

```
System.out.println("Invoking number  
entered");  
System.out.println("n/n")
```

```
}
```

```
}
```

Output:

Enter the account number:

Bank

Enter the account number:

123456789

Deposit a minimum of 1000 for the current account

-----

Enter 1 to deposit money, 2 to withdraw money, 3 to

Calculate interest for savings account, 4 to display the  
details and balance and 5 to exit.

Enter your choice

Enter to deposit

Enter 1 to deposit in the current account and enter 2  
to deposit in the savings account

Enter the amount to deposit into the current account

200

----- MENU -----  
Enter 1 to 11 11 12 13  
11 12 11 12 + 13  
11 12 11 12 13

Enter your choice

Enter 1 to deposit in the current account and enter 2 to  
deposit in the savings account

2

Enter the amount to deposit into the savings account

500

----- MENU -----  
Enter 1 to 11 11 12 13  
11 12 11 12 11 11  
12 12 12 12 13

Enter your choice A

ACCOUNT DETAILS:

Customer Name: Ram

Account Number: 123456789

Savings Account Balance: 500.0

Current Account Balance: 500.0

----- MENU -----  
Enter 1 to 11 11 12 13  
11 11 12 12 13  
11 12 12 12 13

Enter your choice

2

Enters 10 withdrawal from the current account and enters 2 to withdraw from the savings account

1

Enters the amount to withdraw from the current account  
300

The current account balance is less than Rs 500 Assume  
Money M to 50 is implicit

--- MENU ---  
Enter 1 to withdraw  
" " " " "  
" " " " "  
" " " " "

2 Enter your choice 2

Enter 1 to withdraw  
" " " " "

" " 2 to withdraw from the savings account

2

Enter the amount to withdraw from the savings account  
300

--- MENU ---  
Enter 1 to withdraw  
" " " " "  
" " " " "  
" " " " "

3 Enter your choice

3 Enter the number of years to compound

3

The balance in your savings account at the end of 3  
years after compounding annually at a rate of 6% is Rs 138.2852

--- MENU ---  
Enter 1 to withdraw  
Enter your choice 5

## Source Code:

```
import java.util.Scanner;
import java.lang.Math;
class Account
{
    String customername;
    int accountnumber;
    float balance=0;
    void deposit(float amount)
    {
        balance=balance+amount;
    }
}
class Cur_Acct extends Account
{
    int penaltyimposed=0;
    void check_minimum_balance()
    {
        if(balance<300)
        {
            System.out.println("Your current account balance is
less than Rs.300. A service charge of Rs.50 is imposed. ");
            balance=balance-50;
            penaltyimposed=1;
        }
        else
        {
            penaltyimposed=0;
        }
    }
    void withdrawal(float amount)
    {
        if((balance-amount)>=0)
        {
            if(penaltyimposed==0)
            {
                balance=balance-amount;
                check_minimum_balance();
            }
            else
            {
```

```

        System.out.println("The penalty is already
imposed due to lack of minimum balance. Cannot withdraw.");
    }

}

else
    System.out.println("Balance is too less to withdraw.");
}

}

class Sav_Acct extends Account
{
    double interestrate=0.06;
    double calculate_interest(int years)
    {
        return(Math.pow(1.06,years)*balance);
    }
    void withdrawal(float amount)
    {
        if(balance<amount)
        {
            System.out.println("Balance is too less to withdraw
the required amount.");
        }
        else
        {
            balance=balance-amount;
        }
    }
}
class Bank
{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        String customername;
        Cur_Acct current=new Cur_Acct();
        Sav_Acct savings=new Sav_Acct();
        int choice;
        int accounttype;
        System.out.println("Enter the customer name.");
        current.customername=savings.customername=s.nextLine();
        System.out.println("Enter the account number.");
    }
}

```

```

        current.accountnumber=savings.accountnumber=s.nextInt();
        System.out.println("Deposit a minimum balance of Rs.300 for
the current account.");
        current.balance=300;
        while(true)
        {
            System.out.println("-----MENU-----");
            System.out.println("ENTER 1 TO DEPOSIT MONEY, 2
TO WITHDRAW MONEY, 3 TO CALCULATE INTEREST FOR SAVINGS
ACCOUNT, 4 TO DISPLAY THE DETAILS AND BALANCE AND 5 TO
EXIT.");
            System.out.println("ENTER YOUR CHOICE");
            choice=s.nextInt();
            if(choice==1)
            {
                float amount;
                System.out.println("ENTER 1 TO DEPOSIT IN
THE CURRENT ACCOUNT AND ENTER 2 TO DEPOSIT IN THE SAVINGS
ACCOUNT");
                accounttype=s.nextInt();
                if(accounttype==1)
                {
                    System.out.println("Enter the amount to
deposit into the current account.");
                    amount=s.nextFloat();
                    current.deposit(amount);
                }
                else if(accounttype==2)
                {
                    System.out.println("Enter the amount to
deposit into the savings account.");
                    amount=s.nextFloat();
                    savings.deposit(amount);
                }
                else
                    System.out.println("Invalid number
entered.");
            }
            else if(choice==2)
            {
                float amount;

```

```

        System.out.println("ENTER 1 TO WITHDRAW
FROM THE CURRENT ACCOUNT AND ENTER 2 TO WITHDRAW FROM
THE SAVINGS ACCOUNT");
        accounttype=s.nextInt();
        if(accounttype==1)
        {
            System.out.println("Enter the amount to
withdraw from the current account.");
            amount=s.nextFloat();
            current.withdrawal(amount);
        }
        else if(accounttype==2)
        {
            System.out.println("Enter the amount to
withdraw from the savings account.");
            amount=s.nextFloat();
            savings.withdrawal(amount);
        }
        else
            System.out.println("Invalid number
entered.");
    }
    else if(choice==3)
    {
        int years;
        System.out.println("Enter the number of years
to compound.");
        years=s.nextInt();
        System.out.println("The balance in your savings
account at the end of "+years+" years after compounding annually at a
rate of 6% is "+savings.calculate_interest(years));
    }
    else if(choice==4)
    {
        System.out.println("ACCOUNT DETAILS:");
        System.out.println("CUSTOMER
NAME:"+current.customername);
        System.out.println("ACCOUNT
NUMBER:"+current.accountnumber);
        System.out.println("SAVINGS ACCOUNT
BALANCE:"+savings.balance);
    }
}

```

```
        System.out.println("CURRENT ACCOUNT  
BALANCE:"+current.balance);  
    }  
    else if(choice==5)  
        break;  
    else  
        System.out.println("Invalid number entered.");  
    System.out.println("\n\n");  
}  
  
}  
}
```

# Lab Program 6

Lab - (D-23) / 17/2024

Date \_\_\_\_\_  
Page \_\_\_\_\_

Solve Ques.

\* Student.java

package L2E1;

import java.util.Scanner;  
public class Student

protected String name = new String();  
protected String gender = new String();  
public void inputStudentDetails()

```
Scanner sc = new Scanner(System.in);
System.out.println("Enter the name of the student");
name = sc.nextLine();
System.out.println("Enter the gender of the student");
gender = sc.nextLine();
sc.close();
}
```

public void displayStudentDetails()

```
System.out.print("The student's
name is " + name);
System.out.print("The student's name is "
+ name);
System.out.print("The student's
gender is " + gender);
}
```

## \* Internals.java

package LIE;

import java.util.Scanner;

public class Internals extends Student

{ protected int marks[] = new int [5]; }

public void input (Scanner s)

int i;

Scanner sc = new Scanner (System.in);

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

System.out.println ("Enter the LIE

marks for subject " + (i+1));

marks[i] = sc.nextInt());

}

}

}

## \* Externals.java

package SE;

import LIE.Internals;

import java.util.Scanner;

public class Externals extends Internals

{ protected int marks[] = new int [5]; }

protected int finalmarks[] = new int [5];

public void input (Scanner s)

int i;

Scanner sc = new Scanner (System.in);

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

System.out.println ("Enter the SEE marks  
for subject " + (i+1));  
marks[i] = s.nextInt();

```
}
```

~~public void~~

```
public void calculateFinalMarks()
```

```
int i
```

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

finalMarks[i] = (this.marks[i]  
+ tempMarks[i]) / 2;

```
}
```

~~public void~~ displayFinalMarks()

displayStudentDetails()

```
int i
```

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

System.out.println ("Final marks  
of subject " + (i+1) + " is " + finalMarks[i])

```
}
```

```
}
```

\* Internal.java

\* Main.java

import SEE.External;

class Main

{

    public static void main (String args [ ])

        int numofStudents = 2;

        External finalMarks [ ] = new External  
            [ numofStudents ];

        int i;

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

            finalMarks [ i ] = new External();

            finalMarks [ i ].inputStudentDetails();

            System.out.println ("Enter CIE marks");

            finalMarks [ i ].input ( IEMarks );

            System.out.println ("Enter SEE marks");

            finalMarks [ i ].inputSEEmarks();

        }

        System.out.println ("Displaying data");

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

            finalMarks [ i ].calculateFinalMarks();

            finalMarks [ i ].displayFinalMarks();

}

3

Output:

Enter the vno of the student:

1BM22LS173

Enter the name of the student:

Praveen Ramchandray

Enter the semester of the student:

3

Enter CIE marks:

Enter the CIE marks for subject 1:

49

Enter the CIE marks for subject 2:

47

Enter the CIE marks for subject 3:

50

Enter the LIE marks for subject 4:

40

Enter the LIE marks for subject 5:

42

Enter SBE marks:

Enter SBE marks for subject 1:

95

11 11 11 2

47

11 11 11 3

94

11 11 11 4

87

11 11 11 5

40

Enter the vno of the student:

1BM22LS173

Enter the name of the student:

Ganesh Subramanian

Enter the semester of the student.

3

Enter LIE marks

Enter the marks for subject)

38

11

11

2

47

11

11

3

45

11

11

4

42

11

11

5

49

Enter SEE marks

Enter the SEE marks for subject)

79

~~79~~

11

11

\*

11

2

81

11

11

3

93

11

11

4

98

11

11

5

91

Displaying info.

The student's name is 1BM22LS133

The student's name is Pranam Ranawat

The student's semester is 3

Final marks of subject 1 is 96

11 11 11 2 is 95

12 11 11 3 is 99

12 11 11 4 is 83

12 11 11 5 is 87

The student's name is BM2167173 BM2267173

The student's name is GANAV Subramanian

The student's semester is 3

Final marks of subject 1 is 77

11 11 " 2 7 9 )

11 11 " 3 0 9 )

11 11 " 4 0 9 )

11 11 " 5 0 9 4

20  
21  
22

# Source Code

## Internals.java

```
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 the CIE marks for subject
"+(i+1));
            marks[i]=s.nextInt();
        }
    }
}
```

## Student.java

```
package CIE;
import java.util.Scanner;
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 the usn of the student.");
        usn=s.nextLine();
        System.out.println("Enter the name of the student.");
        name=s.nextLine();
    }
}
```

```

        System.out.println("Enter the semester of the student.");
        sem=s.nextInt();
    }
    public void displayStudentDetails()
    {
        System.out.println("The Student's usn is "+usn);
        System.out.println("The Student's name is "+name);
        System.out.println("The Student's semester is "+sem);
    }
}

```

### **Externals.java**

```

package SEE;
import CIE.Internals;
import java.util.Scanner;
public class Externals extends Internals
{
    protected int marks[]=new int[5];
    protected int finalMarks[]=new int[5];
    public void inputSEEmarks()
    {
        int i;
        Scanner s=new Scanner(System.in);
        for(i=0; i<5; i++)
        {
            System.out.println("Enter the SEE marks for subject
p"+(i+1));
            marks[i]=s.nextInt();
        }
    }
    public void calculateFinalMarks()
    {
        int i;
        for(i=0; i<5; i++)
        {
            finalMarks[i]=this.marks[i]/2+super.marks[i];
        }
    }
    public void displayFinalMarks()
    {
        displayStudentDetails();
    }
}

```

```
        int i;
        for(i=0; i<5; i++)
        {
            System.out.println("Final marks of subject "+(i+1)+" is
"+finalMarks[i]);
        }
    }
```

### Main.java

```
import SEE.Externals;
class Main
{
    public static void main(String args[])
    {
        int numOfStudents=2;
        Externals finalMarks[]=new Externals[numOfStudents];
        int i;
        for(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(i=0; i<numOfStudents; i++)
        {
            finalMarks[i].calculateFinalMarks();
            finalMarks[i].displayFinalMarks();
        }
    }
}
```

# Lab Program 7

Lab - (8) - 30/11/2021 to write a program that demonstrates handling of exceptions in inheritance  
Sonnie code:

```
import java.util.Scanner  
class WrongAge extends Exception  
{  
    WrongAge(String s)  
    {  
        super(s);  
    }  
}  
class InputScanner  
{  
    Scanner s;  
    InputScanner()  
    {  
        s = new Scanner(System.in);  
    }  
    int FatherEntersInputScanner  
{  
        int fatherAge;  
        FatherEntersWrongAge  
        {  
            System.out.println("Enter the age of the  
            Father: ");  
            fatherAge = nextInt();  
            if (fatherAge < 0)  
                throw new WrongAge("Age cannot  
                be negative.");  
        }  
    }  
}
```

with display()

System.out.println("The father's age is " +  
fatherAge);

}

else Son extends Father

{

int sonAge;  
sonAge = throw WrongAge;

System.out.println("Enter the age of the  
son:");

sonAge = nextInt();

if (sonAge > fatherAge)

throw new WrongAge("Son's age is

cannot be greater than father's age.");

else if (sonAge < 0)

throw new WrongAge("Age cannot  
be negative.");

}

void display()

System.out.println("The son's age is " +  
sonAge);

}

Class Main

{  
    public static void main (String args[]){  
        System.out.println ("Creating a father class  
            object. ");  
    }  
}

    Father fatherObject = new Father();  
    fatherObject . display();

}  
    {  
        System.out.println ("e");  
    }

}  
    System.out.println ("Creating a Son  
            class object. ");  
    Son  
    {  
        Son sonObject = new Son();  
        sonObject . display();  
    }

}  
    {  
        System.out.println ("e");  
    }

}  
    System.out.println ("");

}

Output:

java Main

Creating a Father class object

Enter the age of the Father:  
-45

Wrong Age: Age cannot be negative

Creating a Son class object

Enter the age of the Father:  
-50

Wrong Age: Age cannot be negative

java Main

Creating a Father class object

Enter the age of the Father:  
43

The father's age is 43

Creating a Son class object

Enter the age of the Father:  
43

Enter the age of the son:

50

Wrong Age: Son's age cannot be greater than father's  
age.

~~Run~~  
~~301120m~~

~~java Main~~

~~Creating a Father class object~~

~~Enter the age of the Father~~

~~43~~

~~The father's age is 43~~

~~Creating a Son class object~~

~~Enter the age of the Father~~

~~50~~

~~Enter the age of the son~~

~~The son's age is 32~~

# Source Code

```
import java.util.Scanner;
class WrongAge extends Exception
{
    WrongAge(String s)
    {
        super(s);
    }
}
class inputscanner
{
    Scanner s;
    inputscanner()
    {
        s=new Scanner(System.in);
    }
}
class Father extends inputscanner
{
    int fatherAge;
    Father() throws WrongAge
    {
        System.out.println("Enter the age of the Father.");
        fatherAge=s.nextInt();
        if(fatherAge<0)
        {
            throw new WrongAge("Age cannot be negative.");
        }
    }
    void display()
    {
        System.out.println("The father's age is "+fatherAge);
    }
}
class Son extends Father
{
    int sonAge;
    Son() throws WrongAge
    {
```

```
System.out.println("Enter the age of the son.");
sonAge=s.nextInt();
if(sonAge>fatherAge)
{
    throw new WrongAge("Son's age cannot be greater
than father's age");
}
else if(sonAge<0)
{
    throw new WrongAge("Age cannot be negative");
}
}
void display()
{
    System.out.println("The son's age is "+sonAge);
}
}
class Main
{
    public static void main(String args[])
    {
        System.out.println("Creating a Father class object.");
        try
        {
            Father fatherobject=new Father();
            fatherobject.display();
        }
        catch(WrongAge e)
        {
            System.out.println(e);
        }
        System.out.println("Creating a Son class object.");
        try
        {
            Son sonobject=new Son();
            sonobject.display();
        }
        catch(WrongAge e)
        {
            System.out.println(e);
        }
    }
}
```

}

# Lab Program 8

Lab-(7)-6/2/2021

Q

Write a program which has two threads, one displaying  
"BMS College of Engineering" one every ten seconds and another  
displaying "CS" one every ten seconds.

Solution:

Using my own extends Thread

String message;

int waitTime;

myThread (String message int waitTime)

{this.message = message}

this.waitTime = waitTime

}

public void run()

while (true)

System.out.println(message)

try

Thread.sleep(waitTime)

catch (InterruptedException e)

System.out.println("The  
thread is interrupted")  
break;

}

}

}

)

Using myThread - main

public static void main (String args [] )

myThread thread1 = new myThread ("Bm")

( "My of Engineering" , 10000 ) ;

myThread thread2 = new myThread ("CSE")

( 2000 ) ;

thread1.start () ;

thread2.start () ;

try

{ Thread.sleep ( 5000 ) ; }

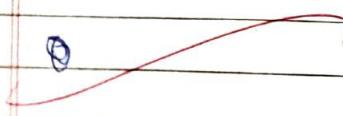
} catch ( InterruptedException e )

System.out.println (" Main thread  
is interrupted " ) ;

}  
thread1.interrupt () ;  
thread2.interrupt () ;

)

o



Turn  
Page

Output:

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

CSE

The Aword is interrupted.  
The Aword is interrupted.

By 6/2/2021

## Source code:

```
class myThread extends Thread
{
    String message;
    int wait_Time;
    myThread(String message, int wait_Time)
    {
        this.message=message;
        this.wait_Time=wait_Time;
    }
    public void run()
    {
        while(true)
        {
            System.out.println(message);
            try
            {
                Thread.sleep(wait_Time);
            }
            catch(InterruptedException e)
            {
                System.out.println("The thread is interrupted.");
                break;
            }
        }
    }
}

class My_Thread_Main
{
    public static void main(String args[])
    {
        myThread thread1=new myThread("BMS College of
Engineering", 10000);
        myThread thread2=new myThread("CSE", 2000);
        thread1.start();
        thread2.start();
    }
}
```

```
try
{
    Thread.sleep(30000);
}
catch(InterruptedException e)
{
    System.out.println("Main thread is interrupted");
}
thread1.interrupt();
thread2.interrupt();

}
```

# Lab program 9

Date \_\_\_\_\_  
Page \_\_\_\_\_

2. Synchronized program

Source code:

class A

{

```
    int n;
    boolean valueSet = false;
```

```
    synchronized (int get){}
```

```
        while (!valueSet)
```

```
            try {
```

```
                wait();
```

```
}
```

```
        catch (InterruptedException e)
```

```
    }
```

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

```
    System.out.println ("but :" + n);
```

```
    valueSet = true;
```

```
    notify();
```

```
    return n;
```

```
} // synchronized void get (int n)
```

```
while (valueSet)
```

```
try {
```

```
    wait();
```

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

```
    }
```

```
System.out.println ("Interrupted
```

Execution caught "])

}  
this.n=n)

valuetSet = this.

System.out.println("Int: " + n))  
n++])

}

class Producer implements Runnable

{

(Q q)  
Producer(Q q)  
{

this.q=q

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

}

public void run()

int i=0;

while(i<15)

{

q.put(i+1)

}

}

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 n = f.get()  
i++
```

}

]

else TestSyncronized

public static void main (String args [])

```
Q q = new (Q);  
new Producer (q);  
new Consumer (q);  
System.out.println ("Press Control-L to  
stop.");
```

J

J

Output:

Int: 0

Int: 0

Int: 1

Int: 1

Int: 2

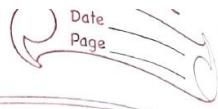
Int: 2

Int: 3

Int: 3

Int: 4

Int: 4



Part : 5  
GPart : 5  
Part : 6  
GPart : 6  
Part : 7  
GPart : 7  
Part : 8  
GPart : 8  
Part : 9  
GPart : 9  
Part : 10  
GPart : 10  
Part : 11  
GPart : 11  
Part : 12  
GPart : 12  
Part : 13  
GPart : 13  
Part : 14  
GPart : 14

Page No. 1 / 2024

Q Write a given program to demonstrate the deadlock.

Sample code:

class A

{

    synchronized void foo(B b)

        System.out.println("Thread " + Thread.currentThread().getName() + " got lock  
        + " + b.foo() + ")")  
    try

    {

        Thread.sleep(1000)

    } catch (InterruptedException e)

        System.out.println("A interrupte")

        System.out.println("B trying to access B.list()")  
        b.list()

    } synchronized void bar()

        System.out.println("Inside A.list()")

    }

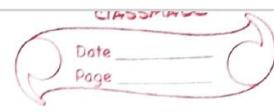
class B

{

    synchronized void bar(A a)

        System.out.println("Inside B.list()")  
        System.out.println("A.list() : " + a.list())  
        + " is accessing B.list()")  
        a.list()

```
try
{
    Thread sleep(1000);
}
catch(InterruptedException)
{
    System.out.println("B interrupted");
}
System.out.println("Trying to access
        A's code");
A code();
}
synchronization void test()
{
    System.out.println("Inside B's code");
}
class DeadlockMainProgram
{
    A a = new A();
    B b = new B();
    DeadlockMainProgram()
    {
        Thread t1 = new Thread(() -> a.stat());
        Thread t2 = new Thread(() -> b.stat());
        t1.start();
        t2.start();
    }
    synchronized void stat()
    {
        System.out.println("B's lock is maintained");
        b.baz();
        System.out.println("B's lock is released");
    }
}
```



private static void main(String args[]){}

new DoublyLinkList.mainProgram()

Output:

mainThread is calling B.list()

mainThread is calling A.list()

Calling Thread is calling B.list()

Trying to access A.list

Trying to access B.list

Re  
2/2/2024

# Source code:

**Deadlock\_Main\_Program.java**

```
class A
{
    synchronized void foo(B b)
    {
        System.out.println(Thread.currentThread().getName()+" is
accessing A.foo()");
        try
        {
            Thread.sleep(1000);
        }
        catch(InterruptedException e)
        {
            System.out.println("A interrupted.");
        }
        System.out.println("Trying to access B.last()");
        b.last();
    }
    synchronized void last()
    {
        System.out.println("Inside A.last()");
    }
}
class B
{
    synchronized void bar(A a)
    {
        System.out.println(Thread.currentThread().getName()+" is
accessing B.foo()");
        try
        {
            Thread.sleep(1000);
        }
        catch(InterruptedException e)
        {
            System.out.println("B interrupted.");
        }
    }
}
```

```

        System.out.println("Trying to access A.last()");
        a.last();
    }
    synchronized void last()
    {
        System.out.println("Inside B.last()");
    }
}
class Deadlock_Main_Program implements Runnable
{
    A a=new A();
    B b=new B();
    Deadlock_Main_Program()
    {
        Thread.currentThread().setName("MainThread");
        Thread t=new Thread(this, "RacingThread");
        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_Main_Program();
    }
}

```

### TestSynchronization1.java

```

class Q
{
    int n;

    boolean valueSet = false;

    synchronized int get()

```

```

    {
        while(!valueSet)
        try {
            wait();
        }
        catch(InterruptedException e)
        {
            System.out.println("InterruptedException caught");
        }

        System.out.println("Got: " + n);
        valueSet = false;
        notify();

        return n;
    }
}

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

}

class Producer implements Runnable
{
    Q q;
    Producer(Q q)
    {

```

```
    this.q = q;
    new Thread(this, "Producer").start();

}

public void run()
{
    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 TestSynchronization1
```

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

# Lab program 10

Lab (10) - 20/2/2024

Q 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 num1 / num2 is displayed in the result field when the divide button is clicked. If num1 or num2 are not integers, the program will throw a `NumberFormatException`. If num2 has zero, the program will throw an `ArithmeticException`, displaying the exception in a message dialog box.

Solve code:

```
import javax.swing.*;  
import java.awt.*;  
import java.awt.event.*;  
class SwingDemo {  
    SwingDemo() {  
        JFrame frame = new JFrame("Divide App");  
        frame.setSize(275, 350);  
        frame.setLayout(new FlowLayout());  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        JLabel num1 = new JLabel("Enter the  
dividend and divisor.");  
        JTextField num1f = new JTextField("0");  
        JTextField num2f = new JTextField("0");  
        JButton divide = new JButton("Divide");  
        divide.addActionListener(new ActionListener() {  
            public void actionPerformed(ActionEvent e) {  
                String num1Str = num1f.getText();  
                String num2Str = num2f.getText();  
                int num1Int = Integer.parseInt(num1Str);  
                int num2Int = Integer.parseInt(num2Str);  
                if (num2Int == 0) {  
                    JOptionPane.showMessageDialog(frame,  
                        "Division by zero is not allowed.",  
                        "Error", JOptionPane.ERROR_MESSAGE);  
                } else {  
                    int result = num1Int / num2Int;  
                    resultLabel.setText(String.valueOf(result));  
                }  
            }  
        });  
        resultLabel = new JLabel("Result");  
        frame.add(num1);  
        frame.add(num1f);  
        frame.add(num2f);  
        frame.add(divide);  
        frame.add(resultLabel);  
        frame.setVisible(true);  
    }  
}
```

y)   
 a) tf: add Antecedent(s) { }  
 b) tf: add Antecedent(s) { }  
 button: add Antecedent(s) {new Antecedent} { }  
  
 further work action required  
(prioritize out)  
 try 1

int A::Integer::Int  
{ int getTest(); }

int biIntegers::sortInt  
(bigInt &list))

into  $\frac{a}{b}$

$$w_{AB} \sin \theta = "t+a"$$

black dot test (in)

$\text{arg} \frac{1}{z_1 z_2}$  (in  $\text{arc tan}$ )

$$(\mathcal{A}^{\alpha_1}, \mathcal{B}^{\alpha_1})$$

1 2 3 4 5

Number Four Forester

anke set test (111)

black - set Text ())) ))

$\text{g/mol} \times \text{Molar Mass}$

models get lost ("")

exist list ("B should  
be non zero"))

})  
})  
} final Variable (final)

} public ActionView min (String arg) {  
StringList invokeList (new Runnable {  
public void run () {  
new String ());  
}}})

Output:

Driver App      -  X

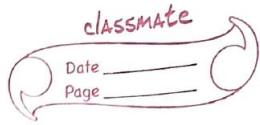
Prints the driver and driver

56

32

calculus

D-36 15:32 AM



Dihedral Angle -  X

~~→~~) Finds only integers

Enters the divisor and dividend.

abc | def

Cultivate

Dindin App -  X

## Bhāskarānugraha

Point to the driver and market

25

6

Chimptec

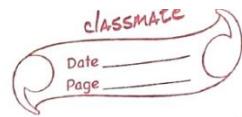
Polymer

# Report writing



Print on the function used in the program

- \* JFrame(): It is a constructor for the JFrame class which is part of javax.swing package (which is part of the Java Swing Standard Widget Toolkit). It name starts with a non-Space with the specific tool.
- \* setSize(): The setSize method of a JFrame object is used to object.setSize (int width, int height) and sets the size of the JFrame window as width pixels wide and height pixels high.
- \* setLayout(): The setLayout method sets the layout of the JFrame object, taking a LayoutObject as a parameter. The layout object can be:
  - \* FlowLayout(): arranges components one after the other in a sequence. The default alignment is center and the horizontal and vertical gaps will be zero.
- \* setDefaultCloseOperation(): This method is specific what action is supposed to be performed when the close button is clicked. The options are:
  - \* Frame.EXIT\_ON\_CLOSE: Exit the application
  - \* Frame.HIDE\_ON\_CLOSE: Hide the frame but keeps the application running.
  - \* Frame.DISPOSE\_ON\_CLOSE: Dispose of the frame object but keep the application running.
  - \* Frame.DO\_NOTHING\_ON\_CLOSE: Ignore the click.



- \* Label): It is used to create an object of a component for placing text in a container. It is used to display a single line of read-only text. The text can be changed by an application but a user cannot edit it directly. It extends the Component class.
- \* Textfield): It is used to create an object for an editable text box. It takes the number of columns as an argument. For example, TextField(8) creates a text box of 8 columns.
- \* Button: Creates a Button object with an optional argument of the text to be displayed inside the button.
- \* ActionListener): It is used to add an action event to check for form a particular object lesser than object is clicked. It takes an ActionListener object as argument and when the object is clicked the action performed method is executed.
- \* setText(): The setText() method of a Label object sets the text content of that label to the specified text.

✓ Improve your handwriting

Rohan

# Source code

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class SwingDemo{
    SwingDemo(){
        JFrame jfrm=new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JLabel jlab=new JLabel("Enter the divider and dividnt:");
        JTextField ajtf=new JTextField(8);
        JTextField bjtf=new JTextField(8);
        JButton button=new JButton("Calculate");
        JLabel err=new JLabel();
        JLabel alab=new JLabel();
        JLabel blab=new JLabel();
        JLabel anslab=new JLabel();
        jfrm.add(err);
        jfrm.add(jlab);
        jfrm.add(ajtf);
        jfrm.add(bjtf);
        jfrm.add(button);
        jfrm.add(alab);
        jfrm.add(blab);
        jfrm.add(anslab);
        ActionListener l=new ActionListener(){
            public void actionPerformed(ActionEvent evt){
                System.out.println("Action event from a text
field");
            }
        };
        ajtf.addActionListener(l);
        bjtf.addActionListener(l);
        button.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent evt){
                try{
                    int a=Integer.parseInt(ajtf.getText());

```

```
        int b=Integer.parseInt(bjtf.getText());
        int ans=a/b;
        alab.setText("\nA="+a);
        blab.setText("\nB="+b);
        anslab.setText("\nAns="+ans);
    }
    catch(NumberFormatException e){
        alab.setText("");
        blab.setText("");
        anslab.setText("");
        err.setText("Enter only integers");
    }
    catch(ArithmaticException e){
        alab.setText("");
        blab.setText("");
        anslab.setText("");
        err.setText("B should be non zero");
    }
}
});
jfrm.setVisible(true);
}
public static void main(String args[]){
    SwingUtilities.invokeLater(new Runnable(){
        public void run(){
            new SwingDemo();
        }
    });
}
}
```