

22/12/23

Lab program - 1

→ Develop a Java program that prints all real solutions to the quadratic equation $an^2 + bn + c = 0$. 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.

Sol.

```
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 non-zero value for a");
            Scanner s = new Scanner(System.in);
            a = s.nextInt();
        }
        d = b * b - 4 * a * c;
        if (d < 0) {
            System.out.println("No real solutions");
        } else if (d == 0) {
            r1 = -b / (2 * a);
            System.out.println("One real solution: " + r1);
        } else {
            r1 = (-b + Math.sqrt(d)) / (2 * a);
            r2 = (-b - Math.sqrt(d)) / (2 * a);
            System.out.println("Two real solutions: " + r1 + " and " + r2);
        }
    }
}
```

$$d = b * b - 4 * a * c;$$

if (d == 0) {

$$x_1 = (-b) / (2 * a);$$

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

System.out.println("Root1 = Root2 = " + x1);

} else if (d > 0) {

$$x_1 = ((-b) + (\text{math.sqrt}(d))) / (2 * a);$$

$$x_2 = ((-b) - (\text{math.sqrt}(d))) / (2 * a);$$

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

System.out.println("Root1 = " + x1 + " Root2 = " + x2);

} else if (d < 0) {

System.out.println("There are no real
solutions");

Class Quadratic main() {

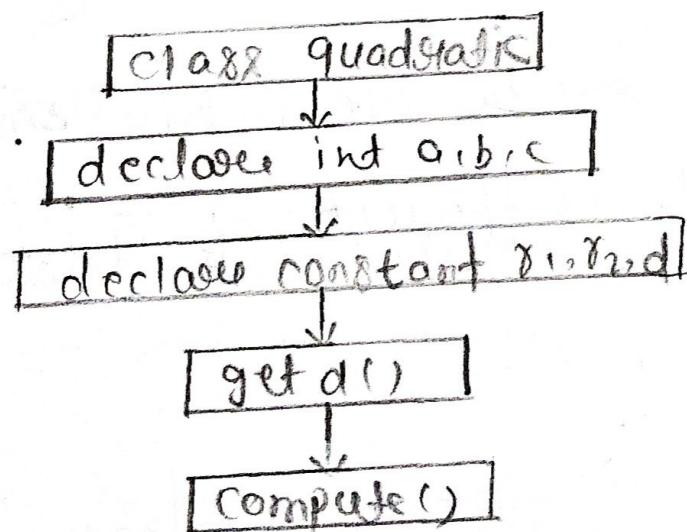
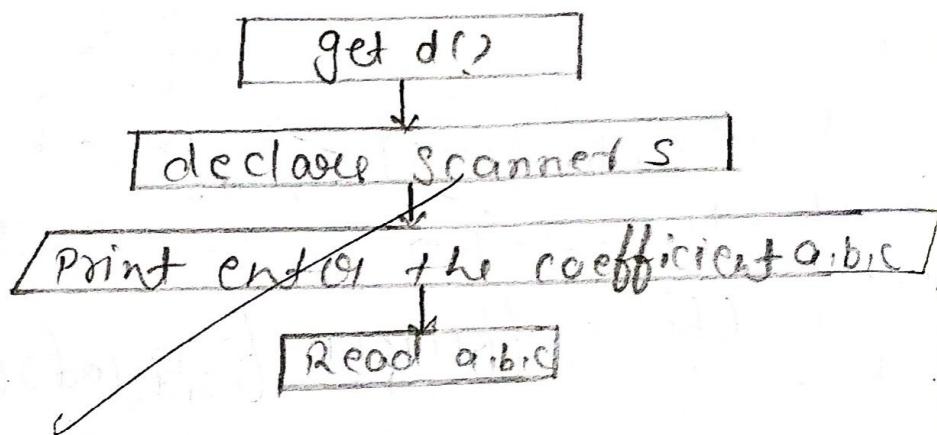
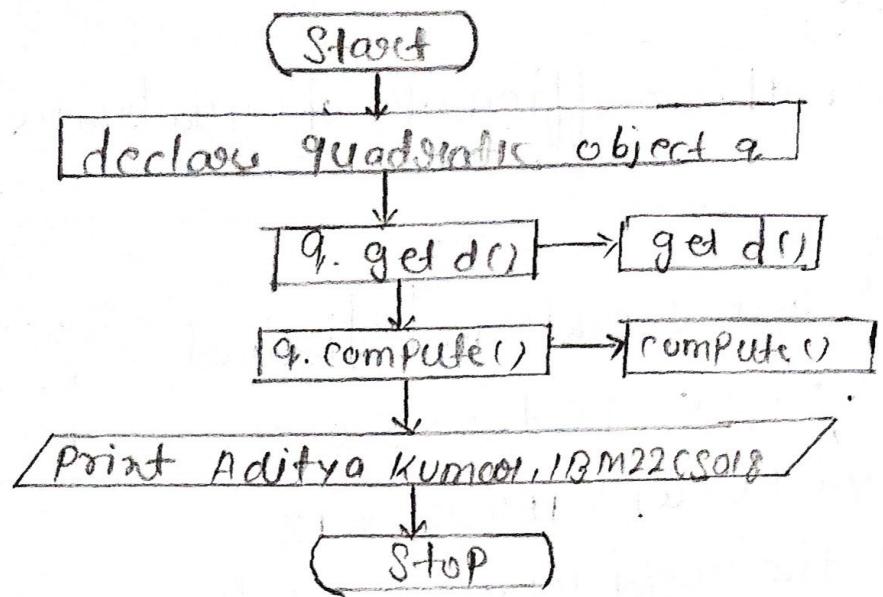
Public static void main (String args[]) {

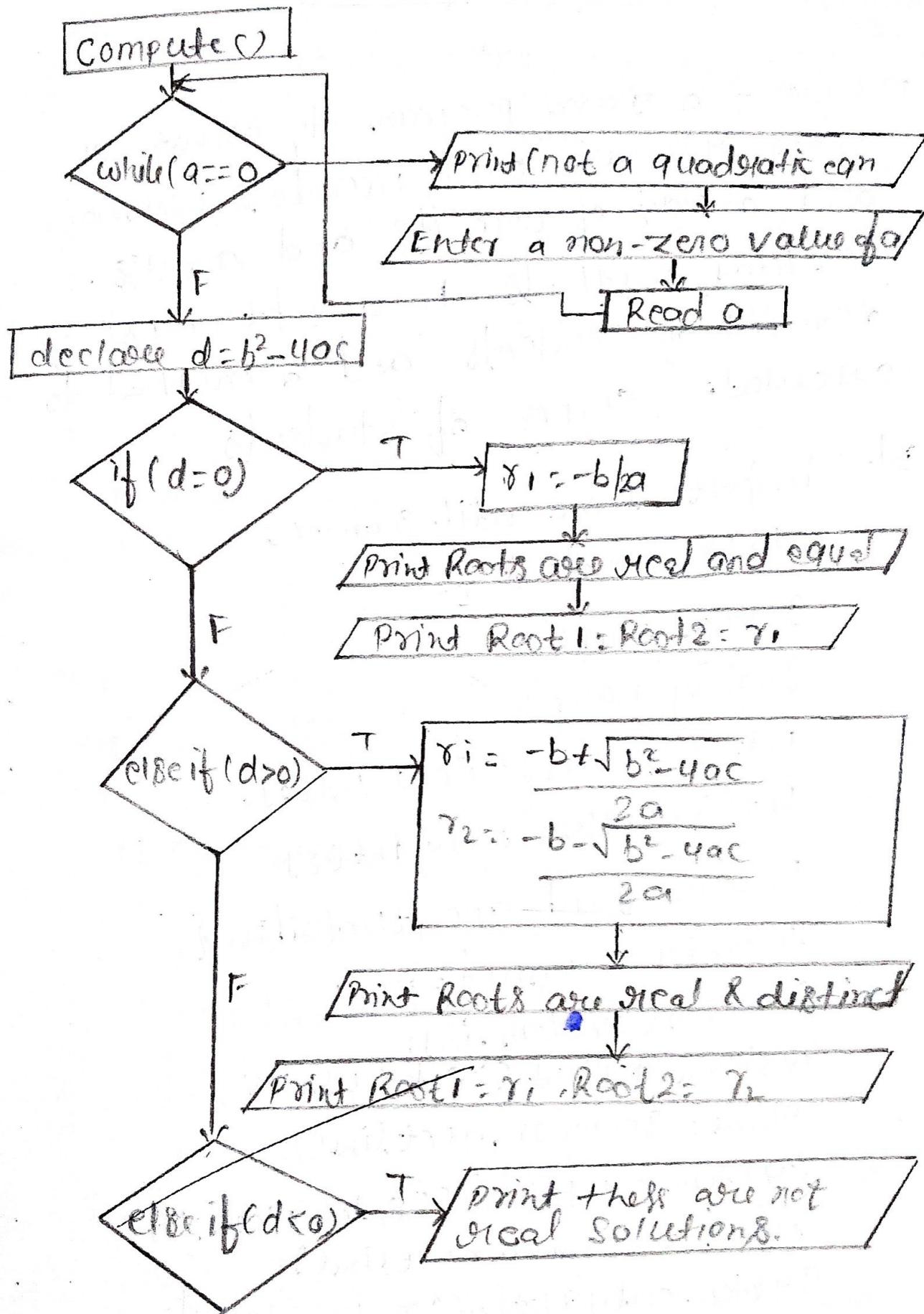
Quadratic q = new Quadratic();

q.getd();

q.compute();

System.out.println("Aditya Kumar", 13M2001);





C:\Windows\System32\cmd.e X + v

C:\Users\singh\Downloads\00J>java Quadratic
ADITYA KUMAR
1BM22CS018

enter the value of a
2
enter the value of b
4
enter the value of c
5
roots are imaginary

C:\Users\singh\Downloads\00J>javac Quadratic.java

C:\Users\singh\Downloads\00J>java Quadratic
ADITYA KUMAR
1BM22CS018

enter the value of a
1
enter the value of b
6
enter the value of c
4
roots are real and unequal
roots are:-0.7639320225002102,-5.23606797749979

20/12/23

Lab program-2

⇒ Develop a Java program to create a class Student with members USN, name and array of credits and marks. It includes methods to accept, display details of students and a method to calculate CGPA of students.

```
Set. import java.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:");  
        name = scanner.nextLine();  
        System.out.print("Enter name:");  
        name = scanner.nextLine();  
        System.out.print("Enter details for  
        subject: ");  
        for (int i=0; i<credits.length; i++) {  
            System.out.println("Enter credits for  
            subject " + (i+1) + ": ");
```

```
credits[i] = scanner.nextInt();
System.out.print("Enter marks for
subject " + (i + 1) + ": ");
marks[i] = scanner.nextInt();
}
Scanner.close();
}

public double calculate GPA() {
    int totalcredits = 0;
    int weightedsum = 0;
    double avg;
    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];
    }
    avg = (double) weightedsum / (double) totalcredits;
    return avg;
}

public void displayDetails() {
    System.out.println("Student Details:");
}
```

```
4 &tem. out. println(" USN:" + USN );
```

```
4 &tem. out. println(" Name:" + name);
```

3

```
public class Sgpa {
```

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

```
Scanners Scanner = new Scanners();
```

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

```
student . acceptDetails();
```

```
student . displayDetails();
```

```
double Sgpa = student . calculateSGPA();
```

```
4 &tem. out. println("In SGPA:" + Sgpa);
```

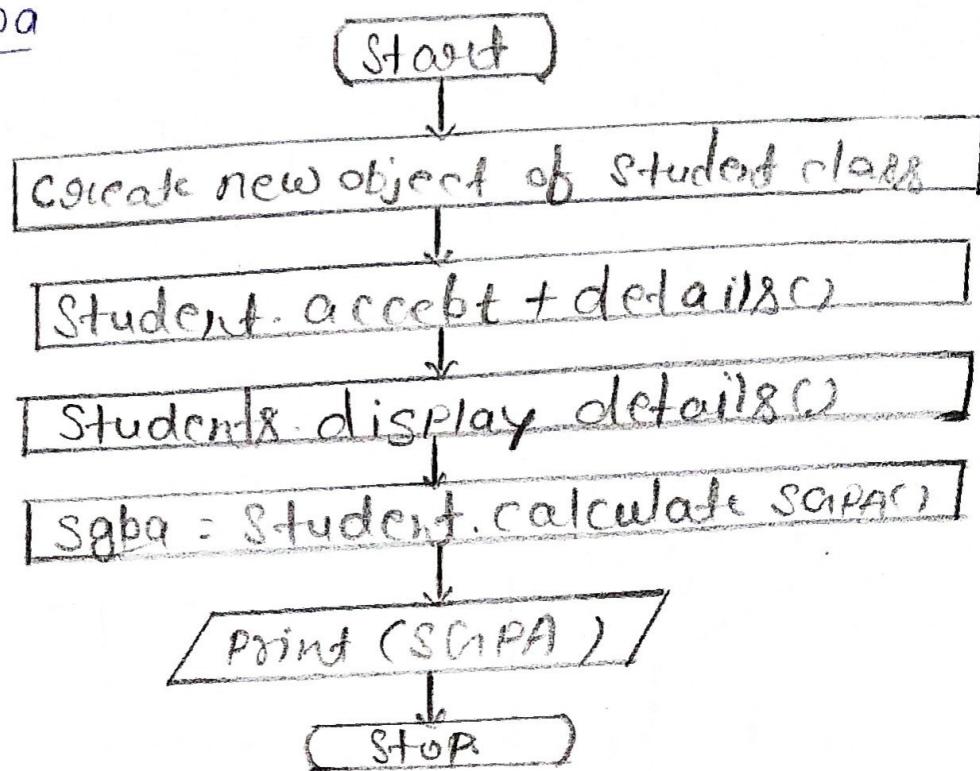
```
Scanners . close();
```

Output:

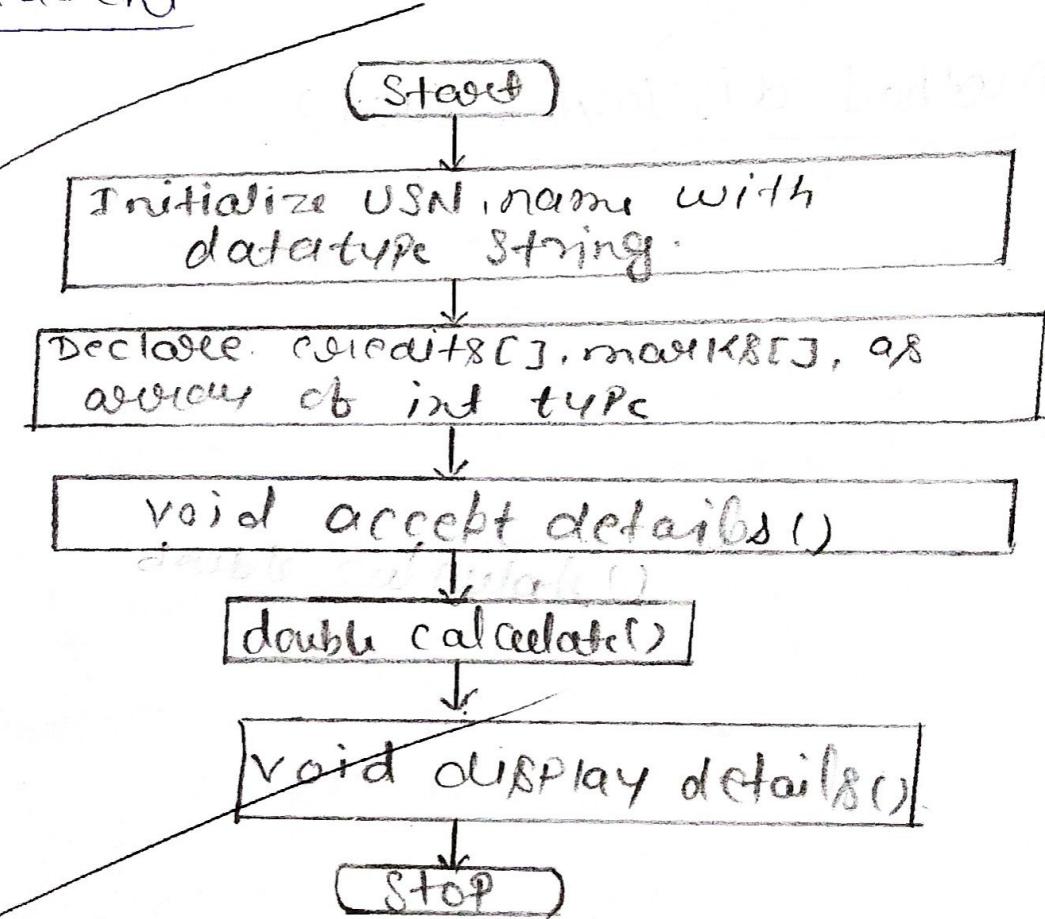
- 123456

flowchart

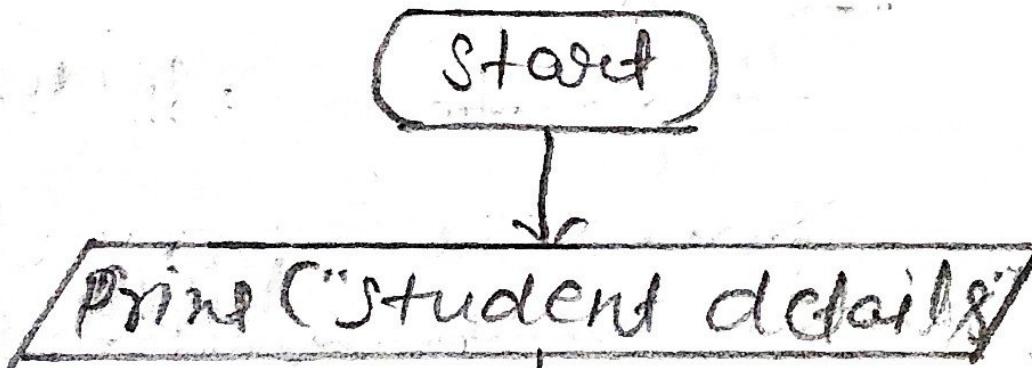
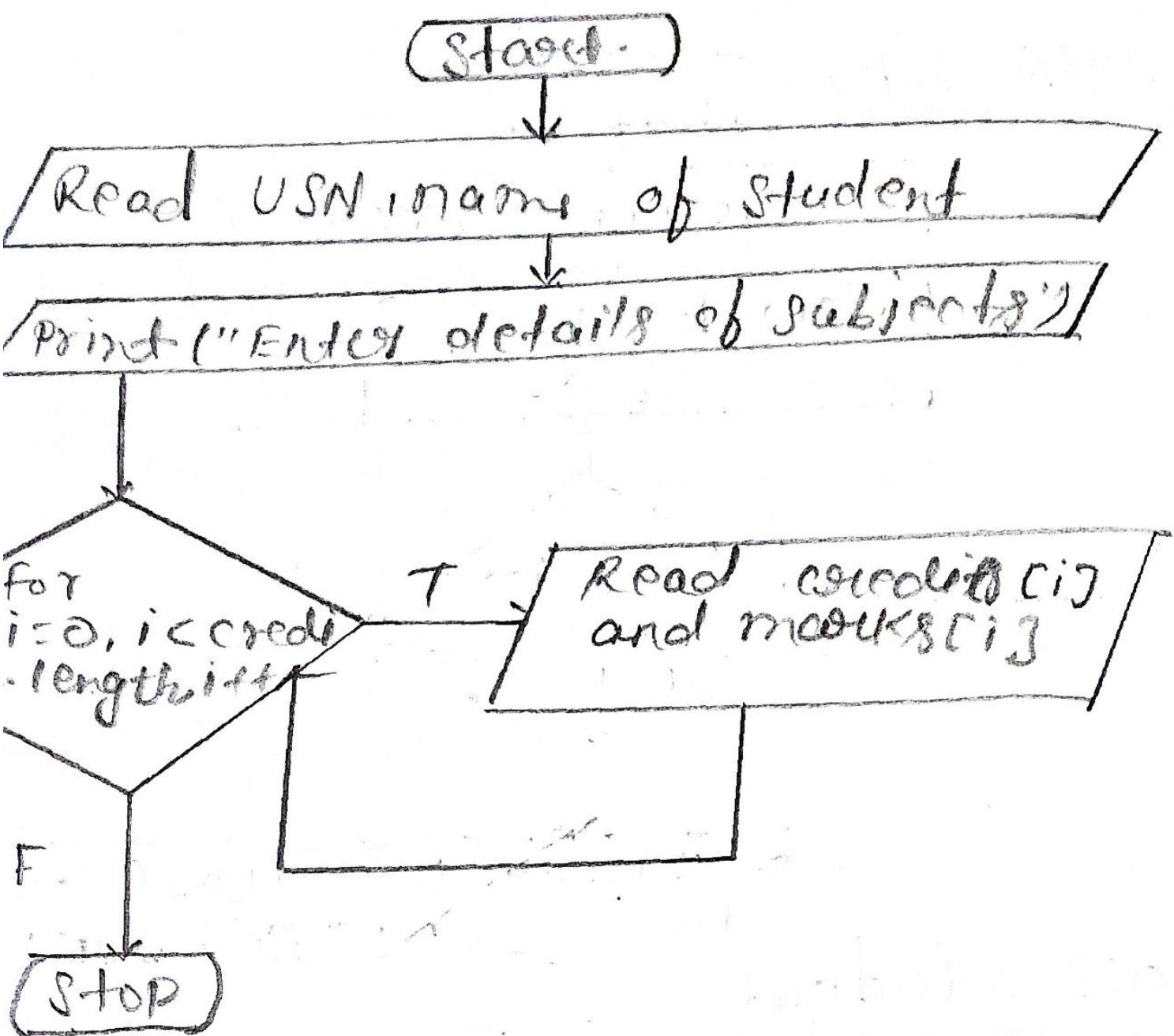
class Sgpa



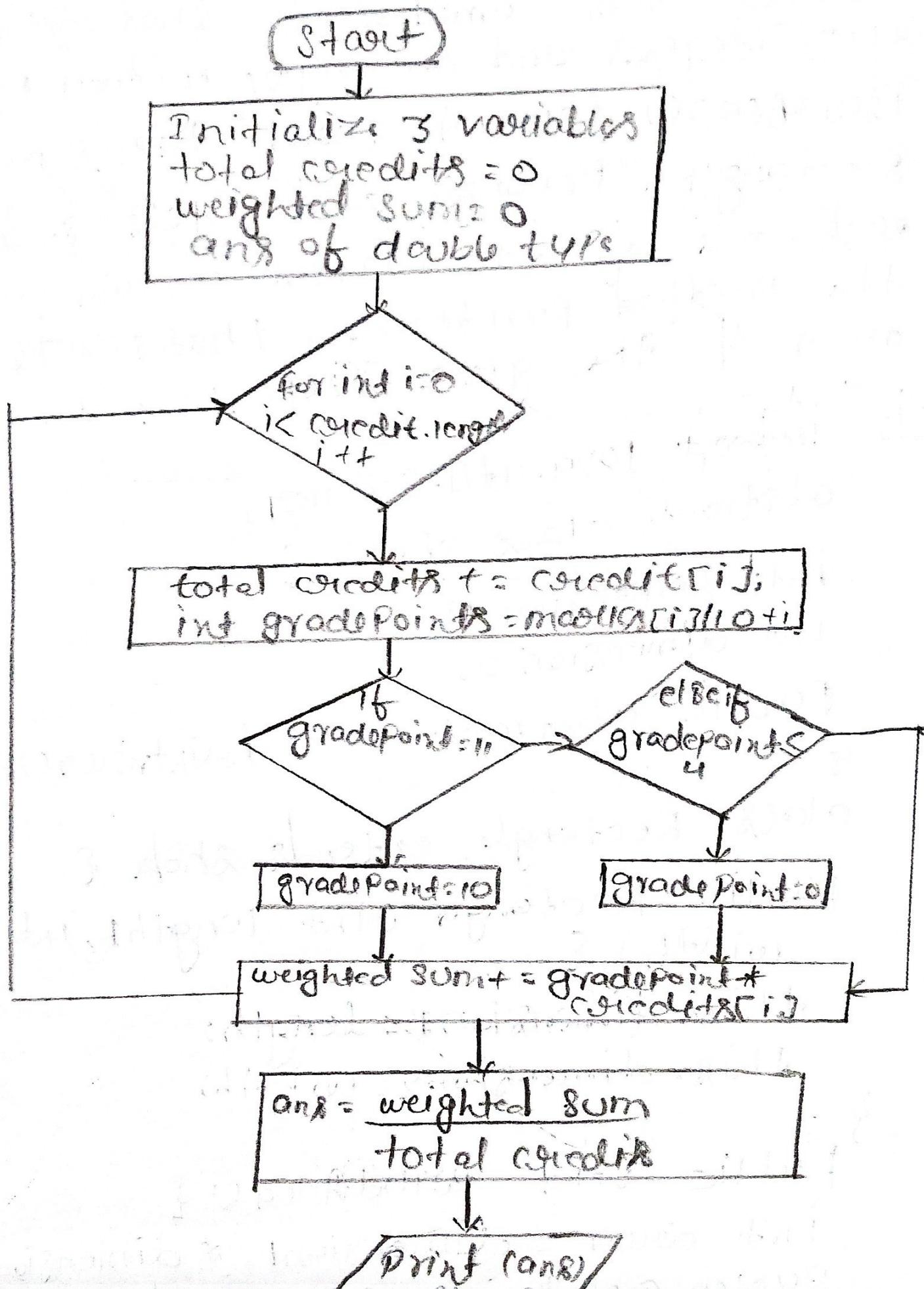
class Student



had acceptability()



method calculateGPA()



Enter USN: 1BM22CS018
Enter Name: ADITYA KUMAR
Enter details for each subject :

Enter credits for Subject 1: 4

Enter marks for Subject 1: 98

Enter credits for Subject 2: 4

Enter marks for Subject 2: 87

Enter credits for Subject 3: 4

Enter marks for Subject 3: 85

Enter credits for Subject 4: 3

Enter marks for Subject 4: 82

Enter credits for Subject 5: 3

Enter marks for Subject 5: 90

Enter credits for Subject 6: 3

Enter marks for Subject 6: 79

Enter credits for Subject 7: 1

Enter marks for Subject 7: 90

Enter credits for Subject 8: 1

Enter marks for Subject 8: 87

Student Details :

USN : 1BM22CS018

Name : ADITYA KUMAR

SGPA: 9.217391304347826

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

12/1/24

Sol:

```
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("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();

}

Output:

Choose a shape to calculate area:

1. Rectangle
2. Triangle
3. Circle
4. exit

Enter choice:

2

Enter base of triangle: 4

Enter height of triangle: 5

Area of triangle: 10.0

variables
dimension1, dimension2
of int type

↓
Create a method void
PrintArea()

↓
Stop.

Inherited class Rectangle

(Start)

↓
Public Rectangle (int length, int width)

{ this. dimension1 = length;

this. dimension2 = width; }

↓
Public void PrintArea()

↓
int Area() = dimension1 * dimension2

↓
Print (Area)

↓
(Stop)

Inherited class

(Start)

public circle (int radius)

{ this.dimension1 = radius }

public void Print Area

int area = Math.PI * dimension1 * dimension1

Point area

(Stop)

Inherited class Triangle

(Start)

public Triangle (int base, int height)

{ this.dimension1 = base, this.dimension2 = height }

public void PrintArea()

~~int area = 0.5 * dimension1 * dimension2~~

Print Area

(Stop)

C:\Windows\System32\cmd.e X +

Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\singh\OneDrive\Desktop\jav1>javac areas.java

C:\Users\singh\OneDrive\Desktop\jav1>java areas

Choose a shape To Calculate Area :

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

Enter Choice :

2

Enter base of Triangle: 24

Enter height of Triangle: 12

Area of Triangle: 144.0

Enter Choice :

1

Enter length of Rectangle: 12

Enter width of Rectangle: 14

Area of Rectangle: 168.0

Enter Choice :

3

Enter radius of Circle: 21

Area of Circle: 1385.442360233099

Program 4

13/11/24

Create a class 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 object. Include `toString()` method of the book. Develop a Java program to create 5 book objects.

```
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: " + num-
pages + " 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
 OK 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 book" + i);
 for (int j = 0; j < i; j++)
```

{

```
 system.out.println("In Book" + (i + 1) + "
 " + h + book[i].toString());
```

{

```
 s.close();
```

{

{

Output :-

Enter no. of books : 1

Enter details for book,

Name : Nuclear

Author : Agam

Price : 101

Number of Pages : 500

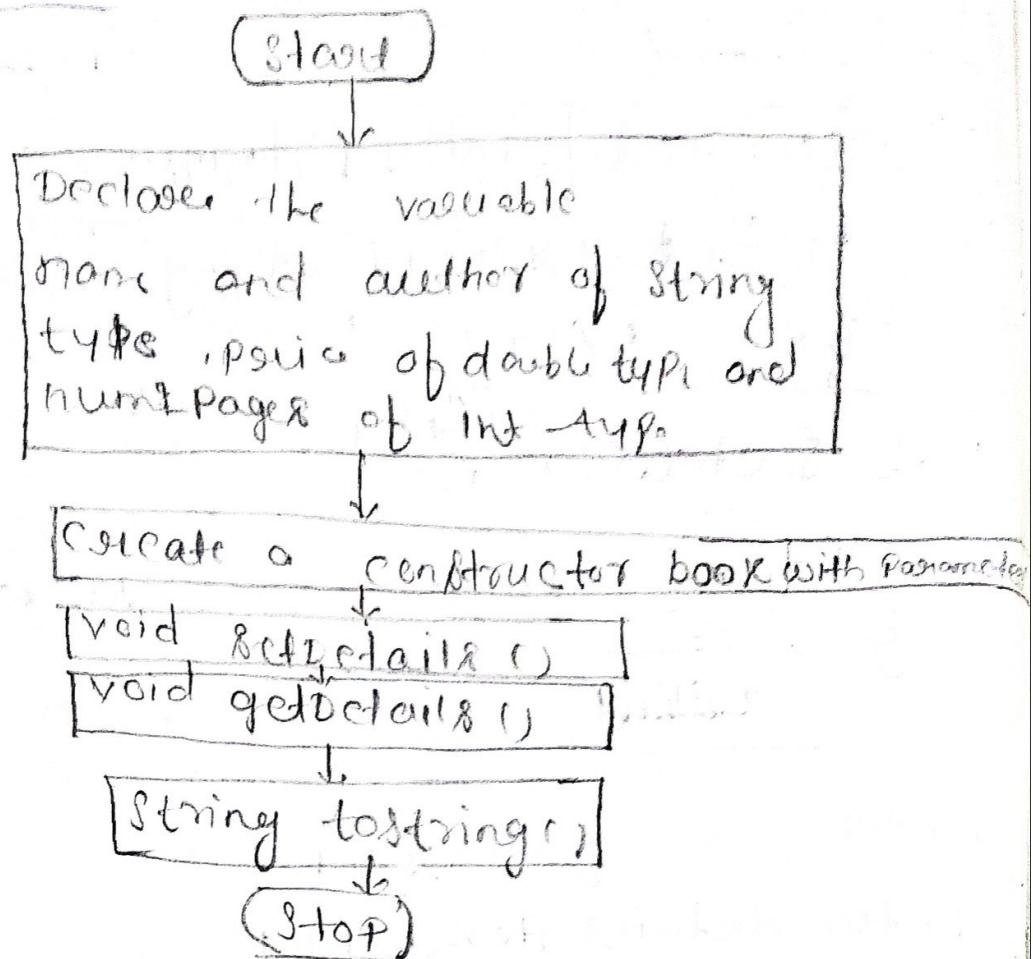
~~Details of book:~~

~~Name : Nuclear , Author : Agam , Price : 101 ,~~

~~Number of Pages : 500 .~~

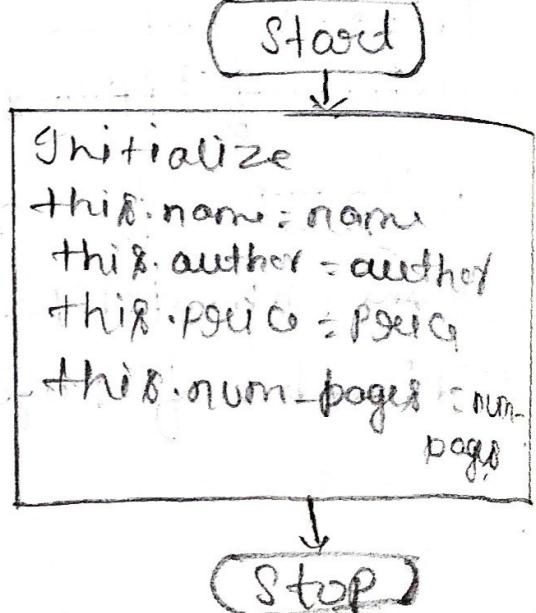
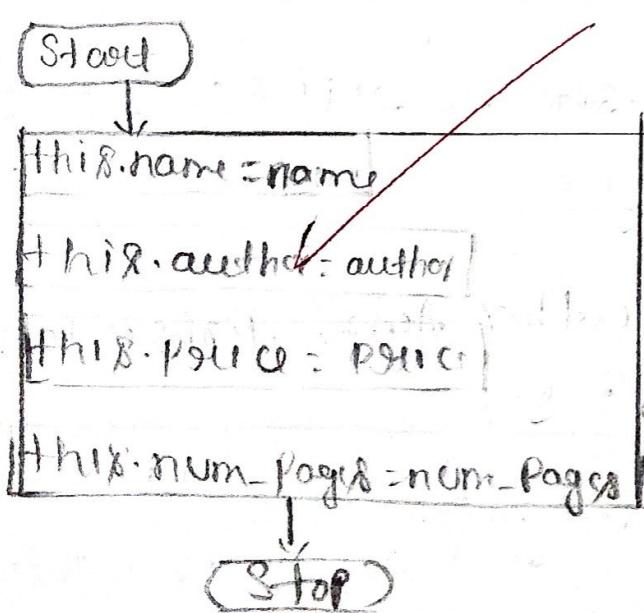
## Flowchart:

### Class Book



### Constructor book

### method setDetails



## Method CreateDetails(.)

## method to string

**Start**

**Start**

String S = "Book :" + name +  
"by author :" + author +  
"with Pages :" + numPages +  
"is of price :" + price;

String S = "Book :" + name +  
"by author :" + author +  
"with Pages :" + num-  
Pages + " is of :" +  
price

**Print(S)**

**Print(S)**

**Stop**

**Stop**

## class main

**Start**

**Create Scanner S**

~~Point : Enter the no. of books to create~~

~~Read No. of books~~

**Books [ ] books = new Books [n]**

**for index = 0; index < n; index++**

**yes**

**Print Enter details of book/**

**Read name, author, Price,  
number of pages**

Calling constructor

book[i] = new Book

(Name, author, price)

details of book are

2115

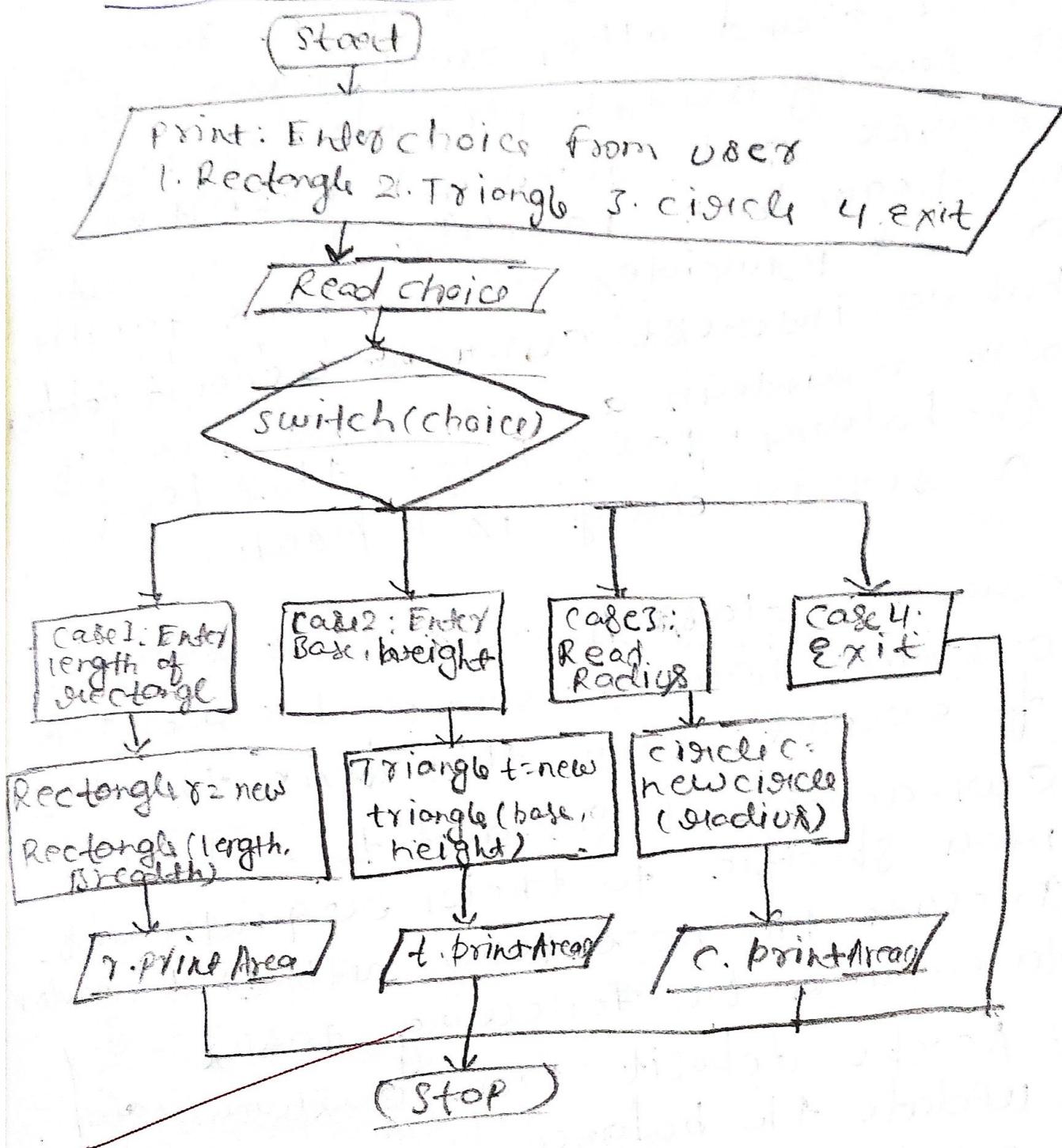
Yes  Paint: Book (i). today

三

# Lab Program #4

## Flowchart

### class main



Mr. Salolay  
Obj Seer

C:\Windows\System32\cmd.e X

+ ▾

C:\Users\singh\Downloads\00J>javac Main.java

C:\Users\singh\Downloads\00J>java Main

ADITYA KUMAR

1BM22CS018

Enter the number of books to create: 2

Enter details for Book 1:

Name: Concept of Physics

Author: HC Verma

Price: 345

Number of Pages: 600

Enter details for Book 2:

Name: Mathematical Foundation

Author: RS Aggarwal

Price: 125

Number of Pages: 235

Details of the Books:

Book 1:

Book : Concept of Physics by author : HC Verma with pages : 600 is of price : 345.0

Book 2:

Book : Mathematical Foundation by author : RS Aggarwal with pages : 235 is of price : 125.0

14/01/2024

Lab-Program

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

Creates a class Account that stores customer name, account number and type of account. From this derive the class Curr-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

- Accept deposit from customer and update the balance.
- Display the balance.
- compute and deposit interest
- Permit withdrawl and update the balance.

Check for the minm balance, impose penly

if necessary and update the balance

code

```
class Account {
```

```
 String customerName;
```

```
 long accountNumber;
```

```
 String accountType;
```

```
 double balance;
```

```
 public Account (String customerName, long
 accountNumber, String accountType, double
 balance) {
```

```
 this.customerName = customerName;
```

```
 this.accountNumber = accountNumber;
```

```
 this.accountType = accountType;
```

```
 this.balance = balance;
```

```
}
```

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

```
 balance += amount;
```

```
 System.out.println ("In Deposit of Rs." + amount +
```

~~"Successful. => Updated balance: Rs." + balance;~~

```
}
```

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

19/e    accountType + " Account (Account Number  
    > T    accountNumber + ") -> " + balance);  
    Bc    {  
    fc    a    public void depositInterest (double  
    Tt    if (accountType == "Savings") {  
    in    doubleInterest  
    n    double interest = balance \* (Rate / 100);  
    c    System.out.println ("In - Interest deposited  
    b    deposit (interest);  
    c    }  
    else {  
    c    System.out.println ("Interest is not  
    applicable for current Account.");  
    c    }  
    }  
    }

public void withdraw (double amount){  
    if (balance >= amount) {  
        balance -= amount;  
        System.out.println ("In withdrawal of R. " +  
                          amount + " successful. -> Updated balance R. " +  
                          + balance);  
    }  
}

```
else {
```

```
 System.out.println("Insufficient funds
withdrawal not allowed.");
```

```
}
```

```
{
```

```
}
```

```
class CurrAcct extends Account {
```

```
private double minBalance;
```

```
private double serviceCharge;
```

```
public CurrAcct (String customerName, long
accountNumber, double balance, double minBalance,
double serviceCharge) {
```

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

```
 this.serviceCharge = serviceCharge;
```

```
{
```

```
public void checkminBalance () {
```

~~balance = service~~

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

```
 balance += serviceCharge;
```

```
 System.out.println("In minimum Balance
equivalent not met. service charge of Rs. 11
```

191

→ T

B.

fi

q

T

ii

r

c

1

System.out.println("⇒ updated  
" + balance);

} else {

System.out.println("In minimum balance  
requirement is met. No service charge  
");

}

balance int chequenumber = 1001;

public void issuecheque(double amount)  
if (balance >= amount) {  
balance -= amount;

System.out.println("In Cheque of Rs." +  
+ " issued. updated balance => Rs." + balance);

System.out.println("In Cheque Number " +  
chequenumber++);

} else {

System.out.println("In Withdrawal is not  
possible");

class SavAcct extends Account {

    double interestRate;

    public SavAcct(String customerName,  
        long accountNumber, double balance, double  
        interestRate) {

        super(customerName, accountNumber,  
            " Savings", balance);

        this.interestRate = interestRate;

}

}

public class main {

    public static void main (String [] args)  
    {

        SavAcct savingsAccount = new SavAcct(  
            " John Doe ", 123456789, 5000, 5);

        CurtAcct currentAccount = new CurtAcct("

        Tom Smith ", 987654321, 7000, 1000, 20);

        System.out.println("In === Operations on Savings  
        Account. === In");

        savingsAccount.display();

        savingsAccount.deposit(2000);

        savingsAccount.depositInterest(5);

        savingsAccount.displayBalance();

        savingsAccount.withdraw(1000);

```
24/07/2014 11:00:00
savingsAccount. displayBalance();
current
currentAccount. deposit(1500);
currentAccount. checkMinBalance();
currentAccount. withdraw(800);
currentAccount. displayBalance();
currentAccount. checkMinBalance();
System.out.println("In cheque issuance for
current Account:");
currentAccount. issueCheque(300);
currentAccount. issueCheque(500);
currentAccount. displayBalance();
```

4

3  
output:

Enter customer name: Aditya Kumar

Enter account Number: 123456

Balance amount: L 100000/- (Rupees One Lakh)

```
public class Cart {
```

```
 private String itemName;
```

```
 private int price;
```

```
 private int quantity;
```

```
 public void setItemName (String itemName)
```

```
{
```

```
 this.itemName = itemName;
```

```
}
```

```
 public String getItemName () {
```

```
 return itemName;
```

```
}
```

```
 public void setPrice (int price) {
```

```
 this.price = price;
```

```
}
```

```
 public int getPrice () {
```

```
 return price;
```

```
}
```

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

```
 Cart obj = new Cart ();
```

```
 obj.setItemName ("Butter"); // Setting item name
```

```
 obj.setPrice (50); // Setting item quantity.
```

workout

class Account

(Start)

Declare

```
String customerName;
long accountNumber;
String accountType;
double balance;
```

↓

Create a parameterized constructor  
to initialize the variables.

↓

Create deposit (double amount)  
method of type void

Create displayBalance method  
of type void

↓

Create void withdraw (double amount)  
method

void checkMinBalance ()

void computeInterest ()  
method of public specifies

(Stop)

Start

Declare two variables

min balance, servicecharge of type

Construct constructor to initialize  
the variable

Super (customerName, accountNumber,  
savings, balance)

void checkminimumBalance

Stop

Inherited class SonAcct

Start

Declare of a variable of interest  
of type double

Construct sonAcct to declare  
variable  
this. interestRate = interestRate

Super (customerName, accountNumber,  
"savings")

void computeInterest()

Stop



old deposit (Amount)

(Start)

{balance + amount}

Print "successful"

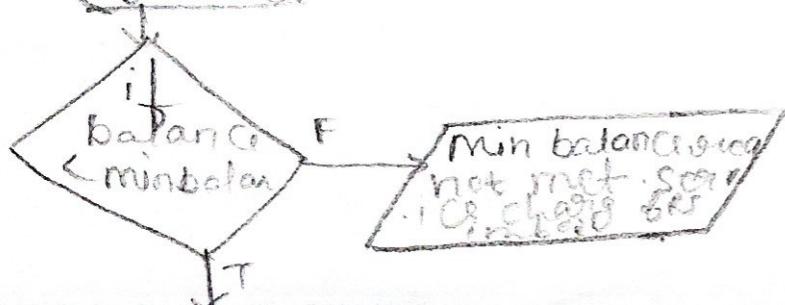
Print "updated balance", balance

checkMinBalance

(STOP)

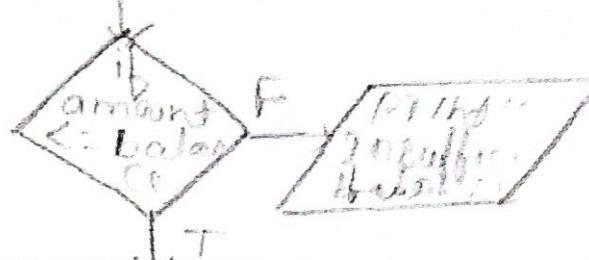
checkMinBalance

(Start)



2) void withdraw (amount)

(Start)



balance -= amount

Print "updated balance"

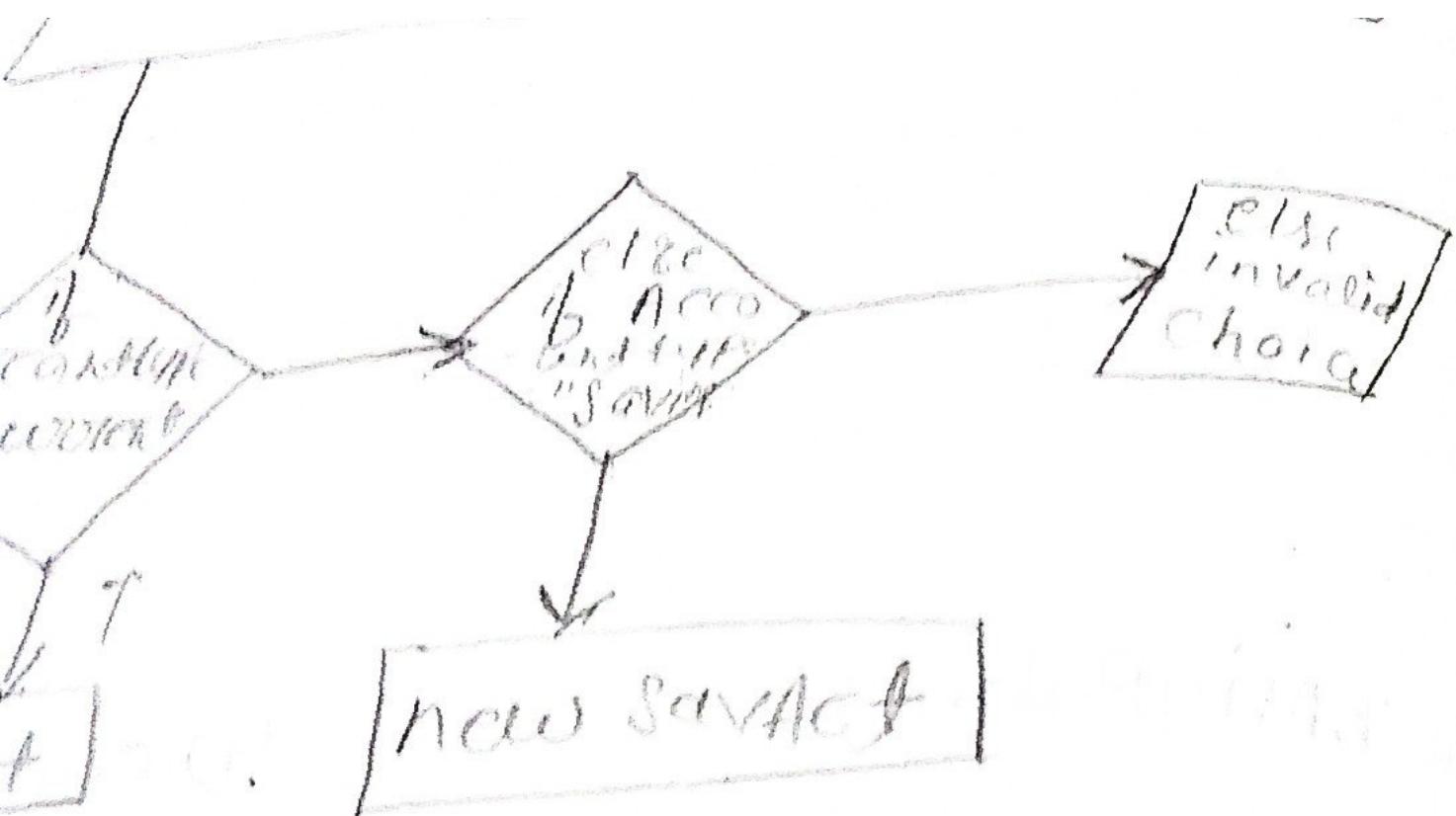
checkMinBalance

(STOP)

4) computeInterest()

(Start)

double interest  
= balance \* interest  
- 100



choose 1. deposit 2. withdraw  
play 4. interest Rate 5. exit



C:\Users\singh\OneDrive\Desktop\00J>javac Bank.java

C:\Users\singh\OneDrive\Desktop\00J>java Bank

Enter customer name: Aditya Kumar

Enter account number: 8765432

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: 100000

Deposit successful. Updated balance: 100000.0

Choose an option:

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

Enter your choice: 3

Account Balance: 100000.0

Date: 2/2/24

Lab Program-6

Q1. Create a package CIE which has two classes - Student and Internship. The class ~~person~~ <sup>Student</sup> has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SRF which is derived class of student. This class has an array that stores the SRF marks scored in five courses of the current sem. of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

```
Q1. 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);
```

10

```
System.out.println("Enter USN, name
sem id:");
USN = Scan.nextLine();
name = Scan.nextLine();
SEM = Scan.nextLine();
```

11

```
}
package CIE;
public class internal
{ public int intIn[] = new int[5];
```

```
package SEE;
import CIE.Student;
public class EXternal extends SEE
{ public int sem[] = new int[5];
import java.util.*;
import SEE.*;
import CIE.*;
public class FindMarks;
{
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.int[n];
for (int i = 0; i < n; i++) {
st[i] = new SEE.External();
s[i] = new CIE.internal();
st[i].semIn = new int[5];
s[i].semIn = new int[5];
for (int j = 0; j < 5; j++) {
st[i].semIn[j] = sc.nextInt();
s[i].semIn[j] = sc.nextInt();
}
}
for (int i = 0; i < n; i++) {
for (int j = 0; j < 5; j++) {
fm[i] = fm[i] + (st[i].semIn[j] * s[i].semIn[j]);
}
}
for (int i = 0; i < n; i++) {
System.out.println("Sum of marks for student " + i + " is " + fm[i]);
}
}}
}
```

## Algorithm :-

Step 1: Start

Step 2: Create a package CIE

Step 3: import java.util.\*

Step 4:

Step 5: Create a class student with three public variable and a public method accept

public int sem;

public String USN;

public String name;

Step 6: Public void accept();

{ Read sem;

Read USN

Read name }

Step 7: Create another public class Internals with a public array variable

Ex

Step 8: Create a package SIE;

Step 9: import CIE.student

Step 10: Create an inherited class Student from External

Step 11: import java.util.\*;

Step 12: import SIE.\*

Step 13: import CIE.\*;

1) value from 0 to 100  
2) instances of each  
cate. (class as Student &  
Employee)

create a for loop from  
0 to 5 and increment  
of details of (i+1) Student  
(i).accept();

2) (initialise 5) for

Read internal marks and  
external marks. 98

(i).im[i] and st[i].sm[i];

sum - Add the marks of student  
in final marks.

3. exed.

Name:Aditya Kumar

USN:1BM22CS018

Enter n:

1

Enter details 1

Enter USN, Name, Sem:

1BM18

Aditya

3

Enter internal and external of sub 1

32

43

Enter internal and external of sub 2

32

34

Enter internal and external of sub 3

45

45

Enter internal and external of sub 4

43

43

Enter internal and external of sub 5

34

21

Final marks of Aditya

Course 1 = 75

Course 2 = 66

Course 3 = 90

Course 4 = 86

Course 5 = 55

16/2/24

## Program 7:

Write a program that demonstrates handling of exceptions in inheritance. 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 takes both Father and Son's age and throws an exception if Son's age is  $\geq$  Father's age.

$\Rightarrow$  class WrongAge extends Exception  
public WrongAge (String message) {  
super (message);

class Father {  
private int age;  
~~public~~ Father (int age) throws WrongAge  
{ if (age < 0) {

throw new WrongAge ("Age can't  
be negative");

this.age = age;

public int getAge () {  
return age;

class Son extends Father {  
 private int sonAge;

public Son(int fatherAge, int  
 theAge) {  
 super(fatherAge);  
 if (sonAge >= fatherAge) {  
 throw new WrongAge("Son's  
 can't be greater than or equal  
 father's age");  
 }  
 this.sonAge = sonAge;  
 }

public int getSonAge() {

return sonAge;

public class ExceptionInheritance  
{  
 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 the Program.

Step 2: Create the class WrongAge which extends exception.

Step 3: Create a public class WrongAge

Step 4: Declare an age variable

Step 5: Pass the message function in super.

Step 6: If age is less than zero then print Father's age can't be negative

Step 7: If son's age is greater than or

Step8: create a public class

exceptionInheritanceDemo

10

➤

Step9: create a try and catch  
method and in this send  
father's age as 45 and son's  
age ~~10~~ 10.

- Step10: print father's age and  
son's age by using get()  
method.

Step11: In catch print throw  
exception e if any error is  
incountered.

Step12: print exception caught.

Step13: End

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

→ class DisplayMessage extends Thread  
String message;  
int interval;

```
public void displayMessage (String message ,
 int interval) {
```

this.message = message;

this.interval = intervals;

3

Public void sun() {

White (tree) §

tony 8

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

~~Thread.sleep(interval \* 1000);~~

~~3~~ catch (InterruptedException e) {

e. PrintStackTrace();

45° 110°

3/17/2

list of classes they can learn

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

Display message thread i = new

DisplayMessage ("BMS College of Engineering")

10  
- age("CSE", 2);  
-> thread1.start();  
thread2.start();  
4

7 } output:

BMS college of Engineering

CSE

CSE

CSE

CSE

### Algorithm:

Step1: Start the program.

Step2: Create a class thread which is extended by a class display.

Step3: Initial declare variable of String type, interval of int type.

Step4: Create a public class display, under it pass arguments message, interval.

Step5: Declare type variable message of interval message, interval of type.

Step6: Create a public class run() &

void type

Step 7: Create a while loop and set i is of  
type in all cases

Step 8: Pass a try and catch function

Step 9: Print the message

Step 10: Set the sleep time to 1000

Step 11: If any error is caught print  
InterruptedException exception

Step 12: Create a public class Thread1

Step 13: Print BMS college Engineering  
under DisplayMessage thread1

Step 14: Print CSE under DisplayMessage  
thread2.

Step 15: Pass thread1.start();

Step 16: Pass thread2.start();

Step 17: end

16/02/24

C:\Windows\System32\cmd.e X + ▾

C:\Users\singh\Downloads\00J>javac ExceptionInheritanceDemo.java

C:\Users\singh\Downloads\00J>java ExceptionInheritanceDemo

Father's age: 45

Son's age: 40

C:\Users\singh\Downloads\00J>javac ExceptionInheritanceDemo.java

C:\Users\singh\Downloads\00J>java ExceptionInheritanceDemo

Exception caught: Son's age cannot be greater than or equal to father's age

C:\Users\singh\Downloads\00J>javac ExceptionInheritanceDemo.java

C:\Users\singh\Downloads\00J>java ExceptionInheritanceDemo

Exception caught: Age cannot be negative

C:\Users\singh\Downloads\00J>

```
C:\Windows\System32\cmd.e X + ^

Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\singh\Downloads\00J>javac threadDemo.java

C:\Users\singh\Downloads\00J>java threadDemo
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
CSE
CSE
```

## Lab Program 9

Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the 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 ("Divide App");
 jfrm.setSize (275, 150);
 jfrm.setLayout (new FlowLayout ());
 jfrm.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
 JLabel jlab = new JLabel ("Enter the divisor and dividend");
 JTextField aif = new JTextField (8);
```

```
jtextfield bjtf = new JTextField(8);
button button = new JButton("calculate");
label err = new JLabel();
label alab = new JLabel();
label blab = new JLabel();
label anslab = new JLabel();
jfrm.add (err);
jfrm.add (ilab);
jfrm.add (ajtf);
jfrm.add (bjtf);
jfrm.add (button);
jfrm.add (alab);
jfrm.add (blab);
jfrm.add (anslab);

ActionListener I = new ActionListener() {
 public void actionPerformed(ActionEvent evt) {
 System.out.println("Action event from a
text field");
 }
};

ajtf.addActionListener(I);
bjtf.addActionListener(I);
button.addActionListener(new ActionListener() {
 public void actionPerformed(ActionEvent evt) {
 }
});
```

```
int a = Integer.parseInt(jTextField1.getText());
int b = Integer.parseInt(jTextField2.getText());
int 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!");
```

```
}
```

```
});
```

```
frame.setVisible(true);
```

```
public static void main (String args[])
 SwingUtilities.invokeLater (new Runnable () {
 public void run () {
```

```
new SwingDemo();
```

```
}
```

Output

Enter the divisor

and dividend

A : 12

B : 4

Ans : 3

C:\Windows\System32\cmd.e X +

C:\Users\singh\Downloads\00J>javac SwingDemo.java

C:\Users\singh\Downloads\00J>java SwingDemo

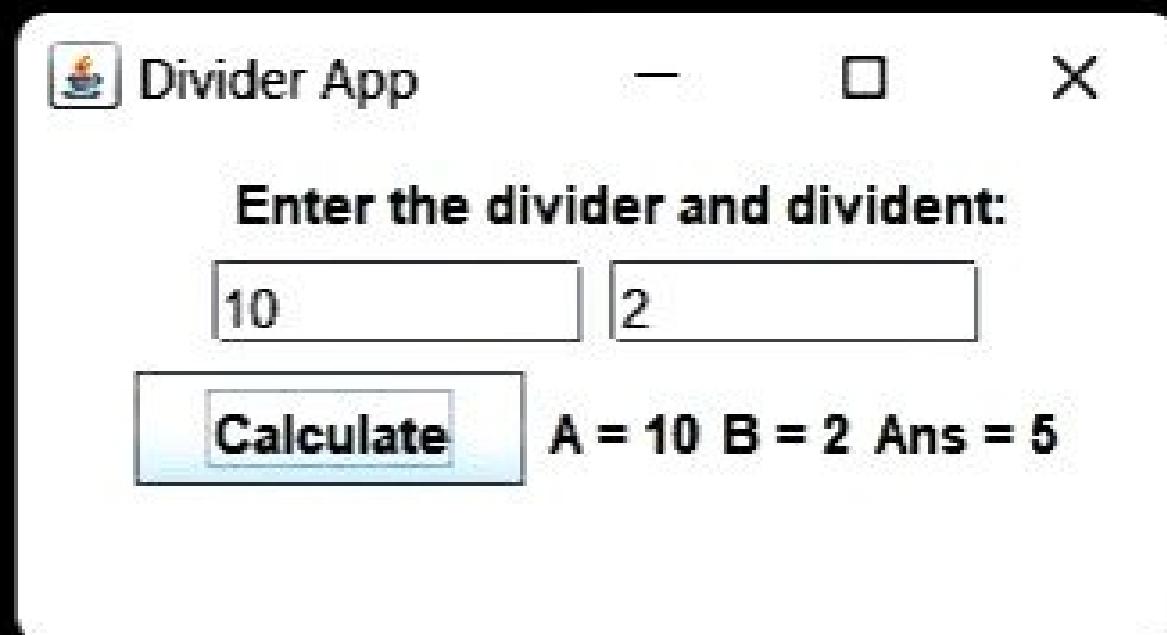
ADITYA KUMAR  
1BM22CS018

Divider App

Enter the divider and divident:

10 2

**Calculate** A = 10 B = 2 Ans = 5



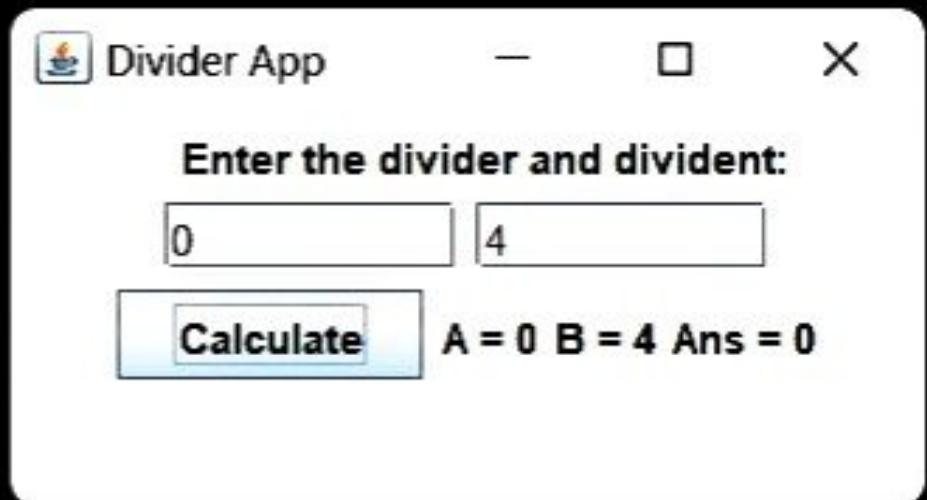
```
C:\Windows\System32\cmd.e X + ^
```

```
C:\Users\singh\Downloads\00J>javac SwingDemo.j
```

```
C:\Users\singh\Downloads\00J>java SwingDemo
ADITYA KUMAR
1BM22CS018
```

```
C:\Users\singh\Downloads\00J>javac SwingDemo.j
```

```
C:\Users\singh\Downloads\00J>java SwingDemo
ADITYA KUMAR
1BM22CS018
```



C:\Windows\System32\cmd.e X + -

C:\Users\singh\Downloads\00J>javac SwingDemo.java

C:\Users\singh\Downloads\00J>java SwingDemo  
ADITYA KUMAR  
1BM22CS018

C:\Users\singh\Downloads\00J>javac SwingDemo.java

C:\Users\singh\Downloads\00J>java SwingDemo  
ADITYA KUMAR  
1BM22CS018

C:\Users\singh\Downloads\00J>javac SwingDemo.java

C:\Users\singh\Downloads\00J>java SwingDemo  
ADITYA KUMAR  
1BM22CS018

