

B.M.S. COLLEGE OF ENGINEERING
Basavanagudi, Bengaluru- 560019
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



LAB REPORT

On

Object Oriented Java Programming
(23CS3PCOOJ)

Submitted By:

ABHISHEK YADAV
1BM22CS009

In partial fulfilment of
BACHELOR OF ENGINEERING
In
COMPUTER SCIENCE AND ENGINEERING
2023-24

Faculty-In-Charge
Swathi Sridharan
Assistant Professor
Department of Computer Science and Engineering

INDEX

| Sl-No | Title Name | Date |
|--------------|-------------------|-------------|
| 1 | Lab Program 1 | 22-12-2023 |
| 2 | Lab Program 2 | 29-12-2023 |
| 3 | Lab Program 3 | 12-01-2024 |
| 4 | Lab Program 4 | 12-01-2024 |
| 5 | Lab Program 5 | 19-01-2024 |
| 6 | Lab Program 6 | 02-02-2024 |
| 7 | Lab Program 7 | 16-02-2024 |
| 8 | Lab Program 8 | 16-02-2024 |
| 9 | Lab Program 9 | 23-02-2024 |

Date - 22/12/23

Q1 Develop a Java program that prints all real solutions to the quadratic equation $ax^2 + bx + c$.
Read 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.

```
Ans. 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 quadratic");  
        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 nonzero value for a");  
            Scanner s = new Scanner(System.in);  
            a = s.nextInt();  
        }  
        d = b * b - 4 * a * c;  
        if (d == 0) {
```

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

```
System.out.println("Roots are real and equal");  
System.out.println("Root1 = Root2 = " + r1);  
}
```

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

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

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

```
System.out.println("Roots are real and distinct");  
System.out.println("Root1 = " + r1 + " Root2 = " + r2);  
}
```

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

```
System.out.println("There are no real solution");  
}
```

```
class QuadraticMain {
```

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

```
Quadratic q = new Quadratic();
```

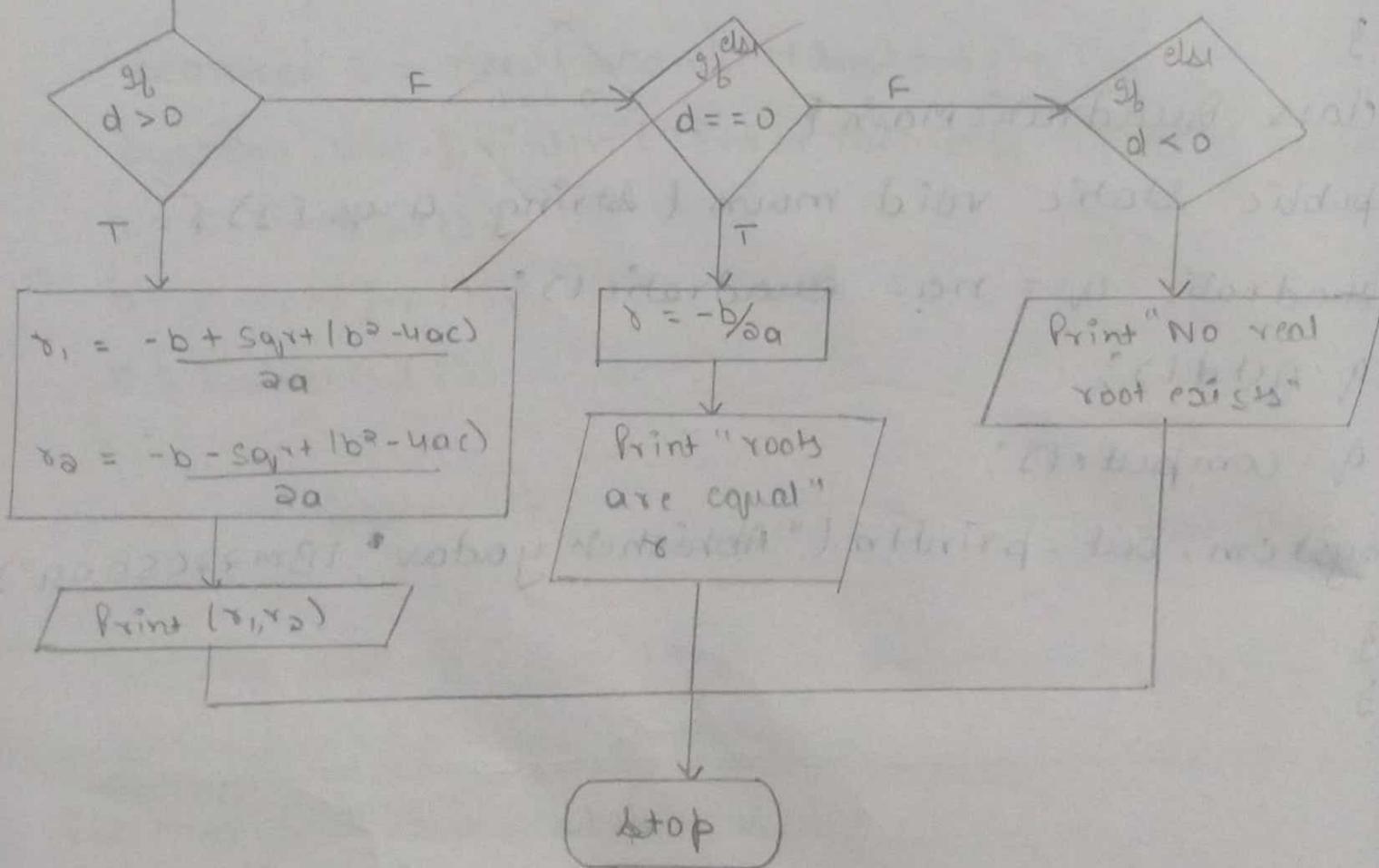
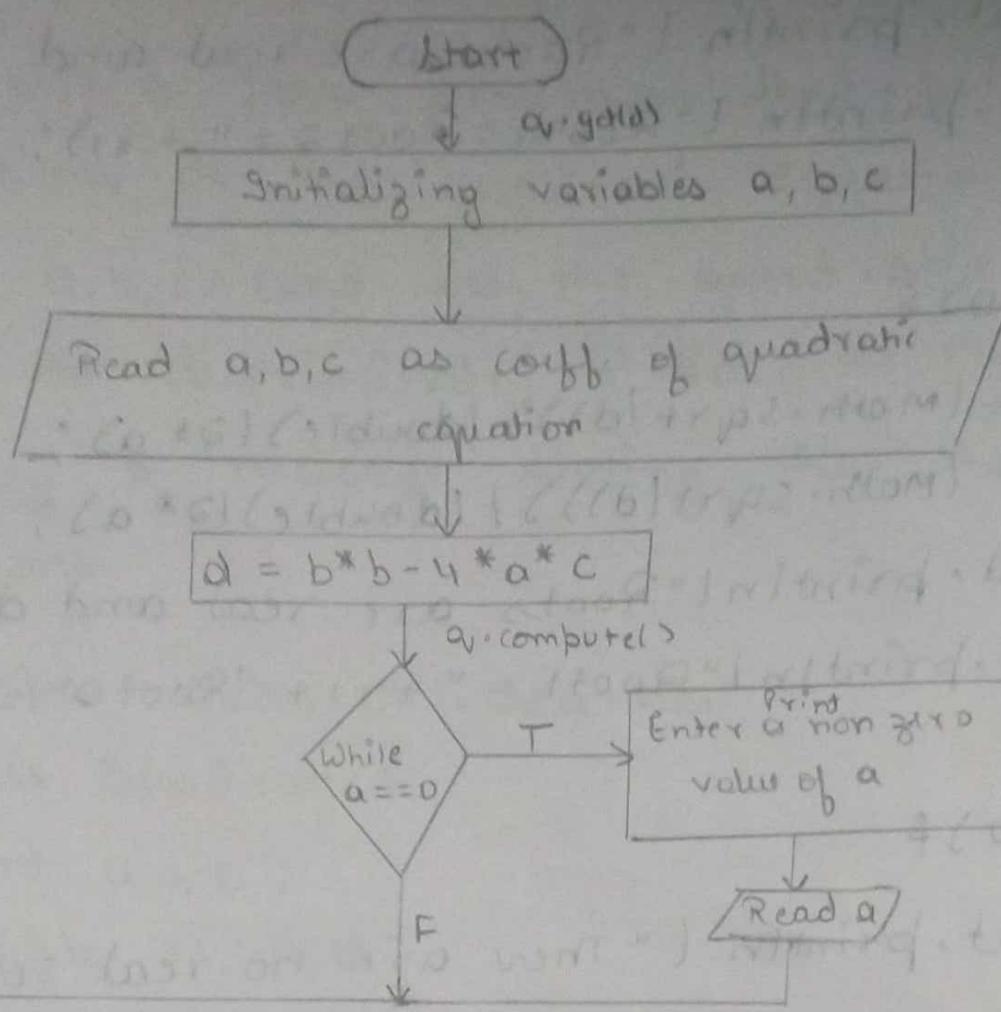
```
q.getd();
```

```
q.comput();
```

```
System.out.println("Abhishek Yadav • 13M22CS009")  
}
```

```
}
```

flowchart



```
C:\Users\Abhishek Yadav\OneDrive\Desktop\Java>javac QuadEqn.java
```

```
C:\Users\Abhishek Yadav\OneDrive\Desktop\Java>java QuadEqn
```

```
Enter the co-efficient of x^2:
```

```
2
```

```
Enter the co-efficient of x:
```

```
5
```

```
Enter the value of the constant: 3
```

```
The roots are:-1.0 and -1.5
```

```
Abhishek Yadav 1BM22CS009
```

```
C:\Users\Abhishek Yadav\OneDrive\Desktop\Java>
```

Date - 29/12/23

Q. Develop a Java program to create a class Student with members USN, name and array of credits and marks.

Include methods to accept, display details of student and a method to calculate SGPA of student

Sol. Java.

```
import 'util. Scanner';  
class Student {  
    String USN;  
    String name;  
    int [] credits = new int [8];  
    int [] marks = new int [8];  
  
    public void acceptDetails () {  
        Scanner scanner = new Scanner (System.in);  
        System.out.print ("Enter USN: ");  
        USN = scanner.nextLine();  
        System.out.print ("Enter Name: ");  
        name = scanner.nextLine();  
        System.out.println ("Enter details for subject: \n");  
        for (int i=0; i < credits.length; i++) {  
            System.out.println ("In Enter credits for subject " + (i+1) + ": ");  
            credits[i] = scanner.nextInt();  
            System.out.print ("In Enter marks for subject " +  
                (i+1) + ": ");  
            marks[i] = scanner.nextInt();  
        }  
    }  
    public void displayDetails () {  
        System.out.println ("USN: " + USN);  
        System.out.println ("Name: " + name);  
        System.out.println ("Credits: ");  
        for (int i=0; i < credits.length; i++) {  
            System.out.println ("Subject " + (i+1) + " : " + credits[i]);  
        }  
        System.out.println ("Marks: ");  
        for (int i=0; i < marks.length; i++) {  
            System.out.println ("Subject " + (i+1) + " : " + marks[i]);  
        }  
    }  
    public float calculateSGPA () {  
        float totalCredits = 0;  
        float totalMarks = 0;  
        for (int i=0; i < credits.length; i++) {  
            totalCredits += credits[i];  
            totalMarks += marks[i] * credits[i];  
        }  
        float SGPA = totalMarks / totalCredits;  
        return SGPA;  
    }  
}
```

```
        scanner.close();
```

```
}
```

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

```
    int totalCredits = 0;
```

```
    int weightedSum = 0;
```

```
    double ans;
```

```
    for (int i = 0; i < credits.length; i++) {
```

```
        totalCredits += credits[i];
```

```
        int gradePoints = (marks[i] / 10) + 1;
```

```
        if (gradePoints == 11)
```

```
            gradePoints = 10; }
```

~~```
 else if (gradePoints == 4)
```~~~~```
        gradePoints = 0; }
```~~

```
        weightedSum += gradePoints * credits[i];
```

```
}
```

```
    ans = (double) weightedSum / (double) totalCredits;
```

```
    return ans;
```

```
}
```

```
public void displayDetails() {
```

```
    System.out.println("Student Details:");
```

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

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

```
}
```

```
public class SGPA {
```

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

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

```
 Student student = new Student();
```

```
student.acceptDetails();
student.displayDetails();
double sgpa = student.calculateSGPA();
System.out.println("In SGPA: " + sgpa);
Scanner.close();
```

y

y

Output:-

```
Enter USN: 18M22CS009
Enter name: Abhishek yadav
Enter details for subject:
Enter credit for subject1: 4
Enter marks for subject1: 92
Enter credit for subject2: 4
Enter marks for subject2: 86
Enter marks for subject3: 4
Enter marks for subject3: 92
Enter credits for subject4: 3
Enter marks for subject4: 82
Enter credits for subject5: 3
Enter marks for subject5: 86
Enter credits for subject6: 2
Enter marks for subject6: 90
Enter credits for subject7: 1
Enter marks for subject7: 98
Enter credits for subject8: 1
Enter marks for subject8: 94
```

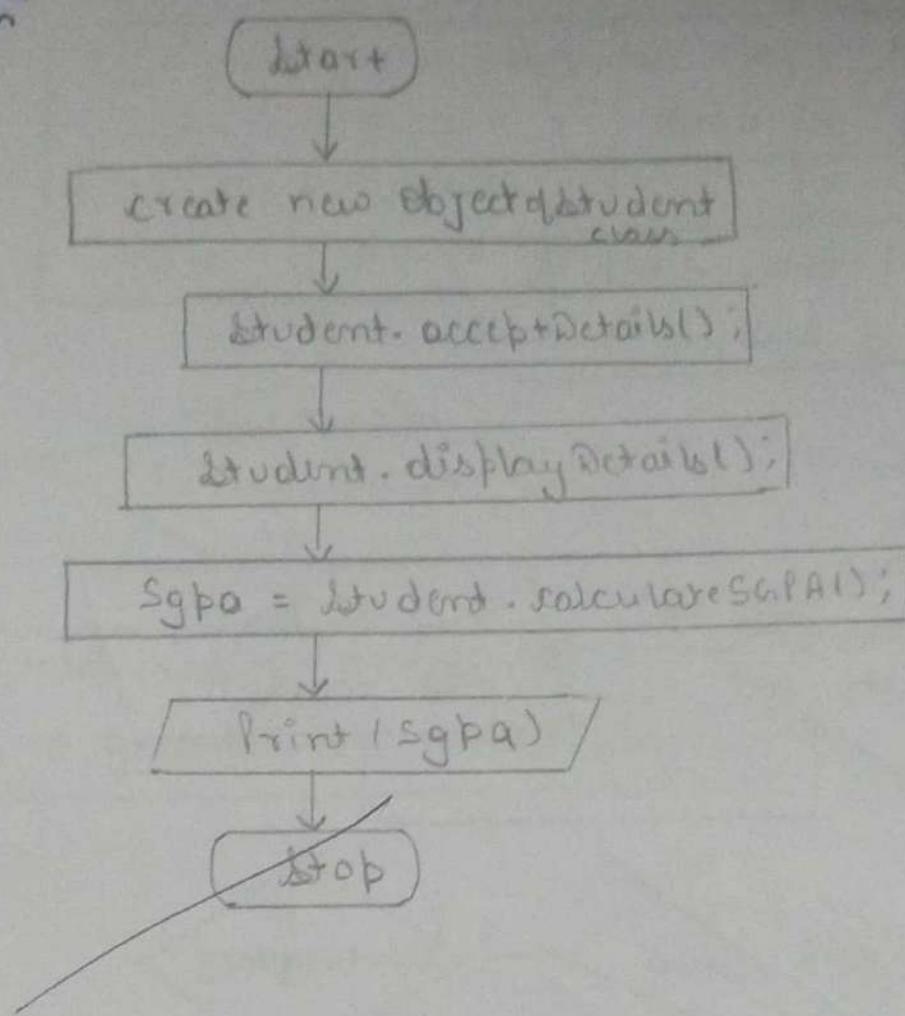
Student Details USN: 18M22CS009

Name: Abhishek yadav

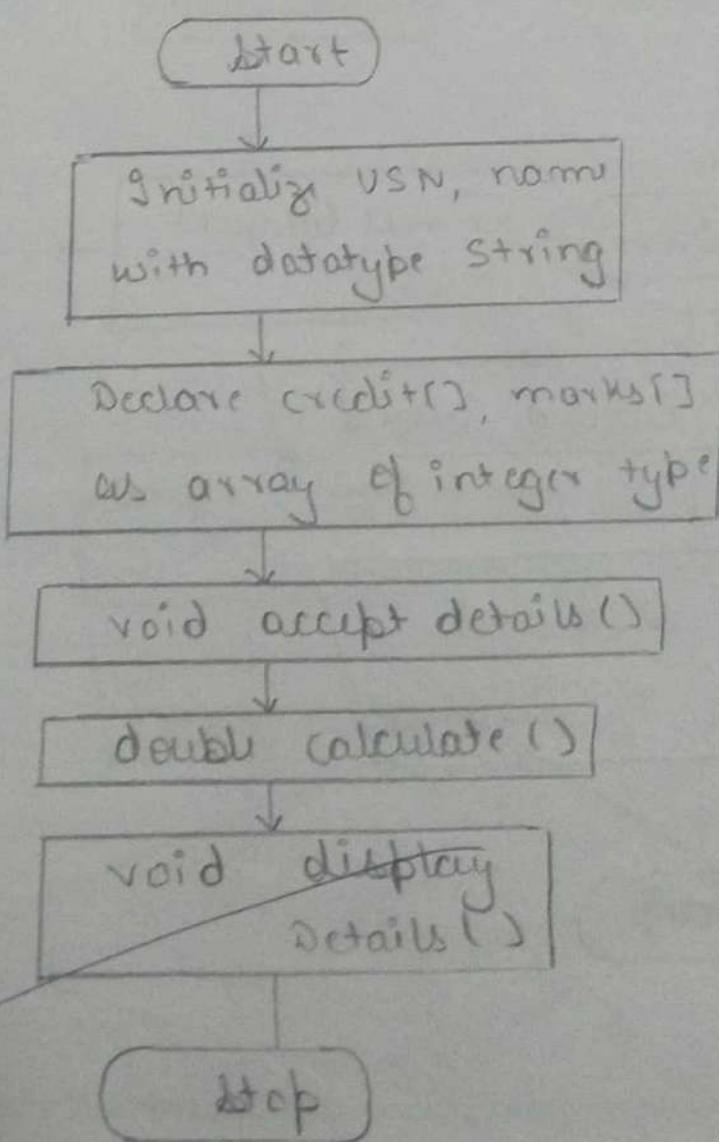
SGPA: 9.545454.

## Flowchart

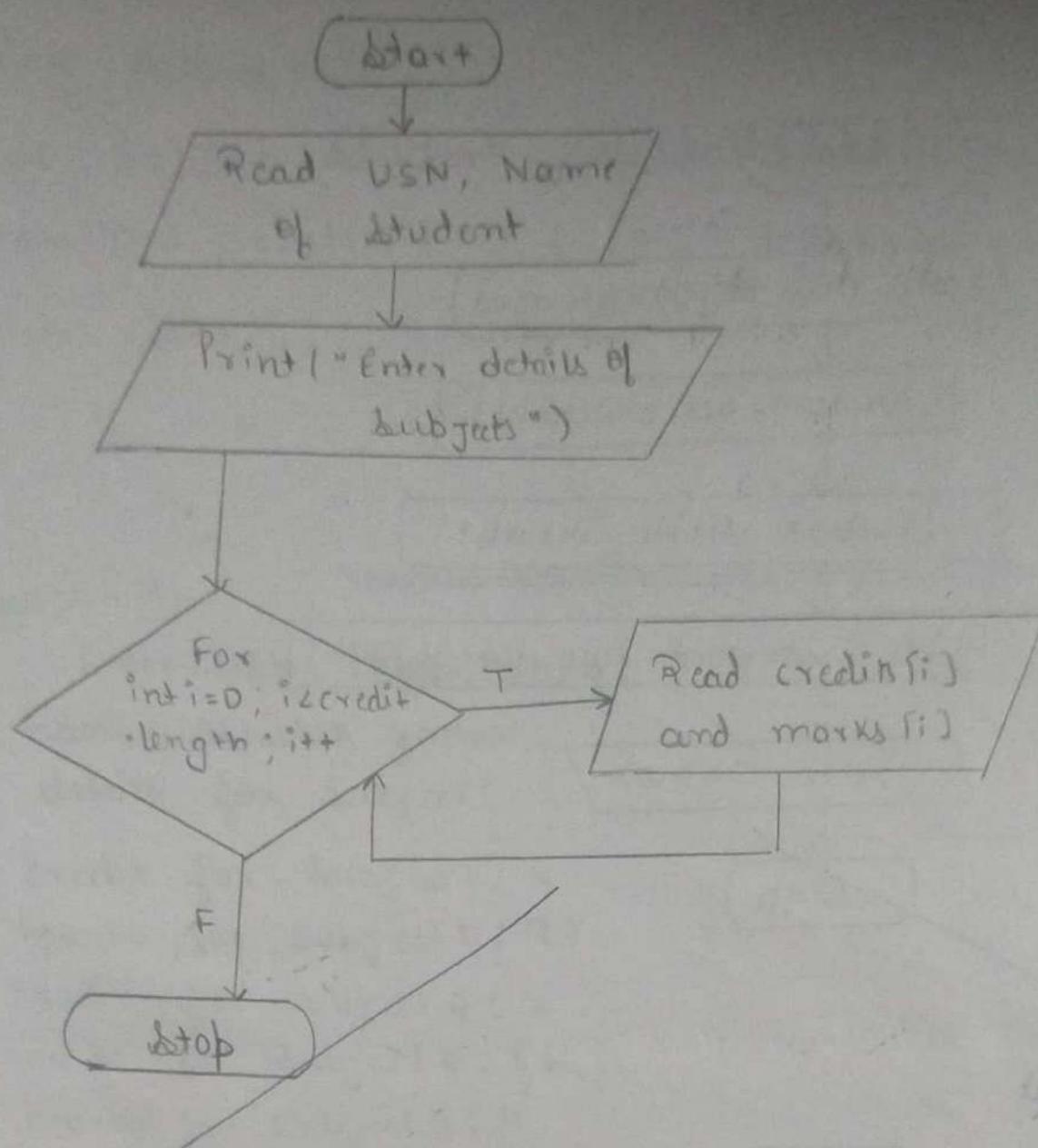
• class bgpa



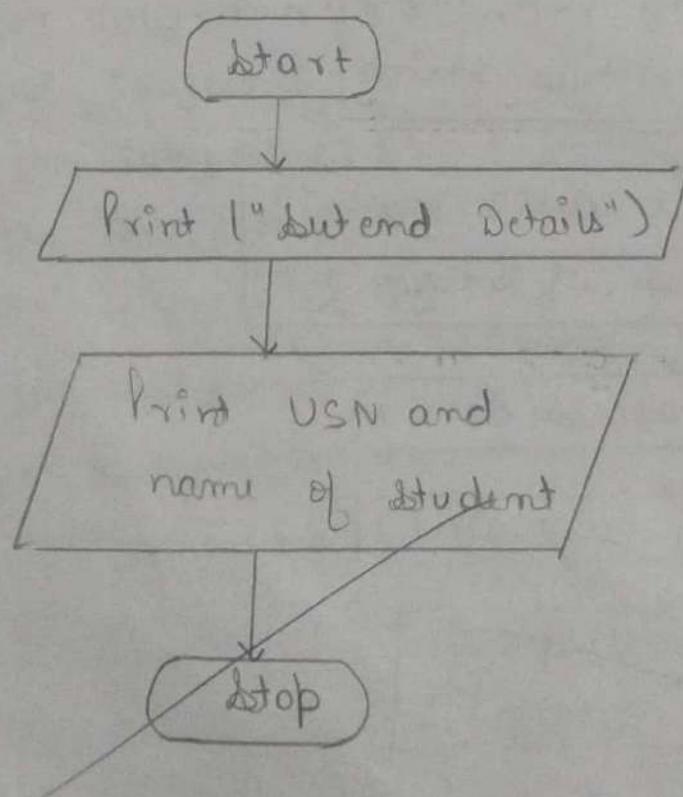
• class student



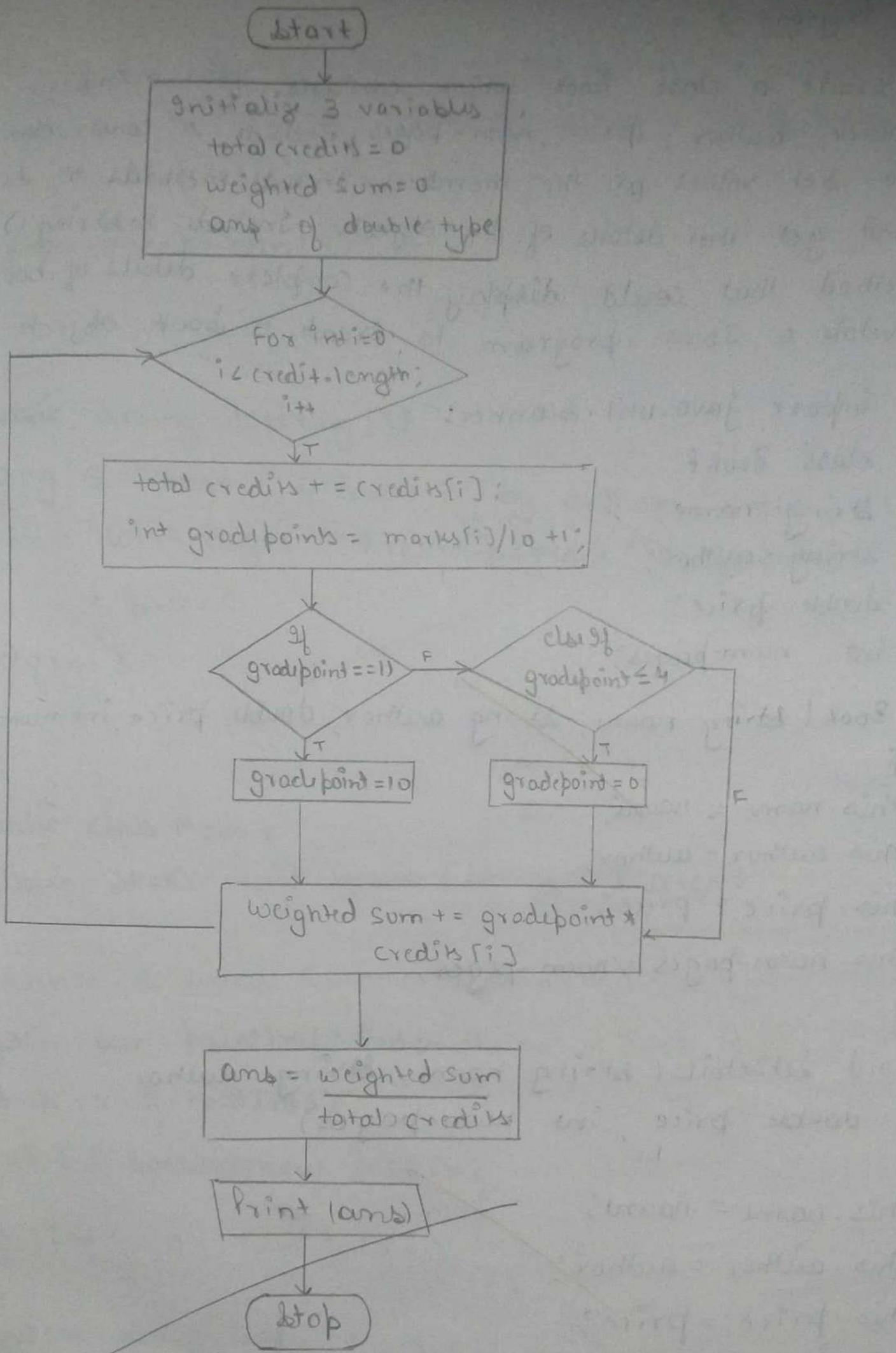
### Method acceptDetails()



### • Method displayDetails()



• Method (calculate SGPA)



K  
29/12/23

C:\Users\Abhishek Yadav\OneDrive\Desktop\Java>java sgpa

Enter USN: 1BM22CS009

Enter Name: Abhishek Yadav

Enter details for each subject :

Enter credits for Subject 1: 4

Enter marks for Subject 1: 90

Enter credits for Subject 2: 4

Enter marks for Subject 2: 90

Enter credits for Subject 3: 4

Enter marks for Subject 3: 95

Enter credits for Subject 4: 3

Enter marks for Subject 4: 88

Enter credits for Subject 5: 3

Enter marks for Subject 5: 86

Enter credits for Subject 6: 3

Enter marks for Subject 6: 92

Enter credits for Subject 7: 1

Enter marks for Subject 7: 95

Enter credits for Subject 8: 1

Enter marks for Subject 8: 96

Student Details :

USN : 1BM22CS009

Name : Abhishek Yadav

SGPA: 9.73913043478261

Date - 12/1/2024

## Lab Program - 2

Q.1 Create a class Book which contains four members name, author, price, num-pages. Include a constructor to set values for the members. Include methods to set and get the details of the objects. Include toString() method that could display the complete details of book. Develop a Java program to create n book objects.

Sol:-

```
import java.util.Scanner;
class Book {
 String name;
 String author;
 double price;
 int numPages;
```

Book( String name, String author, double price, int numPages )

{  
 this.name = name;

this.author = author;

this.price = price;

this.numPages = numPages;

}

void setDetails( String name, String author,  
 double price, int numPages )

{

this.name = name;

this.author = author;

this.price = price;

this.numPages = numPages;

}

```
void getDetails()
{
 String s = "Book:" + name + "by author:" + author +
 "with pages:" + numPages + "is of price:" +
 + price;
 System.out.println(s);
}

public String toString()
{
 String s = "Book:" + name + "by author:" + author +
 "with pages:" + numPages + "is of price:" +
 + price;
 return s;
}

public class Main
{
 public static void main (String [] args)
 {
 Scanner s = new Scanner(System.in);
 System.out.println("Enter the no. of book to create");
 int n = s.nextInt();
 Book [] books = new Book[n];
 for (int i=0; i<n; i++)
 {
 System.out.println("Enter details for Book " + (i+1));
 System.out.print("Name: ");
 String name = s.nextLine();
 System.out.print("Author: ");

```

```
String author = scanner.nextLine();
System.out.print("Price : ");
double price = s.nextDouble();
System.out.print("Number of Pages : ");
int numPages = s.nextInt();
books[i] = new Book(name, author, price, numPages);
}
System.out.println("In Details of books");
for (int i=0; i<n; i++)
{
 System.out.println("In Book " + (i+1) + " In " + books[i]
 .toString());
}
s.close();
}
```

Output

Enter no' of books : 1

Enter details for Book 1

Name : Nuclear

Author : Agam

Price : 101

Number of Pages : 12

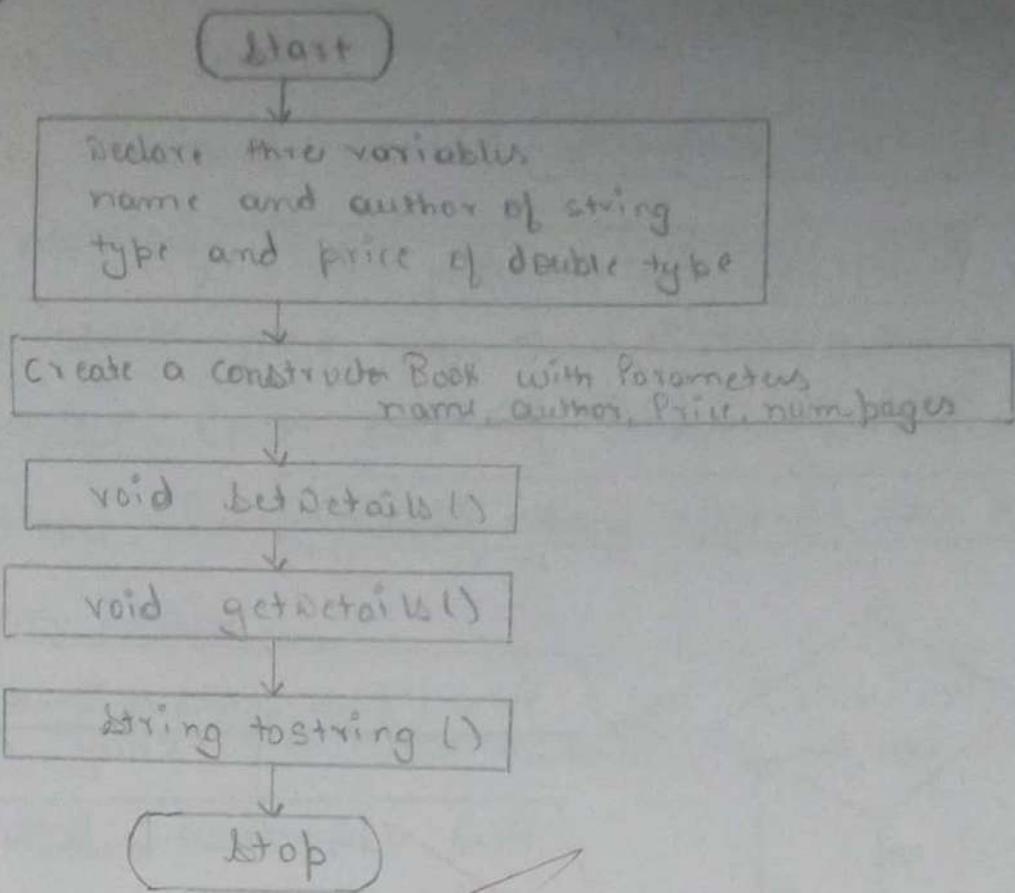
Details of Books :

BOOK 1:

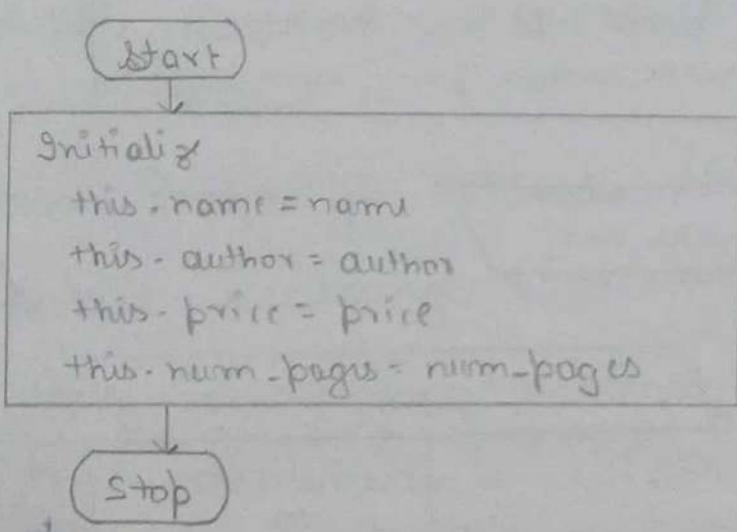
BOOK 1: Nuclear by author: Agam with pages : 12  
is of price : 101.0

# flowchart

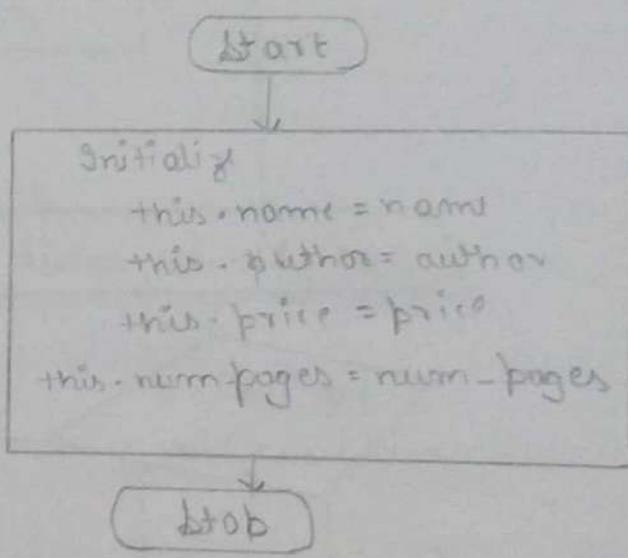
## class Book



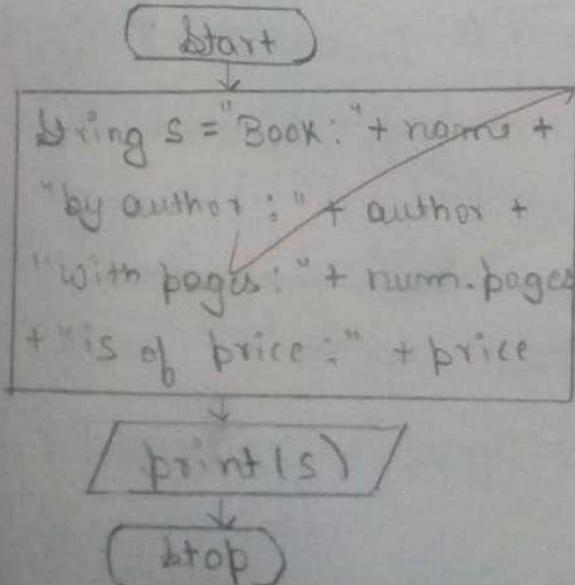
## constructor Book



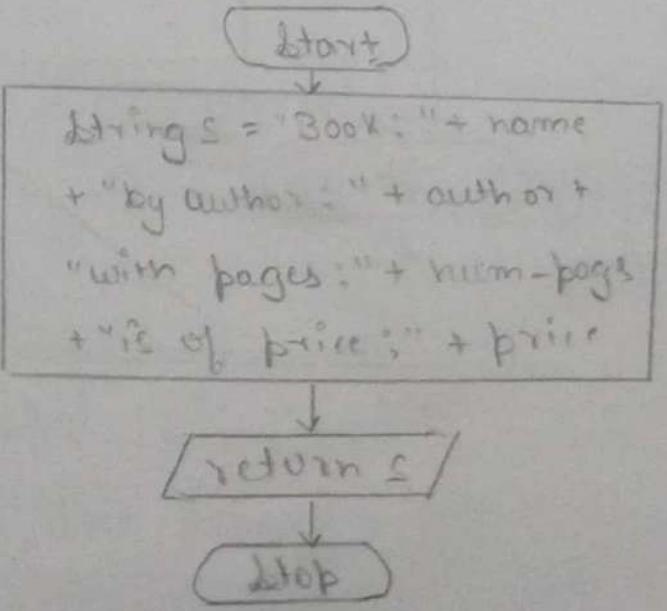
## Method SetDetails()



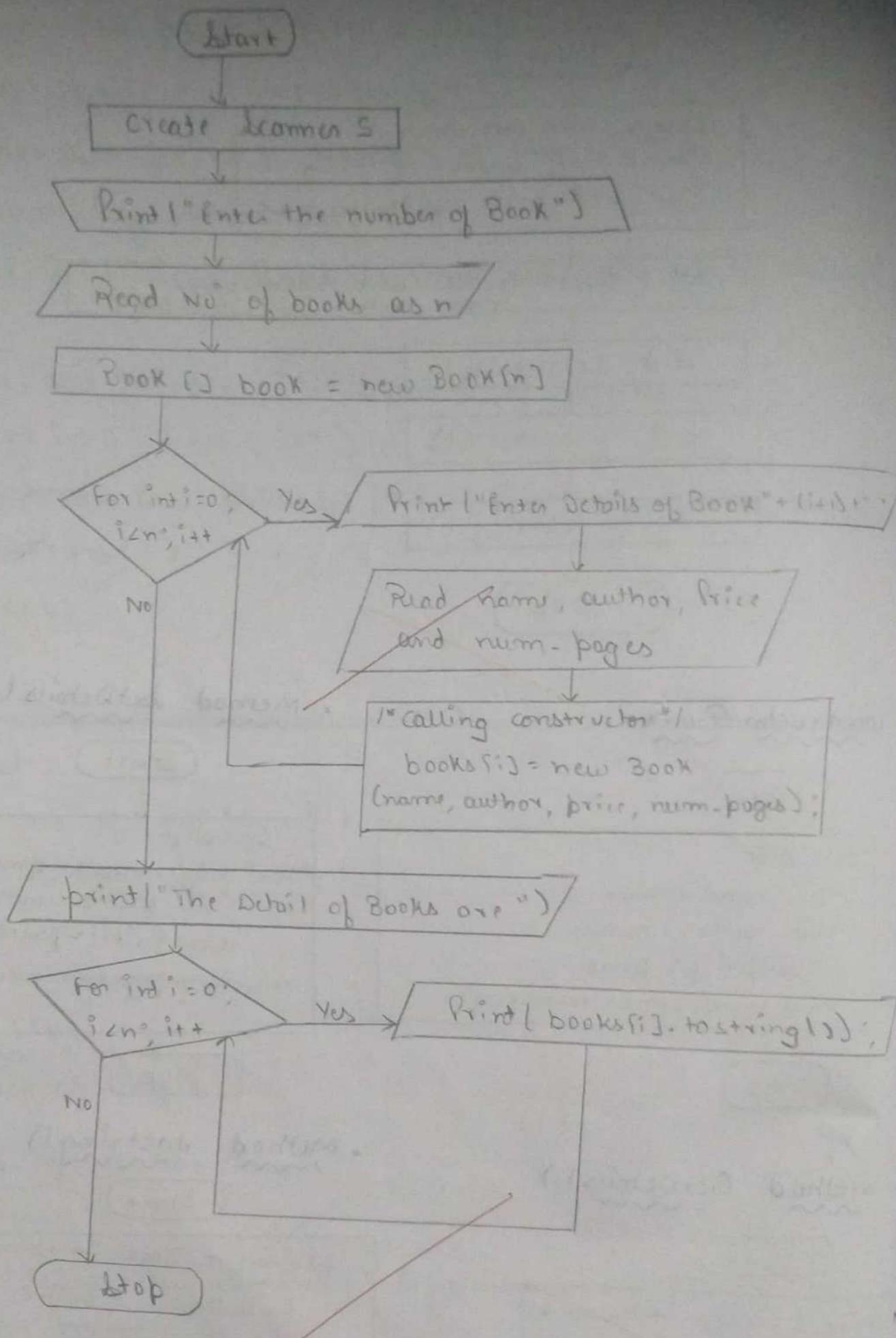
## Method GetDetails()



## Method toString()



## • class Main



C:\Users\Abhishek Yadav\OneDrive\Desktop\Java>java BookProg

Enter the value of n:

2

Enter the name, author, price, number of pages in order:

Nuclear

Abhishek

299

121

Enter the name, author, price, number of pages in order:

Reaction

Yadav

399

151

Details of book:

Name: Nuclear

Author: Abhishek

Price: 299

Num pages: 121

Details of book:

Name: Reaction

Author: Yadav

Price: 399

Num pages: 151

1BM22CS009 Abhishek Yadav

C:\Users\Abhishek Yadav\OneDrive\Desktop\Desktop\Java>[]

Lab Program - 4

Develop a Java program to create an abstract class named Shape that contain two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that the each one of the classes extends the class shape. Each one of classes contain only the method printArea() that prints the area of the given shape.

```
import java.util.Scanner;

abstract class Shape {
 int dimension1;
 int dimension2;
 public abstract void printArea();
}

class Rectangle extends Shape {
 public Rectangle (int length, int width) {
 this.dimension1 = length;
 this.dimension2 = width;
 }

 public void printArea() {
 int area = dimension1 * dimension2;
 System.out.println ("Area of Rectangle: " + area);
 }
}

class Triangle extends Shape {
 public Triangle (int base, int height) {
 }
}
```

this.dimension1 = base;

this.dimension2 = height;

}

public void PrintArea()

{ int area = 0.5 \* dimension1 \* dimension2;

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

} }

class Circle extends Shape {

public Circle (int radius) {

this.dimension1 = radius;

}

public void PrintArea()

int area = Math.PI \* dimension1 \* dimension1;

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

}

}

public class Main {

public static void main (String [] args)

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

System.out.println("Choose a shape to calculate

area in 1. Rectangle in 2. Triangle in 3. Circle

in 4. Exit ");

int run = 1

while (run == 1)

{

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

int choice = scanner.nextInt();

switch (choice) {

```
Case 1 : System.out.print("Enter length of rectangle:");
int length = scanner.nextInt();
System.out.print("Enter width of rectangle:");
int width = scanner.nextInt();
Rectangle r = new Rectangle(length, width);
r.printArea();
break;
```

```
Case 2 : System.out.print("Enter base of Triangle:");
int base = scanner.nextInt();
System.out.print("Enter height of Triangle:");
int height = scanner.nextInt();
Triangle t = new Triangle(base, height);
t.printArea();
break;
```

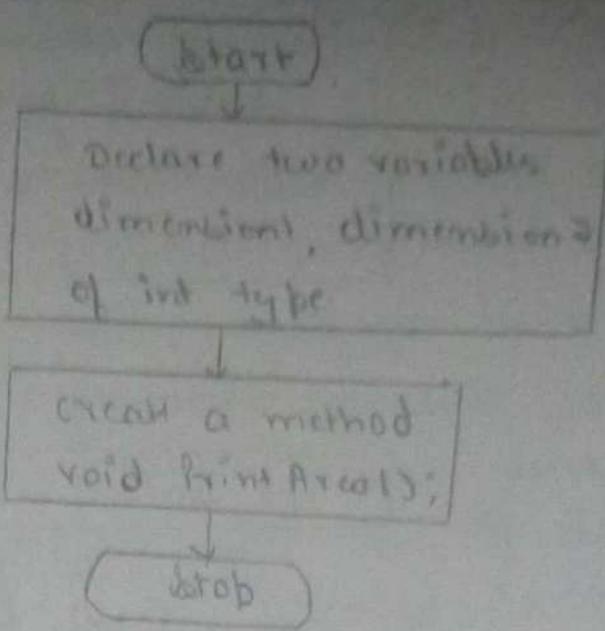
```
Case 3 : System.out.print("Enter radius of circle:");
int radius = scanner.nextInt();
Circle c = new Circle(radius);
c.printArea();
break;
```

```
Case 4 : num = 0;
break;
```

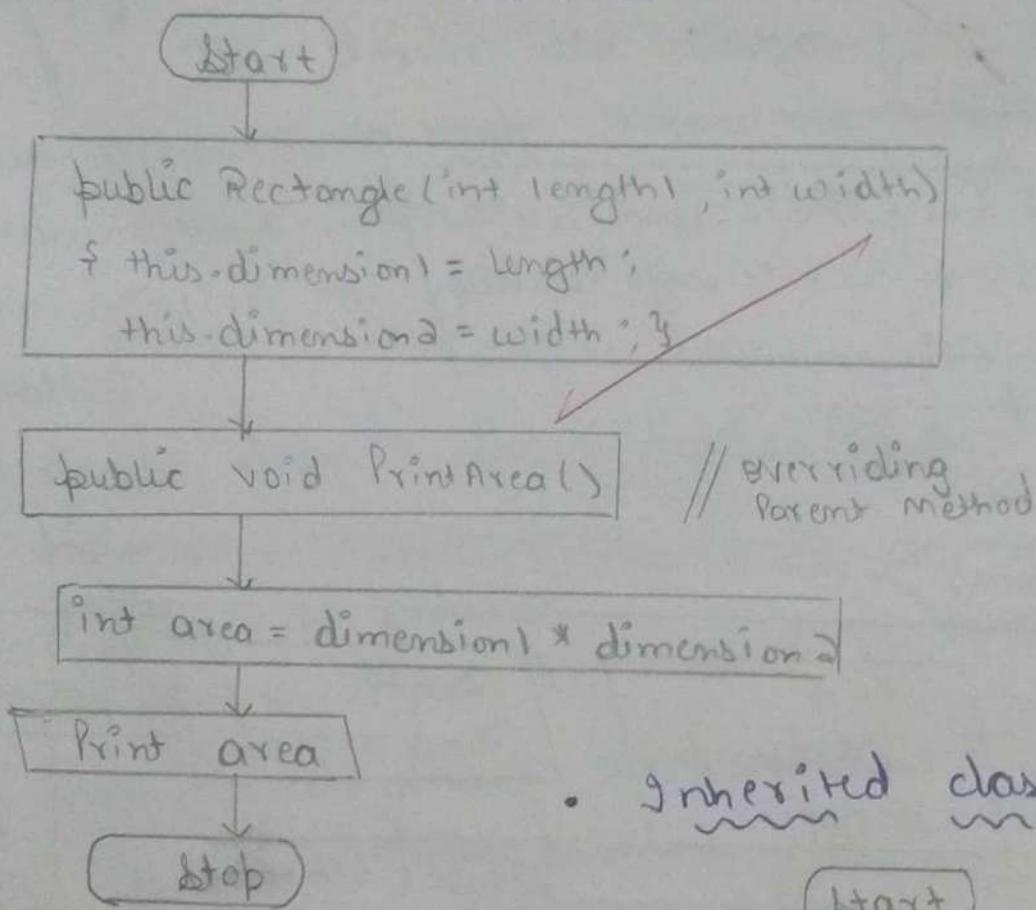
```
default : System.out.print("Invalid choice");
}
} scanner.close();
}
```

# Flowchart

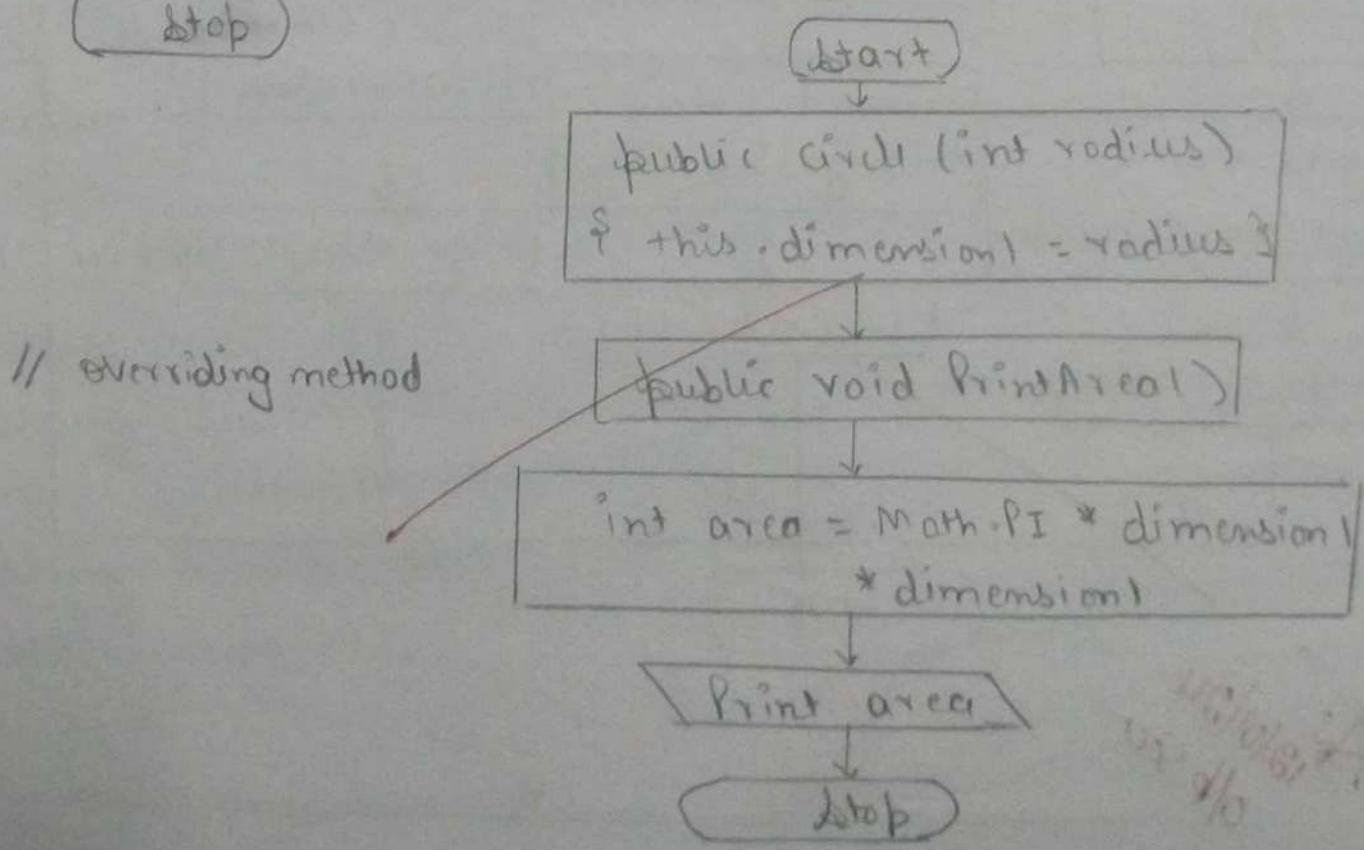
- Abstract class shape



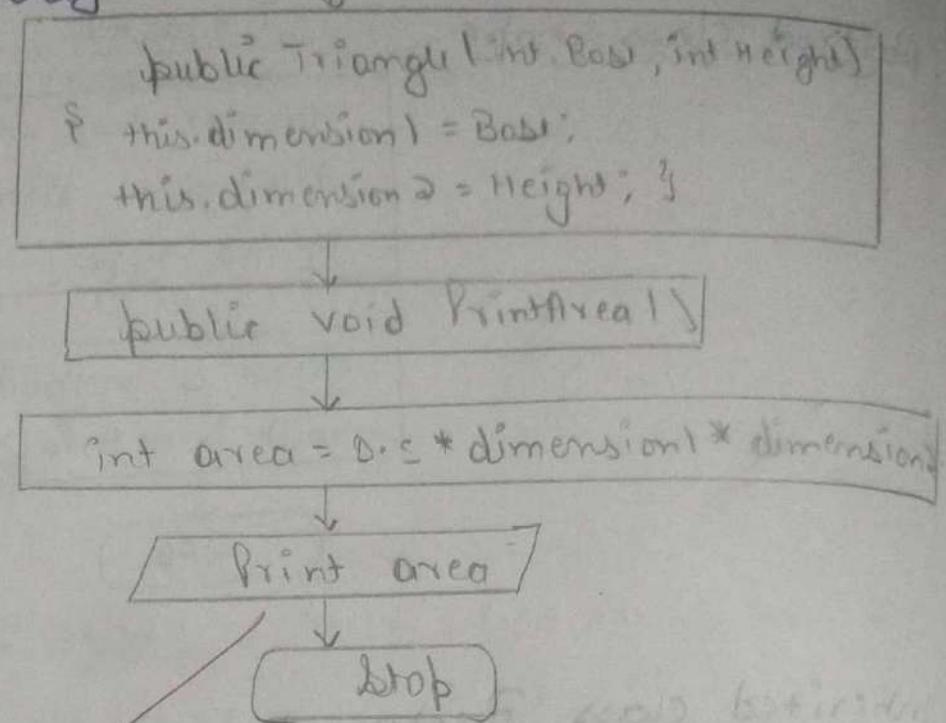
- Inherited class Rectangle



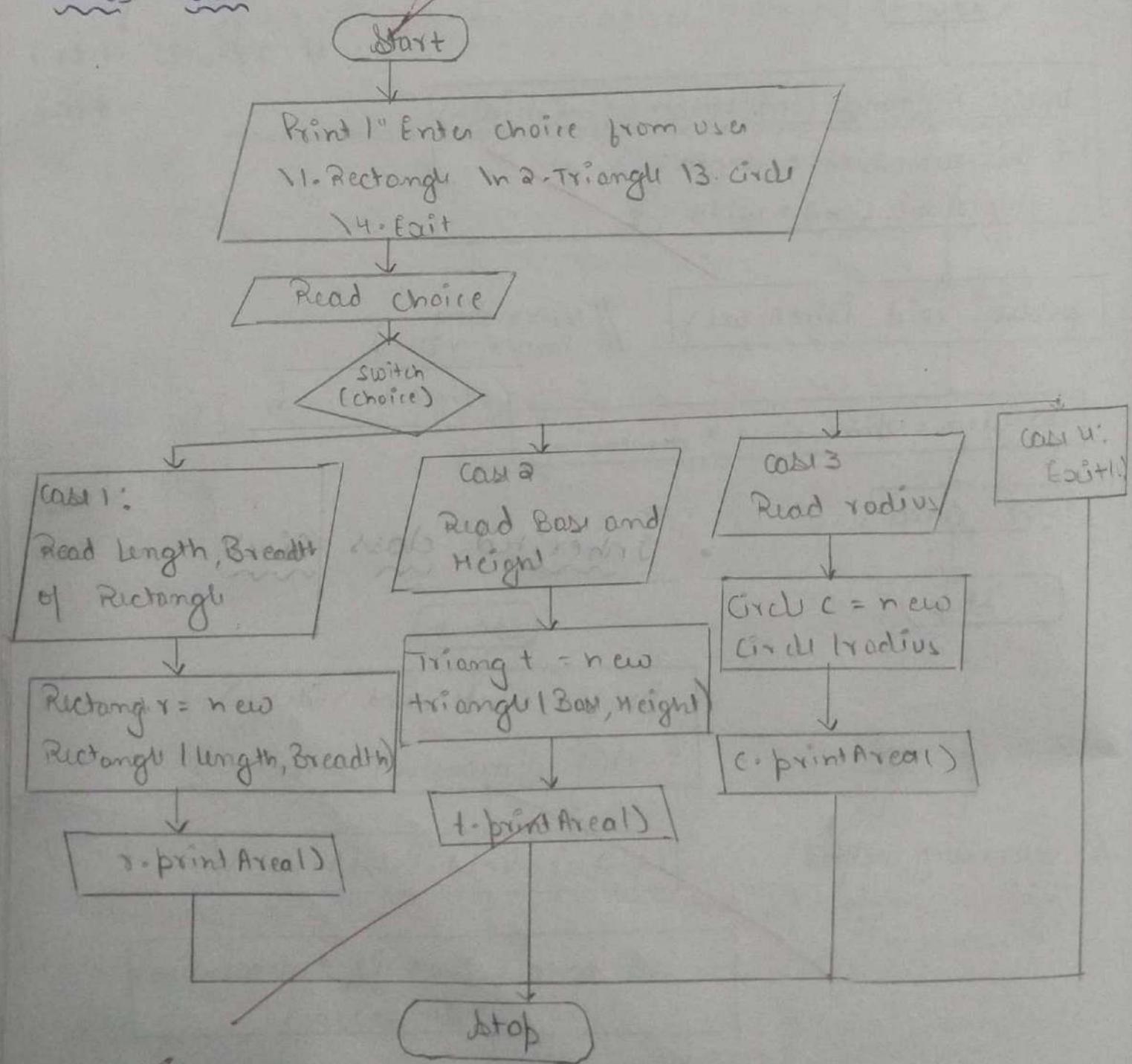
- Inherited class Circle



## Inherited class Trianglu



## Class Main



4.12.01/24  
16 Sec

```
C:\Users\Abhishek Yadav\OneDrive\Desktop\Java>javac areas.java
```

```
C:\Users\Abhishek Yadav\OneDrive\Desktop\Java>java areas
```

```
Abhishek Yadav
```

```
1BM22CS009
```

```
Choose a shape To Calculate Area :
```

- 1. Rectangle
- 2. Triangle
- 3. Circle
- 4. ...Exit..

```
Enter Choice : 1
```

```
Enter length of Rectangle: 2
```

```
Enter width of Rectangle: 2
```

```
Area of Rectangle: 4.0
```

```
Enter Choice : 2
```

```
Enter base of Triangle: 2
```

```
Enter height of Triangle: 2
```

```
Area of Triangle: 2.0
```

```
Enter Choice : 3
```

```
Enter radius of Circle: 2
```

```
Area of Circle: 12.566370614359172
```

```
Enter Choice : 4
```

```
Invalid choice. Please choose a valid choice....
```

```
C:\Users\Abhishek Yadav\OneDrive\Desktop\Java>[]
```

Date: 19/11/2024

b) Develop a Java program to calculate a class Bank that maintains 2 kinds of account for its customer, one called saving and the other called current account. The saving account provides compound interest and withdraw facilities but no cheque book facility but no interest. The current account provides cheque book facility. Current account holder should also maintain a min balance and if 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-account and sav-account to make them more specific to their requirement. 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 the deposit interest.
- d) Permit withdrawal and update the balance. Also check the min balance, impose the penalty if necessary and update the balance.

Ans:- import java.util.Scanner;

class Account {

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

```
public Account (String customerName, int accountN
 String accountType) {
```

```
 this.customerName = customerName;
 this.accountNumber = accountNumber;
 this.accountType = accountType;
 this.balance = 0.0;
}
```

```
public void deposit (double amount) {
```

```
 balance += amount;
```

```
 System.out.println ("Deposit Successful. Updated
balance: " + balance);
```

```
 checkMinimumBalance();
```

```
}
```

```
public void displayBalance () {
```

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

```
}
```

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

```
 if (amount <= balance) {
```

```
 balance -= amount;
```

```
 System.out.println ("Withdraw Successful. Updated
balance: " + balance);
```

```
 checkMinimumBalance();
```

```
}
```

```
 else
```

```
 System.out.println ("Insufficient funds for
withdrawl.");
```

```
}
```

```
}
```

```
public void checkMinimumBalance () { }
```

```
public void computeInterest () { }
```

```
}
```

```
class CurAcct extends Account {
 double minimumBalance;
 double serviceCharge;
}

public CurAcct (String customerName, int accountNumber)
{
 super (customerName, accountNumber, "Current");
 this.minimumBalance = 1000.0;
 this.serviceCharge = 50.0;
}
```

```
public void checkMinimumBalance () { // override
 if (balance < minimumBalance) {
 balance -= serviceCharge;
 System.out.println ("Service charge imposed. Updated
 balance: " + balance);
 }
}
```

```
class SavAcct extends Account {
 double interestRate;
}

public SavAcct (String customerName, int accountNumber)
{
 super (customerName, accountNumber, "Savings");
 this.interestRate = 0.05;
}
```

```
public void computeInterest () {
 double interest = balance * interestRate;
 balance += interest;
 System.out.println ("Interest computed and deposited
 update balance: " + balance);
}
```

```
public class Bank {
 public static void main (String [] args) {
 Scanner scanner = new Scanner (System.in);
 System.out.print ("Enter Customer Name : ");
 String customerName = scanner.nextLine ();
 System.out.print ("Enter account number : ");
 int accountNumber = scanner.nextInt ();
 System.out.print ("Enter account type : ");
 String accountType = scanner.nextLine ();

 Account bankAccount;
 if (accountType.equalsIgnoreCase ("Current"))
 {
 bankAccount = new CurAcct (customerName, accountNumber);
 }
 else if (accountType.equalsIgnoreCase ("Savings"))
 {
 bankAccount = new SavAcct (customerName, accountNumber);
 }
 else
 {
 System.out.println ("Invalid account type");
 scanner.close ();
 }

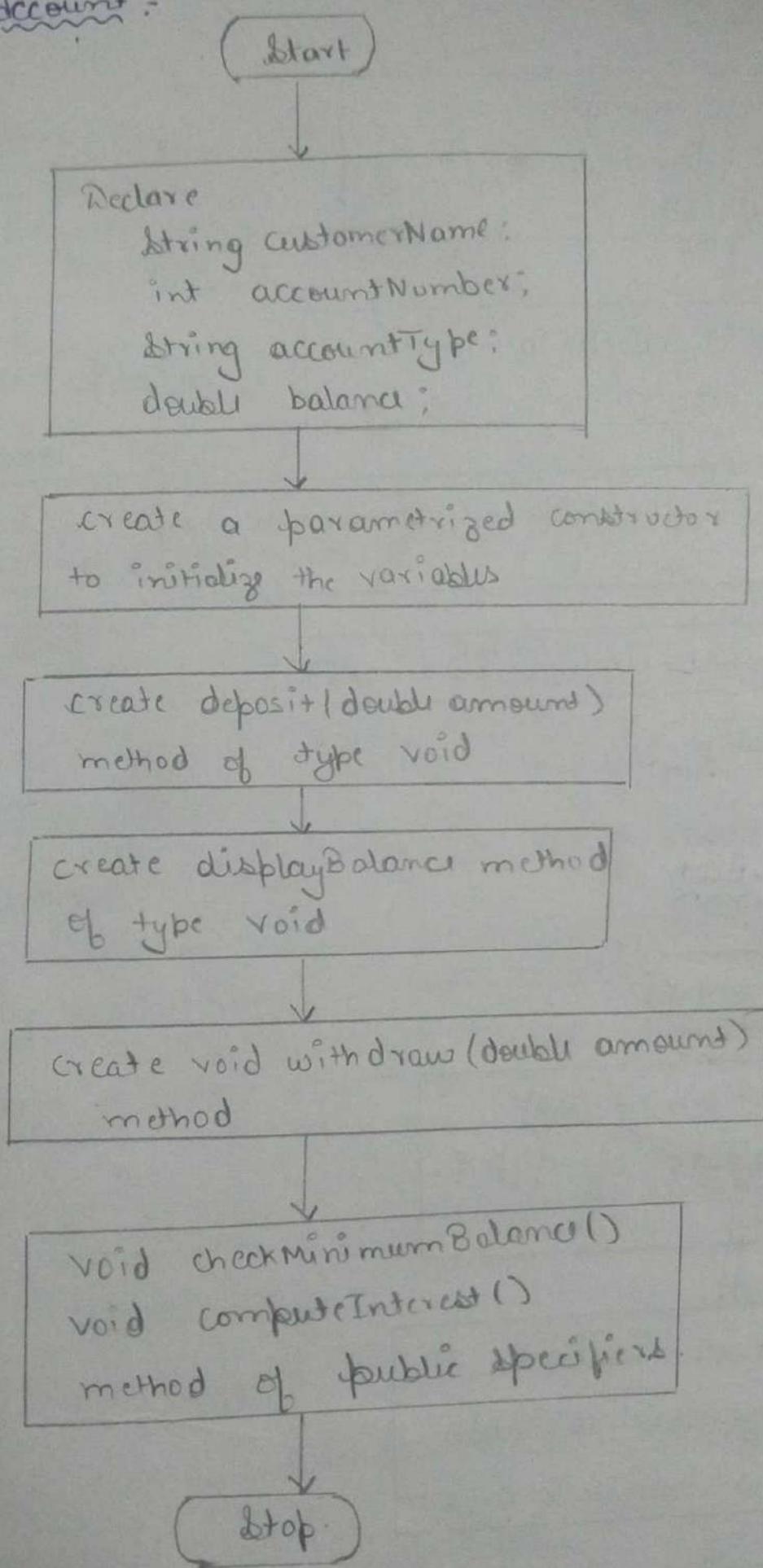
 int choice;
 do
 {
 System.out.println ("Choose an option : ");
 System.out.println ("1. Deposit");
 System.out.println ("2. Withdraw");
 System.out.println ("3. Display Balance");
 System.out.println ("4. Compute Interest");
 System.out.println ("5. Exit");
 }
```

```
System.out.println("Enter your choice");
choice = Scanner.nextInt();
switch(choice)
{
 case 1: System.out.print("Enter deposit Amount");
 double depositAmount = scanner.nextDouble();
 bankAccount.deposit(depositAmount);
 break;
 case 2: System.out.print("Enter withdrawl Amount:");
 double withdrawlAmount = scanner.nextDouble();
 bankAccount.withdraw(withdrawlAmount);
 break;
 case 3: bankAccount.displayBalance();
 break;
 case 4: if(bankAccount instanceof SavAcct) {
 bankAccount.computeInterest(7);
 }
 else
 {
 System.out.println("Applicable to Saving
account only");
 }
 break;
 case 5: exit(0);
 default: System.out.println("Invalid choice");
}
}
while(choice != 0);
scanner.close();
}
```

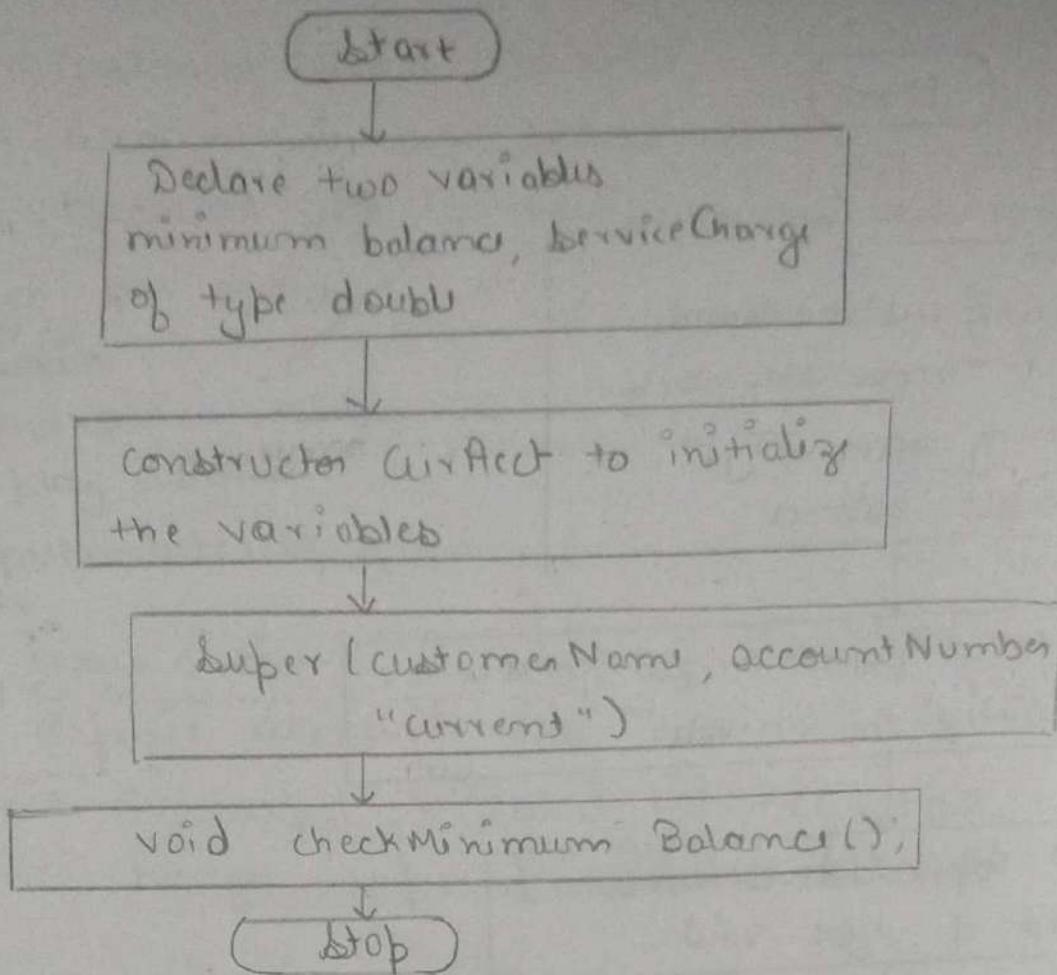
Pratik Sali

# Flowchart

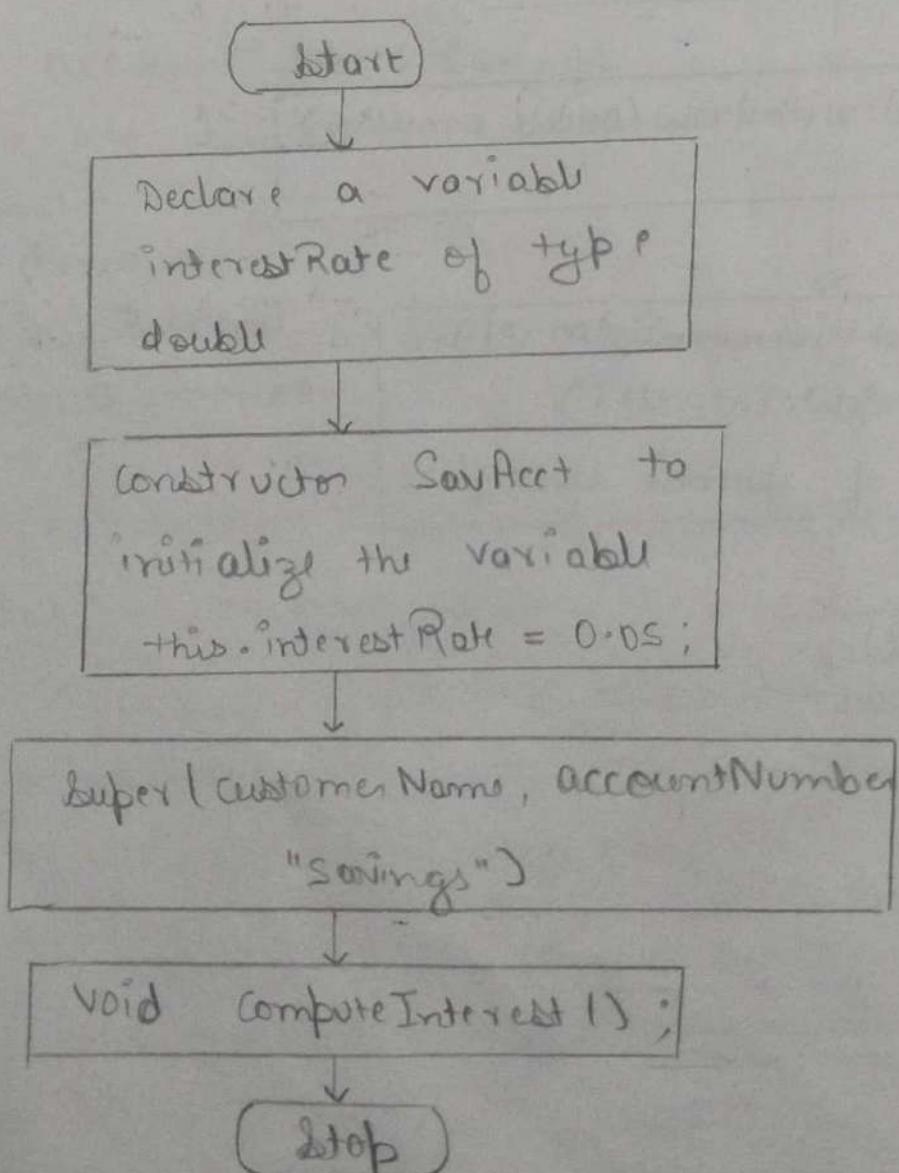
class Account :



## Inherited class CurrAcct

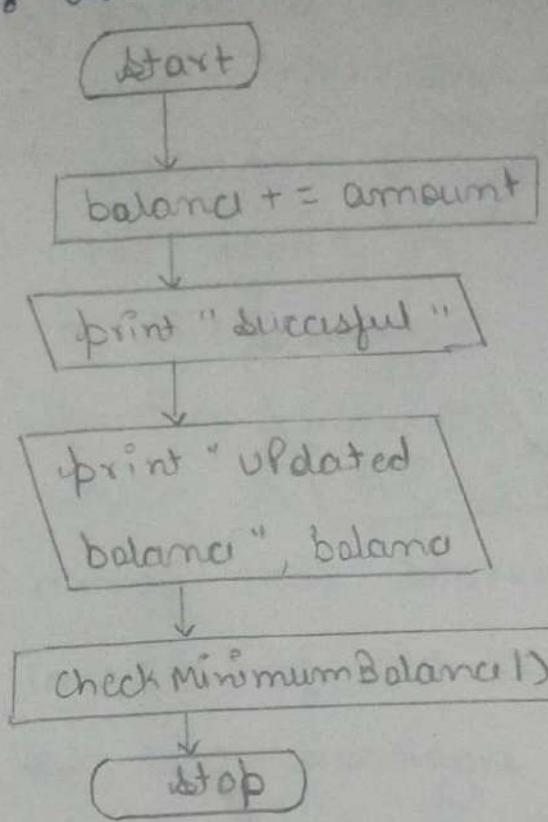


## Inherited class SavAcct

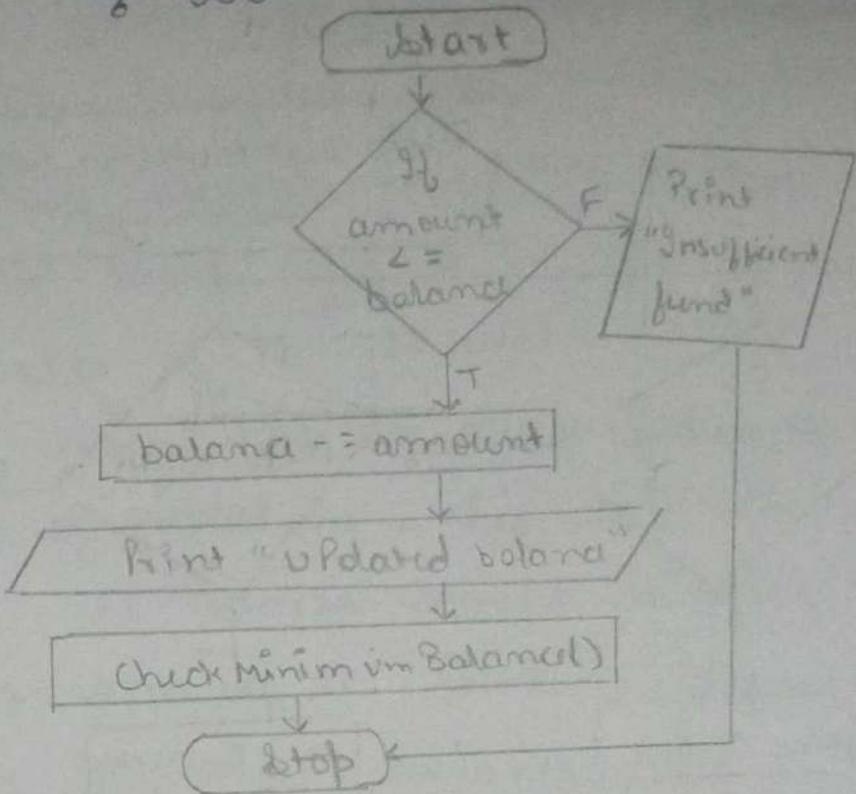


## Methods

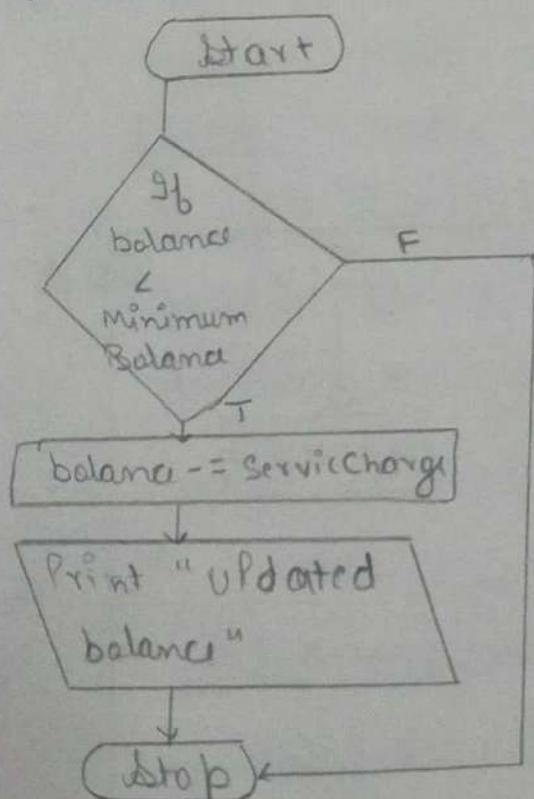
1) void deposit(amount)



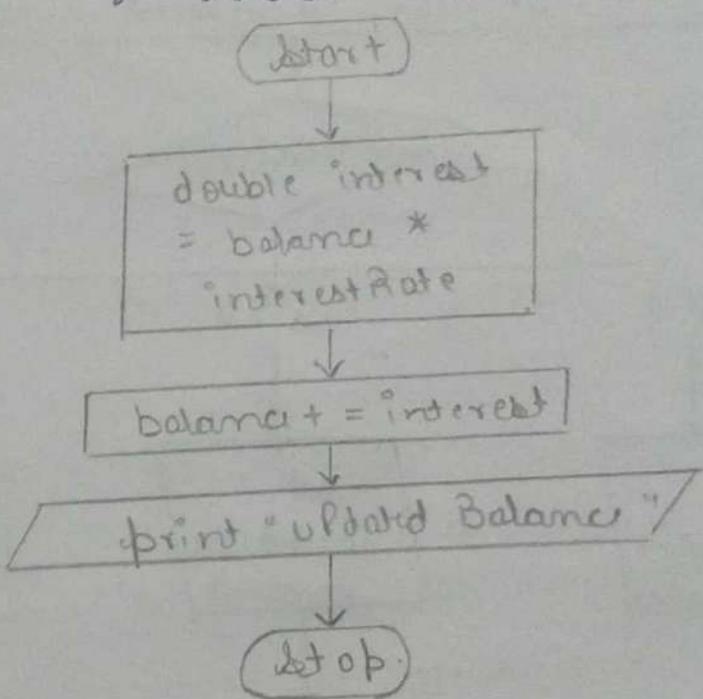
2) void withdraw(amount)



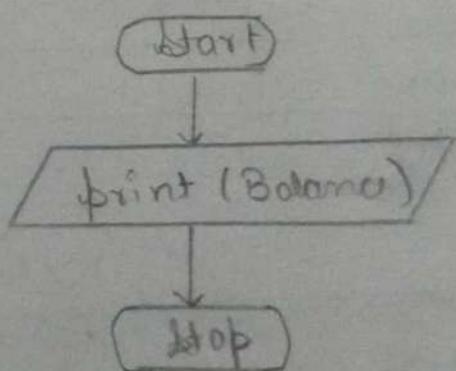
3) Check MinimumBalance()



4) Compute Interest



5) Display Balance



# Class Main - Bank

Start

Read customerName, accountNumber  
accountType

if  
AccountType  
= "Current"

new CurAcct

else if  
AccountType  
= "Savings"

new SavAcct

else  
"Invalid  
Choice"

Print "choose 1. Deposit 2. withdraw  
3. Display 4. Interest 5. exit"

switch [choice]

case: 1

Read  
deposit  
Amount

Deposit( );

break;

case: 2

Read  
withdraw  
Amount

withdraw();

break;

case: 3

display  
Balance();

break

case: 4

Interest();

break

stop

Final

C:\Users\Abhishek Yadav\OneDrive\Desktop\Java>javac Bank.java

C:\Users\Abhishek Yadav\OneDrive\Desktop\Java>java Bank

Enter customer name: Abhishek yadav

Enter account number: 12345

Enter account type (Current/Savings): Current

Choose an option:

1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings)
0. Exit

Enter your choice: 1

Enter deposit amount: 1500

Deposit successful. Updated balance: 1500.0

Choose an option:

1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings)
0. Exit

Enter your choice: 2

Enter withdrawal amount: 800

Withdrawal successful. Updated balance: 700.0

Service charge imposed. Updated balance: 650.0

Choose an option:

1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings)
0. Exit

Enter your choice: 0

Exiting...

C:\Users\Abhishek Yadav\OneDrive\Desktop\Java>[]

Date - 2/2/20

Ques: Create a package CIE which has two classes - Student and Internals. The class Personal has members like USN, name, Sem. The class Internals 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 derived class of Student. This class an array that stores the SEE marks scored in five courses of the current term of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

Ans:

- package CIE;

```
import java.util.*
public class Student
```

```
{ public int Sem;
 public String USN;
 public String name;
 public void accept()
```

```
{ Scanner scan = new
 Scanner (System.in);
```

```
 System.out.println ("Enter USN, name, Sem : ");
```

```
 USN = scan.nextLine();
```

```
 name = scan.nextLine();
```

```
 Sem = scan.nextInt();
```

```
}
```

```
}
```

- package CIE;

```
public class Internals
```

```
{ public int arr[] = new int [5]; }
```

```
• package SEE;
import CIE.Student;
public class External extends Student
{ public int sm[] = new int [5]; }

import Java.util.*;
import SEE.*;
import CIE.*;
public class FinalMarks
{
 public static void main (String args[])
 {
 int fm[] = new int [5];
 Scanner sc = new Scanner (System.in);
 System.out.println ("Enter n. ");
 int n = sc.nextInt();
 SEE.External st[] = new SEE.External [n];
 CIE.Internal s[] = new CIE.Internal [n];
 for (int i=0; i<n; i++)
 {
 st[i] = new SEE.External ();
 s[i] = new CIE.Internal ();
 System.out.println ("Enter details " + (i+1));
 st[i].accept();
 }
 for (int j=0; j<5; j++)
 {
 System.out.println ("Enter internal marks and
sum end marks of sub " + (j+1));
 s[i].im[j] = sc.nextInt();
 st[i].sm[j] = sc.nextInt();
 fm[j] = s[i].im[j] + st[i].sm[j];
 }
 System.out.println ("Final marks of " + st[i].name);
 for (int k=0; k<5; k++) System.out.println ("Course
" + (k+1) + " = " + fm[k]);
 }
}
```

Algorithm:

Start

Step 1: Create a package CIE // 1st Package

Step 2: import java.util.\*

Step 3: Create a class Student with three public variables  
and a public method accept

public int idm

public String USN

public String name

Step 4: public void accept()

{ Read idm

Read USN

Read name }

Step 5: Create another public class Internals with  
a public array variable

public int im[] = new int[5];

Step 6: Create a package SEE // 2nd Package

Step 7: Create a class External that extends class  
Student with 1 array variable.

public int sm[] = new int[5];

// Main class

Step 8: import SEE.\*

import CIE.\*

Step 9: Create a public class Final Marks

Step 10: Create a public static method void main  
(String args[])

Step 11: declare an array variable,

int fm[] = new int[5];

Step 12: Create Scanner sc

Step 13: Read number of students

Step 14 :- Create Instances of Internal and External classes as

SEE::External st[] = new SEE::external[n]

CIE::Internal si[] = new CIE::internal[n]

Algo 15 :- For int i=0 ; i < n ; i++

Step 16 :- Read details of (i+1) student as  
st[i].accept()

Step 17 :- For int j=0 ; j < 5 ; j++

Step 18 :- Read Internal marks and Sem and marks  
as  
st[i].im[j] and ~~st[i].~~ st[i].sm[j]

Step 19 :- Sum or Add the mark and store it in the  
variable fm

Step 20 :- For int k=0 ; k < 5 ; k++

Step 21 :- Print final marks as fm[k]

Step 22 :- Stop Algorithm

---

Output: Enter n : 1

Enter details 1

Enter USN, Name and Sem:

123 Abhishek 1

Enter Internal-marks and Sem-marks of Subject 1

48

49

Subject 2

30

Subject 3

30 30

Subject 4

48 50

Subject 5

23 47

Final Marks

Course 1 = 97

Course 2 = 70

Course 3 = 60

Course 4 = 98

Course 5 = 70

08.02.2k  
Sri

C:\Users\Abhishek Yadav\OneDrive\Desktop\Java>java FinalMarks

Enter n:

1

Enter details 1

Enter U, N, S:

123

Abhishek

1

Enter im and sm of sub 1

47

48

Enter im and sm of sub 2

42

48

Enter im and sm of sub 3

50

50

Enter im and sm of sub 4

39

41

Enter im and sm of sub 5

29

50

Final marks of Abhishek

Course 1 = 95

Course 2 = 90

Course 3 = 100

Course 4 = 80

Course 5 = 79

C:\Users\Abhishek Yadav\OneDrive\Desktop\Java>

Date - 16/2/2024

Q. 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.

Sol. class DisplayMessage extends Thread

```
{
 String message;
 int interval;
```

```
 public DisplayMessage (String message, int interval) {
 this.message = message;
 this.interval = interval;
```

}

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

```
 while (true)
```

```
 { try { System.out.println(message);
 Thread.sleep (interval * 1000); } }
```

```
 catch (InterruptedException e)
```

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

}

}

}

```
public class ThreadDemo {
```

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

```
 {
 DisplayMessage thread1 = new DisplayMessage
```

```
 ("BMS College of Engineering", 10);
```

```
 DisplayMessage thread2 = new DisplayMessage
```

```
 ("CSE", 2);
```

~~```
        thread1.start();  
        thread2.start();
```~~

}

Algorithm

- Step 1 : Start Algorithm
- Step 2 : Create a class DisplayMessage that extends Thread
- Step 3 : declare 2 variables message and interval
of type String and int respectively
- Step 4 : Create a method displayMessage with parameter
(String message, int interval) as construction
to assign the value of message and interval
- Step 5 : Create a method run of type void
- Step 6 : while (true)
Step 7 : try {
 System.out.println("Message")
 Thread.sleep(interval * 1000)
}
Step 8 : catch (InterruptedException e)
{
 System.out.println("Error");
}
- Step 9 : Create a ~~new~~ class ThreadDemo
- Step 10 : Create a method main of type void
- Step 11 : Create instances of thread
as thread1 and thread2
- Step 12 : thread1("BMS college of Engineering", 10);
thread2("CSE", 2);
- Step 13 : Run thread1.start()
 thread2.start()
- Step 14 : Stop Algorithm

Output : BMS college of Engineering
CSE ...

```
C:\Users\Abhishek Yadav\OneDrive\Desktop\Java>java threadDemo
Abhishek Yadav
1BM22CS009
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
^C
C:\Users\Abhishek Yadav\OneDrive\Desktop\Java>
```

Q2 Write a program that demonstrates handling of exception in Inheritance tree. Create a base class called "Father" and derived class called "Son" which extends the base class. In Father class, implement a const which takes the age and throws the exception WrongAge() when the input age < 0. In Son class implement a constructor that takes both father and son's age and throw an exception if Son's age is \geq father's age.

```
#include <iostream>
#include <exception>

class WrongAge : public std::exception {
public:
    WrongAge(const string& message) : message_(message) {}
    const string& what() const { return message_; }
private:
    string message_;
}
```

```
class Father {
private:
    int age;
public:
    Father(int age) throws WrongAge {
        if (age < 0) {
            throw new WrongAge("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
```

```
WrongAge {
```

```
super(fatherAge);
```

```
if (sonAge >= fatherAge)
```

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

}
this.sonAge = sonAge;

public int getSonAge() {
 return sonAge;
}

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

 try {

 Father father = new Father(45);

 Son son = new Son(45, 40);

 System.out.println("father's age: " + father.
 getAge());

 System.out.println("Son's age: " + son.getSonAge());

}

 catch (WrongAge e)

 {

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

}

}

Output:

father's age: 45

Son's age: 40

Algorithm:

Step 1 :- Start Algorithm

Step 2 :- Create a class WrongAge that extends Exception

Step 3 :- Create a public constructor WrongAge with a parameter i.e. (String message)

Step 4 :- Calling the constructor of exception class using super (String) by passing the variable string.

Step 5 :- Create a class Father

Step 6 :- declare a private variable age of type int

Step 7 :- Create a public method father with parameter int age that throws an exception.

Step 8 :- If (age < 0)

 throw new WrongAge ("Age cannot be negative")

Step 9 :- Initialize the variable age

 this.age = age

Step 10 :- Create a method toGetAge()

Step 11 :- print ("age");

Step 12 :- Create another public class Son that extends Father

Step 13 :- declare a private variable sonage of type int.

Step 14 :- Create a public constructor Son with parameters (int age, int sonage) that throws an exception

Step 15 :- call the construct of Parent class by passing the argument age
super (age)

Step 16 :- If (age <= sonage)

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

Step 17: Initialize SonAge = Sonage;

Step 18: Create a method to get SonAge() that prints SonAge

Step 19: Create a class ExceptionInheritanceDemo

Step 20: Create a method public main()

Step 21: try {

Create instances of class Son and Father;

Print 1st SonAge and FatherAge");

}

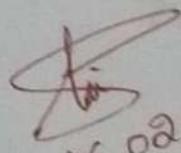
Step 22: catch (WrongAge e)

{

Print error or exception

}

Step 23: End Algorithm


16.02.24

```
C:\Users\Abhishek Yadav\OneDrive\Desktop\Java>java ExceptionInheritanceDemo  
Abhishek Yadav  
1BM22CS009  
Father's age: 45  
Son's age: 40
```

```
C:\Users\Abhishek Yadav\OneDrive\Desktop\Java>
```

Date: 23/2/2024

B.1 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 ArithmeticException. Display the exception in a message dialog box.

Sol:

```
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);
        JLabel jlab = new JLabel("Enter the dividend and
                                -           divisor : ");
        JTextField ajtf = new JTextField(18);
        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(jlab);
        jfrm.add(ajtf);
        jfrm.add(bjtf);
        jfrm.add(button);
        jfrm.add(alab);
        jfrm.add(blab);
        jfrm.add(anslab);
        jfrm.add(err);

        button.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent e)
            {
                int a = Integer.parseInt(ajtf.getText());
                int b = Integer.parseInt(bjtf.getText());
                if (b == 0)
                {
                    JOptionPane.showMessageDialog(null, "Division by zero is not allowed");
                }
                else
                {
                    int result = a / b;
                    anslab.setText("Result: " + result);
                }
            }
        });
    }
}
```

```
jfrm.add(terr);
jfrm.add(ljlab);
jfrm.add(lajtf);
jfrm.add(lbjtf);
jfrm.add(lbutton);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(amslab);
```

ActionListener I = new ActionListener()

{

public void actionPerformed(ActionEvent evt)

{

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

}

,

ajtf.addActionListener(I);

bjtf.addActionListener(I);

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("In A = " + a);

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

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

}

Catch (NumberFormatException e)

{

alab.setText(" ");

blab.setText(" ");

amslab.setText(" ");

```

err.setText ("Enter Integer Only! ");
y
catch (ArithmaticException e)
{
    alab.setText ("");
    blab.setText ("");
    anslab.setText ("");
    err.setText ("The B value Should not Equal to zero");
    y
    y
    y);
    jfrm.setVisible (true);
    y
public static void main (String args[])
{
    SwingUtilities.invokeLater (new Runnable () {
        public void run()
        {
            new SwingDemo ();
            y
            y);
            y
            y
}

```

Output

Enter the dividend and divisor
 calculate

$$A = 12 \quad B = 2 \quad Ans = 6$$

Terms & Components

1. JFrame - The Java Swing. JFrame or JFrame class is a type of container. JFrame works like the main window where components like labels, buttons, text fields are added to create a GUI.
2. JLabel - JLabel class is a component for placing text in a container. It is used to display a single line of text.
3. JTextField - JTextField class is a text component that allows the editing of a single line text.
4. JButton - This is used to create a labeled button that has platform independent implementation.
5. ActionListener - It is notified whenever you click on button. It is notified against actionPerformed with a method actionPerformed.

~~23.02.24~~
23.02.24

Microsoft Windows [Version 10.0.22621.3155]

(c) Microsoft Corporation. All rights reserved.

C:\Users\Abhishek Yadav\OneDrive\Desktop\Java>java SwingDemo

Abhishek Yadav

1BM22CS009

