Date: / /

## Program - 1

Develop a Java Program that prints all real solutions to the quadratic eqn $ax^2+bx+c=0$. Read in 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 solution.

### Algorithm

Step 1 : Start

Step 2 : Initialise valiable a, b, c, d & read a, b, c

Step 3 : If (a = 0) Print "invalid input" & goto step 8.

Step 4 : d = b*b - 4*a*c

Step 5 : if d > 0

Print "roots are real"

r1 = (-b + sqrt(d)) / (2*a)

r2 = (-b - sqrt(d)) / (2*a)

Print (r1, r2) goto step 8.

Step 6 : if d < 0

Print ("Roots are ~~real~~ Imaginary. There are no real solution")

r1 = -b / (2*a)

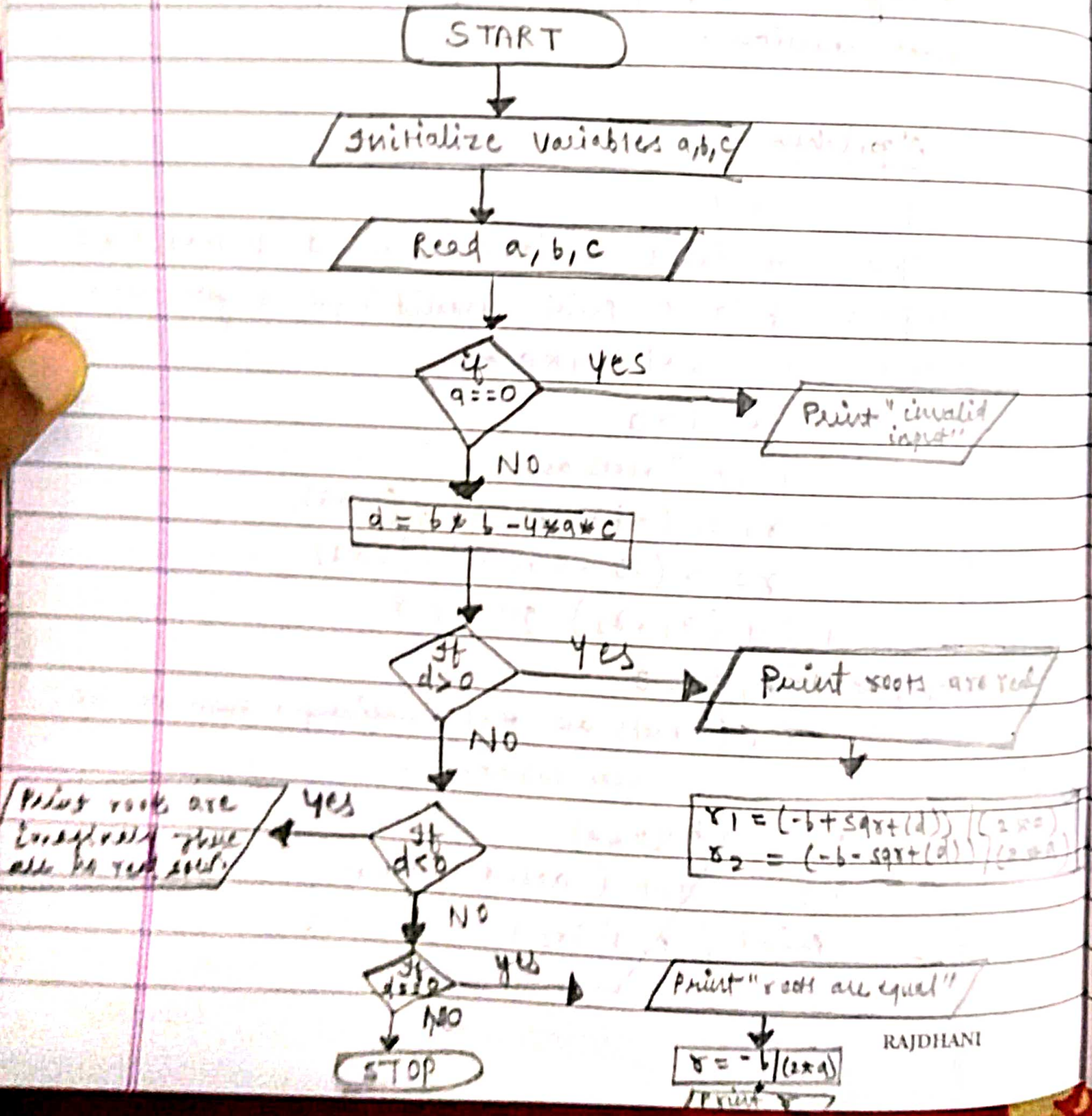
r2 = sqrt ( abs(d)) / (2*a)

Print ( r1 + ir2 )

Date: / /

Step 7 : If d = 0
Print " Roots are equal "
$r_1 = r_2 = -b / (2 * a)$
Print $r_1$

Step 8 : Stop




RAJDHANI

Date: / /

```
import java.util.Scanner;
class Quadratic
{
public static void main (String args[])
{
float a, b, c, d;
double x1, x2; // math.sqrt returns double
Scanner obj = new Scanner (System.in);
System.out.println ("enter the value of a");
a = obj.nextFloat ();
System.out.println ("enter the value of b");
b = obj.nextFloat();
System.out.println ("enter the value of c");
c = obj.nextFloat();
d = b*b - (4*a*c);
if (a==0) {
System.out.println ("input invalid");
  }
else {
if (d==0)
{
System.out.println ("roots are real & equal");
x1 = x2 = (-b)/(2*a);
System.out.println ("roots are :" +x1+ "," +x2);
  }
```

Date: / /

```
else if (d>0)
{
System.out.println("roots are real & unequal");
x1 = ((-b) + Math.sqrt(d))/(2*a);
x2 = ((-b) - Math.sqrt(d))/(2*a);
System.out.println("roots are : " + x1 + "," + x2);
}
else {
System.out.println("roots are imaginary");
}}}}
```

Output :- enter the value of a

1

enter the value of b

-8

enter the value of c

12

roots are real and unequal

roots are : 6.0 , 2.0

Date: / /

# Program - 2

Develop a Java Program to create Student with members USN, name, an array . credits and an array marks . Include methods to accept and display details and a method to calculate SGPA of a studnt .

Algorithm

→ step 1 : Start

step 2 : Initialize variable name, usn, marks [], credits [] , number

step 3 : Take input from the users for the above variables .

step 4 : Make a fn called grade in which all the grade points for certain marks is defined .

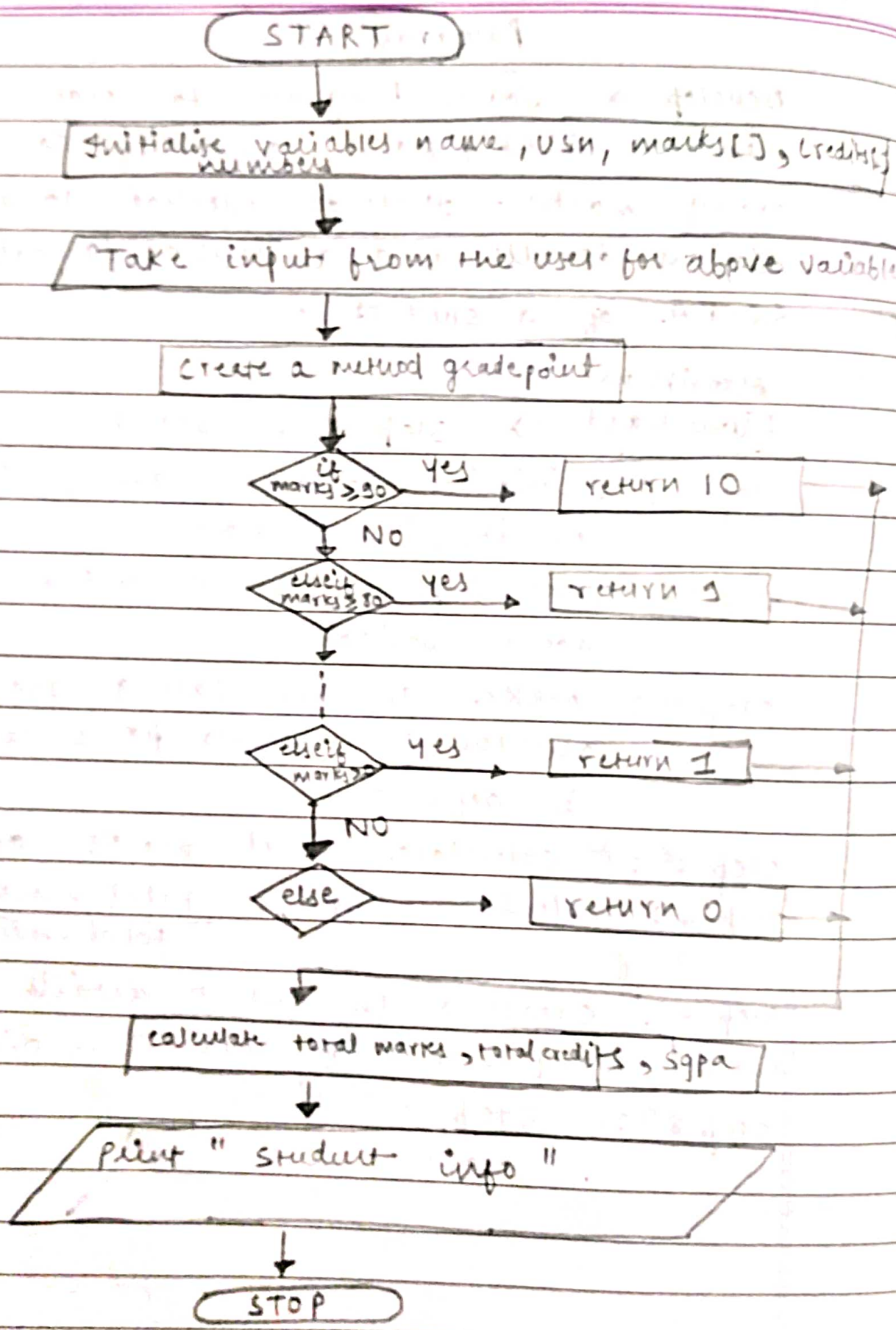
step 5 : calculate total marks and total credit

step 6 : calculate sgpa : $\frac{\text{total marks}}{\text{total credits}}$

step 7 : Create a fn student_details where information of student is displayed .

step 8 : Stop .

Date: / /



Date: / /

```
import java.util.Scanner ;
class Student
 {
String usn ;
String name ;
int[] marks ;
int[] credits ;
int number ;
float total_credits = 0 ;
float total_marks = 0 ;
float sgpa ;
Public void ~~fo~~ info ()
{
Scanner obj = new Scanner (System.in) ;
System.out.println (" enter name ");
name = obj.nextLine () ;
System.out.println (" enter USN ") ;
usn = obj.nextLine () ;
System.out.println (" enter number of subjects ");
number = obj.nextInt () ;
credits = new int [number] ;
 marks = new int [number] ;
System.out.println (" enter the subject marks and
                    its credit ") ;
for ( int i = 0 ; i < number ; i++)
```

Date: / /

```
{
  System.out.println("enter the marks of
                    subject" + (i+1) + ":");
marks[i] = obj.nextInt();
System.out.println("enter the credit of subj
                    +(i+1)+":");
credits[i] = obj.nextInt();
  }
  }
Public int grade (int marks)
{
if (marks >= 90)
{
  return 10;
}
else if (marks >=80) {
  return 9; }
elseif (marks >=70) {
  return 8; }
elseif (marks >=60) {
  return 7; }
else if (marks >=50) {
  return 6; }
elseif (marks >=40) {
  return 5;
  }
```

Date: / /

```
else if ( marks >= 30 ) {
    return 4; }
else if ( marks >= 20 ) {
    return 3; }
else if ( marks >= 10 ) {
    return 2; }
else if ( marks > 0 ) {
    return 1 ; }
else {
    return 0 ;
    }
Public void Calculate ( ) {
for ( int j = 0 ; j < number ; j++)
{
total_marks = total_marks + grade ( marks[j]) *
                                    credits [j];
total_credits = total_credits + credits [j] ;
}
sgpa = total_marks / total_credits ;
}
Public void student_details ( ) {
System.out.println ("details of the student");
System.out.println ("name :" + name);
System.out.println ("USN :" + usn);
for ( int k = 0 ; k < number ; k++) {
```

Date: / /

```
System.out.println("subject" +(k+1)+":
        credits =" +(credits[k])+", marks ="
                        +(marks[k]));
    }
System.out.println("sgpa:" +sgpa);
    }}

class Main
{
public static void main(String args[]){
Student obj2 = new Student();
obj2.info();
obj2.calculate();
obj2.student_details();
    }
    }
```

**output :**

```
enter name
shipra
enter USN
1BM22EC230
enter number of subjects
8
enter the subject marks and its credit
enter the marks of subject1:
97
```

Date: / /

enter the credit of subject 1 :
4
enter the marks of subject 2 :
91
enter the credit of subject 2 :
4
enter the marks of subject 3 :
96
enter the credit of subject 3 :
3
enter the marks of subject 4 :
90
enter the credit of subject 4 :
3
enter the marks of subject 5 :
94
enter the credits of subject 5 :
3
enter the marks of subject 6 :
99
enter the credits of subject 6 :
1
enter the marks of subject 7 :
84
enter the credit of subject 7 :
1

RAJDHANI

Date: / /

enter the marks of subject 8:
82
enter the credit of subject 8:
1
details of the student
name: shipra
usn: 1BM22EC230
subject 1: Credits = 4, marks = 97
subject 2: Credits = 4, marks = 91
subject 3: Credits = 3, marks = 96
subject 4: Credits = 3, marks = 90
subject 5: Credits = 3, marks = 94
subject 6: Credits = 1, marks = 99
subject 7: Credits = 1, marks = 84
subject 8: Credits = 1, marks = 82
Sgpa: 9.9

Date: / /

## Program - 3

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

**Algorithm:** step 1: start

step 2: Initialise variables name, author, price, num_pages, n.

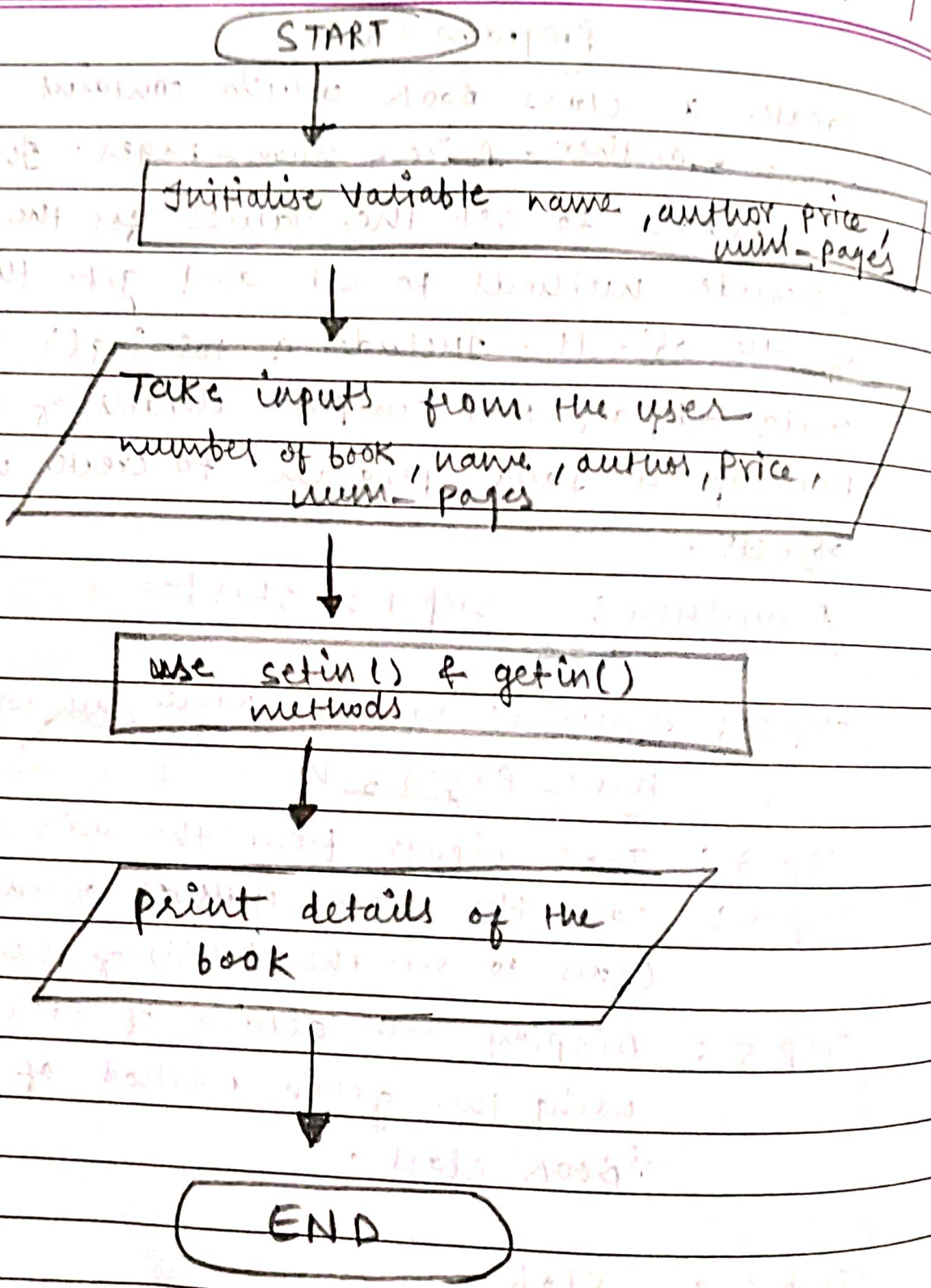
step 3: Take inputs from the user.

step 4: call the setin method of the "Book" class to set the details of the book.

step 5: Display the details of each book using the getin method of the 'Book' class.

step 6: stop

Date: / /


START
Initialise Variable name, author, price, num_pages
Take inputs from the user number of book, name, author, Price, num_pages
use setin() & getin() methods
print details of the book
END

Date: / /

```
import java.util.Scanner;
class Book {
  String name;
  String author;
  double Price;
  int num_Pages;
Public void set-in(int i)
{
  Scanner in = new Scanner(System.in);
  System.out.Println("Enter ~~daits~~ details of book "
                     + (i+1) + ":");
System.out.print("Name:");
  name = in.nextline();
System.out.print("Author:");
  author = in.nextLine();
System.out.Print("Price:");
  Price = in.nextDouble();
System.out.Print("Number of Pages:");
  num_Pages = in.nextInt();
  }
Public String get-in(int i){
   return "Details of book" + (i+1) + "\n" +
"Name:" + name + "\n" +
"Author:" + author + "\n" +
"Price: $" + Price + "\n" +
"Number of Pages:" + num_Pages;
}}
```

RAJDHANI

Date: / /

```
class Main {
public static void main(String args[]) {
  int n;
Scanner in = new Scanner(System.in);
System.out.println("Enter number of book:");
n = in.nextInt();
BOOK[] books = new BOOK[n];
for(int i=0; i<n; i++) {
    books[i] = new Book();
    books[i].setin(i);
  }
for(int i=0; i<n; i++) {
    System.out.println(books[i].getin(i));
  }}}
```

Output: Enter number of books:
1
Enter details of book 1:
Name : godan
Author : Premchandra
Price : 169
Number of pages : 412
Details of book 1
Name : godan
Author : Premchandra
Price : $169.0
Number of pages : 412

RAJDHANI

Date: / /

## Program - 4

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 extends the class shape. Each one of the classes contain only the method printArea() that prints the area of the given shape.

Algorithm : step 1 : start

step 2 : ~~Include 2 in~~ Create abstract class named shape.

step 3 : Declare abstract method area();

step 4 : create sub-class Rectangle that extends shape.

step 5 : Override area method to calculate area of rectangle.

step 6 : Repeat steps 4 and 5 for triangle and circle.

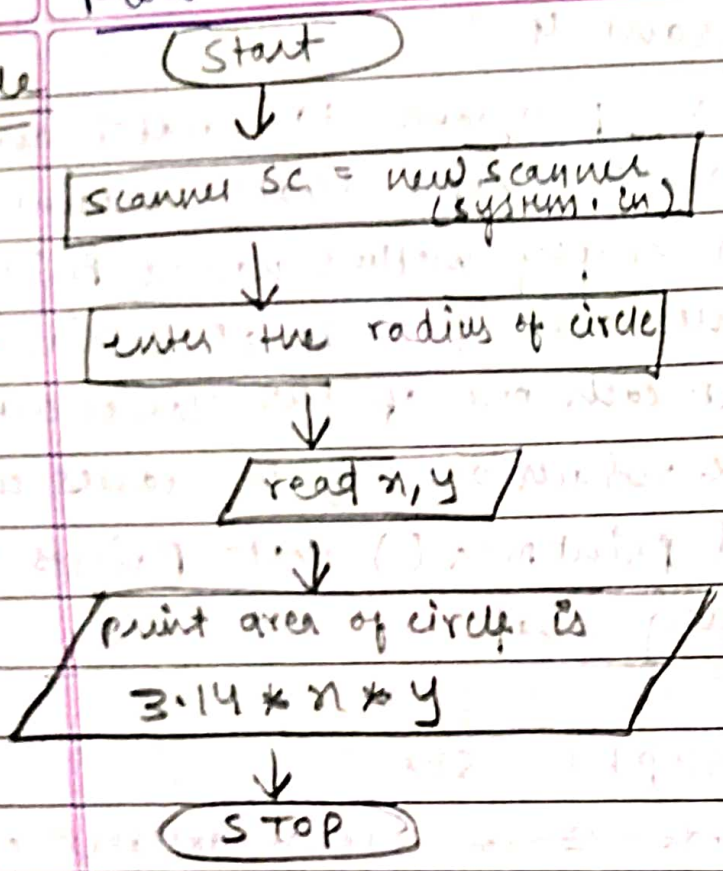
step 7 : In main method, create object rectangle, triangle and circle.

step 8 : Stop.

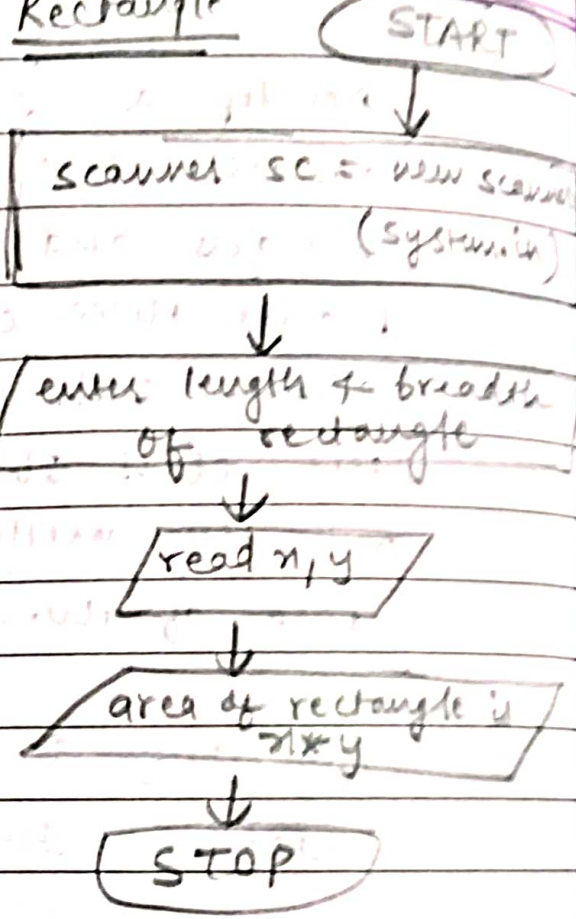
Date: / /

# Flow charts

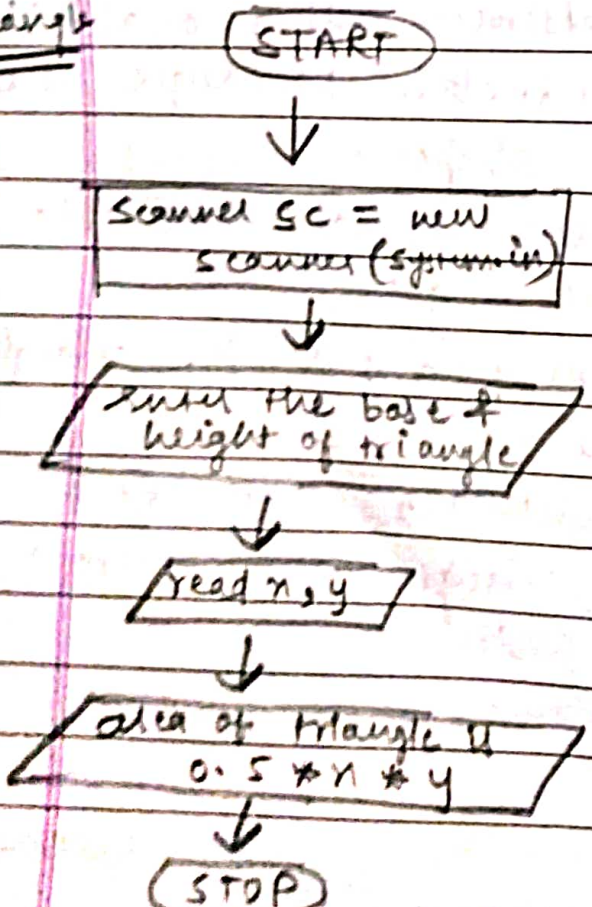
## Circle



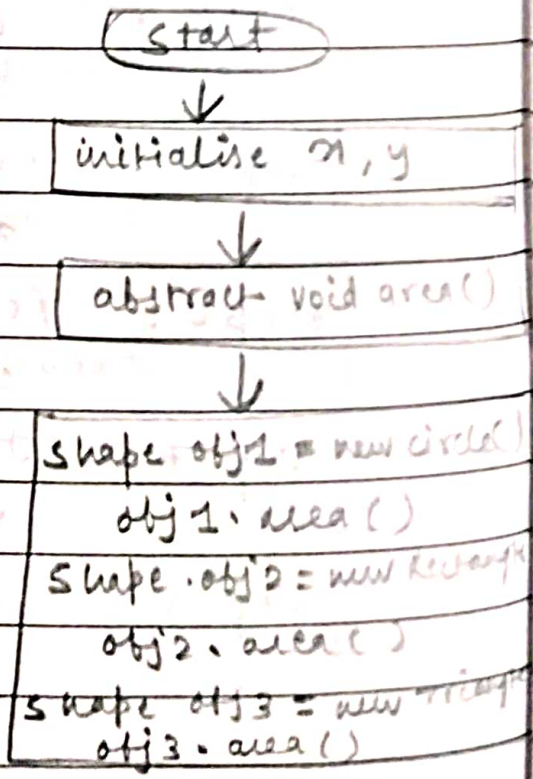

## Rectangle




## Triangle




## class shape




RAJDHANI

Date: / /

```
import java.util.Scanner;
abstract class Shape {
int x, y;
abstract void area();
Public static void main(String args[]) {
Shape obj1 = new Circle();
obj1.area();
Shape obj2 = new Rectangle();
obj2.area();
Shape obj3 = new Triangle();
obj3.area();
}
}
class Circle extends Shape {
Circle() {
Scanner Sc = new Scanner(System.in);
System.out.println("Enter the radius of the
                    circle");
x = Sc.nextInt();
y = x
}
Void area() {
System.out.println("Area of circle is"
                    + 3.14*x*y));
}
}
```

Date: / /

```
class Rectangle extends Shape {
  Rectangle () {
  Scanner sc = new Scanner (System.in);
System.out.println (" enter the length and
                     breadth of the rectangle");
x = sc.nextInt ();
y = sc.nextInt ();
  }
void area()
{
 System.out.println (" Area of rectangle is "
                     + x*y);
}
  }
class Triangle extends Shape {
 Triangle () {
Scanner sc = new Scanner (System.in);
System.out.println (" enter the base and height
                     of triangle :");
x = sc.nextInt ();
 y = sc.nextInt ();
}
 void area () {
  System.out.println (" Area of triangle is: "
                     + 0.5*x*y)
 }
  }
```

RAJDHANI

Date: / /

output :

```
Enter the radius of the circle :
1
Area of circle is 3.14
Enter the length and breadth of the rectangle:
2
5
Area of rectangle is 10
Enter the base and height of triangle :
2
6
Area of triangle is : 6.0
```

Date: / /

# Program - 5 / 6

Develop a Java Program to create a class Bank that maintains two kinds of account for its customers, one called saving account and the other current account. The savings account provided compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. current account holders should also maintain a minimum balance & if the balance falls below this level, a service charge is imposed create a class Account that stores customer name, account number and type of account. From this derive the classes cur-acct and sav-acct to make them more specific to their requirements. 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 and deposit interest.

d.) Permit withdrawl and update the balance check for the minimum balance, impose penalty if necessary & update the balance.

Date: / /

```
import java.util.scanner ;
class Account {
String customerName ;
long accno ;
String accountType ;
double balance ;
Public Account ( String customerName , long
                 accno , String accountType )
{
  this.customerName = customerName ;
  this.accno = accno ;
  this.accountType = accountType ;
  this.balance = 0.0;
}
public void displayBalance ( ) {
System.out.println ("Account number :" +accno);
System.out.println (" customername :" + customername);
System.out.println ("Account Type :" + accountType);
System.out.println (" Balance : $" + balance);
  }
}
class curAcct extends Account {
double minBalance ;
double serviceCharge ;
Public curAcct ( String customername , long accno)
{
```

Date: / /

```
super ( customerName , accno , "current" );
  this.minBalance = 500.0 ;
  this.serviceCharge = 50.0 ;
}
public void withdraw ( double amount ) {
if (balance - amount >= minBalance) {
    balance -= amount ;
System.out.println ("withdrawl successful,
                    current Balance : $." + balance);
}
else {
 System.out.println ("Insufficient funds,
withdrawl not allowed " ) ;
  }
  }
public void imposeservicecharge () {
 if ( balance < minBalance ) {
    balance -= service charge ;
System.out.println ("service charge imposed.
  current Balance : Rs." + balance);
} }
  }
class SavAcct extends Account {
   double interestRate ;
public SavAcct ( String customerName ,
                 long accno)
{
```

RAJDHANI

Date: / /

```
super ( customerName , accNo , " savings " );
this . interestRate = 0.05;
}
public void depositInterest ( ) {
double interest = balance * interestRate ;
balance += interest ;
System.out. println (" Interest deposited . current
    Balance : $ " + balance );
}
public void compoundInterest ( double initial
        Amount , int term ) {
double compoundInterest = initialAmount *
    math. pow ( 1 + interestRate ) , term ) - initial
balance += compoundInterest ;                    Amount ;
System . out . println (" compount interest deposited
 current Balance : Rs " + balance );
    }
}
public class Bank {
public static void main ( String[] args [] ) {
Scanner scanner = new Scanner ( System . in );
System . out . println (" choose account type :" );
System . out . println (" 1. current " );
system . out . println (" 2. savings " );
System . out . println (" Enter choice ( 1 or 2 ) : " );
```

Date: / /

```
int choice = scanner.nextInt();
System.out.println("Enter customer name:");
String customerName = scanner.nextLine();
System.out.print("Enter account number:");
long accno = scanner.nextLong();
if (choice == 1) {
CurAcct curAccount = new curAcct(customerName,
                                 accno);
System.out.print("Enter initial balance: $");
double initialBalance = scanner.nextDouble();
curAccount.balance = initialBalance;
System.out.println("Enter the withdrawl amount:$");
double withdrawlAmount = scanner.nextDouble();
curAccount.withdraw(withdrawlAmount);
curAccount.imposeServiceCharge();
curAccount.displayBalance();
}
else if (choice == 2) {
savAcct savAccount = new savAcct(customer
                                 Name, accno);
System.out.println("Enter initial balance: $");
double initialBalance = scanner.nextDouble();
savAccount.balance = initialBalance;
System.out.print("enter withdrawl amount: $");
double withdrawlAmount = scanner.nextDouble();
```

RAJDHANI

Date: / /

```
savAccount.balance -= withdrawalAccount;
System.out.println("withdrawal successful. current
        Balance : $" + savAccount.balance);
System.out.print("Enter interest Rate :");
double interestRate = scanner.nextDouble();
savAccount.interestRate = interestRate;
savAccount.displayBalance();
system.out.print("Enter term (in years) for
  compound interest calculation :");
int term = scanner.nextInt();
savAccount.compoundInterest(initialBalance.term);
savAccount.displayBalance();
}
else
{
system.out.println("invalid choice");
}
  }
    }
```

output: choose account type :

1. current 2. saving

enter choice (1 or 2) : 1

Enter customer name : Shilpa

Enter account number : 1234

Enter initial balance : $ 5000

Date: / /

Enter withdrawl amount : $ 500
withdrawl successful, current Balance : $ 4500.0
Account Number : 1234
customer Name : Shipra
Account Type : current
Balance : $ 4500.0

Algorithm → Step 1 : Create a class Account
step 2 : Initialise variables customername, accno, accountype, balance.
step 3 : Input : Enter customer name, accno, balance, account type as saving or current account from the user.
step 4 : Enter choice of savings & current account
step 5 : Read the choice if Choice is saving step 6 for current step 7.
step 6 : Enter initial balance, withdrawl amount of min balance from user. Check condition for min balance ie initial balance withdrawl should be >= min ie 800 then new Balance = balance - withdrawl
then print current balance, enter interest rate and time to calculate CI.
CI - balance * Power (1 + intrest rate, time) - initial - balance
Step 7 : else if choice = 2
create object of saving account.

RAJDHANI

Date: / /

## Program - 6

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 a derived class of student. This class has an array that stores the SEE marks scored in 5 courses of the current sem of student. Import the two packages in a file that declares the final marks of n students in all 5 courses

```
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:\n");
                                          ) ;
```

Date: / /

```
usn = scan.nextline();
name = scan.nextline();
sem = scan.nextInt();
}
}
Public class internal {
  public int in[] = 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 Final marks
{
  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();
```

Date: / /

```
SEE. External st[] = new SEE. External[n];
CIE. Internals S[] = new CIE. Internals[n];
for (int i=0 ; i<n ; i++)
{
  st[i] = new SEE. External();
  S[i] = new CIE. Internals();
System. out. println("Enter details" + (i+1));
 st[i]. accept();
for (int j=0 ; j<5 ; j++)
{
 System. out. println("Enter in and sm 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]);
 }
}
}
```

Date: / /

Output:

```
1BM22CS341
Enter no. of students : 1
Enter details 1
Enter sem, usn and name :
Enter internal and see marks of sub 1 : 45
                                        45
Enter internal and see marks of sub 2 : 48
                                        46
Enter internal and see marks of sub 3 : 47
                                        47
Enter internal and see marks of sub 4 : 48
                                        48
Enter internal and see marks of sub 5 : 49
                                        49
Final marks of 1BM22CS341
    course 1 = 90
    course 2 = 92
    course 3 = 94
    course 4 = 96
    course 5 = 98
```

Algorithm

Step 1 : Create package CIE. Declare a public class Student with members USN, name, Sem.

Step 2 : A method input allows user to give inputs for variables USN, name and sem.

Step 3 : Create another class internals. In same package CIE. It consists of an array int[] that stores internal marks scored in 5 courses.

Step 4 : Create package SEE. import Student class from package CIE. Create a class external which extends Student class.

Step 5 : Import package CIE & SEE. Create array int[] that stores total marks. Read user input.

Step 6 : Add internal & external marks.

Step 7 : Display total marks of all students

Step 8 : Stop.

RAJDHANI

Date: / /

# Program - 7

Q) write a program that demonstrate handling of exceptions in inheritance tree. 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 & throws the exception wrongAge() when the input age < 0. In son class, implement a constructor that cases both father & son's age and throws an exception if son's age is >= father's age.

```
import java.util.Scanner;
class WrongAge extends Exception
{
  Public WrongAge ( String message )
  {
   super ( message) ;
  }
    }
class Father
{
   int fatherAge ;
Public Father ( int fatherAge) throws WrongAge
    {
```

Date: / /

```
if (fatherAge < 0) {
    throw new WrongAge ("Age cannot be
                        negative");
  }
this.fatherAge = fatherAge;
  }
  }
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 must be
                     less than father's age");
  }
this.sonAge = sonAge;
}
  }
Public class fatherson {
  public static void main (String [] args)
{
  Scanner sc = new Scanner;
  Scanner (system.in);
System.out.println ("enter father's age and
                     son's age");
```

RAJDHANI

Date: / /

```
int fa = sc.nextInt();
int fa = sc.nextInt();
try {
        Son s = new Son(fa, sa);
System.out.println("Father's age:" + s.fatherAge);
System.out.println("son's age:" + s.sonAge);
  }
catch (WrongAge e) {
   System.out.println("Error" + e.getMessage());
  }
  }
  }
```

output:

```
Enter father's age and son's age:
50
20
Father's age : 50
son's age : 20
```

Date: / /

## Algorithm

step 1: Create an exception WrongAge which extends exception.

step 2: create base class Father

step 3: Make a constructor Father(age) which inputs age of father.

step 4: If age < 0; throw exception WrongAge.

step 5: Create son class derived from Father

step 6: Take both father and son's age in constructor.

step 7: If son's age >= father's age, throw exception WrongAge.

Date: / /

## Program-8

write a Program which creates two threads, one thread displaying "BMS college of Engineering" once every ten seconds & another displaying "CSE" once every two seconds.

```
class A extends Thread
{
  int t1 = 0 , time ;
A()
 {
    t1 = 10000 ;
   time = 21000 ;
}
Public void run()
{
  while ( t1 <= time)
{
   System.out.println ("BMS college of engineering");
try {
    sleep (10000) ;  }
catch ( Exception e)
{
   System.out.println ("error");
   }
```

Date: / /

```
        t1 t = 10000;
    }
}
}
class B extends Thread {
    int t2 = 0, time;
    B() {
        time = 21000;
        t2 = 2000;
    }
    public void run() {
        while (t2 <= time) {
            System.out.println ("CSE");
            try {
                Sleep (2000);
            }
            catch (Exception e) {
                System.out.Println ("error");
            }
            t2 t= 2000;
        }
    }
}
class th
{
    public static void main (String args[])
    {
```

1000701

Date: / /

```
A a = new A();
B b = new B();
a.start();
b.start();
}
}
```

Output :-

```
BMS college of engineering
CSE
CSE
CSE
CSE
CSE
BMS college of engineering
CSE
CSE
CSE
CSE
CSE
```

Date: / /

## Algorithm

Step 1 : Start

Step 2 : Create a class A that extends to thread.

Step 3 : Initialise f1 = 0 , time = 3000 in constructor of A.

Step 4 : write a method run.

Step 5 : use a while loop which runs until the condition t1 <= time becomes false.

Step 6 : Print BMS college of Engineering.
Include try block and use sleep method for 10000. In catch block, print the statement error if it encounters exception e. Increment t1 = t1 + 10000;

Step 7 & 8 : create a class B that extends to thread.
Initialise t2 = 0 , time = 3000 in constructor of B.

Step 8 : Create a method run.

Step 9 : use a while loop which runs until the condition t2 <= time became false..
print CSE.

Step 10 : Include try block and call sleep method for 2000. In catch block, Print the statement error if it encounters exception e.
Increment t2 = t2 + 2000;

Step 11 : create a class Demo.

Date: / /

Step 12 : Initialise object of datatype A, b of datatype B.

Step 13 : Call method start of object A

Step 14 : Call method start of object B.

Step 15 : Stop.