

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:

Pranav Hegde

(1BM22CS202)

Department of Computer Science and Engineering, B.M.S

College of Engineering,

Bull Temple Road, Basavanagudi, Bangalore, 560 019 2023-2024.

Quadratic Equation $an^2 + bn + c = 0$

```
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("This is not quadratic  
Equation");
```

```
    System.out.println("Enter non zero  
values 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)  
{
```

$$r1 = ((-b) + (\text{Math.sqrt}(d))) / (\text{double}(2 * a));$$

$$r2 = ((-b) - (\text{Math.sqrt}(d))) / (\text{double}(2 * a));$$

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

System.out.println("Root 1 = " + r1 + " and "
+ r2);

```
}
```

```
else if (d < 0)  
{
```

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

$$r1 = (-b) / (2 * a);$$

$$r2 = \text{Math.sqrt}(-b) / (2 * a);$$

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();

```
}
```

```
}
```

```
}
```

Output:

(i) Enter the coefficients of a, b, c

0 2 3

Not a quadratic equation

Enter a non-zero value of a.

(ii) Enter the coefficients of a, b, c

1 2 1

Roots are real and Equal

Root 1 = Root 2 = -1

(iii) Enter the coefficients of a, b, c

1 - 3 2.

Roots are real & distinct

Root 1 = 2 Root 2 = 1

(iv) Enter the coefficient of a, b, c

1 1 2.

Roots are Imaginary

Root 1 = 0.0 + i 0.322875

Root 2 = 0.0 - i 0.322875

10/10/23

Calculation CGP & SGPA

```
Import java.util.Scanner;
class Subject
{
    int Subject Marks;
    int Credits;
    int grade;
}
public class Student
{
    Subject [] Subject;
    String name;
    String usn;
    double SGPA;
    Scanner s;
    Student ()
    {
        int i;
        Subject = new Subject [9];
        for (i=0; i<9; i++)
            Subject [i] = new Subject ();
        s = new Scanner (System.in);
    }
    public void get Student Details()
    {
        System.out.println ("Enter Name");
        name = s.nextLine ();
        System.out.println ("Enter USN");
        usn = s.nextLine ();
    }
}
```

```
public void getmarks()
{
    for (int i = 0; i < 8; i++)
    {
        System.out.print("Enter marks for subject " + (i + 1) + ": ");
        subject[i].subjectMarks = sc.nextInt();
        System.out.print("Enter Credits for subject " + (i + 1) + ": ");
        if (subject[i].subjectMarks >= 100)
            System.out.println("Invalid marks entered");
        else if (subject[i].subjectMarks >= 90)
            subject[i].grade = 10;
        else if (subject[i].subjectMarks >= 80)
            subject[i].grade = 9;
        else if (subject[i].subjectMarks >= 70)
            subject[i].grade = 8;
        else if (subject[i].subjectMarks >= 60)
            subject[i].grade = 7;
    }
}
```

else if (subject[i].Subject Marks >= 50)

{ subject[i].grade = 6;

else if (subject[i].Subject Marks >= 40)

{ subject[i].grade = 0;

}

}

public void compute SGPA()

{ double total credits = 0;

double total grade Points = 0;

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

{

total credits += Subject[i].credits;

total gradePoints += Subject[i].grade

subject[i].credits;

}

SGPA = total grade Points / Total Credits;

}

Public static void main (string [] args)

{ Student S1 = new Student();

S1. get student Details();

S1. get + marks();

S1. compute SGPA();

System.out.println (" Name :" + S1.name);

System.out.println (" USN :" + S1.USN);

System.out.println (" SGPA :" + S1.SGPA);

}

Output:

Enter Name: Pavan Hegde
Enter USN: IBM22CS202
Enter Marks for Subject 1: 75
Enter Credits for Subject 1: 4
Enter Marks for Subject 2: 58
Enter Credits for Subject 2: 4
Enter Marks for Subject 3: 75
Enter Credits for Subject 3: 3
Enter Marks for Subject 4: 54
Enter Credits for Subject 4: 3
Enter Marks for Subject 5: 61
Enter Credits for Subject 5: 3
Enter Marks for Subject 6: 76
Enter Credits for Subject 6: 1
Enter Marks for Subject 7: 77
Enter Credits for Subject 7: 1
Enter Marks for Subject 8: 91
Enter Credits for Subject 8: 1

Name: Pavan Hegde

USN: IBM22CS202

SGPA: 7.25

26/12/23

Book ID, Name

import java.util.Scanner;

class Book {

{

String name;

String author;

int price;

int numPages;

~~public Book (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;

{

public class Main

{
 public static void main (String [] args)

 Scanner s = new Scanner (System . in);

 int n;

 String name;

 String author;

 int price;

 int numPages;

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

 n = s . nextInt ();

 Book [] b = new Book [n];

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

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

 name = s . next ();

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

 author = s . next ();

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

 price = s . nextInt ();

 System . out . println ("Enter the number
 of page : ");

 numPages = s . nextInt ();

b[i]) -> new Book (name, author, price,
numPages);

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

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

Output

Enter the number of books:

21

Enter the name of book

ABD

Enter the author of book:

R Krishna

Enter the price of the book:

300

Enter the number of pages of book.

450

Book name: ABD

Author name: R Krishna

Price: 300

Number of pages: 450

Q) Abstract class named shape and method name printArea().

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

```
abstract class Shape
```

```
{
```

```
    int length;
```

```
    int breadth;
```

```
    abstract void printArea();
```

```
}
```

```
class Rectangle extends Shape
```

```
{
```

```
    void printArea()
```

```
{
```

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

```
    System.out.println("Enter length and  
breadth of rectangle :");
```

```
    length = s.nextInt();
```

```
    breadth = s.nextInt();
```

```
    System.out.println("The Area of  
Rectangle is " + (length * breadth));
```

```
}
```

```
}
```

```
class Triangle extends Shape
```

```
{
```

```
    void printArea()
```

```
{
```

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

```
    System.out.println("Enter the dimension  
of Triangle :");
```

~~length = s.nextInt();~~~~breadth = s.nextInt();~~

```
System.out.println("The Area of Triangle is  
+ (0.5 * length * breadth));
```

```
3  
class Circle extends Shape
```

```
{  
    void printArea()
```

```
Scanner s = new Scanner(System.in);  
System.out.println("Enter the dimension  
of circle :");
```

```
length = s.nextInt();
```

```
System.out.println("The Area of Circle  
is " + (3.14 * length * length));
```

```
3  
public class Main
```

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

```
{  
    shape shapes = s1;
```

```
s1 = new Rectangle();
```

```
s1.printArea();
```

```
s1 = new Triangle();
```

```
s1.printArea();
```

```
s1 = new Circle();
```

```
s1.printArea();
```

Output

Enter the dimensions of the Rectangle

7 6

Area of rectangle is 42

Enter the dimensions of Triangle

5 8

Area of Triangle 20

Enter the dimension of circle:

7

Area of Circle is 153.93

a/1/24

Create class Account that stores customer name, account number and type of account.

Check for minimum balance, impose penalty if necessary and update the balance.

import java.util.Scanner;

class Account

{

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

Account (String customerName, int accountNumber, String accountType, double balance)

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

{

void deposit (double amount)

{

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

{

void displayBalance()

{

System.out.println ("Balance: " + balan-

c) i)

3

void withdraw (double amount)

{

if (balance - amount < 0)

{

System.out.println ("Insufficient
balance");
return;

{

balance -= amount;

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

{

3

m) class Saving Account extends Account

{

Saving Account (String customerName,
int accountNumber, String accountType,
double balance)

{

Super (customerName, accountNumber,
accountType, balance);

{

void compoundInterest ()

{

double rate = 0.05;

double time = 1.0;

double interest = balance * Math.

power

new (+rate, time) - balance;

balance + = interest;

System.out.println ("Interest of " + interest
+ " added");

{

void withdraw (double amount)

{

if (balance - amount < 0)

{

System.out.println ("Insufficient
balance");

return;

{

balance - = amount;

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

{

class Current Account extends Account

{

double minimum Balance = 1000;

double service Charge = 50;

Current Account (String Customer Name,
int account Number, String account Type,
double balance)

{

Super (Customer Name, account
Number, account Type, balance);

{

```
void withdraw (double amount)
{
    if (balance - amount < minimum
        Balance)
    {
        System.out.println ("Insufficient
            balance");
        return;
    }
    balance -= amount;
    System.out.println ("Withdrawal
        of " + amount + " Successful");
}
```

```
void impose service charge ()
```

```
{
    if (balance < minimum Balance)
    {
        balance -= Service charges;
        System.out.println ("Service
            charge of " + Service charge + " Imposed");
    }
}
```

```
public class Bank
```

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

```
    Scanner s = new Scanner (System.in);
    System.out.println ("Enter customer
        name:");
    String CustomerName = s.nextLine ();
    System.out.println ("Enter account
        number:");
}
```

```
int accountNumber = scanner.nextInt();
System.out.println("Enter account type
(Savings / current): ");
String accountType = s.next();
System.out.println("Enter initial
balance: ");
double balance = s.nextDouble();
```

Account ac;

if (accountType.equals("Savings"))

ac = new Savings Account
(customerName, accountNumber,
accountType, balance);

}
else

ac = new Current Account
(customerName, accountNumber,
accountType, balance);

while (true)

{

System.out.println("1. Deposit");

System.out.println("2. Display balance");

System.out.println("3. Compute and
deposit interest");

System.out.println("4. Withdrawal");

System.out.println("5. Exit");

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

int choice = s.nextInt();

switch (choice)

{ case 1:

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

double amount = s.nextDouble();

account.deposit(amount);

break;

case 2:

account.displayBalance();

break;

case 3:

if (account instanceof SavingAccount)

((Saving Account) account).compound
Interest();

}

else

{

System.out.println ("Interest not
available for current account");

}

break;

case 4:

System.out.println ("Enter amount
to withdraw");

amount = s.nextDouble();

account.withdraw(amount);

if (account instanceof Current
Account)

{

((Current Account) account).impose
ServiceCharge();

break;

case 5:
System.exit(0);

Output:

Enter customer name : Rajesh
Enter account number 38.
Enter customer name : Achintya
Enter account number 28

MENU

1. Enter the amount to deposit
2. Display Amount
3. Compute and deposit Interest
4. Withdrawal.
5. Exit

16.1.24

Strings

- ① Various string constructors.

Output:

Hello, Earth ! Java :

- ② String length, string literal, string concat

String Name : str.length()
String Name : str.concat(" ")

Output:

Length = 11

Concatenated = Paravartilegde Lost.

- ③ ToString().

this.name = name.

public String toString()

Output:

College : Sport.

- ④ getChars();

Extracting BMSCE from "Welcome to BMSCE"

Using char[] target = new char[5]

Extracting : BMSCE

⑥ `getBytes()`, `toCharArray()`

~~String Name~~

~~bytes [3] bytes = text. getBytes();~~

~~char [7] chars = text. getBytes(). toCharArray();~~

Output: "H, Earth!!"

⑦ Check the following

B msce equals Bmsce

B MSCE equals Bnsce

bmsce equals BMSCe

bm&cc equals Ignoredare BMSCe .

True

False

False

True.

⑧ `regionMatches()`

~~String target = "Bmsce colleg"~~

~~↑
SubString~~

Substring is matched

⑧ startwith() (True / False)

BMSCE College

True

⑨ endwith() (True / False)

BMSECE College

ends with (" COLLEGE")

False.

⑩ equal()

world equal world → True

world == World → False

| | | | |
|----|-------|--------|--------|
| 1) | Apple | Hen | parrot |
| | Bull | Ice | Queen |
| | Cat | Gig | Ring |
| | Dog | Kite | Star |
| | Ear | Man | Yatch |
| | Free | Nel | Zoo |
| | Gun | Orange | |

12) Compare to ()

1

2

3

4

5

6

7

8

9

13) Substring

This is way a text. This is wast, too.
This is a text, This is, too.

14) Concatenate

concat()

Hello World

15) Replace ()

College

16) Trim ()

~~Hello Friends~~

17) Enter details for student:

Registration Number:
Full Name: Bhupendra
Semester: 2
CGPA: 9.2

18) String Buffer

After setLength(5) : Hi! Yo
character at index 0 : H.

After setCharAt(0, 'X') : Xi! Yo
getchar(0, 5) : Xi! Yo

19) Bird Demo

Eagle Action:

Eagle is soaring high in the sky eagle swoops
landly.

Hawk action:

Hawk is gliding through the air. Hawk makes
a sharp turn.

20) Shape Demo.

Area of Circle : 79.539 Perimeter : 31.41
Triangle : 6 Perimeter : 12.

Packages

Student

student.java.

CIE

SEE

package CIE;

final marks.

public class student.

{

public string USN;

public string name;

public int sum;

public student()

{

this("", "", 0);

}

 public student(string USN, string
name, int sum)

{

this.USN = USN;

this.name = name;

this.sum = sum;

{

public void setUSN(string USN)

{

this.USN = USN;

{

public void setName(string name)

{

this.name = name;

{

weeks

to make

1.41

#internal.java

```
package CT;
public class internal {
    private int [] internalmarks = new int [5];
    public internal () {
        {
            public void setinternalmarks (int [] internalmarks) {
                this.internalmarks = internalmarks;
            }
            public int [] getinternalmarks () {
                return internalmarks;
            }
        }
    }
}
```

external.java

```
package SEE; import CT.E.student;
public class external extends student {
    public int [] SEEmarks = new int [5];
    public external () {
        this.USN = "", name = " ", sum = 0, new int [5];
    }
    public external (String USN, String Name, int sum, int [] SEEmarks) {
        super (USN, name, sum);
    }
}
```

line. SEEmarks = SFFmarks;

public int[] getSEEmarks()

return SEEmarks;

}

Final Marks java

internal

import CIF.student;

import CIF.internals;

import SFF.external;

import java.util.Scanner;

public class FinalMarks

{

public static void main(String [] args)

Scanner s = new Scanner(System.in);
System.out.println("Enter the numbers
of students : ");

int n = s.nextInt();

student [] stu = new student [n];

internal [] in = new internal [n];

external [] ex = new external [n];

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

stu[i] = new student();

System.out.println("Enter the USN for
student " + (i+1) + ": ");

stu[i].setUSN (s.nextInt());

System.out.println("Enter the name of
student " + (i+1) + ": ");

stu[i].setname (s.next());

new int[3])

ing Names

```
System.out.println ("Enter semester for  
student "+(i+1)+" :");  
stu[i].setsem(s.nextLine());  
in[i]=new Internals();  
in[i].setinternalmarks (InputMarksWithValidation  
("type : " + "internal", i, s, min:0, max:90));  
ext[i]=new external (stu[i].getUSN(),  
stu[i].getname(), stu[i].getsem(), new  
int[]);  
ext[i].setSEFmarks (InputMarksWithValidation  
("type : " + "external", i, s, min:0, max:100));  
int [5] finalmarks = new int [5];  
for (int j=0; j<5; j++)  
    finalmarks [j] = in[i].getinternalmarks ()[j]+  
    ext[i].getSEFmarks ()[j]/2;  
    }
```

```
System.out.println ("Student "+(i+1)+" Final  
Marks : "+finalmarks [0] + "," + finalmarks  
[1] + "," + finalmarks [2] + "," + finalmarks [3] + "  
" + finalmarks [4] + "," + finalmarks [5]);  
s.close();  
}
```

```
private static int [5] InputMarksWithValidation  
(String type, int studentindex, Scanner s,  
int min, int max)  
{
```

```
int [5] marks = new int [5];  
System.out.println (" Enter "+type+" marks  
of student "+(studentindex+1)+". ");  
for (int i=0; i<5; i++)
```

9
Get mark;

do

9

System.out.println("Subject "+(i+1)+": ");
mark = s.nextInt();
if (mark < 0 || mark > max)

9

System.out.println("Invalid input type");
"marks should be between 0 and "+max +
"Please try again");

9

while (mark < 0 || mark > max);
marks[i] = mark;

9

return marks;

9

9

Output

Enter the USN

18M22CS202

Enter the Student Name

Hegde

Enter the semester.

3.

Enter the CIE marks

Enter the marks of each subject

48

46

45

49

50

Father SEE Marks

Subject 1 : 46

Subject 2 : 43

Subject 3 : 48

Subject 4 : 49

Subject 5 : 44

Wrong Age

31/1

Exception()

```
import java.util.Scanner;
class WrongAge extends Exception
{
    public WrongAge()
    {
        super("Age Error");
    }
    public WrongAge(String message)
    {
        super(message);
    }
}
```

```
class InputScanner
```

```

Scanner s = new Scanner(System.in);
public void closeScanner()
{
    s.close();
}
```

```
class Father extends InputScanner
```

```

int fatherAge;
public Father()
{
    System.out.println("Enter the father's age:");
    fatherAge = s.nextInt();
}
```

if (fatherage < 0)

 throw new WrongAge (message = "Age
 cannot be negative");

}

public void display ()

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

}

Class Son extends Father

{

 int sonAge;

 public Son () throws WrongAge {

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

 sonAge = s.nextInt ();

 if (sonAge >= fatherAge)

 {

 throw new WrongAge (message = "Son's
 age cannot be greater than or
 equal to father's age");

 } else if (sonAge < 0)

 {

 throw new WrongAge (message = "Age cannot
 be negative");

 }

 public void display ()

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

public class Main

{ public static void main(String[] args)

{ try

{ Son s = new Son()

s.display();

s.closeScanner();

} catch (legalArgumentException e)

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

Output:

Enter father's Age : 49

Enter son's age : 50

~~Exception: Son's age cannot be greater than or equal to father's age.~~

Multi Threading

```
public class ThreadExample {  
    public static void main(String[] args) {  
        Thread thread1 = new Thread(new Dispenser());  
        Thread thread2 = new Thread(new Dispenser());  
        thread1.start();  
        thread2.start();  
    }  
}  
class Dispenser implements Runnable {  
    public void run() {  
        try {  
            Thread.sleep(10000);  
            System.out.println("BMS college of  
Engineering");  
        } catch (InterruptedException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

```
class Dispenser implements Runnable {  
    public void run() {  
        //  
    }  
}
```

```
9  
while (true)  
{  
    try
```

```
    Thread.sleep (millis: 2000);
```

```
    System.out.println ("x = " + CSE);
```

```
    catch (InterruptedException e)
```

```
    {  
        e.printStackTrace();  
    }
```

```
3  
Output:
```

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering.

CSE

CSE

Inter process Communication

class A

```
int n;
boolean value set = false;
Synchronized int get()
{
    while (!value set)
        try
        {
            System.out.println ("In consumer
waiting in ");
            wait();
        }
        catch (InterruptedException)
        {
            System.out.println ("Interrupted
exception caught ");
        }
    System.out.println ("got : " + n);
    Value Set = false;
    System.out.println ("In Intimate produ
cer in ");
    notify();
    return n;
}
Synchronized void put(int n)
{
    while (Value set)
        try
        {
            System.out.println ("In Producer waiting
")
```

```
\n");  
wait();  
}  
catch (InterruptedException e)  
{
```

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

```
this.n = n;
```

```
valuetest = true;
```

```
System.out.println("Put " + n);
```

```
System.out.println("\n Intimate consumer
```

```
\n");
```

```
}
```

```
}
```

```
Qq;  
producer(Qq);
```

```
S
```

```
this.q = q;
```

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

```
public void run()
```

~~int i = 0;~~~~while (i < 15)~~~~q.put(i++);~~~~q.put(i++);~~

```
S
```

class consumer implements Runnable

{

Q q;
consumer(Q q)

{

int i = 0;

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

{

public void run()

int i = 0;
while (i < 15)

{

int a = q.get();

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

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");

3

output

Press control c to stop

put : 0
Intimate consumer
producer waiting
get = 0.

Intimate Producer
put : 1

Intimate Consumer
producer waiting
consumed = 0
lot = 1.

Intimate Producer
consumed st
put : ?

Deadlock

13/2/84

class A

```
Synchronized void foo(Bb)
{
    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(Aa)
```

String name =

id

papergrid

Date: / /

Thread. current Thread(). getName()
System.out.println("name + " entered
B. Baa");

try

{

 thread.sleep(1000);

}

catch (Exception e)

{

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

}

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

 a.last();

}

 word last();

{

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

}

Aa = new A();

Bb = new B();

Deadlock = 0;

System.out.println("Main thread started");

Thread t = new Thread(this, "Racing");

Thread t.start();

t.start();

a = new B();

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

papergrid

Date: / /

public void run()

b.baz()

System.out.println(Back in other thread)

public static void main(String args[])

new Thread(lock());

Output

Main thread entered A-fac

Racing thread entered B-baz

Main thread trying to call B-last()

Inside A-last()

Back in main thread

Racing thread trying to call A-last()

Inside B-last()

Back in other thread

Date: / /
2012Integer Divisions

Last Lab

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

```
class Swing Demo
```

```
{  
    Swing Demo()  
}
```

```
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 dividend : ");
```

```
JTextField aJf = new JTextField (8);  
JTextField bJf = 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 (atab );  
jfrm . add (btab );  
jfrm . add (cmstab );
```

Action listener 1 = new Action Listener ()

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

```
        System.out.println ("Action event from  
        a text field ");
```

```
{  
    ajtf . addActionListener (1);  
    bjtf . addActionListener (1);
```

```
    button . addActionListener (new  
    ActionListener ())
```

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

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

```
{  
    try
```

```
    int a = Integer . parseInt (ajtf . get  
    Text ());
```

```

    int b = Integer.parseInt(bjf.getText());
    int ans = a(b);
    awb.setText("A = " + a);
    bwb.setText("B = " + b);
    awlab.setText("Am = " + ans);
    awlab.set("Am = " + ans);
}

catch(NumberFormatException e)
{
    awlab.set("Format");
    bwlab.set("Format");
    awlab.set("Enter only");
    awlab.set("integer");
}

```

catch (Arithmatic Exception)

5
a lab. set Text (quot; quot;);
b lab. set Text (quot; quot;);
ani lab. set Text (quot; quot;);
err set Text (quot; B should be NON
zero !! quot;);

3) ~~if form . set visible (true)~~

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

Saving utilities - invoke later (new)
Runnable().

```
public void event()
```

papergrid

Date: / /

new swing Demo (2)

8)

3

Output

Enter the divisor & dividend

48

16

Calculate $A = 48$ $B = 6$ Ans = 8

Niy
26.02.24