

P1 : WRITE A JAVA PROGRAM TO FIND THE QUADRATIC ROOTS OF THE GIVEN EQUATION.

Q) Quadratic equation (using input)

```
import java.util.*;  
public class Quadratic
```

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

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

```
System.out.print ("Enter value of a: ");
```

```
double a = sc.nextDouble();
```

```
System.out.print ("Enter value of b: ");
```

```
double b = sc.nextDouble();
```

```
System.out.print ("Enter the value of c: ");
```

```
double c = sc.nextDouble();
```

```
double d = b*b - 4.0*a*c;
```

```
if (d > 0.0)
```

```
{ double r1 = (-b + Math.sqrt (d, 0.5)) / (2.0*a);
```

```
double r2 = (-b - Math.sqrt (d, 0.5)) / (2.0*a);
```

```
System.out.println ("The roots are " + r1 + " and " + r2);
```

```
else if (d == 0.0)
```

```
{ double r1 = -b / (2.0*a);
```

```
System.out.println ("The root is " + r1);
```

```
else
```

```
{ System.out.println ("Roots are not real."); }
```

```
}
```

**OUTPUT:**

```
C:\Users\arjun\.jdks\openjdk-21.0.2\bin\java.exe
enter value of a:
3
Enter value of b:
4
Enter value of c
5
roots are not Real
ARJUN 1BM22CS053
```

```
C:\Users\arjun\.jdks\openjdk-21.0.2\bin\java.exe
enter value of a:
1
Enter value of b:
2
Enter value of c
1
The root is -1.0
ARJUN 1BM22CS053
```

## Output

Enter the value of a : 1

Enter the value of b : 1

Enter the value of c : 1

Roots are not real.

## Algorithm

Step 1: Start

Step 2: Declare variables  $a, b, D, d, c, r_1, r_2$

Step 3: Calculate discriminant ( $D = b \times b - 4 \times a \times c$ )

Step 4: if  $D == 0$

roots are real and equal

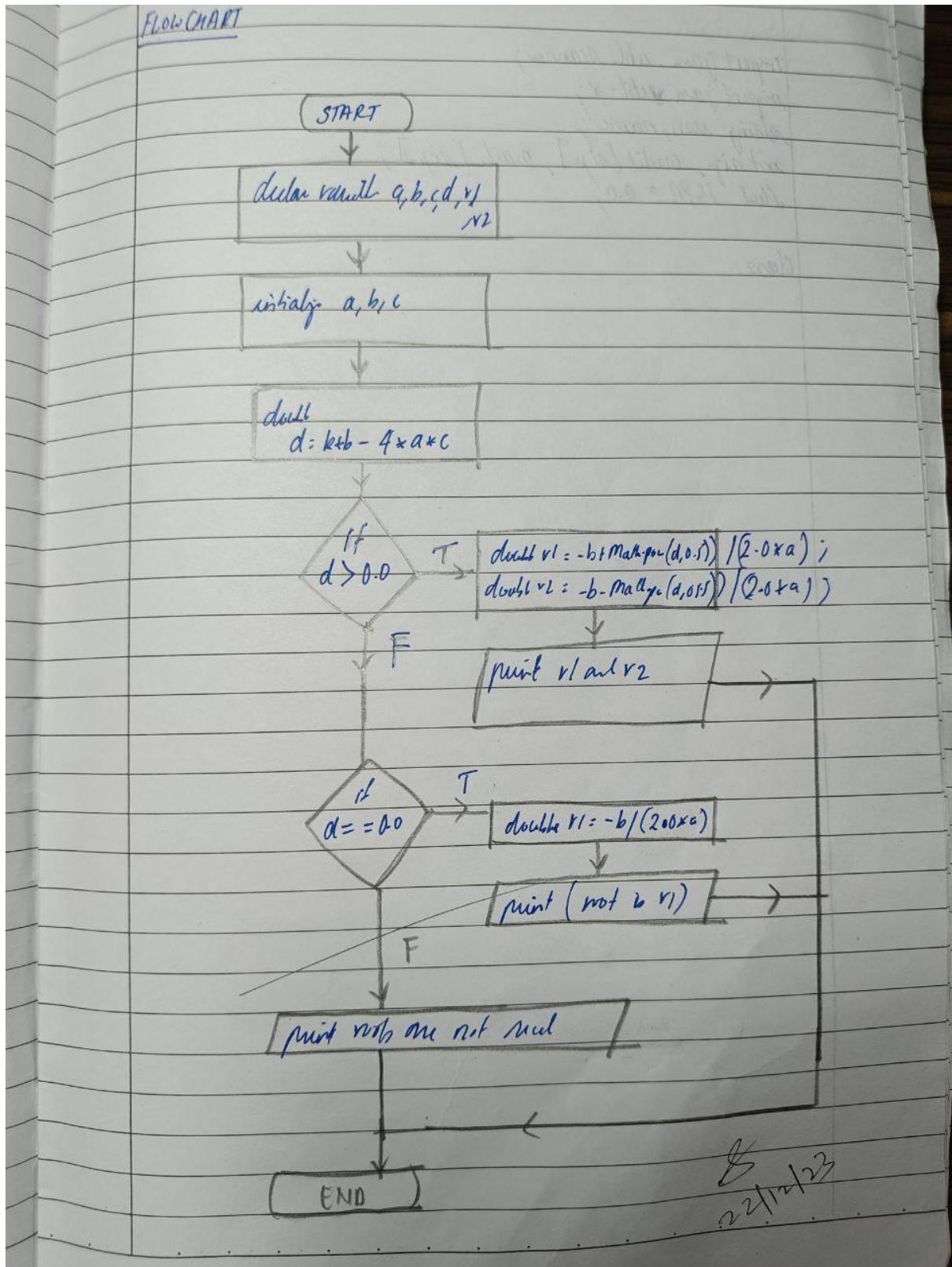
$$r_1 = r_2 = -b/2a$$

Step 5: if  $D > 0$  roots are real and distinct

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

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

Step 6: if  $D < 0$  roots are imaginary  
roots are not real.



PROGRAM 2 : WRITE A JAVA PROGRAM TO FIND THE SGPA OF A STUDENT

LAB-3

SGPA

```
import java.util.*;  
class Student {  
    String name, name;  
    int n, credit[n], marks[n], sum;  
    float SGPA = 0;  
  
    Scanner sc = new Scanner("System.in");  
    void Accept_and_Display() {  
        System.out.println("Enter your name");  
        name = sc.next();  
        System.out.println("Enter your name");  
        name = sc.next();  
  
        for (int i=0; i<n; i++)  
        {  
            System.out.println("Enter credit : ");  
            credit[i] = sc.nextInt();  
            System.out.println("Enter the mark");  
            marks[i] = sc.nextInt();  
        }  
    }  
}
```

```
void calculate () {
```

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

{

$$SGPA = SGPA + credit[i] * (int) marks[i] / 10;$$

```
    sum = sum + credit[i];
```

}

$$SGPA = SGPA / sum;$$

```
Algorithm-Del. print ("SGPA is : " + SGPA);
```

}

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

{

```
System.out.print ("Enter n. of subjects : ");
```

```
n = sc.nextInt();
```

```
Student S = new Student();
```

~~```
S.accepts And - Reply();
```~~~~```
S.calculate();
```~~

}

J

### Algorithm:

(1) Step 1: START

Step 2: declare variables int n, credit[n], marks[n], sum; float = 0.0;

Step 3: (make method Accept and display)

enter name, MNR

for i=0 and i < n

enter credit[i]

enter marks[i]

Step 4: make method calculate()

for i=0 & i < n

SGPA = SGPA + credit[i] \* (int)(marks / 10);

sum = sum + credit[i];

SGPA = SGPA / sum;

print (SGPA)

Step 5: make public static void main (String args[])

(question - output) enter no of subjects

Step 6: display SGPA

Step 7: STOP

### OUTPUT:

Enter no. of subjects: 2

Enter your name: Rohan

Enter your USN: 18M22CS046

Enter credit: 4

Enter marks: 80

Enter credit: 3

Enter marks: 82

SGPA = 8.857

```
C:\Users\arjun\.jdks\openjdk-21.0.2\bin\java.exe
Enter a no: of subjects
2
Enter your name
ARJUN
Enter your USN
7349
Enter credit:
4
Enter the marks
90
Enter credit:
3
Enter the marks
89
8.571428571428571
ARJUN 1BM22CS053
```

PROGRAM 3: WRITE A JAVA PROGRAM WHICH CONTAINS INFO ABOUT BOOK NAME,AUTHOUR,PRICE.

Lab - 4

URBAN  
EDGE

Create a class book that contains four member : name, author, price, num pages. Create a constructor & set the value of the member variables. Set all get the details of the object. Use a function, that contains display, the complete details of the object.

```
import java.util.Scanner;  
  
class Book{  
    private String author;  
    private String name;  
    private int price;  
    private int numPages;  
}  
  
public Book (String name, String author, int price, int pages)  
{  
    this.name = name;  
    this.author = author;  
    this.price = price;  
    this.numPages = numPages;  
}  
  
public void setName (String name)  
{  
    this.name = name;  
}  
  
public String getName () { return name; }  
  
public void setAuthor (String author)  
{  
    this.author = author;  
}  
  
public String getAuthor ()  
{  
    return author;  
}
```

public void setPrice (int price)

{  
    this.price = price;

}  
public int getPrice ()

{  
    return price;

}  
public void setPages (int numPages)

{  
    this.numPages = numPages;

}  
public int getPages ()

{  
    return numPages;

public String toString () {  
    return "Book Details : " + "Name : " + name + "Author : " +  
        author + "Price : \$" + price + "Number of pages : " + numPages;  
}

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

    Scanner sc = new Scanner (System.in);

    System.out.print ("Enter no. of books : ");

    int n = sc.nextInt ();

    Book [] book = new Book [n];

    for (int i = 0; i < n; i++)

{

```
System.out.print ("In Enter detail b book" + (i+1) + ":" );  
Scanner sc = new Scanner (System.in);
```

```
String name = sc.nextLine();
```

```
System.out.print ("Author : ");
```

```
String auth = sc.nextLine();
```

```
System.out.print ("Price : ");
```

```
int price = sc.nextInt();
```

```
System.out.print ("No of page : ");
```

```
int numpage = sc.nextInt();
```

```
book[i] = new Book (name, auth, price, numpage);  
}
```

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

```
System.out.print ("In Details of b book" + (i+1) + ":" );  
System.out.print (book[i]);
```

```
sc.close();
```

OUTPUT:

Enter the name of book : 1

Enter the details of the book 1 :

Name : Harry potter

Auth : JK rowling

price : 250

No of page : 270

Detail b book 1 :

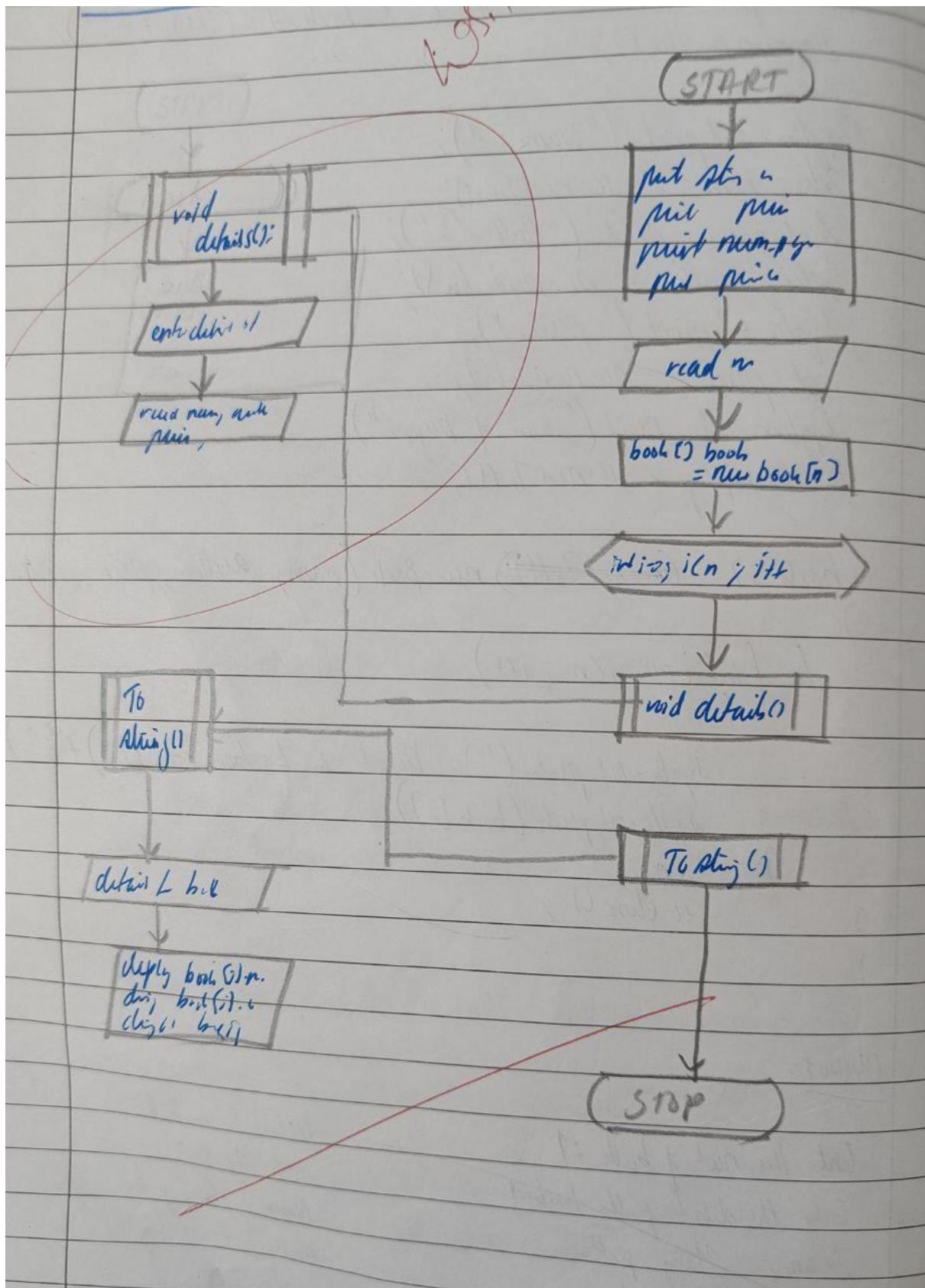
Book Detail :

Name : Harry potter

Auth : JK rowling

price : 250

page : 270



OUTPUT:

```
c:\Users\Arjun\Downloads\openjdk-21.0.2-bin\jave  
Enter the number of books: 2  
  
Enter details for Book 1:  
Name: The Metamorphosis  
Author: Franz Kafka  
Price: rs1000  
Number of Pages: 450  
  
Enter details for Book 2:  
Name: The Brothers Karamazov  
Author: Dostoevsky  
Price: rs950  
Number of Pages: 390  
  
Details for Book 1:  
Book Details:  
Name: The Metamorphosis  
Author: Franz Kafka  
Price: rs1000.0  
Number of Pages: 450  
  
Details for Book 2:  
Book Details:  
Name: The Brothers Karamazov  
Author: Dostoevsky  
Price: rs950.0  
Number of Pages: 390  
ARJUN 1BM22CS053
```

PROGRAM 4: WRITE A PROGRAM USING INHERITANCE TO FIND THE AREA OF CIRCLE, TRIANGLE, RECTANGLE.

Q)

Write a program using inheritance to find the area of the circle, rectangle and triangle.

abstract class Shape {

public int side1;

public int side2;

public Shape (int side1, int side2)

{ this.side1 = side1;

this.side2 = side2;

} public abstract void printArea();

class Rectangle extends Shape { public int height, width;

super (height, width);

public void printArea() {

int area = side1 \* side2;

System.out.println ("Area of rectangle = " + area);

class Circle extends Shape {

public Triangle (int base, int height) {

super (base, height);

public void printArea() {  
 double area = 0.5 + side1 \* side2;  
 System.out.println("Area of trapezoid = " + area);  
}

} class Circle extends Shape {

public Circle (int radius) {  
 super (radius, 0);

} public void printArea() {

double area = Math.PI \* radius \* radius;

} System.out.println("Area of circle = " + area);  
}

public class ShapeTest {

public static void main (String args[]) {

Rectangle rectangl = new Rectangle (4,5);

Triangle triangl = new Triangle (3,8);

Circle circl = new Circle (6);

rectangle.printArea();

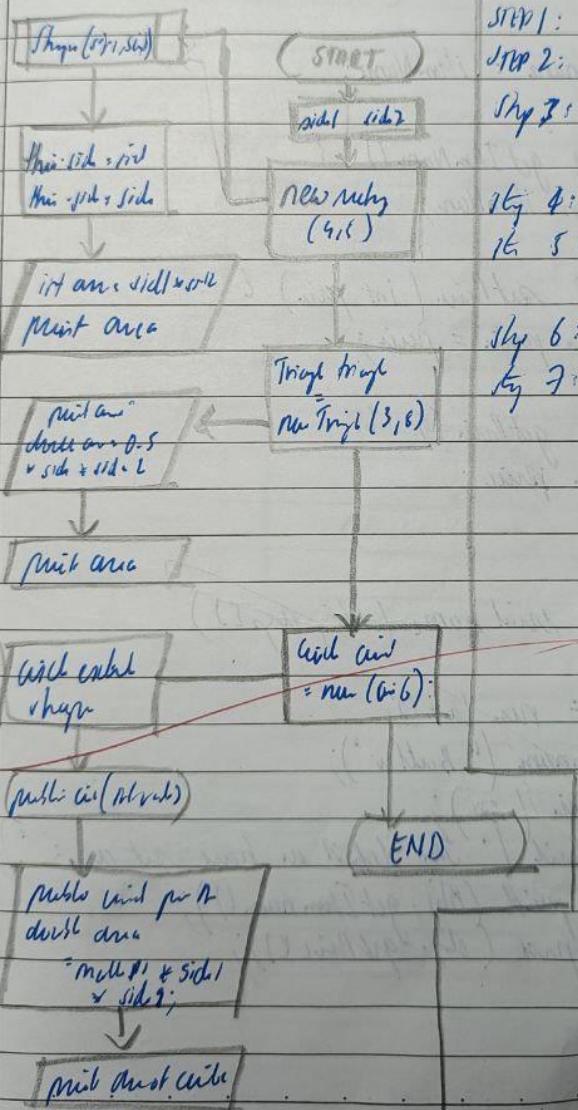
triangl.printArea();

circl.printArea();

OUTPUT:

Area of rectangle = 20  
 Area of right triangle = 12.0  
 Area of circle = 113.097

## FLOWCHART



Step 1: START

Step 2: mult. side1, sides

Step 3: mult. side1, side2

area rectangle

Step 4: abstract class for area

Step 5: class rectangle, triangle, circle

Step 6: print multi, area,

Step 7: exit -

(40)

OUTPUT:

```
C:\Users\arjun\.jdks\openjdk-21.0.2\bin\java.exe
Area of Rectangle: 20
Area of Triangle: 12.0
Area of Circle: 113.09733552923255
ARJUN 1BM22CS053
```

PROGRAM 5: WRITE A JAVA PROGRAM TO CREATE 2 KINDS OF ACCOUNT TO ITS CUSTOMERS  
ONE FOR SAVING ACCOUNT AND THE OTHER FOR CURRENT ACCOUNT.

Q

Develop a Java program to create two kinds of accounts from the customer, one kind of account by the customer on current saving account and the other account will be the saving account and the third account

70

import java.util.Scanner;

class Account {

String customerName;

int accountNumber;

String accountType;

double balance;

Account (String name, int accNo, String accType, double bal)

name = customerName;

accNo = accountNumber;

accType = accountType;

bal = balance;

}

void displayBalance()

~~System.out.println("The balance for the account " + accountNumber + " is " + balance);~~

class CurrentAccount extends Account {

double minBalance;

double annualCharge;

Current (String name, int acNumber, double initialBalance,  
double minBalance, double serviceCharge)  
{  
    Super (name, acNumber, "Current", initialBalance);

This - minBalance = minBalance  
This - ServiceCharge = ServiceCharge;

void withdraw (double amount)

{  
    if (balance <= amount < minBalance)

{  
    System.out.println ("Withdraw not permitted");  
    imposeServiceCharge();

. else  
{

    balance = balance - amount;

    System.out.println (" " + amount + " withdrawn");  
    displayBalance();

]  
]

private void imposeServiceCharge ()

{  
    System.out.print ("Service charge of " + serviceCharge +  
        " imposed");

    balance = balance - serviceCharge;

    displayBalance();

]  
]

class SavAcct extends Account {  
 double interestRate;  
 SavAcct (String name, int accNumber, double initialBalance,  
 double interestRate)  
 {  
 super (name, accNumber, "Savings", initialBalance);  
 this.interestRate = interestRate;  
 }  
}

void depositInterest ()  
{  
 double interest = balance \* interestRate / 100;  
 balance = balance + interest;  
}

System.out.println ("Enter y 2 "+ interest + " deposits")  
, displayBalance();

~~void withdraw (double amount) {  
 if (balance >= amount)  
 {~~

~~balance = balance - amount;  
 System.out.println ("2 " + amount + " withdrawn successfully");  
 displayBalance();  
 }  
}~~

~~else~~

~~{~~

~~System.out.println ("withdrawal not permitted. Insufficient funds.");  
}}~~

public class Bank

{  
public static void main (String args[])

Scanner sc = new Scanner (System.in);

Customer currentAmm = new Current ("John Doe", 12345,  
1000.0, 0.00, 0);

Customer savingAmm = new saving ("Jan Smith", 67890,  
2000.0, 5.0);

ammount. displayBalance();  
ammount. withdraw (200.0);

savingAmm. displayBalance();

savingAmm. deposit Interest (1);

savingAmm. withdraw (1500.0);

### SUMMARY

Balanc in amm 12345 : £ 1000.0

£ 200 withdr ~~withdr~~ withdrawal.

Balanc in amm 12345 : £ 800.0

Balanc in amm 67890 : £ 2000.0

Interest of £ 100.0 display.

Balanc in amm 67890 : £ 2100.0

£ 1500 withdraw ~~withdr~~ withdrawal.

Balanc in amm 67890 : £ 600.0

OUTPUT:

```
C:\Users\arjun\.jdks\openjdk-21.0.2\bin\java.exe
Balance for account 12345: rs 1000.0
rs200.0 withdrawn successfully.
Balance for account 12345: rs 800.0
Balance for account 67890: rs 2000.0
Interest of rs100.0 deposited.
Balance for account 67890: rs 2100.0
rs1500.0 withdrawn successfully.
Balance for account 67890: rs 600.0
```



PROGRAM 6: WRITE A JAVA PROGRAM TO CREATE A PACKAGE CIE WHICH HAS 2 CLASSES STUDENT AND INTERNALS. THE CLASS STUDENT HAS MEMBERS LIKE USN, NAME, MARKS, SEM. THE CLASS INTERNAL HAS AN ARRAY THAT STORES THE INTERNAL MARKS SCORED IN 5 COURSES.

LAB-6

URBAN  
EDGE

⑨ Create a package CIE which has 2 classes Student and Internal. The class Personal has marks like package USN, Name, Sem. The class internal 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 a driver class of Student.

~~package~~ package CIE:

```
public class Student {  
    String USN;  
    String name;  
    int sem; }
```

~~public class Internal extends Student~~

```
public class Internal {  
    public int[] inter = new int[5]; }
```

~~package SEE;~~

```
public class External extends CIE {  
    public int[] exten = new int[5]; }]
```

import C10.Student;  
 import Scanner;  
 import java.util.\*;

public class Finalmark {

public static void main (String [] )

```
Scanner sc = new Scanner (System.in);
System.out.print ("Enter no. of students");
int n = sc.nextInt();
int finalmarks [] [] = new int [n] [5];
for (int i=0; i<n; i++)
```

System.out.print ("Student " + (i+1) + ":" );

System.out.print ("Enter marks in 5 columns :")

Internals (I = new Internals ())

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

I.ints [j] = sc.nextInt();

System.out.print ("Enter exam mark in I column :")

External E = new External ()

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

E.L.I = sc.nextInt();

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

{ finalmarks[i][j] = I.intake[i] + E.extra[j]; }

}

System.out.println ("Final marks : ")

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

{ System.out.print ("Student " + (i+1) + ": "); }

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

System.out.println (finalmarks[i][j] + " ");

}

System.out.println ();

)

OUTPUT:

150  
50  
212121

Enter no. of students

1

Student 1:

Enter intake marks for 5 courses

50

50

50

2/10/17

EDGE

Make a party for 4 math tip class

SD

SD

Enter external mark in 8 column

WD

84

56

55

45

Final marks :

Student 1:

90

95

100

100

95

183  
212/2

OUTPUT:

```
C:\Users\arjun\.jdks\openjdk-21.0.2\bin\java.exe
Enter the number of students: 2
Enter details for Student 1:
USN: 45545
Name: ARH
Semester: 2
Enter Internal Marks for 5 courses:
Course 1: 45
Course 2: 46
Course 3: 47
Course 4: 48
Course 5: 49
Enter External Marks for 5 courses:
Course 1: 49
Course 2: 48
Course 3: 44
Course 4: 40
Course 5: 39
```

```
Enter details for Student 2:  
USN: 54245  
Name: HAJ  
Semester: 2  
Enter Internal Marks for 5 courses:  
Course 1: 45  
Course 2: 40  
Course 3: 50  
Course 4: 60  
Course 5: 45  
Enter External Marks for 5 courses:  
Course 1: 30  
Course 2: 30  
Course 3: 3  
Course 4: 30  
Course 5: 30
```

**Final Marks of Students:**

**Student 1:**

USN: 45545

Name: ARH

Semester: 2

Total Marks:

Course 1: 94

Course 2: 94

Course 3: 91

Course 4: 88

Course 5: 88

**Student 2:**

USN: 54245

Name: HAJ

Semester: 2

Total Marks:

Course 1: 75

Course 2: 70

Course 3: 53

Course 4: 90

Course 5: 75

PROGRAM 7: WRITE A PROGRAM THAT DEMONSTRATES EXCEPTION HANDLING CREATE A BASE CLASS CALLED FATHER AND SAME BASE CLASS EXTENDS FATHER CLASS.

LAB - 7

Q) Write a program to demonstrate exception handling inheritance. Create a base called father and some base class extend father class. class WrongAgeException extends Exception {

if (age < 0) public WrongAgeException (String message) {  
super (message);  
}  
}  
class father {  
int d-age;  
public father (int a) throws WrongAgeException {  
if (a < 0) throw new WrongAgeException ("Age is less than zero!!");  
d-age = a;  
}  
}  
class son extends father {  
int s-age;  
public son (int d-age, int s) throws WrongAgeException {  
super (d-age);  
if (d-age < 0) throw new WrongAgeException ("father age can't be less than son!!");  
s-age = s;  
}  
}

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

System.out.println ("Hello World");

}  
 son s1 = new son (-1, 20);  
} catch (any exception e) {

System.out.println ("Exception : " + e.getMessage());  
System.out.println ("");

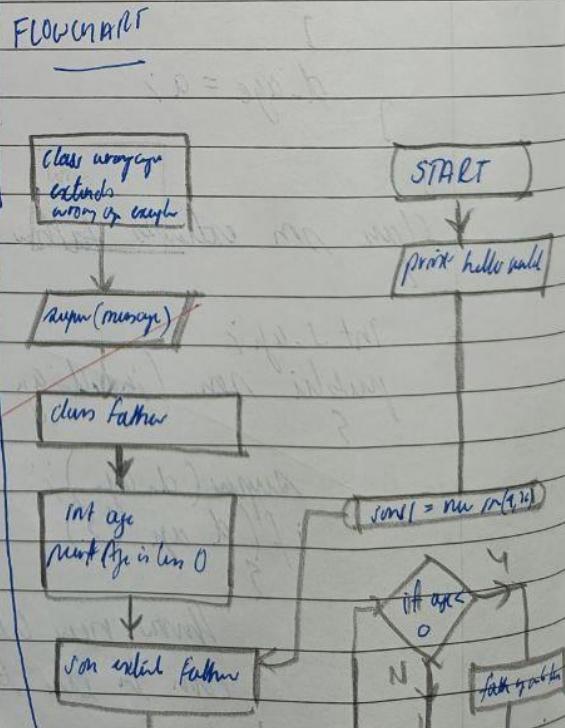
}

)

OUTPUT :

Hello World

Exception : age is less than 0



OUTPUT:

```
C:\Users\arjun\.jdks\openjdk-21.0.2\bin\java.exe
Father's age: 50
Son's age: 25
```

PROGRAM 8 : WRITE A JAVA PROGRAM WHICH CONTAINS TWO THREADS  
ONE THREAD PRINTS BMS COLLEGE OF ENGINEERING EVERY TEN SEC AND  
ONE THREAD PRINTS CSE EVERY TWO SECONDS.

Q B E |  
With a prgr which cont 2 thrd  
print "BMS college of Engg every 10 sec & print  
"CSE" every 2 seconds

```
class ps1 implements runnable {
    public void run() {
        for (int i=0; i<5; i++) {
            try {
                System.out.println (" BMS College of Engineering ");
                Thread.sleep (10000);
            } catch (Exception e) {
                e.printStackTrace ();
            }
        }
    }
}

class ps2 extends ps1 implements runnable {
    public void run () {
        System.out.println (" (CSE) ");
        try {
            Thread.sleep (2000);
        } catch (Exception e) {
            e.printStackTrace ();
        }
    }
}
```

public static void main (String args [] )

p51 p1 = new p50 ();  
p52 p2 = new p52 ();

Thread t1 = new Thread (p1);

Thread t2 = new Thread (p2);

t1.start ();

t2.start ();

### OUTPUT

CSE

BMS college of Engineering

CSE

CSE

CSE

BMS college of Engineering.

BMS college of Engineering.

BMS college of Engineering.

BMS college of Engineering.

## ALGORITHM

STEP 1: START

STEP 2: CREATE class ps1 implements Runnable

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

[ print "BMS college of Engineering" ]

[ Thread.sleep(200) ]

catch exception e and print message.

STEP 3: create class ps2 implements runnable

print (CSE)

Thread.sleep(2000);

catch exception e;

STEP 4 : create object p1, p2

STEP 5: Create thread t1, t2 and pass the objects p1, p2  
respectively

STEP 5: start t1, t2;

STEP 6: STOP

(20)  
16/12/15

OUTPUT:

```
C:\Users\arjun\.jdks\openjdk-21.0.2\bin\java.exe
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
```

PROGRAM 9: WRITE A JAVA PROGRAM THAT CREATES A USER INTERFACE TO PERFORM INTEGER DIVISION. THE USER ENTERS TWO NUMBERS IN THE TEXT FIELD NUM1,NUM2.

(09)

Write a program that creates a user interface to perform integer division. The user enters 2 no. in the text field, num1, num2. division of num1, num2 is displayed in the Result field when the divide button is clicked. If num1 or num2 are not an int, the program will throw a NumberFormatExeption. If num2 is zero, the program will throw an arithmetic exception. Display the 'exception' in a message dialog box.

```
import javax.swing.*;  
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);

see  
JFrame

Font  
label

JLabel jlab = new JLabel("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();

6  
1 a  
sun/1

```
jfm.add(ev);  
jfm.add(jlab);  
jfm.add(ajtf);  
jfm.add(bjtf);  
jfm.add(button);  
jfm.add(alab);  
jfm.add(blah);  
jfm.add(anslab);
```

ActionListener 1 = new ActionListener () {

public void actionPerformed (ActionEvent evt) {

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

};

ajtf.addActionListener (1);

bjtf.addActionListener (1);

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;

(~~try~~)

alab.setText ("\nA = " + a);

blah.setText ("\nB = " + b);

anslab.setText ("\nAns = " + ans);

```
    catch (NumberFormatException e) {
```

```
        alab.setText("");
```

```
        blab.setText("");
```

```
        anslab.setText("");
```

```
        err.setText("Enter only integers!");
```

```
} catch (ArithmaticException e) {
```

```
    alab.setText("");
```

```
    blab.setText("");
```

```
    anslab.setText("");
```

```
    err.setText("0 should be Non zero!");
```

```
}  
});
```

```
} ifrm.setVisible(true);
```

~~```
public static void main (String args[])
```~~~~```
    DemoUtilities.invokeLater (new Runnable () {
```~~~~```
        public void run () {
```~~~~```
            new Demo ();
```~~~~```
} );
```~~~~```
}
```~~

OUTPUT:

Divide App -  X

Enter the divide and dividend

Calculate

A = 10   B = 5   Ans = 2

//

Action Listener:-

AL for textfield ; Print a message to the control when text field  
is interacted with,

ActionListener for button;

Attempts to pass that text from the text fields as int type (a and b)  
if successful, performs division ( $a/b$ ) and displays the results

// Visualizing the Application :-

When you run the code, a simple GUI window appears with  
A label prompting for input.

Labels to display the entered values and result.

An error label for displaying error messages.

Key points:

- The code showcases basic swing component, event handing, exception handing.
- Swing safety is consider a prop GUI creation and interaction.

8  
27/2/2015

OUTPUT:

