

## PROGRAM - 7

- ① Develop a Java program that finds all real solutions to the quadratic equation  $ax^2 + bx + c = 0$ . Read in  $a, b, c$  & 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;  
import static java.lang.Math.sqrt;  
import static java.lang.Math.abs;  
  
public class quad {  
    public static void main (String [] args) {  
        Scanner in = new Scanner (System.in);  
        System.out.println ("Enter coefficients : ");  
        int a = in.nextInt ();  
        int b = in.nextInt ();  
        int c = in.nextInt ();  
        if (a == 0) {  
            System.out.println ("Invalid input");  
        }  
        else {  
            int d = b * b - 4 * a * c;  
            if (d > 0) {
```

```
System.out.println ("Roots are real ");
float r1 = (float) (-b + sqrt (d)) / (2 * a);
float r2 = (float) (-b - sqrt (d)) / (2 * a);
System.out.println (r1);
System.out.println (r2);
```

{

else if (d &lt; 0) {

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

```
float r1 = (float) - b / (2 * a);
float r2 = (float) sqrt (abs (d)) / (2 * a);
System.out.println (r1 + " + i " + r2);
System.out.println (r1 + " - i " + r2);
```

}

else {

```
System.out.print (n ("Roots are equal "));
```

~~float r = (float) - b / (2 \* a);~~~~System.out.println (r);~~

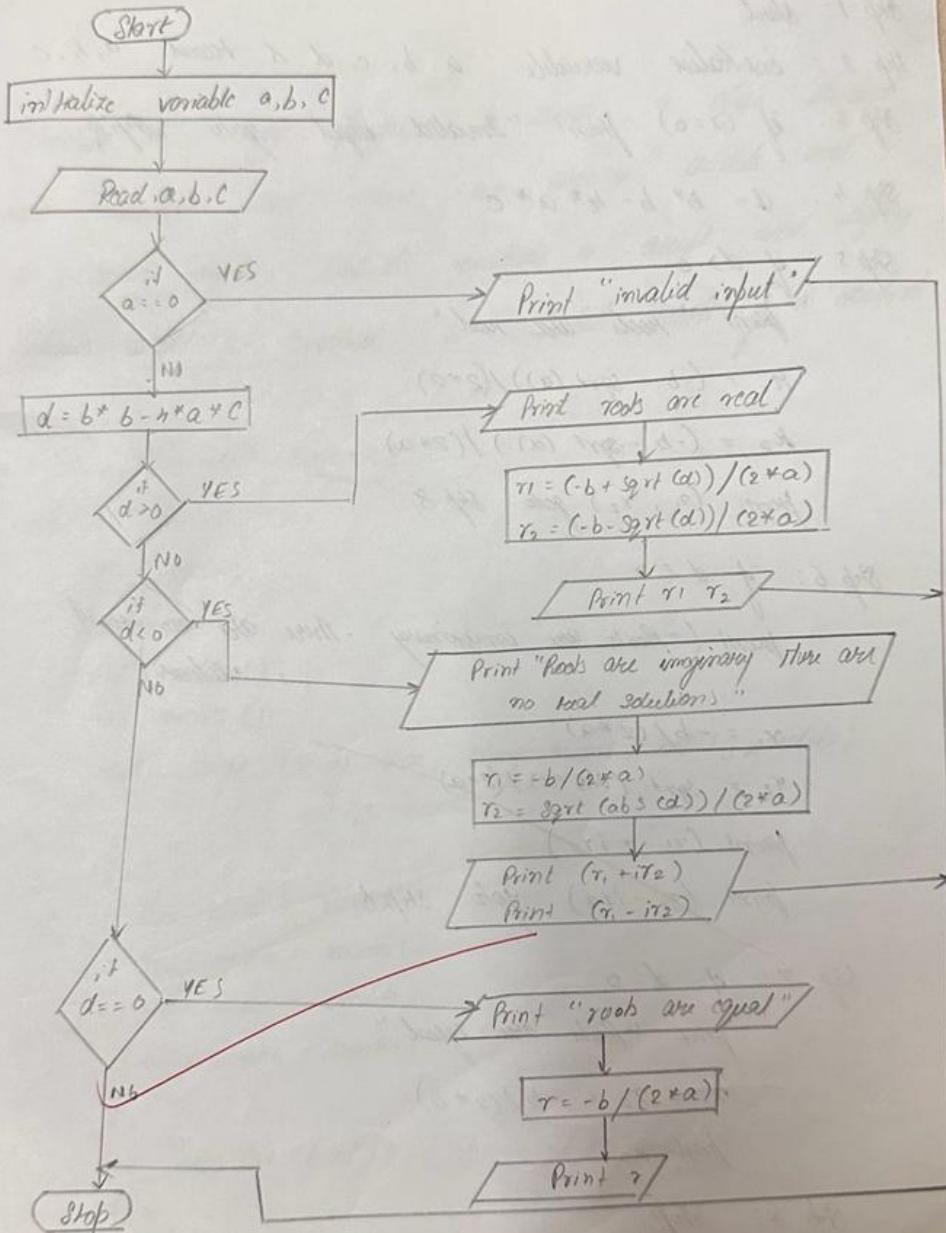
}

}

}

}

Flowchart:



ALGORITHM:

Step 1: Start

Step 2: initialise variable  $a, b, c, d$  & read  $a, b, c$

Step 3: if ( $a=0$ ) print "Invalid input" goto Step 8

Step 4:  $d = b^2 - 4 * a * c$

Step 5: if  $d > 0$

print "roots are real"

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

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

print  $(r_1, r_2)$  goto Step 8

Step 6: if  $d < 0$

print ("Roots are imaginary. There are no real solutions")

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

~~$$r_2 = \sqrt{|d|} * (a / (2 * a))$$~~

~~print  $(r_1 + ir_2)$~~ ~~print  $(r_1 - ir_2)$  goto Step 8~~

Step 7: if  $d = 0$

print "Roots are equal"

$$r_1 = r_2 = -b / (2 * a)$$

print  $r_1$

Step 8: Stop

```
C:\Users\Arin\Desktop\oopreport>java QuadEq
Enter values of a, b, c:
1 2 1
Roots are real and equal
R1= -1.0          R2= -1.0
Name: Arin Dsouza
USN: 1BM22CS052
```

```
C:\Users\Arin\Desktop\oopreport>java QuadEq
Enter values of a, b, c:
6 7 6
Roots are imaginary
Name: Arin Dsouza
USN: 1BM22CS052
```

```
C:\Users\Arin\Desktop\oopreport>java QuadEq
Enter values of a, b, c:
1 6 1
Roots are real and distinct
R1= -0.17157288 R2= -5.8284273
Name: Arin Dsouza
USN: 1BM22CS052
```

20/12/23

- Q) 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 CGPA of a student.

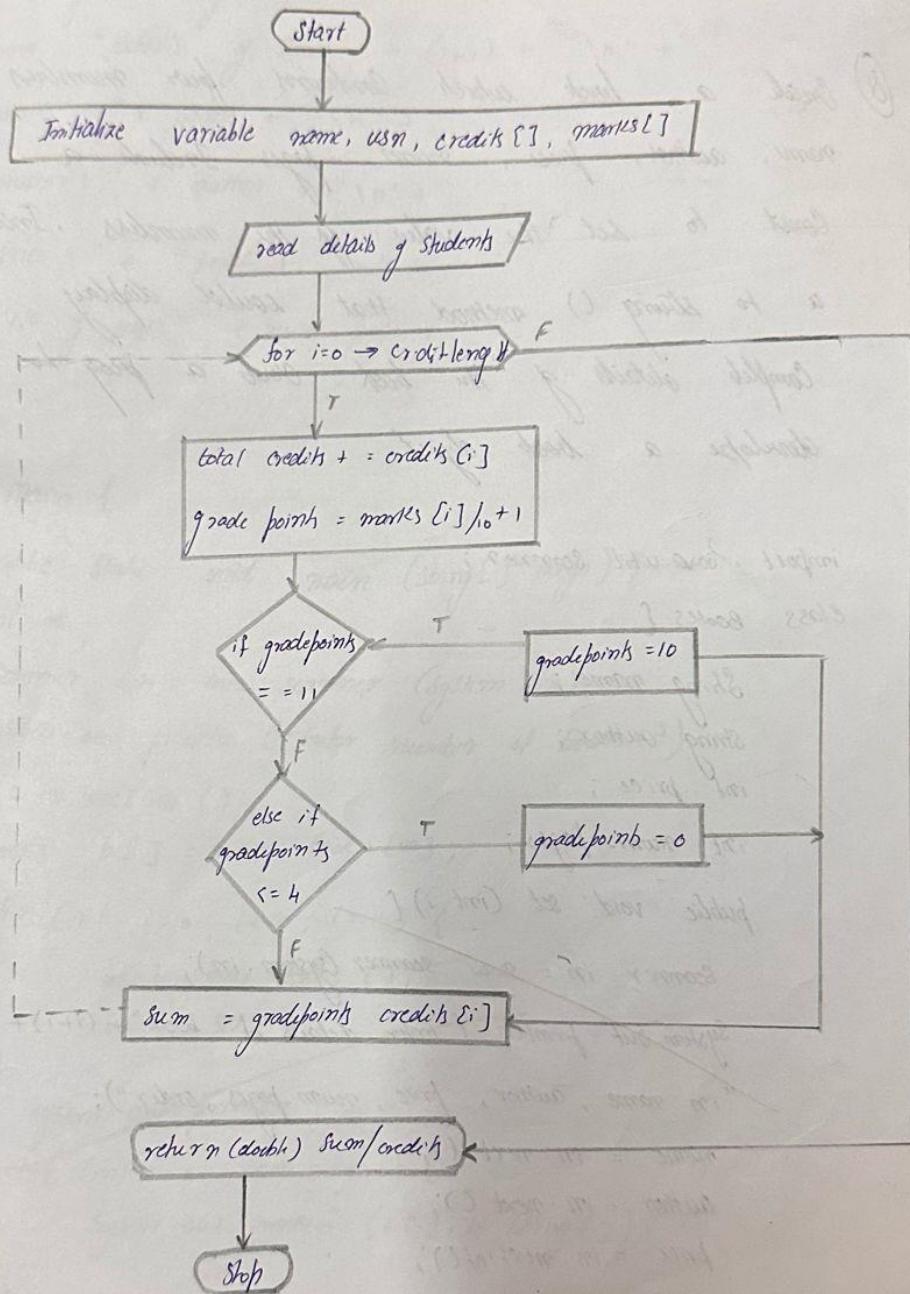
```
import java.util.Scanner;
class Student {
    String usn;
    String name;
    int credits[];
    int marks[];
    void accept (String usn, String name, int [] credits,
                 int [] marks)
    {
        this.usn = usn;
        this.name = name;
        this.credits = credits;
        this.marks = marks;
    }
    int calculate (int [] credits, int [] marks)
    {
        int sum1 = 0, sum2;
```

```
for (int i=0; i<8; i++)  
{  
    sum1 += credits[i] * marks[i];  
}  
sum2 = 20;  
System.out.println("1: " + sum1 + "2: " + sum2);  
return sum1 / sum2;  
}  
void display (float sgpa)  
{  
    System.out.println("usn: " + usn);  
    System.out.println("name: " + name);  
    System.out.println("sgpa: " + sgpa);  
}  
}  
class Main {  
    public static void main (String [] args)  
    {  
        char name, usn;  
        Scanner sc = new  
        Scanner (System.in);  
        System.out.println ("enter size");  
        int size = sc.nextInt();
```

```
int [] marks = new int [size];
int [] credits = new int [size];
float res;
Student st = new Student ();

System.out.println ("enter usn");
String usn = sc.nextLine();
System.out.println ("enter name");
String name = sc.nextLine();
System.out.println ("enter marks considering
    91-100 as 10, 81-90 as 9 and so on");
for (int i=0; i< size; i++)
{
    marks [i] = sc.nextInt();
}
System.out.println ("enter credits in sequential order");
for (int i=0; i< size; i++)
{
    credits [i] = sc.nextInt();
}
st.accept (usn, name, marks, credits);
res = st.calculate (marks, credits);
st.display (res);
}
```

Flowchart:



```
C:\Users\Arin\Desktop\oopreport>java Student

Enter name: aryan

Enter USN: 1BM22CS055

Enter no. of subjects: 4
Enter marks and credits:
Marks for subject 1:
90
Credits for subject 1:
4
Marks for subject 2:
96
Credits for subject 2:
4
Marks for subject 3:
88
Credits for subject 3:
3
Marks for subject 4:
92
Credits for subject 4:
3
Name : aryan
USN : 1BM22CS055
Subject 1 : Marks= 90 Credits= 4
Subject 2 : Marks= 96 Credits= 4
Subject 3 : Marks= 88 Credits= 3
Subject 4 : Marks= 92 Credits= 3

SGPA : 9.785714285714286
Name: Amrutha Ravi
USN: 1BM22CS036
```

⑧ Create a book which contains four numbers  
name, author, price, num - pages. Include a  
Const to set the value for the numbers . Include  
a to string () method that could display  
complet. details of the book. Create a prog to  
develop a book object.

```
import java.util.Scanner;  
class Books {  
    String name;  
    String author;  
    int price;  
    int num - pages;  
    public void set (int i) {  
        Scanner in = new Scanner (System.in);  
        System.out.println ("Enter details of book " + (i+1) +  
            " in name , author , price , num - pages , order ");  
        name = in.next();  
        author = in.next();  
        price = in.nextInt();  
        numpages = in.nextInt();  
    }  
}
```

```
public string to string () {  
    return "Details of Book" + (i+1) + "\n" +  
    "Name : " + name + "\n" +  
    "Author : " + author + "\n" +  
    "Price = " + price + "\n" +  
    "No. of pages = " + num - pages;  
}
```

```
class Main {  
    public static void main (string[] args) {  
        int n;  
        Scanner in = new Scanner (System.in);  
        System.out.println ("Enter number of books :")  
        n = in.nextInt ();  
        Books b[] = new Books [n];  
        for (int i=0; i<n; i++) {  
            b[i] = new Books ();  
            b[i].set (i);  
        }  
        System.out.println ();  
        for (int i=0; i<n; i++) {  
            System.out.println (b[i].to string());  
        }  
    }  
}
```

OUTPUT:

Enter details for book 2:

Price : INR 800

Num of pages : 62

Enter name :

Priy

Book 2:

Enter author :

abc

Name : xyzp

Enter price :

1000

Author : abc

Enter number of pages:

82

price : INR 800

Num of pages : 85

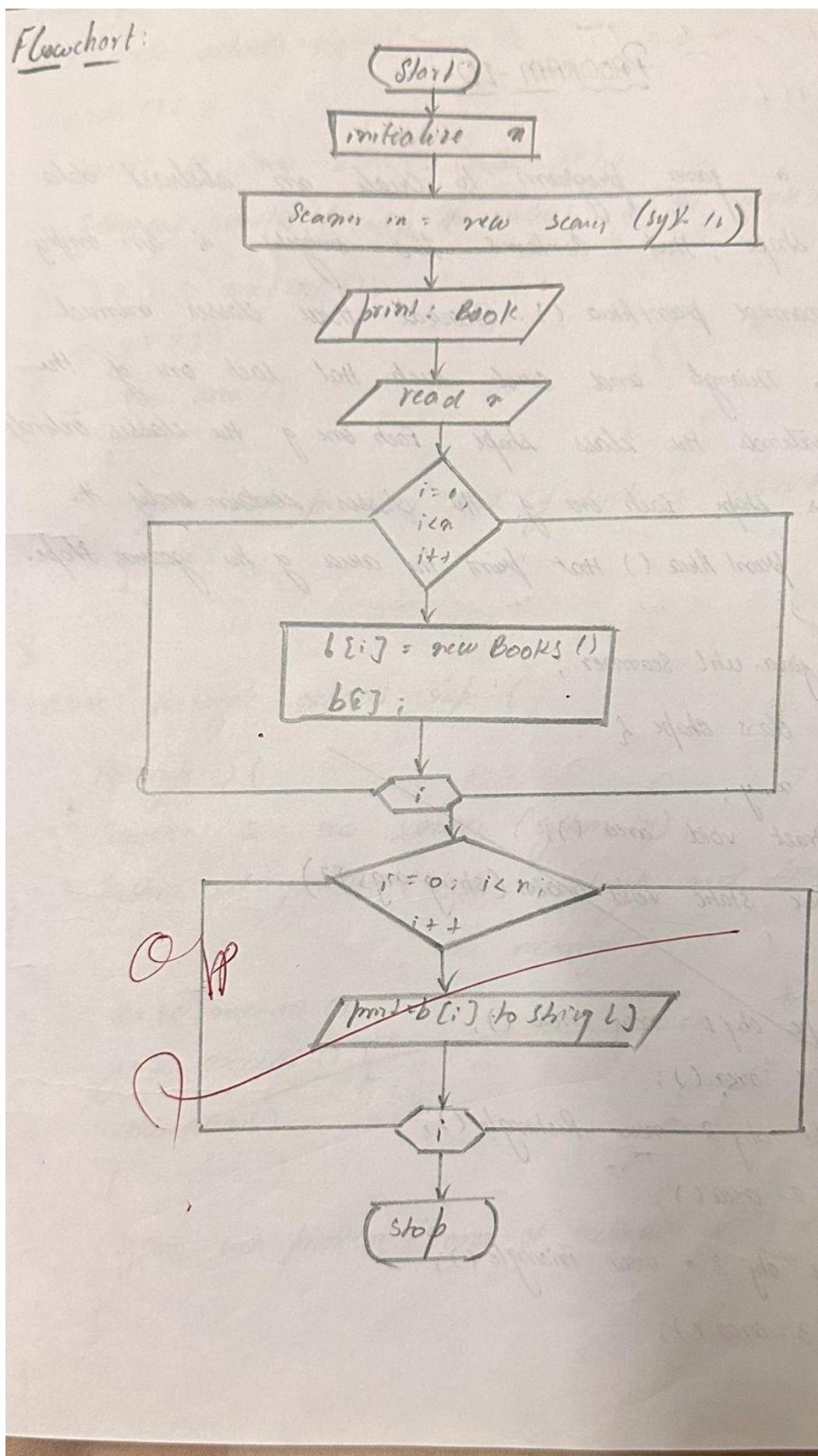
Details of all books :

Book 1:

Name : xyz

Author : xyz

Flowchart:



```
C:\Users\Arin\Desktop\oopreport>java bookdetails
Constructor values :
Name : Mehta
Author : neil
Price : 56
Number of pages : 154
Enter the number of object of books
1
Enter the details of 1book
Enter the name of the book
Time never stops
Enter the name of the author
steph curry
Enter the price
599
Enter the number of pages
400

Details of the book1
Name : Time never stops
Author : steph curry
Price : 599
Number of pages : 400
Arin
1BM22CS052
```

## PROGRAM -10

Q Develop a java program to create an abstract data named shape, that contains two integers & 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 contains only the method printArea() that prints the area of the given shape.

```
import java.util.Scanner;  
abstract class shape {  
    int x,y;  
    abstract void area();  
    public static void main (String args [ ])  
    {  
        shape obj1 = new circle ();  
        obj1.area();  
        shape obj2 = new Rectangle ();  
        obj2.area();  
        shape obj3 = new Triangle ();  
        obj3.area();  
    }  
}
```

class circle extends shape {

circle () {

Scanner sc = new Scanner (System. in);

System.out.println ("enter the radius of the circle");

x = sc.nextInt();

y = x;)

Void area ()

{

System.out.println ("area of circle is "+3.14\*x\*x);

}

}

class Rectangle extends shape {

Rectangle () {

Scanner sc = new Scanner (System. in);

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

x = sc.nextInt();

y = sc.nextInt();}

Void area ()

{

System.out.println ("area of rectangle is "+x\*y);

}

}

class Triangle extends shape {

Triangle () {

Scanner sc = new Scanner (System.in);

System.out.println ("enter the base and height of the triangle");

x = sc.nextInt();

y = sc.nextInt();

void area ()

{

System.out.println ("area of triangle is " + 0.5 \* x \* y);

}

}

OUTPUT:

enter the radius of the circle

2

Area of circle is 12.56

enter the length and breadth of the rectangle

3

4

Area of rectangle is 12

enter the base and height of the triangle

4

4

Area of triangle is 8.0

Algorithm:

- shp① Create abstract class name & shape
- shp② Include 2 numbers,  $x$  &  $y$
- shp③ Place abstract method area()
- shp④ Create sub class rectangle that extends shape
- shp⑤ override area method to calculate area of rectangle

rectangle

- shp⑥ Repeat steps for square & circle
- shp⑦ Create object rectangle in main method area with others
- shp⑧ Rest

```
C:\Users\Arin\Desktop\ojreport>java AreaMain
Arin Dsouza - 1BM22CS052
Enter d1 and d2:
2 4
Enter d1 and d2:
2 6
Enter d1:
3
Area of rectangle is: 8.0
Area of triangle is: 6.0
Area of circle: 28.25999999999998
```

## PROGRAM-1

- Q. 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 facilities 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 ~~class~~ derive the classes cur-account and sav-account to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks
- ① Accept deposit from customer & update the balance.
  - ② Compute and deposit interest
  - ③ Display the balance
  - ④ Permit withdrawal & update the balance check for the minimum balance, impose penalty if necessary & update the balance.

```
import java.util.Scanner;  
  
class Account {  
    String customerName;  
    long accno;  
    String accountType;  
    double balance;  
    public Account (String CustomerName, long accno, String  
                    accountType) {  
        this.customerName = CustomerName;  
        this.accno = accno;  
        this.accountType = accountType;  
        this.balance = 0.0;  
    }  
    public void displayBalance () {  
        System.out.println ("Account Number : " + accno);  
        System.out.println ("Customer Name : " + customerName);  
        System.out.println ("Account Type : " + accountType);  
        System.out.println ("Balance : # " + balance);  
    }  
}  
class CurAccet extends Account {  
    double minBalance;  
    double serviceCharge;
```

```
public Car Act (String CustomerName, long accno) {  
    super (CustomerName, accno, "Current");
```

```
    this. minBalance = 500.0;
```

```
    this. servicecharge = 50.0;
```

```
}
```

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

```
{
```

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

```
        balance -= amount >= minBalance) {
```

```
            balance -= amount;
```

```
            System.out.println ("withdrawal successful. Current
```

```
Balance : $" + balance); }
```

```
else
```

```
{
```

~~System.out.println ("withdrawal successful. Current~~~~Balance : \$" + balance); }~~~~else~~~~{~~~~System.out.println ("Insufficient funds. withdrawal~~~~not allowed.");~~

```
}
```

```
}
```

```
public void imposeServiceCharge () {
```

```
    if (balance < minBalance) {
```

balance = Service Charge;  
System.out.println ("Service Charge imposed - Current Balance:  
Rs. " + balance);

{

}

{

class SavAcct extends Account {

double interestRate;

public SavAcct (String CustomerName, long AcuNo) {

{

super (CustomerName, acno, "savings");

this . interestRate = 0.05;

{

public void depositInterest () {

double interest = balance \* interestRate;

balance += interest;

~~System.out.println ("Interest deposited . Current Balance:  
\$" + balance);~~

{

public void CompoundInterest (double initialAmount, int for,

{

double CompoundInterest = initialAmount \* Math.pow ((1 +

```
interestRate), term) -> minia(amount ;  
balance += CompoundInterest ;  
System.out.println ("Compound Interest deposited - Current  
Balance : Rs " + balance);
```

{

{

```
public class Bank {  
    public static void main (String [] args) {  
        Scanner Scanner = new  
        Scanner (System.in);  
        System.out.println ("Choose account type:");  
        System.out.println ("1. Current");  
        System.out.println ("2. Savings");  
        System.out.println ("Enter choice (1 or 2):");  
        int choice = Scanner.nextInt ();  
        System.out.println ("Enter customer name:");  
        String CustomerName = Scanner.next ();  
        System.out.print ("Enter account number:");
```

```
long acno = scanner.nextInt();
if (choice == 1) {
    CusAcct curAccount = new CusAcct (Customer Name, acno);
    System.out.println ("Enter initial balance : $");
    double initialBalance = scanner.nextDouble();
    curAccount.setBalance (initialBalance);
    System.out.print ("Enter withdrawal amount : $");
    double withdrawalAmount = scanner.nextDouble();
    curAccount.withdraw (withdrawalAmount);
    curAccount.imposeServiceCharge ();
    curAccount.displayBalance ();
}
else if (choice == 2) {
    ServAcct servAccount = new ServAcct (Customer Name, acno);
    System.out.print ("Enter initial balance : $");
    double initialBalance = scanner.nextDouble();
    servAccount.setBalance (initialBalance);
    System.out.print ("Enter withdrawal amount : $");
}
```



```
C:\Users\Arin\Desktop\oopreport>java Bank
Name : Arin Dsouza
USN:IBM22CS052
Enter the number of users: 1

User 1
Enter customer name: joshua
Enter account number: 12345
Enter initial deposit amount: INR 7000
Enter account type (Savings/Current): Current

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 1
Enter account number: 12345
Enter deposit amount: INR 500
Deposit of INR 500.0 successful

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 2
Enter account number: 12345
Enter withdrawal amount: INR 1000
Withdrawal of INR 1000.0 successful

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 3
Enter account number: 12345
Account Number: 12345
Customer Name: joshua
Account Type: Current
Balance: INR 6500.0

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 4
Enter account number (for Savings account): 12345

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 5

C:\Users\Arin\Desktop\oopreport>
```

```
C:\Users\Arin\Desktop\oopreport>java Bank
Name : Arin Dsouza
USN:1BM22CS052
Enter the number of users: 1

User 1
Enter customer name: joshua
Enter account number: 12345
Enter initial deposit amount: INR 7000
Enter account type (Savings/Current): Current

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 1
Enter account number: 12345
Enter deposit amount: INR 500
Deposit of INR 500.0 successful

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 2
Enter account number: 12345
Enter withdrawal amount: INR 1000
Withdrawal of INR 1000.0 successful

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 3
Enter account number: 12345
Account Number: 12345
Customer Name: joshua
Account Type: Current
Balance: INR 6500.0

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 4
Enter account number (for Savings account): 12345
```

## PROGRAM - 12

With a program that demonstrates handling of exceptions in inheritance true. 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 uses both father and son's age and throws an exception if son's age is  $\geq$  father's age.

Ans:-

```
Import java.util.Scanner;  
class wrongAgeException extends Exception {  
    public wrongAgeException (String message) {  
        super (message);  
    }  
}  
class Father {  
    private int age;  
    public Father (int age) throws wrongAgeException {  
        if (age < 0) {
```

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

}

this.age = age;

```
public int getAge() {  
    return age;  
}
```

}

```
class Son extends Father {
```

private int SonAge;

Public Son (int fatherAge, int sonAge) throws WrongAgeException {

super(fatherAge);

if (sonAge >= fatherAge) {

throw new wrongAgeException ("sons age should be less  
than fathers age");

}

this.SonAge = SonAge;

public int getSonAge() {

return SonAge;

}

}

```

public class FatherSon {
    public static void main (String args) {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter father and son age");
        int fa = sc.nextInt ();
        int sa = sc.nextInt ();
        try {
            Sons s = new Son (fa, sa);
            System.out.println ("Son's age : " + s.sonAge);
        }
    }
}

```

```

    catch (WrongAge e) {
        System.out.println ("Error : " + e.getMessage ());
    }
}

```

OUTPUT:	
<u>case:1</u> Enter father's age : 50 Enter son's age : 20 Father's age : 50 Son's age : 20	<u>case:2</u> Enter father's age : 40 Enter son's age : 45 Exception Caught : wrong Age: Son's Age Cannot be greater or equal to father's age
Exception Caught : Son's Age cannot be greater than or equal to father's age	

```
C:\Users\Arin\Desktop\oopreport>java AgeTest
Enter age of father and son

50
19
Name: Arin Dsouza
USN: 1BM22CS052

C:\Users\Arin\Desktop\oopreport>java AgeTest
Enter age of father and son

-10
30
Age cant be less than 0
Caught
Name: Arin Dsouza
USN: 1BM22CS052

C:\Users\Arin\Desktop\oopreport>javac AgeTest
error: Class names, 'AgeTest', are only accepted if annotation processing is explicitly requested
1 error

C:\Users\Arin\Desktop\oopreport>java AgeTest
Enter age of father and son

20
30
Father can't be younger than son
Caught
Name: Arin Dsouza
USN: 1BM22CS052
```

### PROGRAM - 13

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

# Class A extends Thread  
{  
    int t1, time;  
    A() {  
        t1 = 10,000;  
        time = 21006;  
    }  
    }  
    public void run()  
    {  
        while (t1 <= time)  
        {  
            System.out.println ("BMS COLLEGE OF ENGINEERING");  
            sleep (10000);  
        }  
        try {  
            catch (InterruptedException) {  
                System.out.println ("error");  
        }  
    }

```
t1 += 10000;  
}  
}  
  
class B extends Thread {  
    int t2, time;  
  
    B() {  
        time = 21000;  
        t2 = 2000;  
    }  
  
    public void run() {  
        while (t2 <= time)  
        {  
            System.out.println ("CSE");  
            by();  
            sleep (2000);  
        }  
        catch (Exception e)  
        {  
            System.out.println ("error");  
        }  
        t2 += 2000;  
    }  
}
```

}

Class th

{

public static void main (String args [])

{

A a = new A();

B b = new B();

a.start();

b.start();

}

}

### OUTPUT:

BMS COLLEGE OF ENGINEERING

CSE

### (13) ALGORITHM

Step 1: Define class A.

Initialize variable  $t_1$  & time

Step 2: In constructor, set  $t_1$  to 10000 &  
time to 21000.

Step 3: Override run method, while loop to  
print BMS College of Engineering at interval of  
10 seconds.

Step 4: Define class B extends thread, initialize  
variable  $t_2$  & time.

Step 5: while loop to print cost every 2 sec.

Step 6: Catch exception and print error.

Step 7: Define the main class, create instance  
of class A & B.

Step 8: start both threads.

8/10  
16/2/22

```
C:\Users\Arin\Desktop\oopreport>java Demo
Name:Arin Dsouza
USN:1BM22CS052
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
```

## PROGRAM - 14

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

Sols:

```
import java.awt.*;  
import java.awt.event.*;  
  
class SwingDemo {  
    SwingDemo() {}  
  
    JFrame jfrm = new JFrame ("Divide App");  
    jfrm.setSize(275, 150);  
    jfrm.setLayout (new FlowLayout());  
  
    jfrm.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
```

```
label glab = new label ("Enter the divisor and dividend");  
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 (glab);  
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");  
    }  
}
```

```
};

ajtt.add ActionListener (l);
bjtt.add ActionListener (l);

button.addActionListener (new ActionListener () {
    public void actionPerformed (ActionEvent evt) {
        try {
            int a = Integer.parseInt (ajtt.getText ());
            int b = Integer.parseInt (bjtt.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 (ArithmetricException e) {
            alab.setText ("");
            blab.setText ("");
            ansLab.setText ("");
            err.setText ("B Should be Non Zero !");
        }
    }
})
```

```
});
```

```
    form.setVisible(true);
```

```
}
```

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

```
    swingUtilities.invokeLater (new Runnable) {
```

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

```
            new SwingDemo ();
```

```
}
```

```
});
```

```
}
```

```
C:\Users\Arin\Desktop\oopreport>javac SwingDemo.java  
C:\Users\Arin\Desktop\oopreport>java SwingDemo
```



Divider app



**B should be NON zero!**

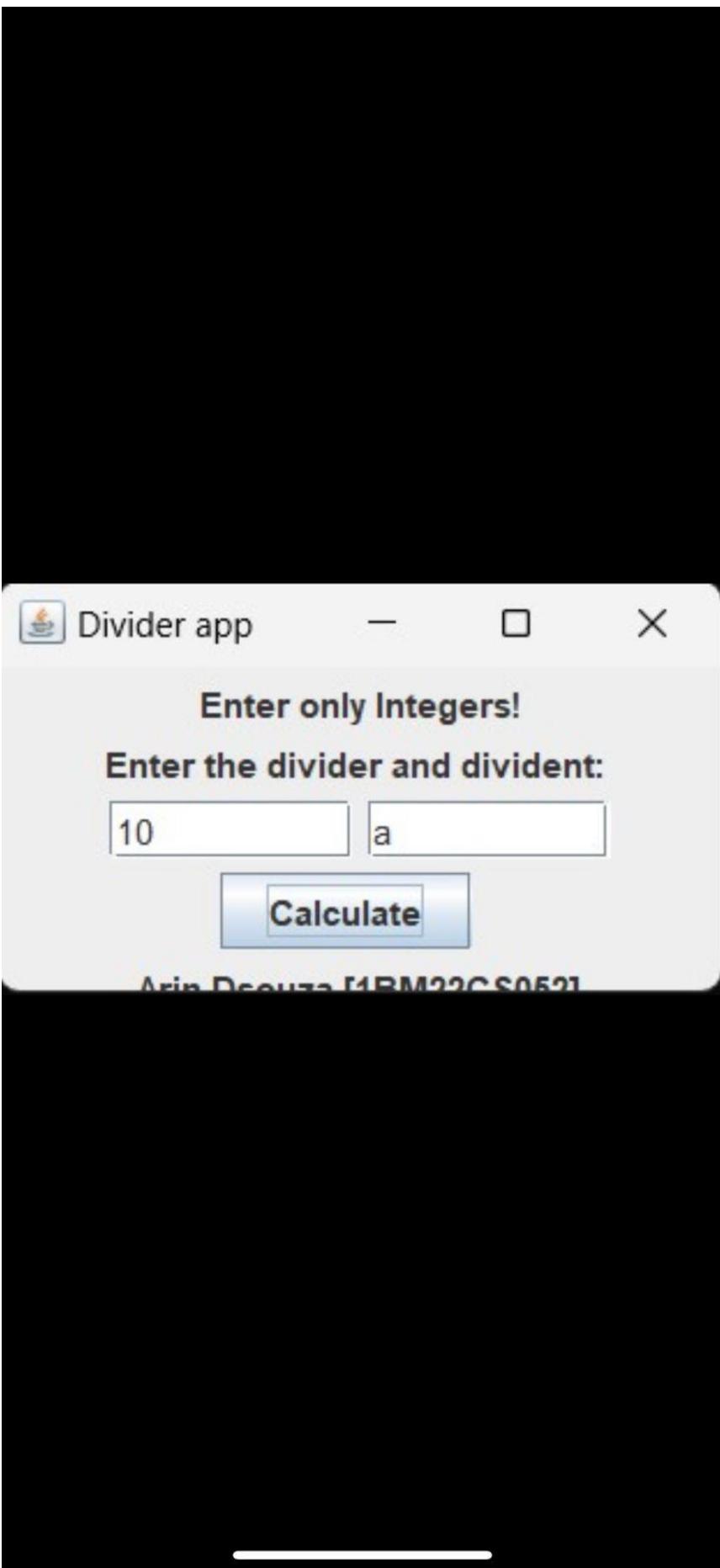
**Enter the divider and divident:**

10

0

**Calculate**

Arin D'Souza MRM22CS0521



 Divider app

Enter the divider and divident:

10	2
----	---

**Calculate**    A= 10 B= 2

Ans= 5 Arin Dsouza [1BM22CS052]

## PROGRAM - 15

Create a package CIE which has two classes -  
Student and Internal . The class Personal has members  
like usn, name , sem . The class internal has an array  
that stores the internal marks scored in four 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 four 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 four courses .

### Program:

Package CIE ;

import java.util\*;

public class Student {

    public String name ;

    public String usn ;

    public int sem ;

```
public void display() {
    Scanner sc = new Scanner(System.in);
    System.out.println("Name");
    name = sc.next();
    System.out.println("USN");
    usn = sc.next();
    System.out.println("Sem:");
    sem = sc.nextInt();
}

package CIE;
import java.util.*;
public class Internals extends Student {
    public double view[] = new double[5];
    public void display() {
        view = new double[5];
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter CIE marks out of 50");
        for (int i=0; i<5; i++) {
            view[i] = sc.nextDouble();
        }
    }
}
```

```
Package SFE;
import CIE.*;
import java.util.*;

public class extends extends CIE.Student {
    public double score [ ];
    public void display () {
        score = new double [5];
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter marks for 5 subjects
out of 100 : ");
        for (int i=0, i<5 ; i++) {
            score[i] = sc.nextDouble ();
        }
    }
}

import CIE.*;
import SFE.*;
import java.util.*;

public class Main {
    public static void main (String [] args) {
        int n;
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter no of students : ");
        n = sc.nextInt ();
        CIE.Student st [] = new CIE.Student [n];
```

CSE. Internal in[i] = new CSE. Internal [n];  
SEE. External ex[i] = new SEE. External [n];

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

st[i] = new CSE. Student();

in[i] = new CSE. Internals();

ex[i] = new SEE. Externals();

st[i]. display();

in[i]. display();

ex[i]. display();

System.out.println("total marks of " + st[i]. name + "\n");

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

System.out.println(in[i]. avg(j) + ex[i]. avg(j)/2);

}

}

OUTPUT:

Total marks of Akash

```
50
Enter the marks for subject 5:
50
Enter SEE marks
Subject1 marks:50
Subject2 marks:50
Subject3 marks:50
Subject4 marks:50
Subject5 marks:50
Displaying data:
```

```
USN: 1BM22CS052
Name: Arin Dsouza
Semester: 3
Subject1: 75
Subject2: 75
Subject3: 75
Subject4: 75
Subject5: 75
USN: AppleBee
Name: Alpha
Semester: 3
Subject1: 75
Subject2: 75
Subject3: 75
Subject4: 75
Subject5: 75
```