

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



OBJECT ORIENTED JAVA PROGRAMMING

LAB RECORD

Bachelor of Engineering in
Computer Science and Engineering

Submitted by:

SHARANABASAPPA KALLAPPA TONDIHAL
USN-1BM22CS253

Department of Computer Science and Engineering B.M.S
College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560 019
2023-2024

INDEX

NAME: Sharana Basappa STD: _____ SEC: CSE-6 ROLL NO: 23

S.No.	Date	Title	Page No.	Teacher's Sign/Remarks
1	12/12	Quadratic Equation	10	✓
2	19/12	SGPA Calculation	10	✓
3	26/12	Books Java	10	26/12/23
4	2/1/24	Shapes Java	10	8/2/2024
5	9/1/24	Bank.java	10	10/1/24
6	16/1	String.java	10	23/1/24
7	23/1	Package.java	10	25/1/24
8	30/1	Exception.java	10	30/1/2024
9	6/2	Threads.java	10	6/2/2024
10	13/2	Deadlock.java	10	✓ 13/2/2024
	13/2	Enterprise Communication.java	10	✓ 13/2/2024
	20/2	User Interface Java	10	✓ 20/2/2024

⑤ Develop a Java program that prints all real solutions to the quadratic equation $ax^2 + bx + c = 0$. Read in a, b , c and use the quadratic formula. If the discriminant $b^2 - 4ac$ is negative, display a message stating that there are no real solutions.

```
> 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 computer()  
    {  
        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 = " + r1);
}

else 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 ("Root1 = " + r1 + " Root2 = " + r2);
}

else
{
    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();

}

}

Output

Enter the coefficient of a,b,c

1 2 3

Roots are imaginary

Root 1 = -1.0 + i 1.41421356

Root 2 = -1.0 - i 1.41421356

Enter coefficient of ab,c

131

Roots are real and distinct

Root 1 = -0.38196601

Root 2 = -2.6180537

Enter coefficient of abc.

5 0 5

Roots are Imaginary

Root 1 = 0.0 + i 0.8

Root 2 = 0.0 - i 0.8

SJ
19/12/2023

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

→ Import java.util.Scanner;
class subject

```
{
    int subjectmarks;
    int credits;
    int grade;
}
```

class Student

```
{
    Subject subject[5];
    String name;
    String id;
    double sgpa=0;
```

Scanner s;

Student ob

```
{
    int i;
    Subject = new Subject[5];
    for (i=0; i<5; 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 (int i=0; i<9; i++)
```

```
{
```

```
    System.out.println("enter marks for subject "+(i+1));
```

```
    subject[i].subjectmarks = s.nextInt();
```

```
    System.out.println("enter credits for subject "+(i+1));
```

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

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

```
    if (subject[i].grade == 11)
```

```
        subject[i].grade = 10
```

```
    if (subject[i].grade < 4)
```

```
        subject[i].grade = 0;
```

```
,
```

```
3
```

```
void computeSGPA()
{
    int total credits = 0;
    for (int i=0; i<9; i++)
    {
        sgpa += (subject[i].credits * subject[i].grade);
        total credits += subject[i].credits;
    }
}
```

$$sgpa = sgpa / total credits;$$

```
}

class main
```

```
{
```

```
    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  
Sharan

Enter your USN  
IBM 22C8253

enter marks of subject 1:

80

enter credits for subject 1:

4

enter marks of subject 2:

70

enter credits for subject 2:

3

enter marks of subjects:

100

enter credits for subject 3:

1

enter marks of Subject 4:

65

enter credits for subject 4:

3

enter marks of Subject 5:

98

enter credits for subject 5:

2

enter marks of Subject 6:

75

enter credits for subject 6:

4

enter marks of Subject 8:

89

enter credits for subject 8:

3

Enter marks for subject 9:

98

Enter credits for subject 9:

3

Name  $\Rightarrow$  Sharda

USN  $\Rightarrow$  18M22CS253

SGPA  $\Rightarrow$  8.84615

Quar  
19-12-13

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

{  
System.out.println("Book Publishing co");

}

3

Output enter the no of books  
2

enter the name of book

biology

enter the price of book

100

enter the author of book

Sharon

enter the no of pages in book

100

enter the next name of book

giology.

enter the price of book

200

enter the author of book

26/12/2023 Ragan

Ragan

enter the no of pages in book

gology 200

book name: biology

author name: Sharon

price: 100

number of pages: 100

book name: giology

author name: ragan

price: 200

number of pages: 200

Develop java program to creates an abstract class named shape that contains two integers and an empty method named printArea(). Provide three classes extends the class shape. Each one of the class contains only the method printArea() that prints the area of the given shape.

import java.util.Scanner;

abstract class shape

{

    int a, b;

    abstract public void printArea();

}

Class rectangle extends shape

{

    int area\_rect;

    public void printArea()

{

        Scanner s = new Scanner(System.in);

        System.out.println("enter the length and breadth of the rectangle");

        a = s.nextInt();

        b = s.nextInt();

        area\_rect = a \* b;

        System.out.println("Enter the length of rectangle:" + a + " breadth of rectangle:" + b);

        System.out.println("Area of rectangle:" + area\_rect);

}

9

class triangle extends shape

{  
public ~~double~~ triangle-area;

public void printarea()

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

Printf("enter the height and base of the triangle");

a = s.nextInt();

b = s.nextInt();

triangle-area = (a \* b) / 2

Printf("height of triangle:" + a + "base of triangle" + b);

Printf("area of triangle" + triangle-area);

}

}

Class circle extends shape

{  
public ~~double~~ circle-area;

public void printarea()

{

Scanner s = new Scanner(System.in);

Printf("Enter the radius of the circle");

a = s.nextInt();

circle-area = 3.14 \* a \* a;

Printf("area of circle" + circle-area);

}

3

class Shapeclass

{ public static void main(String args){}

{ rectangle r = new rectangle();

r.printarea();

triangle t = new triangle();

t.printarea();

Circle c = new circle();

c.printarea();

}

}

Output

Enter the length and breadth of rectangle

2

3

length of rectangle: 2 breadth of rectangle: 3

area of rectangle: 6

Enter the height and base of triangle

2

3

height of triangle: 2 base of triangle: 3

base

area of triangle: 3.0

Enter the radius of circle

3

area of circle: 28.25999

2/1/2024

a) Develop a Java program to develop it create a class bank that maintains two kind of accounts one called savings account and other called current account, the savings account provides compound interest and withdrawal facility but no chequebook facility. The current account provides cheque book facility but no interest. Current account holder also need to maintain minimum balance and if balance falls below this level, a service charge is imposed.

Create a class account that stores customer name, account number, account type. From this derive the classes ~~•~~ curr-account and sav-account to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks.

- a) Accept deposits from customer and update the balance.
- b) display the balance.
- c) Compute and deposit interest
- d) Permit withdrawal and update the balance

```

import java.util.Scanner;
class account
{
 String name;
 int accno;
 String type;
 double balance;

 public account (String name, int accno, String type, double balance)
 {
 this.name = name;
 this.type = type;
 this.accno = accno;
 this.balance = balance;
 }

 void deposit (double amount)
 {
 balance = balance + amount;
 }

 void withdraw (double amount)
 {
 if ((balance - amount) >= 0)
 balance -= amount;
 else
 System.out.println ("Insufficient balance, can't withdraw");
 }

 void display()
 {
 System.out.println ("name" + name + "accno" + accno +
 "type" + type + "balance");
 }
}

```

Class Saving account extends Account {

    private static double rate = 5;

    Saving account (String name, int accno, double balance);

        super (name, accno, "Saving", balance);

}

void interest ()

{

    balance = balance + balance \* (rate) / 100;

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

}

Class Bank {

public static void main (String args)

{

    Scanner sc = new Scanner (System.in);

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

    String name = sc.next();

    System.out.println ("Enter the type (current/saving):");

    String type = sc.next();

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

    int accno = sc.nextInt();

    System.out.println ("Enter the initial balance:");

    double balance = sc.nextDouble();

    int ch;

    double amount1, amount2;

    Account acc = new Account (name, accno, type, balance);

    SavingAccount sav = new SavingAccount (name, accno, balance);

    GorAvt co = new GoraVat (name, accno, balance);

case (true)

{ if (acc.type.equals("Savings"))

{

System.out.println("Enter name to deposit to savings  
3 - compute interest & display");

System.out.print("Enter the choice");

int ch = scanner.nextInt();

switch (ch)

{

case 1:

{ System.out.print("Enter the amount");

amount1 = scanner.nextInt();

sa.deposit(amount1);

} break;

case 2:

{ System.out.print("Enter the amount");

amount2 = scanner.nextInt();

sa.withdraw(amount2);

} break;

case 3:

{ sa.calculate();

} break;

case 4:

{ sa.display();

} break;

Cases :

System.exit(0);

3

3

else

8

System.out.println ("enter the amount :");  
amount1 = s.nextInt();  
carddeposit (amount1);  
break;

case 1: System.out.println ("enter the amounts");  
amount1 = s.nextInt();  
carddeposit (amount1);  
break;

Case 2: System.out.println ("enter the amounts");  
amount 2 = s.nextInt();  
ca.withdraw (amount2);  
ca.check min();  
break;

Case 3: ca.display ();  
break;

Case 4: System.exit (0);

3  
4

Output

Enter the name: Shashi

Enter the type / savings/current

current

Enter the account number

10000

Enter the initial balance

10000

menu

1. deposit 2.withdraw 3.display.

1

enter the amount : 1000

menu

1. deposit 2.withdraw 3.display

2

enter the amount : 500

menu

1. deposit 2.withdraw 3.display.

3

Shashi 10000 current 10500

16/1/2024

## Strings

### String constructors

```
char chars[] = {'b', 'm', 's', 'c', 'e'}
```

```
String s1 = new String(chars)
```

```
String s2 = new String(chars, 1, 3);
```

Output: bmsce

msc

### String length

```
char chars[] = {'p', 'y', 't', 'u', 'o', 'n'}
```

```
String s = new String(chars);
```

```
s.length();
```

Output: 6

### String Literal & Concatenation

```
System.out.println("abc".length());
```

```
String car = "BMW";
```

```
System.out.println("He has " + car + " car");
```

Output: 2

He has a BMW car

### getchar

```
String obj = "Welcome BMSC College";
```

```
getchar(11, 17, Buff, 07);
```

Output: BMSC

equals & equalsIgnoreCase!

BMSC equals BMSC → true

BMSC equals College → false

BMSC equals BMSC → false

BMSC equalsIgnoreCase BMSC → true

### region matches

```
Boolean isMatch = str.regionMatches(11, str2, 0, 13);
```

Output: substring is matched.

startsWith and endsWith?

String game = "Basketball".

```
System.out.println(game.startsWith("Basket"));
```

```
System.out.println(game.endsWith("ball"));
```

Output:

true

true

equals v/s =

Hello equals Hello → true

Hello == Hello → false

Sort

apple ball cat dog end free gun hen ice jug kettle  
lift man net orange parrot queen ring star  
umbrella van watch Xmas tree yak zee

18)

2 3 4 5 6 7 8 9

19

This is a test . This is, too

14

world

15

college

16

Hello friends

17

Student 1

Name: Swaroop

Regno: 253

Sem: 3

Cg PA: 8.135

Student 2

Name: Sureesh

Regno: 230

Sem: 3

Cg PA: 9.5

Write a java program to create a generic class stack  
which holds 5 integer & 5 double values

```
import java.util.*;
class Stack<E> {
 E stk[];
 int top;
 int size = 10;
 Stack() {
 stk = (E[]) new Object[size];
 top = -1;
 }
 void push(E item) {
 if (top == size - 1)
 System.out.println("overflow");
 else
 stk[++top] = item;
 }
 E pop() {
 if (top < 0)
 System.out.println("underflow");
 return null;
 }
 else
 return stk[top--];
 }
}
```

public class TestStackE

publio statt vold mein (String[ ] orgs)

5

```
Stack<Integer> mystack1 = new Stack<Integer>();
Stack<Integer> mystack2 = new Stack<Integer>();
```

Scanner 8=mod 324-Scanner (System 97);

System.out.println("Enter elements into the integer Stack");

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

```
? int n = s.nextInt();
myStack.push(n);
```

3

System-orientation ("factor elements in the double book")

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

5

```
double m = s.nextDouble();
myStack.push(m);
```

三

System "out-of-print" ("Elements of mysticism");

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

5

- System «autoprint» (mystack • paper);

7

~~System-out-printing ("Elements of mysticism");~~

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

2

Systems root-pointed (my stock & paper);

三

卷之三

Create a package OEE which has two classes Student and Internals. The class Student has members USN, name, sem. The class Internals derived from student has an array that stores the internal marks scored in 5 courses of the current semester of the student. Create another package SEE which has the class External which is derived class of student. This class has an array which stores the SEE marks of current semester of student. Import the two packages in file that declares a final marks of a student in all five courses.

```
package OEE;
import java.util.*;
public class Student {
 protected String name = new String();
 protected String USN = new String();
 protected int sem;
 public void InputStudentDetail() {
 Scanner s = new Scanner(System.in);
 System.out.println("enter the name");
 name = s.nextLine();
 System.out.println("enter the sem");
 sem = s.nextInt();
 }
 public void displayStudentDetail() {
 System.out.println("Name" + name);
 System.out.println("USN" + USN);
 System.out.println("sem" + sem);
 }
}
```

### 1) Internals

```
package SEE;
import java.util.*;
public class Internal extends Student {
 protected int marks = new int[5];
 public void inputInternalMarks() {
 Scanner s = new Scanner(System.in);
 for (int i = 0; i < 5; i++) {
 marks[i] = s.nextInt();
 }
 }
}
```

### 2) Externals

```
package SEE;
import SE.Internal;
import java.util.*;
public class External extends Internal {
 protected int marks[];
 protected int finalmarks[];
 public External() {
 marks = new int[5];
 finalmarks = new int[5];
 }
 public void inputExternalMarks() {
 Scanner s = new Scanner(System.in);
 for (int i = 0; i < 5; i++) {
 System.out.println("Enter Subject " + (i + 1) + " Marks");
 marks[i] = s.nextInt();
 }
 }
}
```

public void calculateFinalMarks() {  
 for (int i = 0; i < 5; i++) {  
 finalmarks[i] = marks[i] \* 0.8 + super.marks[i];  
 }  
}

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

//main

import java.io.Externalizable;  
class Main {  
 public static void main(String args[]) {  
 int n = 2;  
 External finalmarks[] = new External[5];  
 for (int i = 0; i < 5; i++) {  
 finalmarks[i] = new External();  
 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");
for (int i=0 ; i<5 ; i++) {
 finalmarks [i] = cal.calculateFinalMarks();
 finalmarks [i].displayFinalMarks();
```

3

Output:

Enter the USN : IBM22CS253

Enter the name : Sharanareshappa

Enter Sem : 3

Enter CIE marks :

Subject 1: 44

Subject 2: 47

Subject 3: 48

Subject 4: 43

Subject 5: 44

Enter SEE marks :

Subject 1: 88

Subject 2: 89

Subject 3: 87

Subject 4: 88

Subject 5: 87

Display Data:

USN : IBM22CS253

Name : Sharanareshappa

Sem 3

30/1/2024

Subject 1: 86

Subject 2: 90

Subject 3: 85

Subject 4: 87

Subject 5: 87

```
import java.util.*;
class Wrongage extends Exception {
 public Wrongage (String s)
 {
 super(s);
 }
}
class Father {
 int fage;
 Father () throws Wrongage
 {
 System.out.println ("Enter father's age");
 Scanner s = new Scanner (System.in);
 fage = s.nextInt();
 if (fage < 0)
 {
 throw new Wrongage ("Age cannot be negative");
 }
 }
 void display()
 {
 System.out.println ("Father's Age is :" + fage);
 }
}
class Son extends Father {
 int sage;
}
```

Son () throws Wrongage

{ System.out.printLn("Enter son's age");

Scanner s = new Scanner (System.in);

sage = s.nextInt();

if (sage > fage)

{

    throw new Wrongage ("Son's age cannot be greater  
    than father's age");

}

else if (sage < 0)

{

    throw new Wrongage ("Age cannot be negative");

}

else if (fage == sage)

{

    throw new Wrongage ("Age cannot be same");

}

3

    wd5display()

{

    System.out.println ("Son's age is: " + sage);

}

3

Class Eraser {

    Public static void main (String args[])

{

try {

Son s = new Son();

s.display();

s.sdisplay();

}

catch (Wrong age e) {

System.out.printLn(e);

}

}

}

### Output

enter father's age

40

enter son's age

36

Father's Age 15:40

Son's Age 15:36

enter father's age

35

enter son's age

40

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

enter father's age

35

enter son's age

35

Age can't be same

30/1/2020

@Write a program which creates two threads, one thread displaying "BMS College of Engineering" once every ten seconds and another displaying "CSE" once every 100 seconds.

```

class Thread implements Runnable {
 Thread t;
 public Thread() {
 t = new Thread(this, "N Thread");
 System.out.println("CT:" + t);
 t.start();
 }
 public void run() {
 try {
 for (int n=5; n>0; n--) {
 System.out.println("CSE " + n);
 Thread.sleep(2000);
 }
 } catch (InterruptedException e) {
 System.out.println("CSE quitting.");
 }
 }
}

```

```

public class PrintCollege {
 public static void main (String args) {
 new Thread () {
 System.out.println ("Back in main");
 try {
 for (int n=0; n>0; n--) {
 System.out.println ("BMSCE " + n);
 Thread.sleep (10000);
 }
 } catch (InterruptedException e) {
 System.out.println ("BMSCE interrupted");
 }
 System.out.println ("BMSCE quitting");
 };
 }
}

```

### Output

GT: Thread [NTThread, 5, main]

Back in main.

CSE 5

BMSCE 2

CSE 4

CSE 3

CSE 2

CSE 1

BMSCE 1

CSE quitting

BMSCE quitting

SK  
6/2/2024

```
class Q {
```

```
 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("InterruptedException Caught");
```

```
}
```

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

```
 valueSet = false;
```

```
 System.out.println("In Enterate Producer\n");
```

```
 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("Interr Exception Caught");
```

```
this->n;
```

```
valueSet = true;
```

```
System.out.println("In Intermate Consumer In'");
```

```
notify();
```

```
}
```

```
}
```

```
class Producer implements Runnable {
```

```
Q 2:
```

```
Producer(Q 2) {
```

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

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

```
}
```

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

```
int i = 0;
```

```
while (i < 10) {
```

```
q.put(i++);
```

```
}
```

```
}
```

```
}
```

```
class Consumer implements Runnable {
```

```
Q 2:
```

```
Consumer(Q 2) {
```

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

~~```
new Thread(this, "Consumer").start();
```~~~~```
}
```~~

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

```
int i = 0;
```

```
while (c != 5) {
 int req, gets;
 System.out.println("Consumed: " + r);
 r++;
 if (r == 5)
 break;
```

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!");
}
```

Output :-

Put : 0

Intrmte Consumer

producer waiting.

press Control -C to stop

Got : 0

~~Intrmte producer~~

put : 1

Intrmte Consumer

producer waiting.

Consumed : 0

Got : 1

In-Hmata Producer

consumed : 1

put : 2

In-Hmata Consumer

producer waiting

Get : 2

In-Hmata Producer

consumed : 2

put : 3

In-Hmata Consumer

producer waiting

Get : 3

In-Hmata Producer

consumed : 3

put : 4

In-Hmata Consumer

producer waiting

Get : 4

In-Hmata Producer

consumed : 4

put : 5

In-Hmata Consumer

producer waiting

Get : 5

In-Hmata Producer

consumed : 5

put : 6

## 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, "RunningThread");

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 ("Breaks in other thread");

}

public static void main (String args) {

new Deadlock();

}

}

Output

Main Thread entered A::foo Racing Thread entered entered

B::bar Main Thread trying to call B::last()

Inside A::last

Back in main thread

Racing Thread trying to call A::last()

Inside A::last

Back in ~~other~~ thread.

S  
13/2/2024

Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the textfields, num1 and num2. The division of num1 and num2 displayed in the result field when the divide button is clicked. If num1 and num2 are not integers, the program should throw NumberFormatException. If num2 were zero, then the program would throw an ArithmeticException. Display the message or exception in a message dialog box.

```

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

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

 public DivisionMain()
 {
 getLayout(new FlowLayout());
 dResult = new Button("Result");
 Label number1 = new Label("Number1:", Label.RIGHT);
 Label number2 = new Label("Number2:", Label.LEFT);
 num1 = new TextField(5);
 num2 = new TextField(5);
 outResult = new Label("Result:", Label.RIGHT);
 }
}

```

```

 add (number1);
 add (num1);
 add (number2);
 add (num2);
 add (cResult);
 add (outResult);

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

public void actionPerformed (ActionEvent ae) {
 int n1, n2;
 try {
 if (ae.getSource () == cResult) {
 n1 = Integer.parseInt (num1.getText ());
 n2 = Integer.parseInt (num2.getText ());
 /* if (n2 == 0)
 throw new ArithmeticException (""); */
 out = n1 + " " + n2;
 out += String.valueOf (resultNum);
 resultLabel.setText (out);
 }
 }
}

```

3

3

catch (NumberFormatException e) {

5

flag = 1;

out = "Number Format Exception! " + e;  
repaint();

3

catch (ArithmaticException e)

5

flag = 1;

out = "Divide by 0 Exception! " + e;  
repaint();

7

public void paint (Graphics g)

if (flag == 0)

g.drawString (out, outResult.getX() + outResult.getWidth(),  
outResult.getY() + outResult.getHeight());

else

g.drawString (out, 100, 200);

flag = 0;

3

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

```
{
```

```
 DivisionMain1 dm = new DivisionMain1();
```

```
 dm.setSize (new Dimension (800, 400));
```

```
 dm.setTitle ("Division Of Integers");
```

```
 dm.setVisible (true);
```

```
}
```

```
3
```

Output:

Number 1:

Number 2:   Result: 10 / 2 = 5.0

8  
aditya

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 the discriminant  $b^2 - 4ac$  is negative, display a message stating 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.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+"+"+r2);
 System.out.println("Root1="+r1+"-"+r2);
 }
}

1
}
class QuadraticMain
{
 public static void main(String[] args)
 {
 Quadratic q=new Quadratic();
 q.getd();
 q.compute();
 System.out.println("Sharana basappa K Tondihal 1BM22CS253"); }
}

```

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

```

import java.util.Scanner;

class Subject{
 int subjectMarks;
 int credits;
 int grades;
}

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

 Student(){
 subject = new Subject[9];

 for(int i = 0;i<9;i++){
 subject[i] = new Subject();
 }
 s= new Scanner(System.in);
 }

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

```

```

 this.usn = s.nextInt();
 }

 void getMarks(){
 for(int i = 0;i<8;i++){
 System.out.println("Enter the marks of the "+(i+1)+" subject");

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

 System.out.println("Enter the credits of the "+(i+ 1)+" subject");
 subject[i].credits = s.nextInt();
 subject[i].grades = (subject[i].subjectMarks/10)+1;

 if(subject[i].grades >10){
 subject[i].grades = 10;
 }
 if(subject[i].grades <4){
 subject[i].grades = 0;
 }
 }
 }

 void computeSGPA(){
 int sum=0;
 int totalCredits = 0;
 for(int i = 0;i<9;i++){
 sum+=(subject[i].grades * subject[i].credits);
 totalCredits += subject[i].credits;
 }
 this.SGPA = (double) sum/totalCredits;
 }
}

public class MainSGPA{
 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);
 System.out.println("Sharana basappa K Tondihal 1BM22CS253"); }
}

```

3)Create a class Book which contains four members: name, author, price, num\_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a `toString()` method that

could display the complete details of the book. Develop a Java program to create n book objects.

```
import java.util.Scanner;

class Books{
 String name;
 String author;

 int price;
 int numPages;

 public 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="No of Pages : " + this.numPages + "\n";
 return name + author + price + numPages;
 }

 String getName(){
 this.name=name;
 }

 String getAuthor(){
 this.author=author;
 }

 int getPrice(){
 this.price=price;
 }

 int getNumPages(){
 this.numPages=numPages;
 }
}

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

 System.out.println("Enter no of books :");
 }
}
```

```

n= s.nextInt();
s.nextLine();

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

for(int i=0;i<n;i++){
 System.out.println("Enter the name of
book"+(i+1)+":");
 name=s.nextLine();
 System.out.println("Enter Author of book"+(i+ 1)+":");
 author=s.nextLine();

 System.out.println("Enter price of book"+(i+
1)+":");
 price=s.nextInt();
 System.out.println("Enter no of pages of book"+(i+1)+":");
 numPages=s.nextInt();
 s.nextLine();
 b[i]=new Books(name,author,price,numPages);
}

for(i=0;i<n;i++){
 String bookDetails=b[i].toString();
 System.out.println(bookDetails);
}
for(int i=0;i<n;i++){
 System.out.println("Book "+(i+1)+":");
 System.out.println("Name :" + b[i].getName());

 System.out.println("Author :" +b[i].getAuthor()); System.out.println("Price
:+b[i].getPrice());
 System.out.println("No of
pages :" +b[i].getNumPages());

 System.out.println("*****");
}

System.out.println("Sharana basappa K Tondihal \n
1BM22CS253");

}
}

```

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

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

```

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

abstract class Shape extends InputScanner{
 double a;
 double b;
 abstract void getInput();
 abstract void displayArea();

 }

class Rectangle extends Shape{
 void getInput(){
 InputScanner sc=new InputScanner();
 System.out.println("Enter the length and breadth of rectangle");
 a=sc.nextInt();
 b=sc.nextInt();
 }
 void displayArea(){
 double area_rect=a*b;
 System.out.println("Area of retangle is :
"+area_rect);
 }
}

class Triangle extends Shape{
 void getInput(){
 InputScanner sc=new InputScanner();
 System.out.println("Enter the base and height of triangle:");
 a=sc.nextInt();
 b=sc.nextInt();
 }
 void displayArea(){
 double area_tri=a*b/2;
 System.out.println("Area of tritangle is :
"+area_tri);
 }
}

class Circle extends Shape{
 void getInput(){
 InputScanner sc=new InputScanner();
 System.out.println("Enter the radius of circle:"); a=sc.nextInt();
 }
 void displayArea(){

```

```

 double area_circle=3.14*a*a;
 System.out.println("Area of circle is :
"+area_circle);
 }
}

public class AbstractMain{
 public static void main(String args[]){
 Rectangle a = new Rectangle();
 a.getInput();
 a.displayArea();

 Triangle b = new Triangle();
 b.getInput();
 b.displayArea();

 Circle c = new Circle();
 c.getInput();
 c.displayArea();

 System.out.println("Sharana basappa K Tondihal 1BM22CS253");
 }
}

```

5)Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

- Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:
- a)Accept deposit from customer and update the balance. • b)Display the balance.
- c)Compute and deposit interest
- d)Permit withdrawal and update the balance
- Check for the minimum balance, impose penalty if necessary and update the balance.

```

import java.util.Scanner;

class Account{
 String name;
 int accNo;
 String type;
 double balance;

```

```

Account(String name,int accNo,String type,double balance) {
 this.name=name;
 this.accNo=accNo;
 this.type=type;
 this.balance=balance;
}

void deposit(double amt){
 balance=balance+amt;
}

void withdraw(double amt){
 if(balance<amt){
 System.out.println("Insufficient Balance");
 }
 else{
 balance=balance-7amt;
 }
}

void display(){
 System.out.println("Name:"+name+"\tAccount
No:"+accNo+"\tAccount Type:"+type+"\tBalance"+balance); }
}

class Savings_acc extends Account{
 private static double rate= 3.5;

 Savings_acc(String name,int accNo,double balance){
 super(name,accNo,"savings",balance);
 }

 void callInt(){
 double interest=(balance*rate)/100;
 System.out.println("Interest is "+interest); }
}

class Current_acc extends Account{
 private double minBal=500;
 double s_charges=50;

 Current_acc(String name,int accNo,double balance){
 super(name,accNo,"current",balance);
 }

 void check_bal(){
 if(balance<minBal){
 System.out.println("Insufficient Balance");
 balance=balance-s_charges;
 }
 }
}

```

```

 System.out.println("Balance =" + balance);
 }
}

public class Bank{
 public static void main(String args[]){
 String name;
 int AccNo;
 String Type;
 double init_bal;

 Scanner s=new Scanner(System.in);
 System.out.println("Enter Customer Name:");
 name=s.nextLine();
 System.out.println("Enter Account No:");
 AccNo=s.nextInt();
 System.out.println("Enter Account Type:");
 Type=s.next();
 System.out.println("Enter Initial Balance:");
 init_bal=s.nextDouble();

 double amt;
 Account a=new Account(name,AccNo,Type,init_bal); Savings_acc
 sv=new Savings_acc(name,AccNo,init_bal); Current_acc ca=new
 Current_acc(name,AccNo,init_bal);

 while(true){
 if(Type.equalsIgnoreCase("savings")){
 System.out.println("----MENU----");
 System.out.println("Enter 1:Deposit 2:Withdra 3:Interest
 4:Display Details 5:Exit");
 int ch=s.nextInt();
 switch(ch){
 case 1:
 System.out.println("Enter The Amount:");
 amt=s.nextDouble();
 a.deposit(amt);
 break;
 case 2:
 System.out.println("Enter the withdrawing amount");
 amt=s.nextDouble();
 a.withdraw(amt);
 break;
 case 3:
 sv.callInt();
 break;
 case 4:
 a.display();
 break;
 case 5:
 System.exit(0);
 }
 }
 }
 }
}

```

```

 default:
 System.out.println("Invalid Choice");
 }

}

else{
 System.out.println("----MENU---");
System.out.println("Enter 1:Deposit 2:Withdraw 3:Display Details 4:Exit");
int ch=s.nextInt();

switch(ch){
 case 1:
 System.out.println("Enter The Amount:");
 amt=s.nextDouble();
 a.deposit(amt);
 break;
 case 2:
 System.out.println("Enter the withdrawing amount");
 amt=s.nextDouble();
 a.withdraw(amt);
 ca.check_bal();
 break;
 case 3:
 a.display();
 break;
 case 4:
 System.exit(0);
default:
 System.out.println("Invalid Choice");
}
}
System.out.println("Sharana basappa K Tondihal 1BM22CS253"); }

}
}

```

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

```

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);
 this.usn=s.nextLine();
 this.name=s.nextLine();
 this.sem=s.nextInt();
}

public void displayStudentDetails(){
 System.out.println(this.usn+" "+this.name+
"+this.sem);
}
}

```

10

```

package CIE;
import java.util.Scanner;
public class Internals extends Student{
 protected int marks[]={};
 public void inputCIEmarks(){
 Scanner s=new Scanner(System.in);
 for(int i=0;i<5;i++){
 marks[i]=s.nextInt();
 }
 }
}

package SEE;
import CIE.Internals;
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]);
 }

import SEE.*;

class Main1{
 public static void main(String args[]){
 int num=2;
 Externals finalMarks[]=new Externals[num];
 for(int i=0;i<num;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<num;i++){
 finalMarks[i].calculateFinalMarks();
 finalMarks[i].displayFinalMarks();
 }
 System.out.println(Sharana basappa K Tondihal 1BM22CS253");
 }
}

```

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

```

import java.util.Scanner;

class WrongAge extends Exception{

```

```

 WrongAge(String s){
 super(s);
 }
 }
 class InputScanner{
 Scanner sc;
 InputScanner(){
 sc=new Scanner(System.in);
 }
 }
 class Father extends InputScanner{
 int fatherAge;
 public Father() throws WrongAge{
 InputScanner sf=new InputScanner();
 fatherAge=sf.sc.nextInt();
 if(fatherAge<0){
 throw new WrongAge("Age cannot be negative");
 }
 }
 void Fdisplay(){
 System.out.println("Father's Age: "+fatherAge); }
 }
 class Son extends Father{
 int sonAge;
 public Son() throws WrongAge{
 InputScanner ss=new InputScanner();

12
 sonAge=ss.sc.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 Sdisplay(){
 System.out.println("Son's Age: "+sonAge);
 }
 }

public class AgeCheck{
 public static void main(String args[]){
 Son a;
 try{
 a=new Son();
 a.Fdisplay();
 a.display();
 }
 }
}

```

```

 catch(WrongAge e){
 System.out.println(e);
 }
 System.out.println("Sharana basappa K Tondihal 1BM22CS253"); }
 }
}

```

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

```

class Thread1 implements Runnable{
 Thread t;
 public Thread1()
 {
 t=new Thread(this, "NThread");

 System.out.println("CT:"+t);

 t.start();
 }
 public void run()
 {
 try
 {
 for(int n=5;n>0;n--)
 {
 System.out.println("CSE "+n);
 Thread.sleep(2000);

 }
 }
 catch(InterruptedException ie) {
 System.out.println("CSE Interrupted"); } System.out.println("CSE quitting");
 }

 }
}

public class PrintColleg {

```

```

public static void main(String ss[])
{
new Thread1();
System.out.println("Back in main");
try
{
for(int n=2;n>0;n--)
{
System.out.println("BMSCE "+n);

Thread.sleep(10000);
}
}
catch(InterruptedException ie){

System.out.println("BMSCE interrupted");
}
System.out.println("BMSCE quitting");
System.out.println("Sharana basappa K Tondihal 1BM22CS253");
}
}

```

10) Demonstrate Inter process Communication and deadlock  
a) Inter process Communication

```

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<4) {

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<4) {
 int r=q.get();
 System.out.println("consumed:"+r); 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.");
 System.out.println("Sharana basappa K Tondihal 1BM22CS253");
}
}

```

### b)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 thisthread.

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

```

public void run() {
 b.bar(a); // get lock on b in otherthread.
 System.out.println("Back in other thread"); }

public static void main(String args[]) {
 new Deadlock();
 System.out.println("Sharana basappa K Tondihal 1BM22CS253"); }

}

```

9)Write a program that creates a user interface to perform integer divisions.The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked.If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class SwingDemo1 {
 SwingDemo1(){
 // create jframe container
 JFrame jfrm = new JFrame("Divider App"); jfrm.setSize(275,
 150);
 jfrm.setLayout(new FlowLayout());
 // to terminate on close
 jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

 // text label
 JLabel jlab = new JLabel("Enter the divider and divident:");

 // add text field for both numbers
 JTextField ajtf = new JTextField(8);
 JTextField bjtf = new JTextField(8);

 // calc button
 JButton button = new JButton("Calculate");

 // labels
 JLabel err = new JLabel();
 JLabel alab = new JLabel();
 JLabel blab = new JLabel();

 JLabel anslab = new JLabel();

 // add in order :)
 jfrm.add(err); // to display error bois 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(ArithmeticException e){
alab.setText("");
blab.setText("");
anslab.setText("");
err.setText("B should be NON zero!"); }
}
});

// display frame
jfrm.setVisible(true);
}

public static void main(String args[]){
// create frame on event dispatching thread
SwingUtilities.invokeLater(new Runnable(){ public void run(){
new SwingDemo1();
}
});
System.out.println("Sharana basappa K Tondihal 1BM22CS253"); }
}

```

