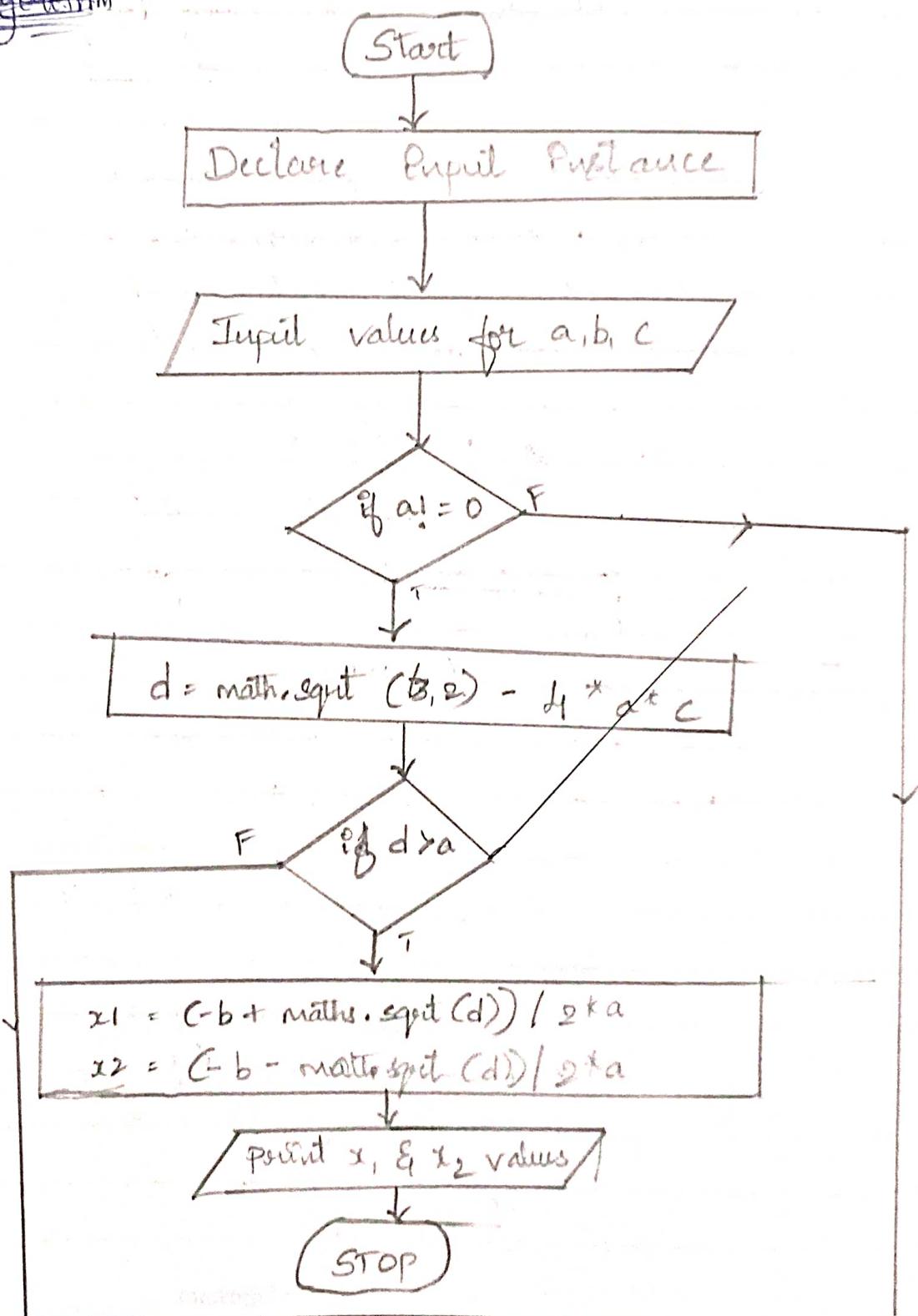
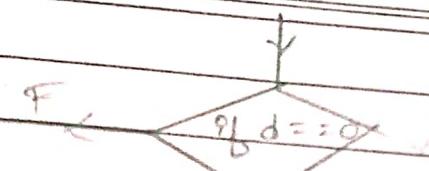


7) Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c=0$.

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.

Flowchart
Algorithm:





$|x = -b/(c_2 + a)$

Point x value

(STOP)

if $d \neq 0$

T

Print "no real solutions"

(STOP)

Print "invalid value for a"

(STOP)

Algorithm

- 1 → Start
- 2 → Define Repeat Block
- 3 → Input values for variables a, b, c.
- 4 → Check if $a \neq 0$, if true goto step 5 else goto step 15
- 5 → $d = \text{math.sqrt}(b^2 - 4ac)$
- 6 → Check if $d > 0$, if true, goto step 7 else goto step 9
- 7 → $x_1 = (-b + \text{math.sqrt}(d)) / 2 * a$
 $x_2 = (-b - \text{math.sqrt}(d)) / 2 * a$
- 8 → Print x_1 and x_2 values, and goto step 15
- 9 → Check if $d = 0$, if true goto step 10 else goto step 12
- 10 → $x = -b / (2 * a)$
- 11 → Print value of x goto step 15
- 12 → Check if $d < 0$, if true goto step 13 else goto step 14
- 13 → Print "no real solutions" then goto step 15
- 14 → Print "Invalid Input for a"
- 15 → Stop.

Java Code:-

```
Import java.util.Scanner;
import static java.lang.Math.sqrt
import static java.lang.math.
```

```
Public class New{
```

```
    public static void main()
        float a, b, c;
```

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

```
    System.out.println ("Enter Co-efficient of
quadratic equation");
```

```
    int a = n.nextInt();
```

```
    int b = n.nextInt();
```

```
    int c = n.nextInt();
```

```
    if (a==0) {
```

```
        System.out.println ("Enter a valid value a");
```

```
    }
```

```
    else {
```

```
        float d = b*b - 4*a*c;
```

```
        if (d>0) {
```

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

```
            float r1 = (float) (-b + math.sqrt (d)) / (2*a);
```

```
            float r2 = (float) (-b - math.sqrt (d)) / (2*a);
```

```
            System.out.println (r1);
```

```
            System.out.println (r2);
```

Teacher's Signature _____

}
else if (d < 0)

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

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

float r2 = (float) sqrt (ab (d)) / (r1 * a);

System.out.println (r1 + " + i " + r2);

System.out.println (r1 + " - i " + r2);

}

else {

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

float r = (float) - b / (a * d);

System.out.println (r);

$\sqrt{ab}/2$

D:\Javalab>java Demo

Name :C Mohan

USN:2023BMS02523

23

43

- 23

Roots are real and distinct

Root 1: 0.43489230360271006

Root 2: -2.3036575299940144

D:\Javalab>java Demo

Name :C Mohan

USN:2023BMS02523

23

45

43

No real solutions exist

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

→ Import java.util.Scanner;
class Student {

```
private String name;
private String usn;
private int [ ] credits;
private int [ ] marks;
```

public Student ()

```
credits = new int [5];
marks = new int [5];
```

y

public void acceptDetails () {

Scanner s1 = new Scanner (System.in);

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

usn = s1.next();

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

name = s1.next();

System.out.println ("Enter credits and marks
for 5 subjects: ");

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

System.out.println ("Subject " + (i+1) + " credit");

credits [i] = s1.nextInt();

```
System.out.println("Subject" + (i+1) + "Marks:");  
marks[i] = sc.nextInt();
```

y

y

```
public void displayDetails() {  
    System.out.println("In Student Details:");  
    System.out.println("In Usn :" + usn);  
    System.out.println("Name :" + name);  
    System.out.println("Credits and Marks:");  
  
    for (int i=0; i<5; i++) {  
        System.out.println("Subject" + (i+1) + ":"  
        + credits[i] + ", Marks -" + marks[i]);  
    }  
}
```

y

y

```
public double SGPA() {
```

double totalCredits = 0;

double weightSum = 0;

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

totalCredits += credits[i];

totalSum += gradePoint(marks[i]) *
credits

return totalSum / totalCredits;

y

```

public float GradePoint (float marks) {
    if (marks == 90) {
        return 10;
    }
    else if (marks >= 80) {
        return 9;
    }
    else if (marks >= 70) {
        return 8;
    }
    else if (marks >= 60) {
        return 7;
    }
    else if (marks >= 50) {
        return 6;
    }
    else {
        return 0;
    }
}

```

```

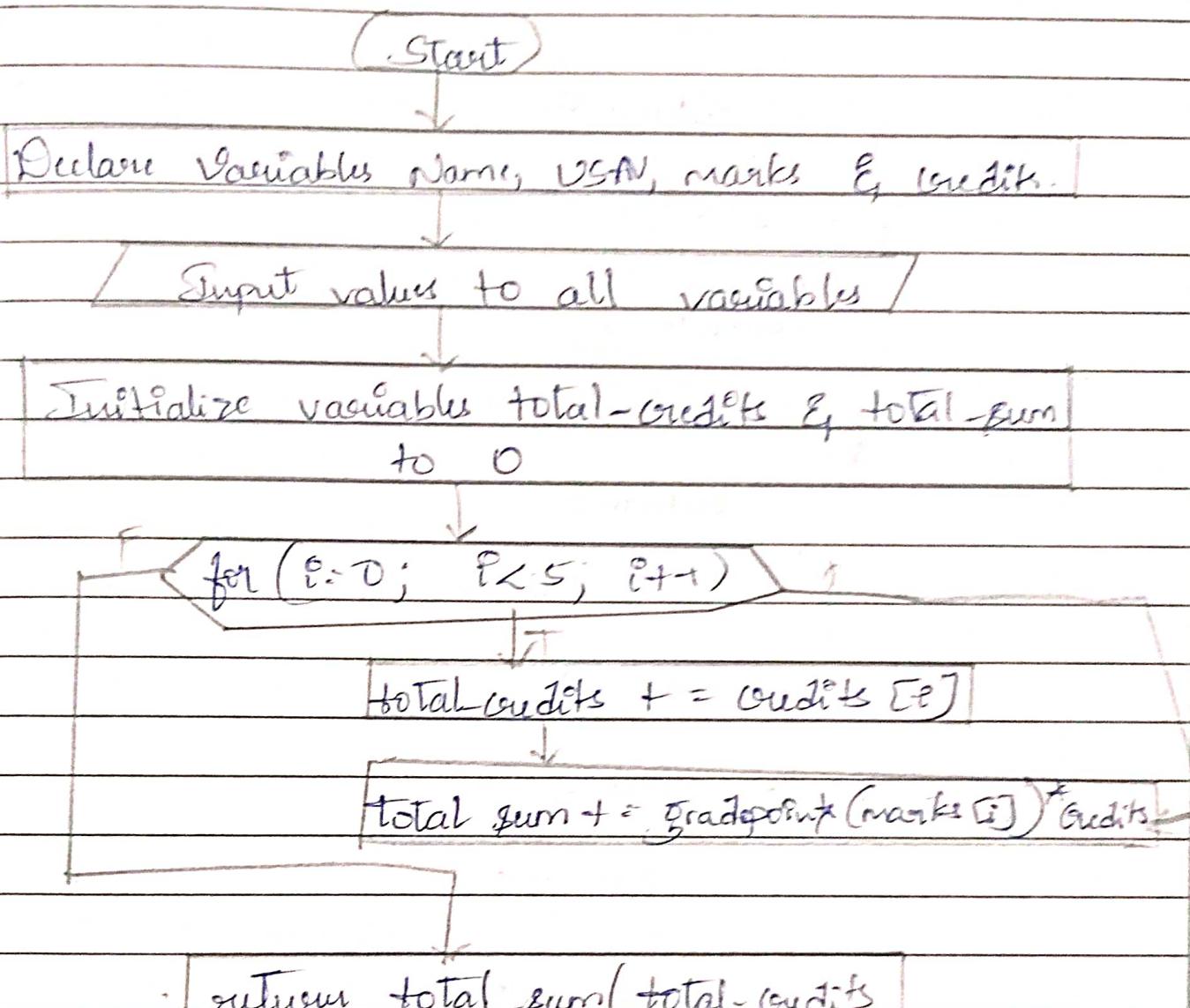
public class Main {
    public static void main (String [] args) {
        Student student = new Student ();
        student . details ();
        System.out.println ("In Student Details");
        student . display ();
        System.out.println ("In SGPA : " +
            student . SGPA ());
    }
}

```

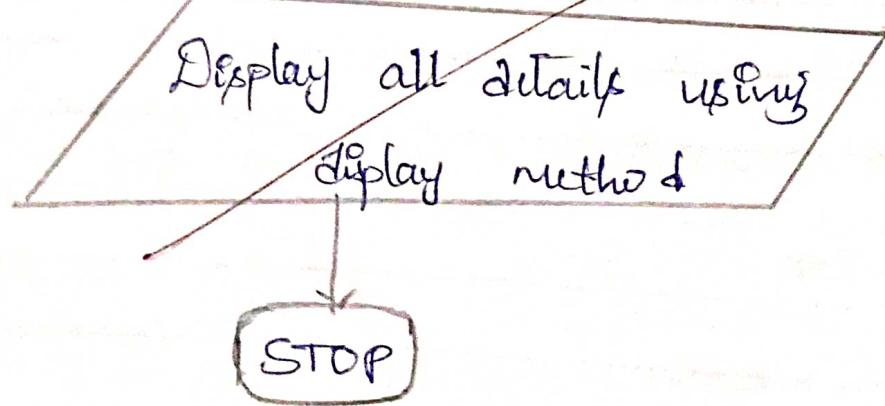
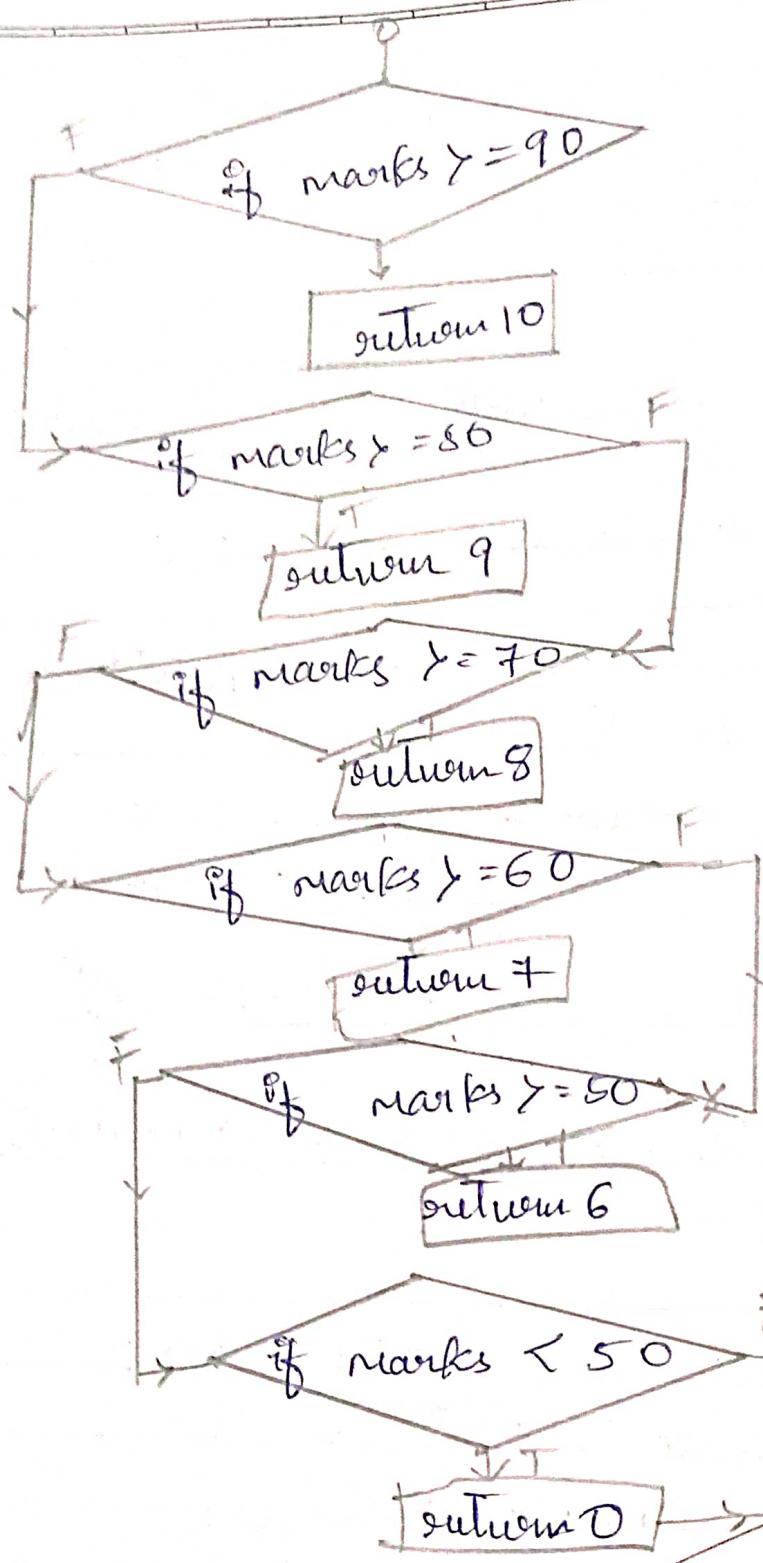
Algorithm:-

- 1) Start
- 2) Declare variables USN, name, credits, marks;
- 3) Input values for USN, name, credits, marks using a method
- 4) Create another method to calculate SGPA
- 5) In SGPA method, Initialize variables, total-credits and total-sum to 0;
- 6) For loop condition ($i = 0; i < 5; i++$)
 - total-credits += credits[i]
 - total-sum += gradepoint(marks[i])
- 7) Return value for SGPA function as total_sum / total-credits
- 8) Create gradepoint method & check for marks collection.
- 9)
 - if marks $\geq 90 \rightarrow$ return 10
 - if marks $\geq 80 \rightarrow$ return 9
 - if marks $\geq 70 \rightarrow$ return 8
 - if marks $\geq 60 \rightarrow$ return 7
 - if marks $\geq 50 \rightarrow$ return 6
 - ~~if marks $\geq 40 \rightarrow$~~
 - ~~else \rightarrow fail \rightarrow return 0~~
- 10) In the main function call the function details to input details & display function to display it.

Flowchart:-



~~Declare function gradepoint~~



C:\Users\DELL>java Main

Name :C Mohan

USN:2023BMS02523

Enter USN: 23

Enter Name: Mohan

Enter credits and marks for 5 subjects:

Subject 1 Credits: 3

Subject 1 Marks: 100

Subject 2 Credits: 4

Subject 2 Marks: 100

Subject 3 Credits: 4

Subject 3 Marks: 100

Subject 4 Credits: 3

Subject 4 Marks: 100

Subject 5 Credits: 2

Subject 5 Marks: 100

Student Details:

USN: 23

Name: Mohan

Credits and Marks:

Subject 1: Credits - 3, Marks - 100

Subject 2: Credits - 4, Marks - 100

Subject 3: Credits - 4, Marks - 100

Subject 4: Credits - 3, Marks - 100

Subject 5: Credits - 2, Marks - 100

SGPA: 10.0

Lab Program

Create a class Book which contains four members: name, author, price, num-page. Include a constructor to set the values for the members. Include methods to set & get the details of the objects. Include a toString() method that could display the details of the book. Develop a Java program to create a book object.

→ Import java.util.*;
class Book {

private String name;

private String author;

private int price;

private int num-page;

public Book (String name, String author, int price,
int num-page) {

this.name = name;

this.author = author;

this.price = price;

this.num-page = num-page;

}

public void setName(String name) {

this.name = name;

y

public ~~String~~ ^{void} getName(String author) {

~~this.author = author;~~ return name;

y

```
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 setNumPages (int numPage) {
    this.numPage = numPage;
}
```

```
public int getNumPages () {
    return numPage;
}
```

```
public String toString () {
    return "name of Book is " + name + " auth
    is: " + author + " the cost of book is " + price +
    " the total no of pages in book is " + numPage;
}
```

~~no class defined~~

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

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

```
int n = Input.nextInt ();
```

```
Book b = new Book (n);
```

```
for (int i = 0; i < n, i++) {  
    System.out.println ("Enter details for book " + (i+1));  
    System.out.print (" ");
```

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

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

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

```
String author = input.nextLine();
```

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

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

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

```
int numPage = input.nextInt();
```

```
b[i] = new Book (name, author, price, numPage);
```

y

```
System.out.println ("The book details are: ");
```

```
for (Book book : b) {
```

```
    System.out.println (book.toString());
```

y

```
input.close();
```

y

dp:

Enter details for book: 1

Name: Java

Author: Mohan

Price: 50000

num of pages: 3

The book details are: name of Book is Java author is
Mohan the cost of Book is 50000 the no of pages is 3.

```
C:\Users\DELL>java Demo1
Name :C Mohan
USN: 2023BMS02523
1
Enter details for book 1:
Name:
java
Author :
Mohan
price is:
50000
number of pages in book is:
3
the book details are:
Name of the book is java
the author of the book is Mohan
the price of the book is 50000
the total number of the pages of the book is3
```

→ Develop a Java Program to create an abstract class named Shape that contains two Integers & an empty method named Rectangle, Triangle & 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.

Code: abstract class shape {

 int dimension1;

 int dimension2;

 void printArea() {

y

y

class Rectangle extends shape {

 public Rectangle (int length, int width) {

 this.dimension1 = length;

 this.dimension2 = width;

y

 void printArea() {

 int area = dimension1 * dimension2;

 System.out.println ("Rectangle Area: " + area);

y

y

class Triangle extends shape {

public Triangle (int base, int height) {

this.dimension1 = base;

this.dimension2 = height;

}

void printArea () {

double area = 0.5 * dimension1 *

dimension2;

System.out.println ("Triangle area: " + area);

}

y

class Circle extends shape {

public Circle (int radius) {

this.dimension1 = radius;

this.dimension2 = 0;

void printArea () {

double area = Math.PI * dimension1

* dimension2;

System.out.println ("Circle area: " + area);

class Program_shape &

```
public static void main(String [ ] args) {
    Rectangle Rec = new Rectangle (5, 10);
    Triangle Tri = new Triangle (4, 7);
    Circle Cc = new Circle (2);
```

Rec. printArea();

Tri. printArea();

Cc. printArea();

}

}

Algorithm:-

1] Start

2] Create abstract class shape & create variables dimension1 & dimension2 & a function printArea

3] Create classes rectangle, triangle & circle that extends to class shape.

4] Specify different printArea method in each to override the existing printArea method.

5] set the area in triangle class as

~~area = 0.5 * dimension1 * dimension2;~~

and area in rectangle class as

~~area = dimension1 * dimension2~~

and area in circle class as

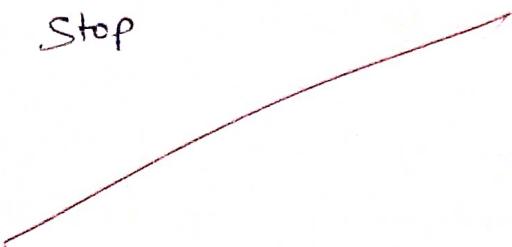
~~area = math.PI * dimension1 * dimension2;~~

Teacher's Signature

6) In main method, create constructor and call the
printArea() method to print the values of the area of
rectangle, circle and Triangle classes

7) ~~End~~

7) Stop



```
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>javac E.java

C:\Users\Admin>java E
Name : Mohan
USN is: 2023BMS02523
Enter length and breadth of a rectangle:
34
43
Area of rectangle: 1462
Enter base and height of a triangle:
23
43
Area of the triangle: 494.5
Enter the radius of a circle:
43
Area of Circle: 5805.860000000001

C:\Users\Admin>
```

```
Lab 5:- import java.util.Scanner;  
class Account {  
    String customerName;  
    int accountNumber;  
    String accountType;  
    double balance;  
    public Account (String customerName, int accountNumber,  
                    String accountType, double balance) {  
        this.customerName = customerName;  
        this.accountNumber = accountNumber;  
        this.accountType = accountType;  
        this.balance = balance;  
    }  
    public void displayBalance() {  
        System.out.println ("Balance for Account Number  
accountNumber + ": " + balance);  
    }  
    public void displayDeposit (double amount) {  
        balance += amount;  
        System.out.println ("Deposit of " +  
amount + " successful");  
    }  
    public void withdrawal (double amount) {  
        if (balance >= amount) {  
            balance -= amount;  
            System.out.println ("Withdrawal of " +  
amount + " successful");  
        }  
    }  
}
```

y
else d

System.out.println ("Insufficient funds for withdrawal");

y
y
y

class Current extends Account {
double minBalance;
double serviceCharge;

public Current (String customerName, int accountNumber,
double balance) {

super (customerName, accountNumber, "Current", balance);
this. minBalance = 1000;
this. serviceCharge = 50;

y

@Override

public void withdrawal (double amount) {
if (balance - amount >= minBalance) {
super.withdrawal (amount);

y

else t

System.out.println ("Insufficient funds

to maintain minimum balance. Service charge of " +
serviceCharge + " imposed.");

balance -= serviceCharge;

y y y

```
class SavAcct extends Account {  
    double interestRate;  
  
    public SavAcct (String customerName, int accountNumber,  
        "Savings", balance);  
        this.interestRate = 0.05;  
    }  
  
    public void computeInterest () {  
        double interest = balance * interestRate;  
        deposit (interest);  
        System.out.println ("Interest of " + interest + "  
        computed and deposited.");  
    }  
  
public class Bank {  
    public static void main (String [] args) {  
        Scanner sc = new Scanner (System.in);  
        CurrentAccount currentAccount = new CurrentAccount ("John  
        Doe", 12345, 1500);  
  
        currentAccount.currentAccount. displayBalance ();  
        currentAccount.withdrawal (2000);  
        currentAccount.displayBalance ();  
  
        SavingsAccount savingsAccount = new SavAcct ("Jane  
        Doe", 67890, 5000);
```

savingsAccount. displayBalance();
 savingsAccount. computeInterest();
 savingsAccount. displayBalance();
 scanner.close();

y

y

~~O/p:~~

→ Balance for Account no. 12345: \$1500.0

Sufficient funds to maintain minimum balance. Service charge of \$50.0 imposed.

Balance for Account no. 12345: \$1450.0

Balance for Account no. 67896: \$5000.0

Deposit of \$250.0 successful

Interest of \$250. computed & deposited

Balance for Account no. 67890: \$5250.0

Algorithm:-

- > Start
- Create class Account and create variables customerName, accountNumber, accountType, balance & constructor & a function displayBalance, deposit, withdrawal.
- Create class CurrentAcct, SavAcct & extends it to class Account.
 - Include additional instance variables for minimum balance and service charge.
 - Override the withdrawal method to check for minimum balance & impose a service charge if necessary.
- Create a class Bank & create instances of current & savAcct.
- Demonstrate the functionality by performing actions such as displaying balances, depositing, withdrawal & computing interest.
- > Stop

Flowchart**Start**

Declare variables customerName, accountNumber,
accountType, balance & define display method

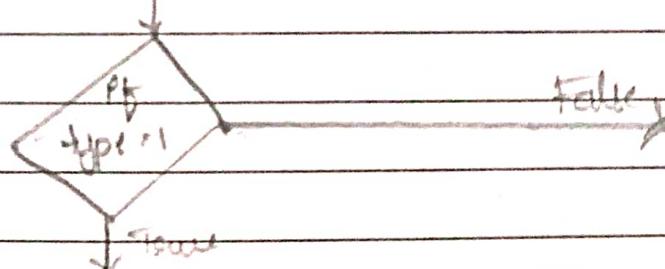
Define class savings which inherits class
Account

Define method withdraw

Define class current which inherits class Account

Define method deposit Interest

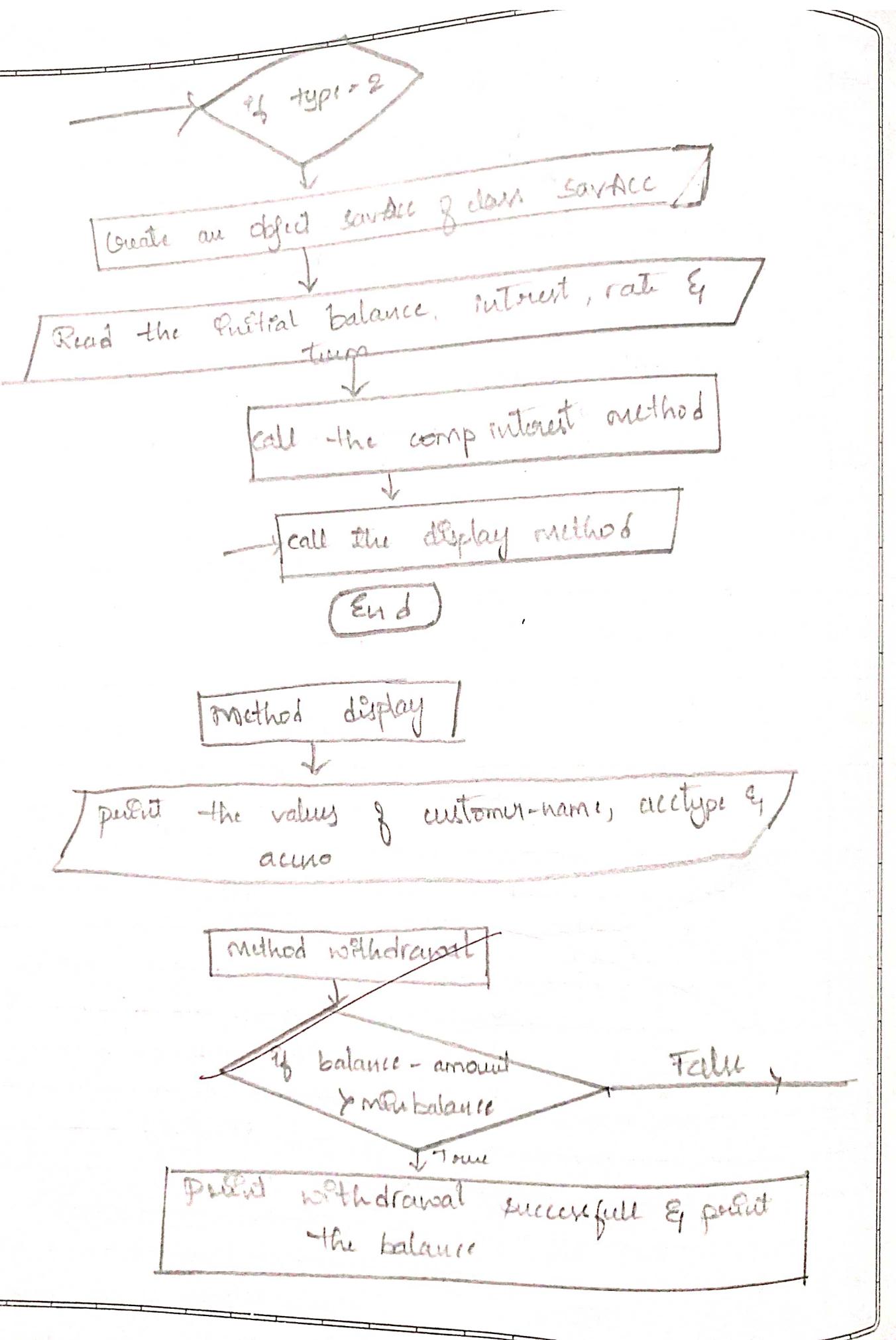
Read the type & acc /



Create an object (instance of class)

Read the initial & withdrawal amt /

call the withdrawal method



Expt. No.:



Print sufficient funds

method PINLIST

computation, fulfillment &
math, socio & extrastatic items

print the compound interest & balance

STOP

100

19/12

```
D:\Javalab>java Bank
Name: C Mohan
USN:2023BMS02523
Balance for Account Number 12345: $1500.0
Insufficient funds to maintain minimum balance. Service charge of $50.0 imposed.
Balance for Account Number 12345: $1450.0
Balance for Account Number 67890: $5000.0
Deposit of $250.0 successful.
Interest of $250.0 computed and deposited.
Balance for Account Number 67890: $5250.0
```

Q:- Create a program package CIE which has two classes Student & Testmarks. The class personal has members like USN, name, Sem. - the class Testmarks 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.

→ Packages :-

```
Package CIE;  
Import java.util *;  
public class student {
```

```
    public int Sem;  
    int USN;  
    String name;
```

```
    public void accept () {  
        Scanner sc = new Scanner (System.in);  
        System.out.println ("Enter USN, name, Sem");  
        USN = sc.nextInt();  
        name = sc.nextLine();  
        Sem = sc.nextInt();  
    }
```

6

```
public class Tutorials {
```

```
    public int summares [] = new int [5];
```

y

```
Package SEE;
```

```
import CIE.student;
```

```
public class External Student {
```

```
    public int ex-marks [] = new int [5];
```

y

```
import java.util.*;
```

```
import SEE;
```

```
import CIE;
```

```
public class FinalMarks {
```

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

```
        int marks [] = new int [5];
```

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

```
        System.out.println ("Enter no of student");
```

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

```
        SEE.External st [] = new SEE.External [n];
```

```
        CIE.Tutorial s [] = new CIE.Tutorial [n];
```

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

```
            st [i] = new SEE.External ();
```

```
            s [i] = new CIE.Tutorial ();
```

```
            System.out.println ("Enter details " + (i+1))
```

```
            ~ s [i].accept ();
```

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

```
                System.out.println ("Enter details " + (i+1))
```

```
                s [i].accept ();
```

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

System.out.println(" Enter internal and
External marks of subject "+ (i+1));

$s[i]$ in-marks[j] = sc.nextInt();

$st[i]$ ex-marks[j] = sc.nextInt();

finals[j] = $s[i]$ in-marks[j] + $st[i]$ ex-marks[j];

y

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

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

System.out.println(" " + " " + (k+1) + " " +
+ "Finals " + s[k]);

y

y

Algorithm :-

- 1) Start
- Create a package (IE)
 - Create public class Student
 - Initialize public variables, user, name, sem
 - Create void method accept to read user, name, sem
 - Create public class External.
 - Initialize public Integer array End [] of 5 elements
 - Create package SEE
 - Create public class External extending class Student from package (IE)
 - Initialize public Integer array sum [] of 5 elements
 - Import packages (IE, SEE)
 - Initialize array fin [] of 5 elements.
 - Read no of students
 - Create array st [] from SEE.External [n]
 - Create array sc [] from (IE.External [n])
 - By using for loop accept the information of student & using another for loop add st[] + sc[] to obtain final marks
 - Display the marks of student
 - Stop

88
2/2/24.

```
D:\JAVA LAB PROGRAM LISTS\lab 6>java FinalMarks
Name:Mohan
USN:2023BMS02523
Enter n:
1
Enter details 1
Enter USN, Name, Sem:
2023BMS02523
Mohan
3
Enter internal and external of sub 1
42
53
Enter internal and external of sub 2
32
45
Enter internal and external of sub 3
43
45
Enter internal and external of sub 4
43
32
Enter internal and external of sub 5
23
12
Final marks of Mohan
Course 1 = 95
Course 2 = 77
Course 3 = 88
Course 4 = 75
Course 5 = 35

D:\JAVA LAB PROGRAM LISTS\lab 6>
```

→ Write a program - that demonstrates handling of exceptions. For instance take Create a base class called "Father" & derived class called "son" which extends the base class. In Father class, implement a constructor which takes the age & throws the exception WrongAge() when the input age < 0. In son class, implement a constructor - that calls both father & son's age & throws an exception if son's age is >= father's age.

→ class WrongAge extends Exception {
public WrongAge (String message) {
super(message);
}

class Father {

int age;

public Father(int age) throws WrongAge {
if (age < 0) {

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

this.age = age;

y

class Son extends Father {
 int sonAge;
 public Son(int fatherAge, int sonAge) throws
 WrongAge {
 super(fatherAge);
 if (sonAge >= fatherAge) {
 throw new WrongAge("Son's age should be less
 than father's age");
 }
 this.sonAge = sonAge;
 }
}

class Main2

public static void main (String args[]) {
 try {
 Father father = new Father (-5);
 }
 catch (WrongAge e) {
 System.out.println ("Exception Caught: "+
 e.getMessage());
 }
 try {
 Father father = new Father (10);
 Son son = new Son (father.age, 45);
 }
 catch (WrongAge e) {
 System.out.println ("Exception Caught: "+
 e.getMessage());
 }
}

try {

Father father = new Father(40);

Son son = new Son(father.age, 20);

y

catch (WrongAge e) {

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

y

y

Algorithm:-

- Start
- Create class WrongAge and extend with Exceptions
- Create a constructor for WrongAge and pass parameters string message & write super(message).
- Create a Super class called Father & declare the int age & create a constructor for Father class
- The constructor throws an WrongAge E, if (age<0) then throw new error Wrong age.
- And Initialize it age = age.
- Create a class sub class called Son & declare the int sonAge. & extend class to Father class.
- Create a constructor & pass Two parameters int Father age, Son age.

- Throw the WrongAge & throw an new WrongAge error for Son class.
- Create a class called Main2 & declare public static void main().
In Test file create an object and the value.
- In catch, catch the WrongAge exception error & run program.
- Stop

Op:-

Father age is Wrong Age cannot be negative
given age is wrong Son's age should be less
than father's age.

```
E:\JAVA>javac Main2.java  
E:\JAVA>java Main2  
Name :Mohan  
USN :2023BMS02523  
Father age is wrong Age cannot be negative  
given age is wrong Son's age should be less than father's age  
E:\JAVA>
```

8) Write a program which creates two threads, one thread displaying "IEMS College of Engineering" once every ten seconds & another displaying "CSE" once every two seconds.

→ class Display extends Thread {

private String message;
private int interval;

public Display (String message, int interval)
this.message = message;
this.interval = interval;

y

public void run() {

try {

while (true) {

System.out.println(message);

Thread.sleep (interval * 1000);

y

catch (InterruptedException e) {

e.printStackTrace();

y

y

~~public class Main {~~

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

```
Display thread1 = new Display("BMS college of  
Engineering", 10);  
Display thread2 = new Display("CSE", 2);  
    -thread1.start();  
    thread2.start();  
}  
y
```

Algorithm:-

- Start
- Create a class Display & extends it with thread.
- Declare 2 variable one was message & interval.
- Create a constructor display and pass parameter.
- Create a method run & In run throw try & catch. Use thread.sleep and pass 1000 milliseconds in the sleep method.
- In catch pass the InterruptedException.
- Create class called Main
- Create 2 objects & pass values.
- Stop.

O/I - BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

~~SCQ
16/12/24~~

Q) Write a program that creates a user interface to perform integer divisions. The user enters two int 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 ArithmeticException. Display the exception in a message dialog box.

Code →

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

class SwingDemo {
    SwingDemo() {
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JLabel glab = new JLabel("Enter the divisor and dividend:");
        JTextField aJTF = new JTextField(8);
        JTextField bJTF = new JTextField(8);
```

```
JButton button = new JButton("Calculate");
```

```
JLabel com = new JLabel();
```

```
JLabel alab = new JLabel();
```

```
JLabel blab = new JLabel();
```

```
JLabel blabandalab = new JLabel();
```

```
frame.add(com);
```

```
frame.add(alab);
```

```
frame.add(cjtf);
```

```
frame.add(bjtf);
```

```
frame.add(button);
```

```
frame.add(alab);
```

```
frame.add(blab);
```

```
frame.add(alab);
```

```
ActionListener l = new ActionListener() {
```

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

System.out.println ("Action event from a

text field");

}

};

```
cjtf.addActionListener(l);
```

```
bjtf.addActionListener(l);
```

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

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

try {

 put a = Integer.parseInt(aJTF.getText());

 put b = Integer.parseInt(bJTF.getText());

 put ans = a/b;

 alab.setText("In A = " + a);

 blab.setText("In B = " + b);

 anslab.setText("In Ans = " + ans);

}

catch (NumberFormatException e) {

 alab.setText(" ");

 blab.setText(" ");

 anslab.setText(" ");

 err.setText("Enter Only Integers!");

}

catch (ArithmaticException e) {

 alab.setText(" ");

 blab.setText(" ");

 anslab.setText(" ");

 err.setText("B Should be Non zero!");

y

y

y);

ifrm.setVisible(true);

y

public static void main (String args[]) {

 SwingUtilities.invokeLater (new Runnable()) {

 public void run() {

 new SwingDemo();

y); y;

y)

Q/P:- ~~Enter in.~~

Divide app

Enter the divisor & dividend:

2000 43

~~Calculate~~

$d = 2000 \quad B = 43 \quad Ans = 46$

$$\begin{array}{r} 46 \\ 23 \overline{) 200 } \\ 18 \\ \hline 20 \\ 18 \\ \hline 2 \end{array}$$

Teacher's Signature _____

```
E:\JAVA>cd java  
The system cannot find the path specified.  
E:\JAVA>javac SwingDemo.java  
E:\JAVA>java SwingDemo  
Name :Mohan  
USN :2023BMS02523
```

