Program-1

Q. "Develop a Java program that prints all real solutions to the quadratic equation $ax^2 + bx + c = 0$. Read in a, b, c and use the quadratic formula. If the discriminate $b^2 - 4ac$ is negative, display a message stating that "there are no real solutions."

```
import java.util.Scanner;
import static java.lang.Math.sqrt;
import static java.lang.Math.abs;

public class squad{
  public static void main (String [] args){
    Scanner in = new Scanner (System.in);
    System.out.println ("Enter coefficients : ");
    int a = in.nextInt();
    int b = in.nextInt();
    int c = in.nextInt();
    if (a==0){
      System.out.println ("Invalid input");
    }
    else{
      int d = b*b - 4*a*c;
      if (d>0) {
        System.out.println ("Roots are real");
        float r1 = (float) (-b+ sqrt (d)) / (2*a);
        float r2 = (float) (-b-sqrt(d)) / (2*a);
        System.out.println (r1);
        System.out.println (r2);
      }
```

```
else if (d<0){
    System.out.println ("Roots are imaginary. There
                        are no real solutions");
    float r1 = (float) -b/(2*a);
    float r2 = (float) sqrt (abs (d))/(2*a);
    System.out.println (r1 + "+i" + r2);
    System.out.println (r1 + "-i" + r2);
    }
    else {
    System.out.println ("Roots are equal");
    float r = (float) -b/(2*a);
    System.out.println (r);
    }
  }
}
```

Output:

Enter coefficients:

4

5

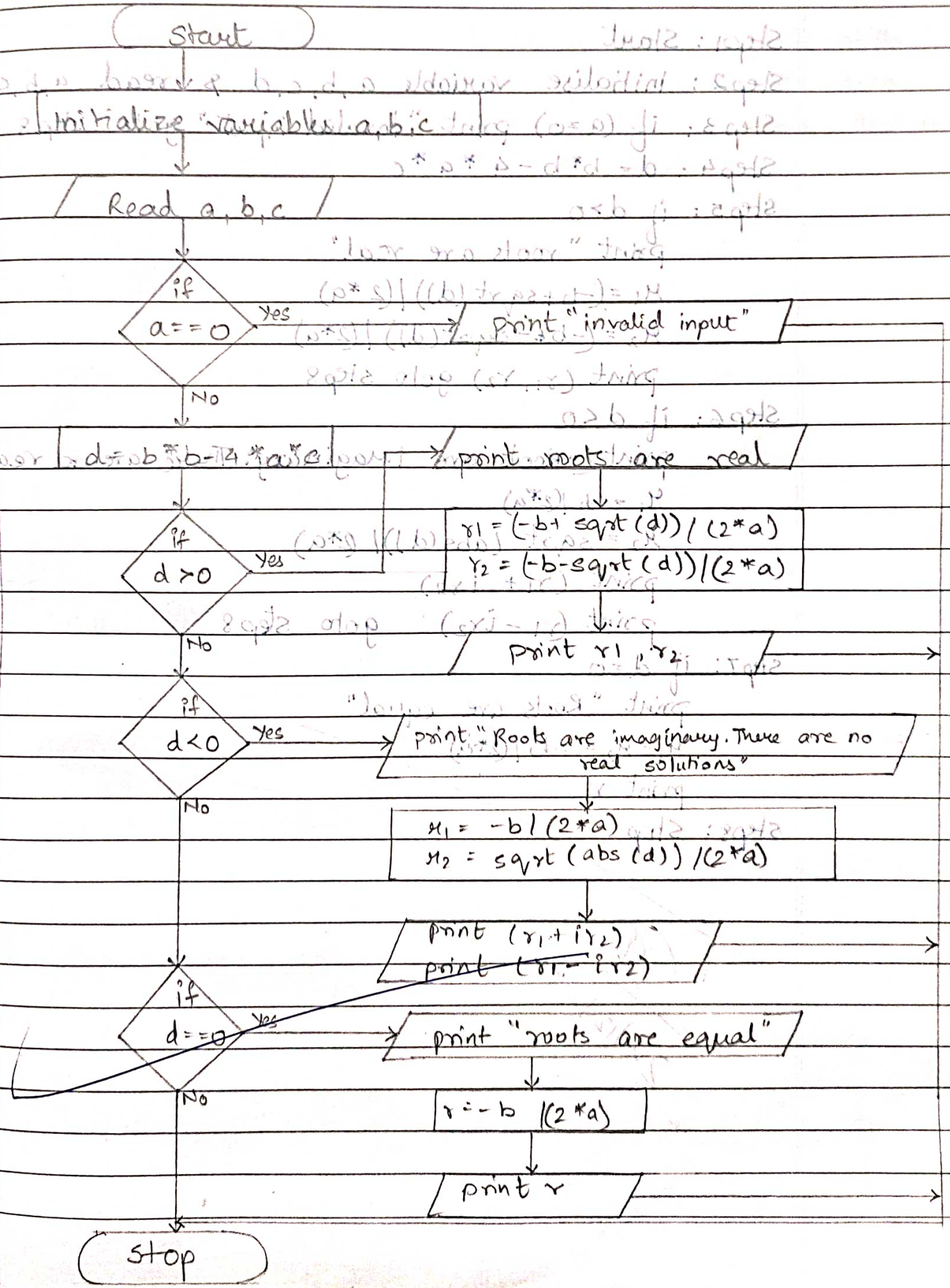
6

Roots are imaginary. There are no real solutions

-0.625 + i1.0532687

-0.625 -i1.0532687

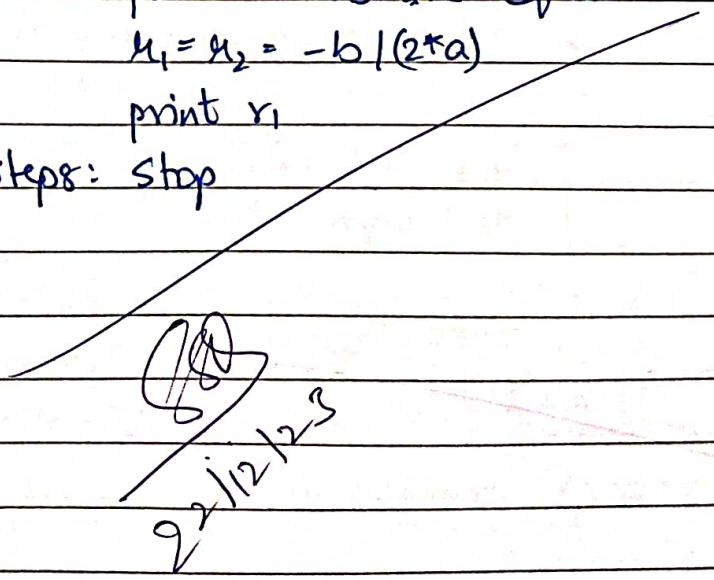
Flowchart:


Start
Initialize variables a,b,c
Read a,b,c
if a==0
Yes
print "invalid input"
No
d=b*b-4*a*c
print roots are real
if d>0
Yes
r1 = (-b+sqrt(d))/(2*a)
r2 = (-b-sqrt(d))/(2*a)
print r1, r2
No
if d<0
Yes
print "Roots are imaginary. There are no real solutions"
No
r1 = -b/(2*a)
r2 = sqrt(abs(d))/(2*a)
print (r1+ir2)
print (r1-ir2)
if d==0
Yes
print "roots are equal"
No
r = -b/(2*a)
print r
Stop

Page No.
Date

Algorithm:

Step1 : Start
Step2 : Initialise variable a, b, c, d & read a, b, c
Step3 : if (a=0) print "Invalid input" goto step8
Step4 : d = b * b - 4 * a * c
Step5 : if d>0
print "roots are real"
$r_1 = (-b + sqrt(d)) / (2 * a)$
$r_2 = (-b - sqrt(d)) / (2 * a)$
print (r_1, r_2) goto step8
Step6 : if d<0
print ("Roots are imaginary. There are no real solution")
$r_1 = -b / (2 * a)$
$r_2 = sqrt(abs(d)) / (2 * a)$
print (r_1 + ir_2)
print (r_1 - ir_2) goto step8
Step7 : if d=0
print "Roots are equal"
$r_1 = r_2 = -b / (2 * a)$
print r_1
Step8 : Stop



Page No.
Date

Program-8

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

```
import java.util.Scanner;
class Subject
{
int subjectMarks;
int credits;
int grades;
}
class Student
{
Subject subject[];
String name;
String USN;
double SGPA;
Scanner s;
Student()
{
int i;
subject = new Subject[9];
for (i=0; i<9; i++)
{
subject[i]= new Subject();
s = new Scanner(System.in);
}}
}
```

```
void getStudentDetails()
{
System.out.print("Enter your name:");
name = s.next();
System.out.println("Enter your USN:");
USN = s.next();
}
voidgetMarks(){
for(int i=0; i<=8; i++)
{
System.out.print("Enter Marks for subject"+(i+1)+":");
subject[i].subjectMarks = s.nextInt();
System.out.print("Enter your credits for subject"+(i+1)+":");
subject[i].credits = s.nextInt();
subject[i].grade = (subject[i].subjectMarks/10)+1;
if (subject[i].grade == 11)
subject[i].grade = 10;
if (subject[i].grade <= 4)
subject[i].grade = 0;
}
}
void computeSGPA()
{
int effectiveScore = 0;
int totalCredits = 0;
for(int i=0; i<9; i++)
{
effectiveScore += (subject[i].grade * subject[i].credits);
totalCredits += subject[i].credits;
}
SGPA = (double) effectiveScore / (double) totalCredits;
```

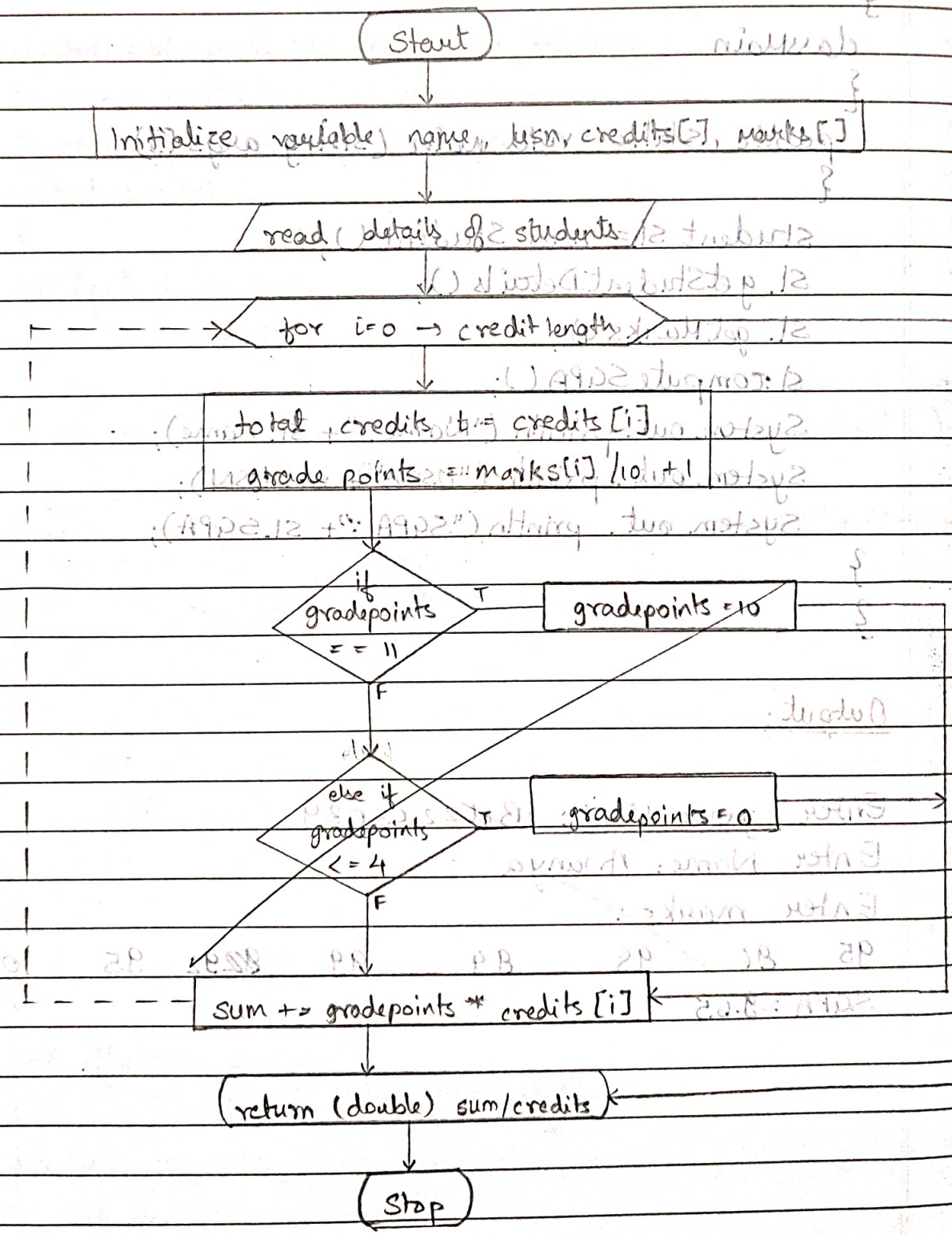
```
}
}
 classMain
 {
  public static void main (String args[])
  {
   Student s1 = new Student();
   s1.getStudentDetails();
   s1.getMarks();
   s1.computeSGPA();
   System.out.println("Name: " + s1.name);
   System.out.println("USN:" + s1.USN);
   System.out.println("SGPA :" + s1.SGPA);
  }
}
```

Output:

```
Enter your USN : 1BM22CS039
Enter Name: Ananya
Enter marks:
95  86  98  84  99  92  95  100
SGPA :9.65
```

Page No.
Date

Flow chart:



Page No.
Date

Algorithm:

Step 1: Start
step 2: Initialize variable arr[], marks, usn, name, sgpa.
step 3: Calling class Student in which calling fun^c first()
step 4: Input : "Enter usn" + usn
step 5: Input : "Enter name" + name
step 6: Print "Enter Marks"
step 7: for (i=0; i<8; i++){
arr(i) = in.nextInt();
}
step 8: Read array credits as {4,4,3,3,3,1,1,1}
step 9: for (i=0; i < arr.length; i++) {
if (arr[i] >= 100) {
arr(i) = arr(i)-10;
}
else if (arr[i] < 40){
arr[i] = 0;
}
marks += credit[i] * ((arr[i]/10) +1)
}
step 10: sgpa = marks/20
step 11: Print "SGPA :-" + sgpa
step 12: Stop

Program-9

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

```
import java.util.Scanner;
class Books{
    String name;
    String author;
    int price;
    int num_pages;
    public void set(int i){
        Scanner in = new Scanner(System.in);
        System.out.println("Enter details of book" + (i+1)+
        "in name, author, price, num_pages, order");
        name = in.next();
        author = in.next();
        price = in.nextInt();
        num_pages = in.nextInt();
    }
    public String toString(){
        return "Details of Book" + (i+1) + "\n" +
        "Name: " + name + "\n" +
        "Author:" + author + "\n" +
        "Price=" + price + "\n" +
        "No. of pages=" + num_pages;
    }
}
```

```
class Main {
    public static void main (String [] args) {
    int n;
    Scanner in = new Scanner (System.in);
    System. out println ("Enter number of books: ")
    n = in. nextInt();
    Books b[] = new Books [n] ;
    for (int i = 0 , i < n ; i++) {
            b[i] = new Books ();
            b[i]. set (i);
    }
    System. out. println ();
    for (int i = 0 ; i < n ; i++){
        System. out. println (b[i]. toString [] ),
    }
  }
}
```

Output:

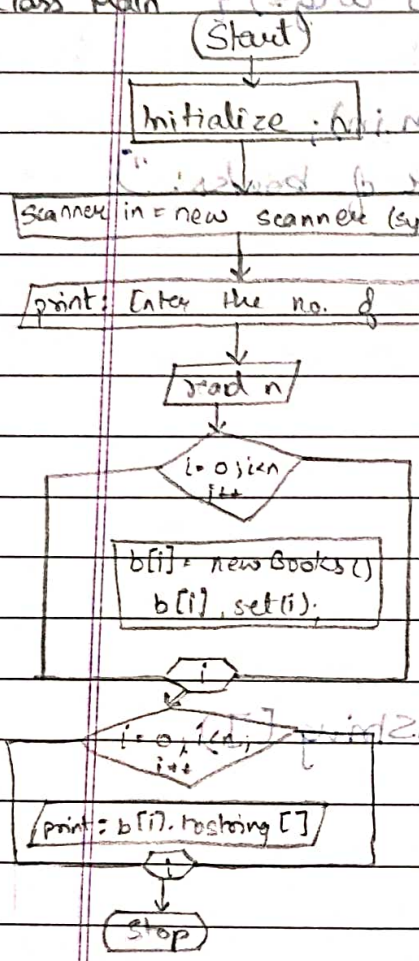
```
Enter the no. of books: 1
Enter name:
abc
Enter author:
xyz
Enter price:
500
Enter number of pages
450
```

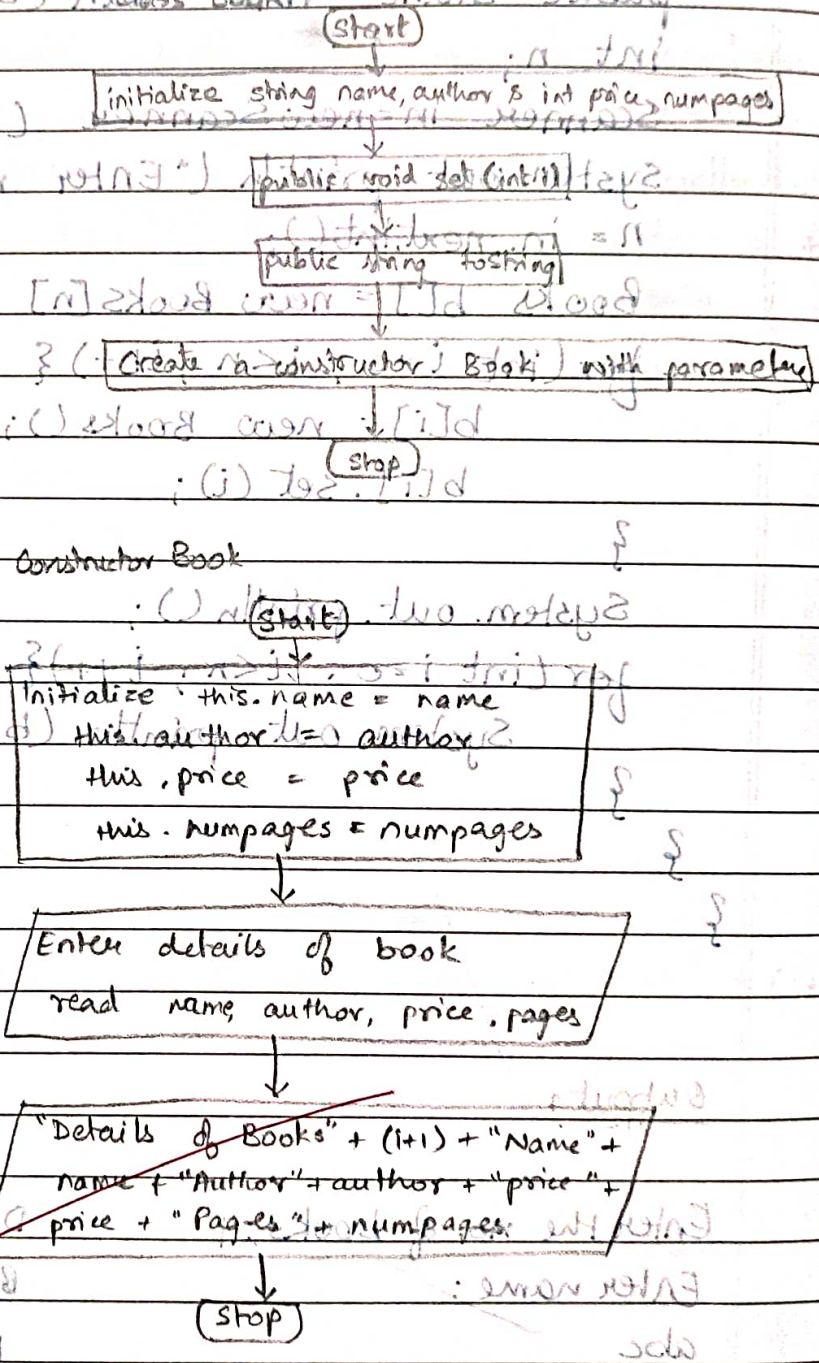
```
Details of all books
Book 1:
Name: abc
Author: xyz
Price: 500
Num of pages: 450
```

Page No.
Date

Flowchart:

Class Main





Page No.
Date

Algorithm:

Step 1: Start
Step 2: Initialize variables, name, author, price, num_pages
Step 3: Enter no. of books
step 4: Enter name, author, price, num_pages
Step 5: for (i=0 ; i<n ; i++)
b [i] = new books ()
b [i] = set [i];
Step 6: display books
for (i=0; i<n ; i++)
print book details
Step 7: stop

Program-10

Q Develop a Java program to create an abstract class named Shape that contains two integers & 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.

```
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 a circle is "+3.14*x*y);
}
}
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 the
        triangle");
     x= sc.nextInt();
     y = sc.nextInt(); }

void area()
 {
  System.out.println("area of triangle is " + 0.5* x*y);
 }
}
```

Page No.
Date

Output:

Enter the radius of the circle
2
area of circle is 12.56
Enter the length & breadth of the rectangle
5
4
area of rectangle is 20
Enter the base & height of the triangle
5
2
area of triangle is 5

12/10/24

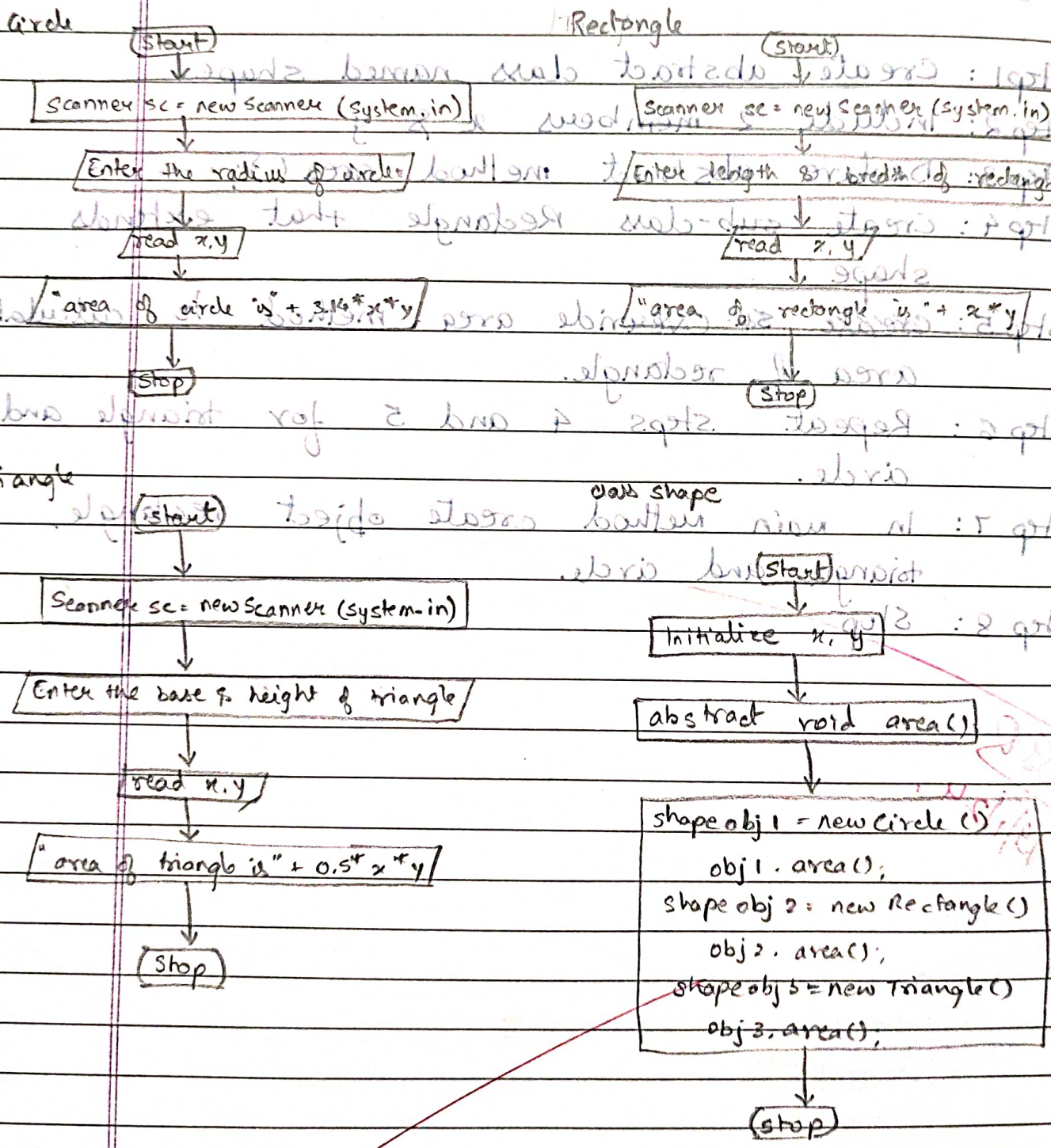
Page No.
Date

Algorithm

Step 1 : Create abstract class named shape
Step 2 : Include 2 members x & y
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

SSD
19/1/24.

Flow chart:


Circle
Start
Scanner sc = new Scanner (System.in)
Enter the radius of circle
read x,y
"area of circle is" + 3.14*x*y
Stop
Rectangle
Start
Scanner sc = new Scanner (System.in)
Enter length & breadth of rectangle
read x, y
"area of rectangle is" + x*y
Stop
Triangle
Start
Scanner sc = new Scanner (System.in)
Enter the base & height of triangle
read x,y
"area of triangle is" + 0.5*x*y
Stop
class shape
Start
Initialize x, y
abstract void area()
shape obj 1 = new Circle ()
obj 1 . area();
shape obj 2 = new Rectangle()
obj 2 . area();
shape obj 3 = new Triangle()
obj 3. area();
Stop

Page No.
Date

Program-12

Q Develop a Java prog to create a class Bank that maintains two kinds of account for its customers, one called savings account & the 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 holders should also maintain a ₹ min. balance & if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account no. & 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 ~~make them~~ achieve the following tasks:

a) Accept deposit from customer and update the balance
b) Display the balance
c) Compute and deposit interest
d) Permit withdrawal & update the balance. Check for min balance, impose ₹ penalty if necessary & update the balance

```
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("Customer Name: " + 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){
        super(customerName, accno, "current");
        this.minBalance = 5000.0; // set min balance
        this.serviceCharge = 50.0; // set service charge
    }
    public void withdraw(double amount) {
        if (balance - amount >= minBalance){
            balance -= amount;
            System.out.println("Withdrawal successful. Current
            Balance: $" + balance);
        }
        else {
            System.out.println("Insufficient funds. Withdrawal
            not allowed.");}}
```

```
public void imposeServiceCharge(){
    if (balance < minBalance){
        balance -= serviceCharge;
        System.out.println("Service charge imposed. Current
            Balance: Rs." + balance);
    }
}

class SavAcct extends Account{
    double interestRate;

    public SavAcct(String customerName, long accno){
        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 initialAmount,
        int term){
        double compoundInterest = initialAmount * Math.pow
            (1+interestRate), term) - initialAmount;
        balance += compoundInterest;
        System.out.println("Compound 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):");
    int choice = scanner.nextInt();
    System.out.print ("Enter customer name:");
    String customerName = scanner.next();
    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.print ("Enter withdrawal amount: $");
        double withdrawalAmount = scanner.nextDouble();
        curAccount.withdraw (withdrawalAmount);
        curAccount.imposeServiceCharge();
        curAccount.displayBalance();
    }
    else if (choice == 2){
        SavAcct savAccount = new SavAcct (customerName,
            accno);
        System.out.print ("Enter initial balance: $");
        double initialBalance = scanner.nextDouble();
        savAccount.balance = initialBalance;
```

```
System.out.print ("Enter withdrawal amount: $");
double withdrawalAmount = scanner.nextDouble();
savAccount.balance -= withdrawalAmount;
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");
}
}
}
```

19/12

Output:

```
Choose account type:
1. Current          2. Savings
Enter choice (1 or 2): 2
Enter customer name: Ananya
Enter account number: 2004
Enter initial balance: $5000
Enter withdrawal amount: $500
Withdrawal successful. Current Balance: $4500.0
```

Account Number : 2004
Customer Name : Ananya
Account Type : current
Balance : ₹ 4500.0

Algorithm:

Step1 : Create a class Account
Step2 : Initialise variables customername, accno, accounttype balance.
Step3 : Input: Enter customername, accno, balance, account type as saving or current account from the user.
Step4 : Enter choice of savings & current account.
Step5 : Read the choice if choice is saving step 6, for current step7.
Step6 : Enter initial balance, withdrawal amount & min balance from user. Check condition for min balance i.e initial balance withdrawal should be >= min i.e 500 then new Balance = balance - withdrawal;

Then, print current balance, enter interest rate & time to calculate CI.
CI - balance * Power (1 + interest rat, time) - initial - balance

Step7 : else if choice = 2
Create object of saving account.
Step8 : Stop

Page No.
Date

Program-13

Q Create a package CIE which has 2 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 5 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 semester of the student. Import the 2 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 U, N, S:\n");
    usn = scan.nextLine();
    name = scan.nextLine();
    sem = scan.nextInt();
  }
}
public class Internals{
  public int im[] = 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();
   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 im 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]);
    }
  }
 }
}
```

Algorithm:

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

Step 2 : A method input allows users to give inputs for variables USN, name & 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 which extends student class.

Step 5 : Import package CIE & SEE, create array m that stores total marks. Read users input.

Step 6 : Add internal & external marks.

Step 7 : Display total marks of the students

Step 8 : Stop.

Page No.
Date

Output:

```
Enter no. of students: 1

Enter details 1
Enter 1sem, USN & Name: 1BM22CS039
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

Final marks of 1BM22CS039
Course 1 : 90
Course 2 : 92
Course 3 : 94
Course 4 : 96
```

Program-14

Q. Write a program that demonstrates 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 & 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{
        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 (System.in);
        System.out.println ("Enter father's age & son's age:");
        int fa = sc.nextInt();
        int sa = 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());
        }
    }
}
```

Page No.
Date

Output:

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

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 & son's age in constructor.

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

Step 8: Stop

Program - 15

Q Write a prog which creates two threads, one thread displaying "BMS College Of Engineering" once every ten seconds & another displaying "CSE" once every two sec.

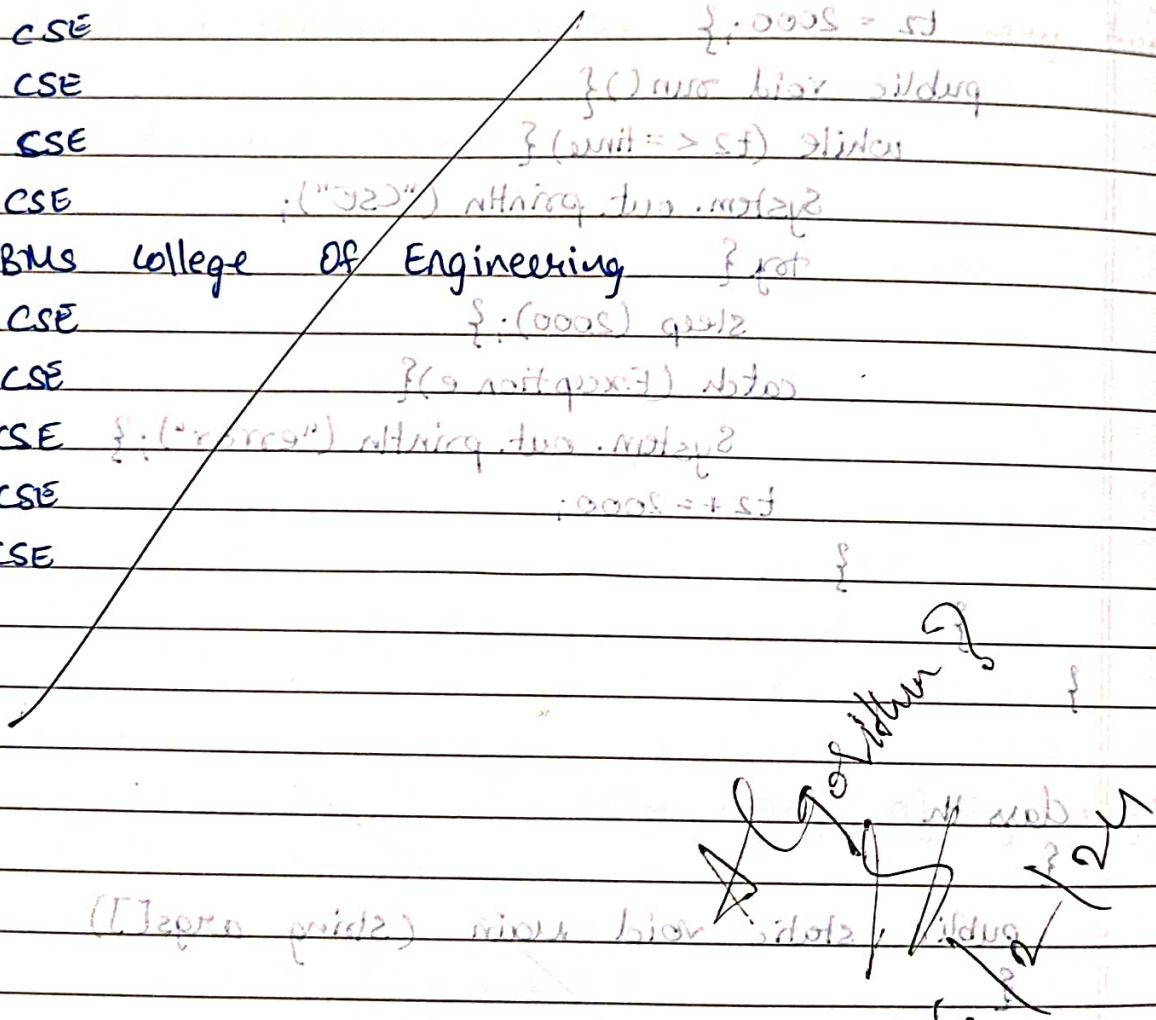
```
class A extends Thread
 {
  int t1=0, time;
  A(){
     t1=10000;
     time = 21000;
  }
  public void run()
   {
   while(t1<=time)
     {
      System.out.println("BMS College of Engineering");
      try{
       sleep(10000);}
       catch(Exception e){
         System.out.println("error");
       }
       t1+=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 += 2000;
          }
    }
}

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



Algorithm:

Step 1 : Start
Step 2 : Create a class A that extends to thread
Step 3 : Initialise tr = 0, time = 3000 in constructor of A,
Step 4 : Write a method run.
Step 5 : Use a while loop which runs until the condition tr <= time becomes false.

Step6: Print BMS college of Engineering.
Include try block & use sleep method for 10000. In catch block, print the statement error if it encounters exception e.
Increment $t_1 = t_1 + 10000$;

Step 7: Create a class B that extends to thread.
Initialise $t_2 = 0$, time = 3000 in constructor of B

Step 8: Create a method run.

Step 9: Use a while loop which runs until the condition $t_2 <= time$ became false. Print CSE.

Step 10: Include try block & call sleep method for 2008. In catch block, print the statement error if it encounters exception e.
Increment $t_2 = t_2 + 2000$;

Step 11: Create a class demo.

Step Initialise object a of datatype A, b of datatype B.

Step 12: Call method start of object A.

Step 13: Call method start of object B.

Step 14: Stop.