OBJECT ORIENTED PROGRAMMING IN JAVA

LAB PROGRAMS

BY:
ANEESH K P
1BM22CS040

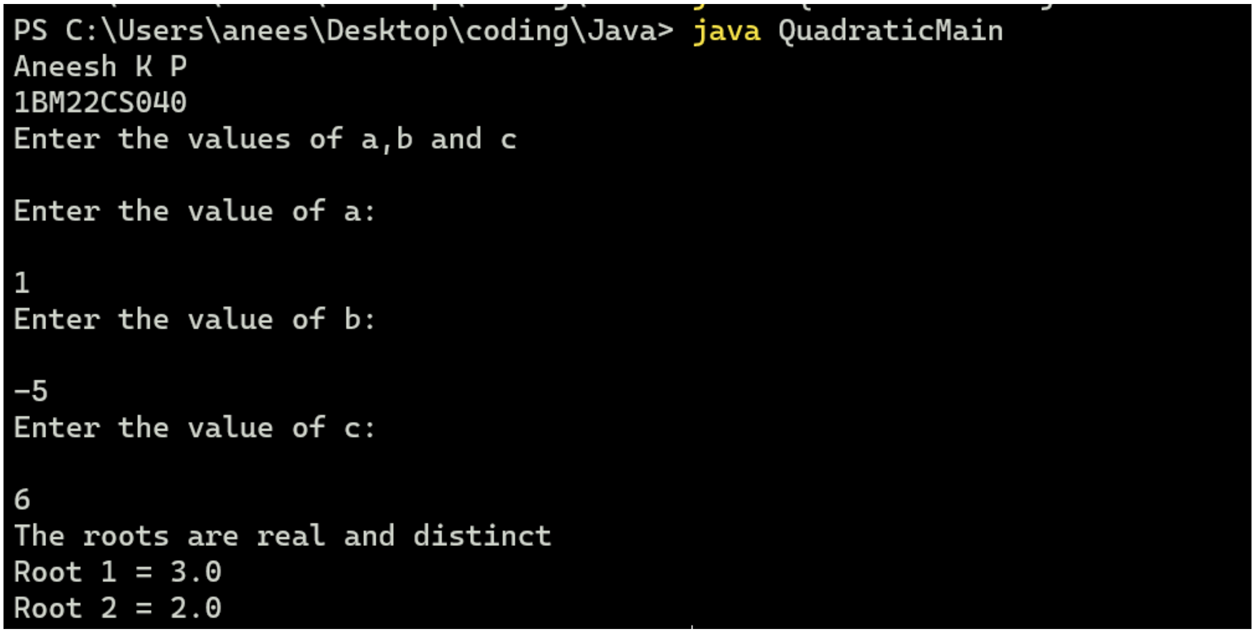
LAB-1

1. Develop a Java program that prints all real solutions to the quadratic equation ax2+bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b2-4ac is negative, displaying a message stating that there are no real solutions.

7. 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 discriminant b^2-4ac is negative, display a message stating that there are no real solutions.

```
import java.util Scanner;

public class QuadraticSolver{

        public static void main(String [] args) {
         Scanner scanner= new Scanner (System. in);

         System.out.println ("Enter the coefficients for
              quadratic equation");

      system. out. println (" Enter a: ");
      double a = scanner. nextDouble();
      System. out. println ("Enter b: ");
      double b= scanner. nextDouble();
      System. out. println ("Enter c:");
      double c = scanner. nextDouble();
      double discriminant = b*b - 4*a*c;
      if ( discriminant >= 0) {
    double root1 = (-b+Math.sqrt (discriminant)) /(2
    double root2 = ( b+Math. sqrt(discriminent)) /(2*

    System. out. println(" Real solutions :");
    System. out. println(" Root 1: " + root1);
    System. out. println (" Root 2: " + root2);
 }
  else {
  System. out. println(" No real solutions ");
 }
  scanner. close();
} }
```

OUTPUT:



```
PS C:\Users\anees\Desktop\coding\Java> java QuadraticMain
Aneesh K P
1BM22CS040
Enter the values of a,b and c

Enter the value of a:

1
Enter the value of b:

-5
Enter the value of c:

6
The roots are real and distinct
Root 1 = 3.0
Root 2 = 2.0
```

LAB-2

2.Develop a Java program to create a class 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 student.

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 and display details and a method to calculate SGPA of student.

```
import java.util.Scanner;

public class Student {
        String usn;
        String name;
        private static int credit [] = {4,4,3,3,3,
                                        1,1,1}
        private int numOfSub = 8;
        int marks[]=new int [numOfSub];

        Scanner s = new Scanner (System.in);

    Student()
    {
        System.out.println("Enter your details below
          to calculate SGPA");
        get_details();
        get_marks();
        display();
    }

public void get_details()
{
 System.out.println ("Enter your USN:");
    usn = s.next();
 System.out.println("Enter your name: ");
public void set_marks()
{
  System.out.println("Enter your marks in
  order");
  for(int i=0; i< numOfSub ; ++i)
  {
      marks[i] = s.nextInt();
  }
}
public double sgpa()
{
  double sgpa =0, temp =0;
  for(int i =0; i< numOfSub; ++i)
  {
   if(marks[i] >=40) {
   if(marks[i] == 100)
   {
      temp += credit[i] *((int)
       (marks[i])/10 ));
   }
   else
   {
    temp += credit[i] * ((int)(marks[i]
         /10) +1);
   }
  }
   else
   {
      temp +=0;
   }
  }
  sgpa = cgpa /20;
  return sgpa;
}
```

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

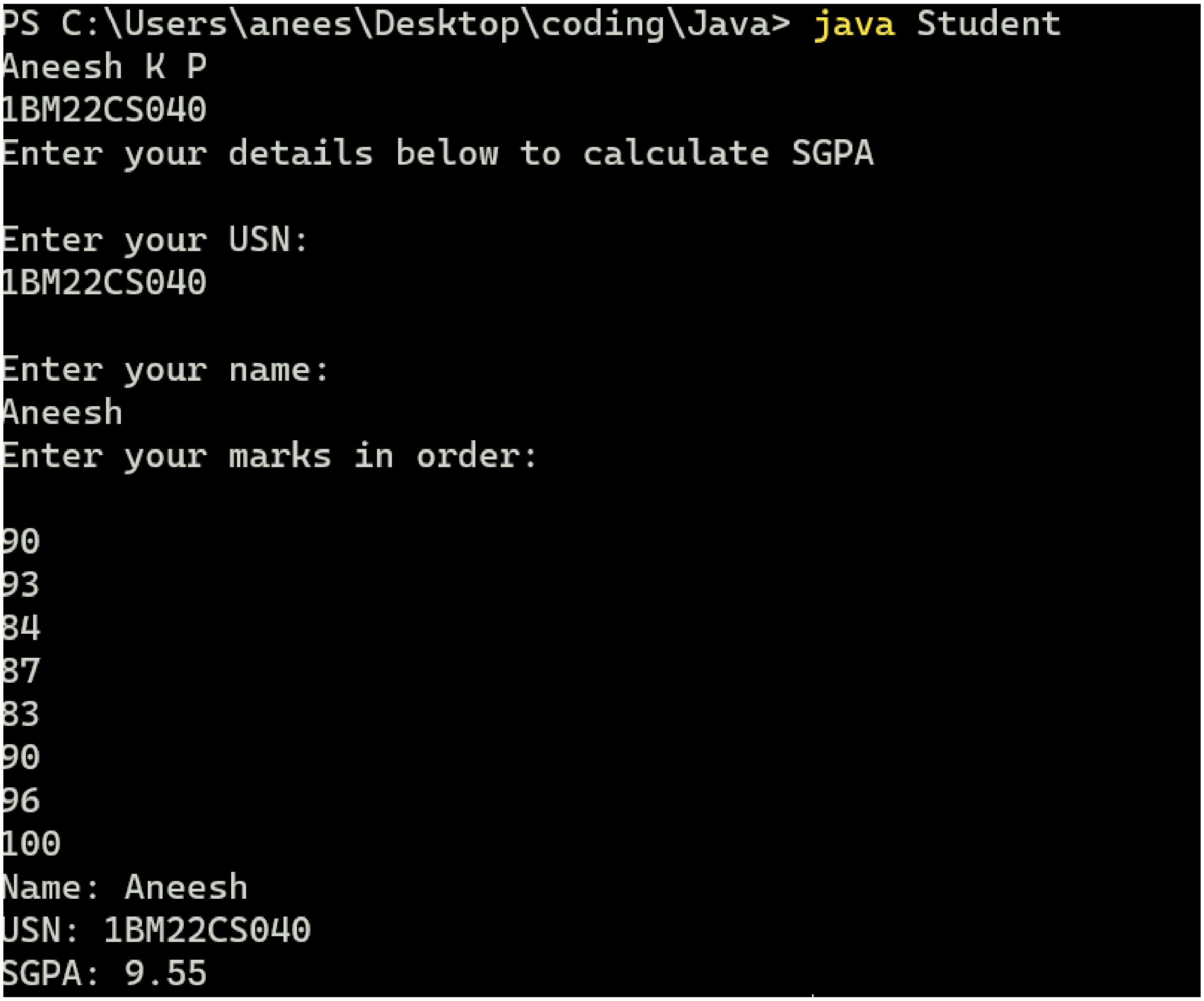
Algorithm:

```
Step1: Start
Step2: Initialize USN, name and an array containing the
       credits of each course in order
Step 3: Define a constructor of the class to ~~display~~ get details
        by calling get_details() function.
Step4: Enter USN, Name and marks of the student
Step5: Read array credits {4,4,3,3,3,1,1,1}
Step6:    for i < array length
              marks += credit[i] * (array[i]/10)+1)
Step 7:     sgpa = marks/20
Step 8: Print SGPA
Step 9: Stop.
```

Output:

```
Enter your details below to calculate your s
 Enter your USN: 1BM40
   Enter your name: Aneesh
   Enter your marks in order:
   90 93 84 87 83 90 96 100
     SGPA: 9.55
```

OUTPUT:



```
PS C:\Users\anees\Desktop\coding\Java> java Student
Aneesh K P
1BM22CS040
Enter your details below to calculate SGPA

Enter your USN:
1BM22CS040

Enter your name:
Aneesh
Enter your marks in order:

90
93
84
87
83
90
96
100
Name: Aneesh
USN: 1BM22CS040
SGPA: 9.55
```

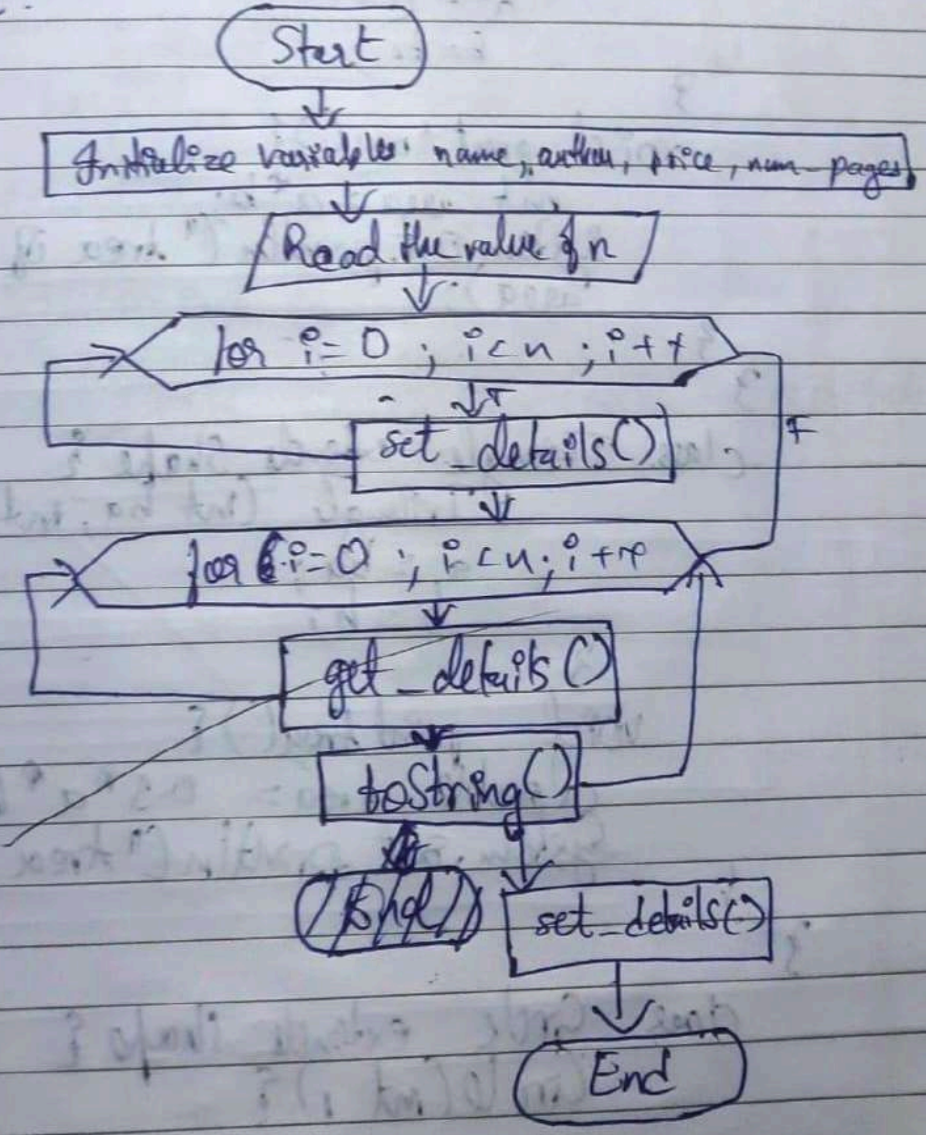
LAB-3

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 objects. Include a toString( ) method that could display the complete details of the book. Develop a Java program to create n book objects.

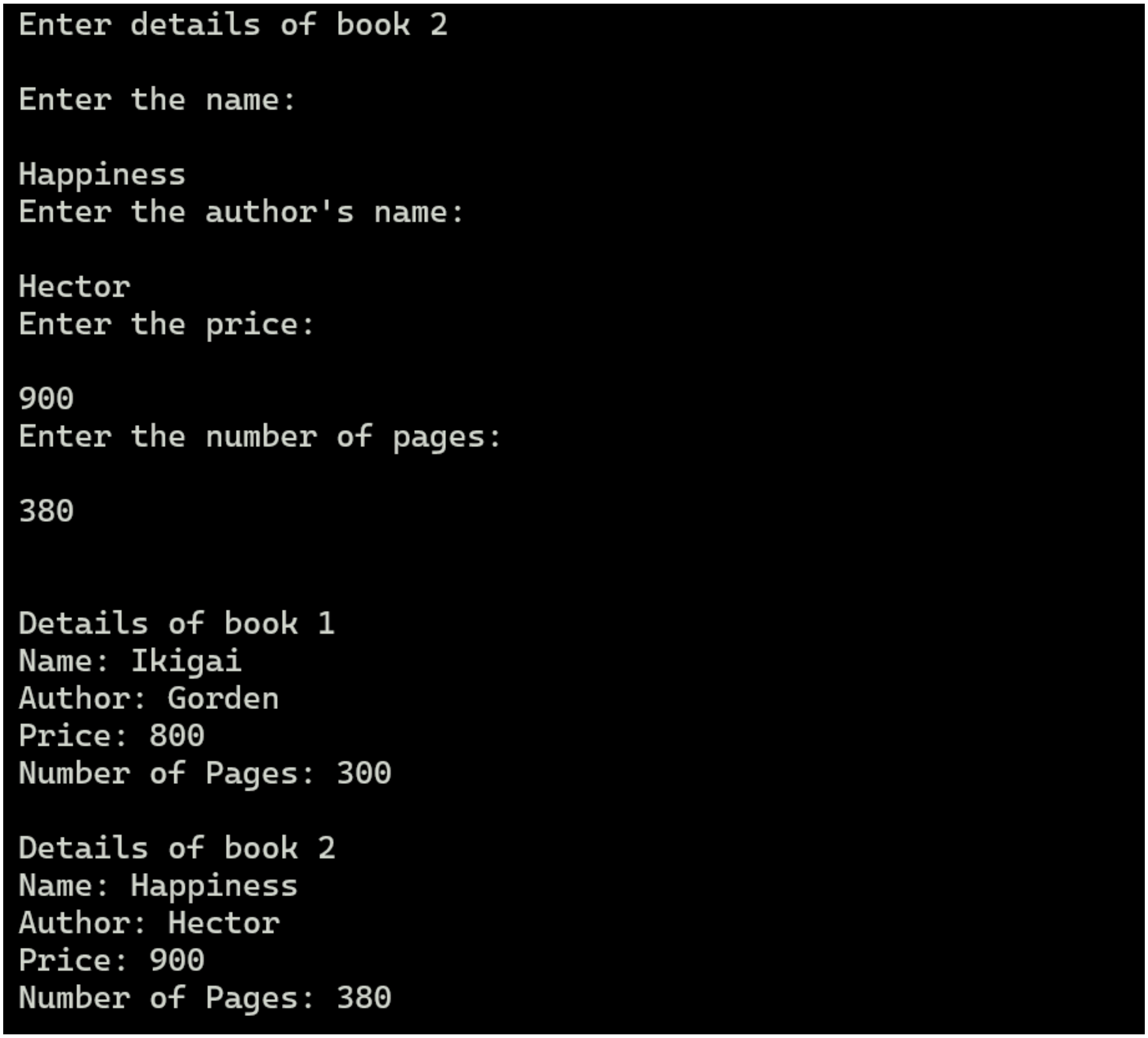
9. WAP to take and print the details of a book

```
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 the details of
         the book" + (i+1) + "\n name, autor,
        price , num of pages in 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 the No. 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());
}
}
}
```

Flowchart:



OUTPUT:



```
Enter details of book 2

Enter the name:

Happiness
Enter the author's name:

Hector
Enter the price:

900
Enter the number of pages:

380



Details of book 1
Name: Ikigai
Author: Gorden
Price: 800
Number of Pages: 300

Details of book 2
Name: Happiness
Author: Hector
Price: 900
Number of Pages: 380
```

LAB-4

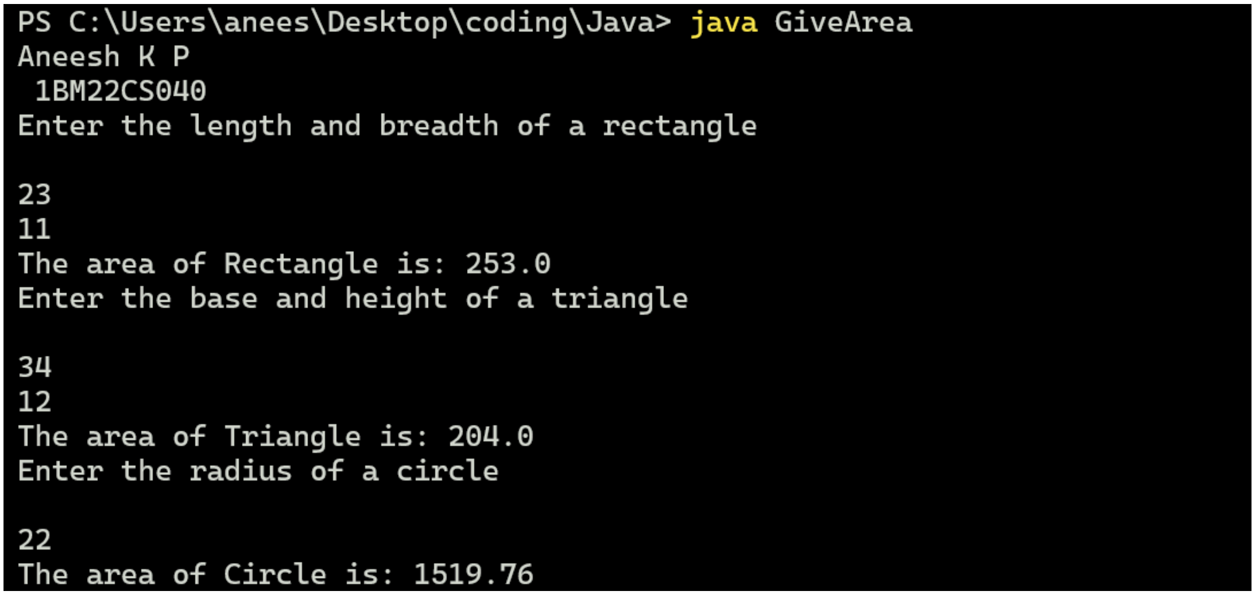
4.Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea(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

10. WAP to create an abstract clas shape and 3 child classes Tringle, Circle and Rectangle calculating the area.

```
import java.util.Scanner;

abstract class Shape{
    int a,b;
    abstract void printArea();
 }
class  Rectangle extends Shape{
        Rectangle(int l, int br){
            a = l;
             b = br;
    }
    void printArea(){
          int area = a*b;
       System.out.println("Area of rectangle "+
       area);
    }
 }
 class Triangle extends Shape {
          Triangle (int ba, int h){
             a = ba;
              b = h;
      }
     void  printArea(){
       double area = 0.5*a*b;
       System.out.println("Area of triangle "+area
    }
}
 class Circle extends Shape {
      Circle(int r){
       a = r;
  }
  void printArea(){
  double area = 3.14*a*a;
```

```
        System.out.println("Area of Circle : "+area);
    }
}
class E {
    public static void main(String[] args){
        Scanner m = new Scanner(System.in);
    System.out.println("Enter length and
        breadth of a rectangle");
            Rectangle rec = new Rectangle
        (m.nextInt(), m.nextInt());
                    rec.printArea();
            System.out.println("Enter base and
        height of a triangle");
            Triangle tri = new Triangle (m.
    nextInt(), m.nextInt());
                tri.printArea();
    System.out.println("Enter radius of
            the circle");
            Circle cir = new Circle(m.nextInt());
            cir.printArea();
    }
}.
```

OUTPUT:



```
PS C:\Users\anees\Desktop\coding\Java> java GiveArea
Aneesh K P
 1BM22CS040
Enter the length and breadth of a rectangle

23
11
The area of Rectangle is: 253.0
Enter the base and height of a triangle

34
12
The area of Triangle is: 204.0
Enter the radius of a circle

22
The area of Circle is: 1519.76
```

LAB-5

5.Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and 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 minimum balance and 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 withdrawal and update the balance

Check for the minimum balance, impose penalty if necessary and update the balance.

```
import java.util Scanner
class Account{
    String customerName;
    long accno;
    String accountType; double balance;
    Account(String customerName, long accno,
          String accountType){
      this.accno = accno;
      this.accountType = accountType;
      this.balance =0.0;
  }
    public void displayBalance(){
     System.out.println("Accont Number"+accno);
     System.out.println("Customer Name:" + customerName
    System.out.println("Account Type:"+accountType);
    System.out.println("Balance :₹" + balance);
   }
}
Class CorrAcct extends Accant{
    double minBalance;
    double serviceCharge;
    ~~public~~ CurAcct(String customerName, long
  {
     super(customerName, accno, "Current");
   this.minBalance = 500.0;
    this.serviceCharge=50.0;
 }
  public void withdraw(double amount){
    if (balance - amount >=minBalance)
    {
```

```
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: ₹ "+ balance);
     }
  }
}
  class SavAcct extends Account{
        double intrestRate;
    SavAcct (String customerName, long accno){
     super( customerName, accno, "Savings");
     this. intrestRate = 0.05;
  }
  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: ₹ "+ balance);
   }
}
  public class Bank{
  public static void main(String[] args){
   Scanner scanner = new Scanner(System.in);
   System.out.println("choose account t    ")
System.out.println("Enter choice (1 or 2)");
    int choice = scanner.nextInt();
  System.out.println(" Enter customer name");
      String customerName = scanner.next();
System.out.println( "Enter account number");
     ~~String~~ long accno = scanner.nextLog();
   if (choice == 1) {
   CurrAcct currAccount = new CurrAcct(customerName,
                 accno);
  System.out.print("Enter initial balance: ₹");
  double initialBalance = scanner.nextDouble();
  currAccount.balance = initialBalance;
  System.out.println("Enter withdrawal amount: ₹");
  double withdrawalAmount = scanner.nextDouble();
currAccount.withdraw(withdrawalAmount);

 currAccount.imposeServiceCharge();
   currAccount.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.println("Enter the withdrawal amount: ₹");
double withdrawalAmount = scanner.nextDouble();
 savAccount.balance -= withdrawalAmount;
System.out.println("Withdrawal successful. Current
   balance: ₹" + balance);
 system.out.println("Enter interest rate");
 double interestRate = interestRate;
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");
}

}

}
```

Algorithm:

ep 1: Start

2: Define ~~Acc~~ class Account and define a constructor to initialize the instance variables acc. no., account type and balance.

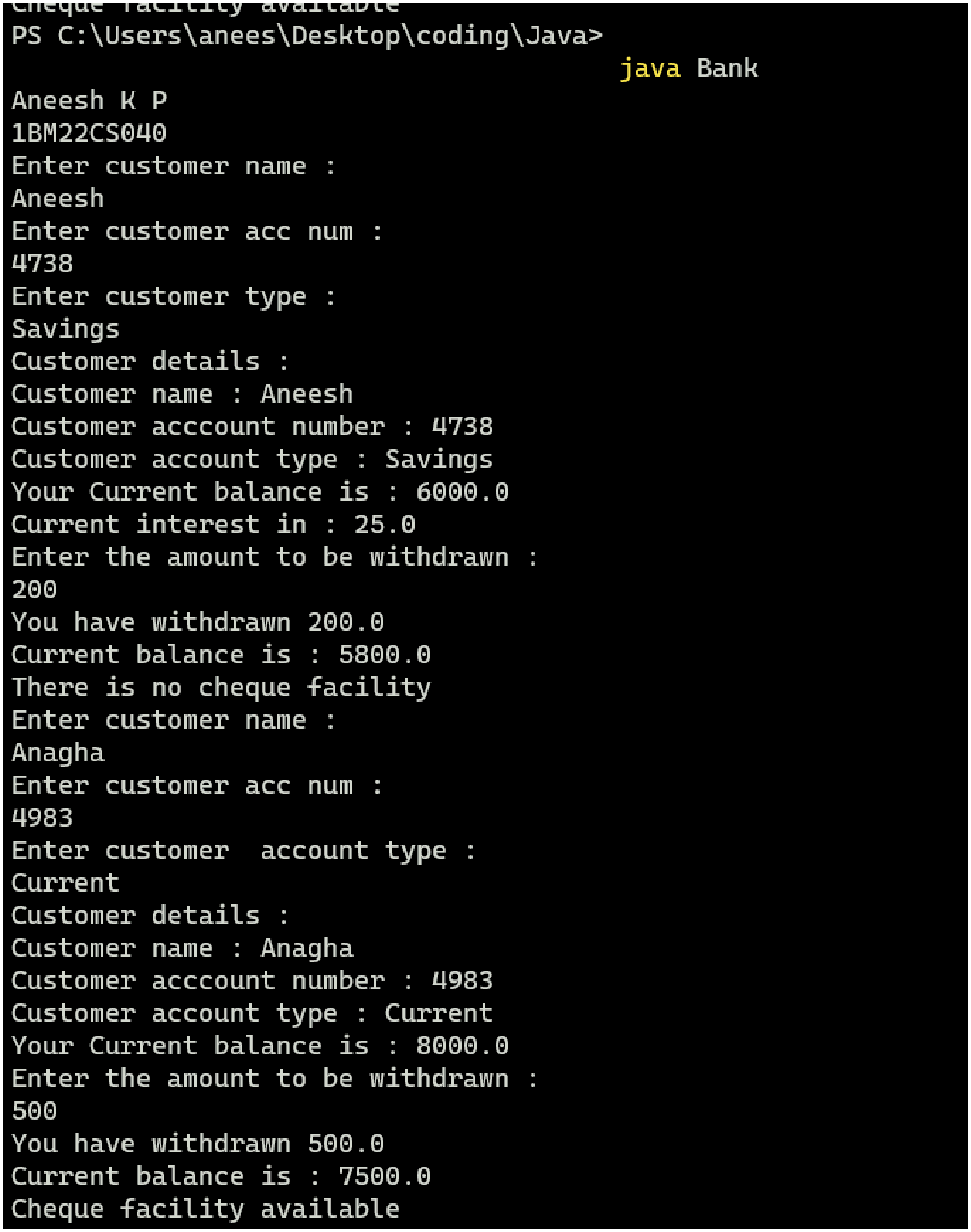
p3: Define a function display Balance, which prints the Customer name, Account ~~name~~ number and type and Balance.

4: Inherit Account to CurAcct class and define a constructor to initialize the instance variables

5: Create method withdraw which handles withdrawals

```
        if (balance - amount > minBalance)
        {
                balance -= amount;
        }
        else
        {
            Print "Insufficient Balance"
        }
```

OUTPUT:



```
Cheque facility available
PS C:\Users\anees\Desktop\coding\Java>
                                                java Bank
Aneesh K P
1BM22CS040
Enter customer name :
Aneesh
Enter customer acc num :
4738
Enter customer type :
Savings
Customer details :
Customer name : Aneesh
Customer acccount number : 4738
Customer account type : Savings
Your Current balance is : 6000.0
Current interest in : 25.0
Enter the amount to be withdrawn :
200
You have withdrawn 200.0
Current balance is : 5800.0
There is no cheque facility
Enter customer name :
Anagha
Enter customer acc num :
4983
Enter customer  account type :
Current
Customer details :
Customer name : Anagha
Customer acccount number : 4983
Customer account type : Current
Your Current balance is : 8000.0
Enter the amount to be withdrawn :
500
You have withdrawn 500.0
Current balance is : 7500.0
Cheque facility available
```

LAB-6

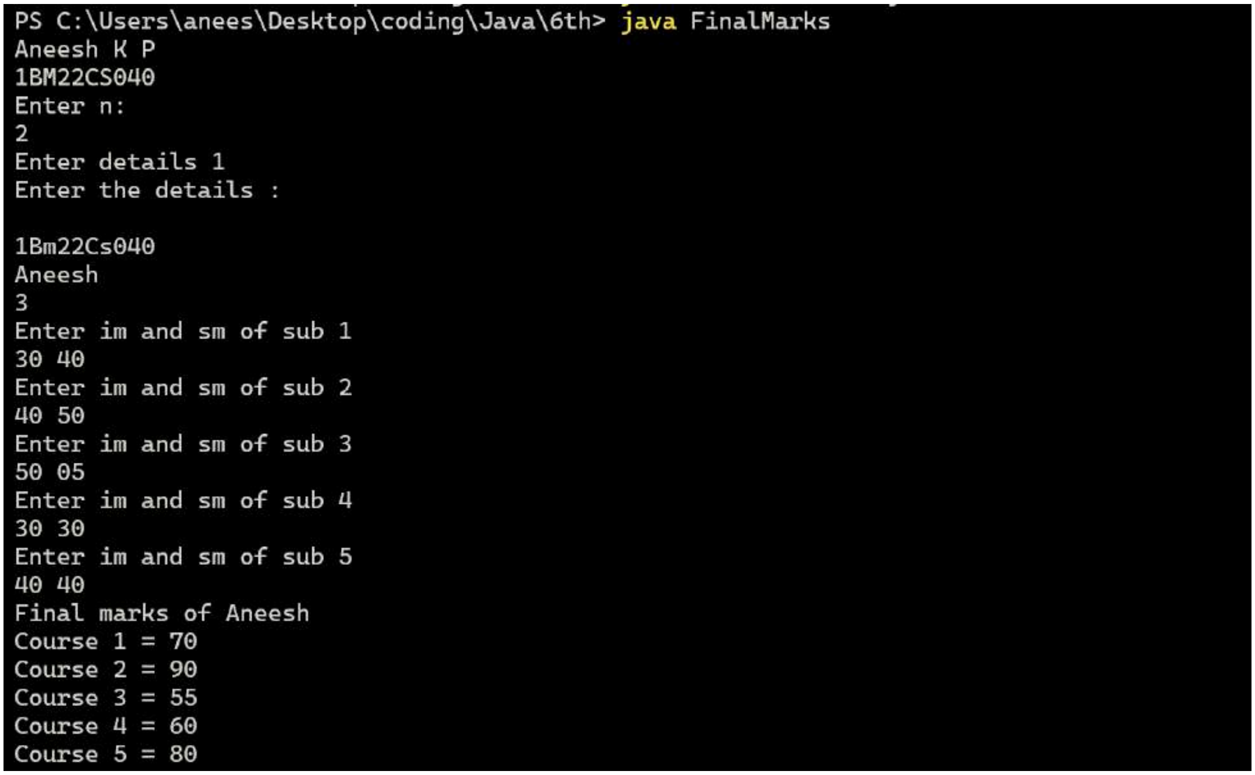
6.Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

6. WAP to create CIE package class student and Internals (USN, name, sem) -> student and Internals ->(array that stores marks). Create another package see which has the class external which is derived from class student.

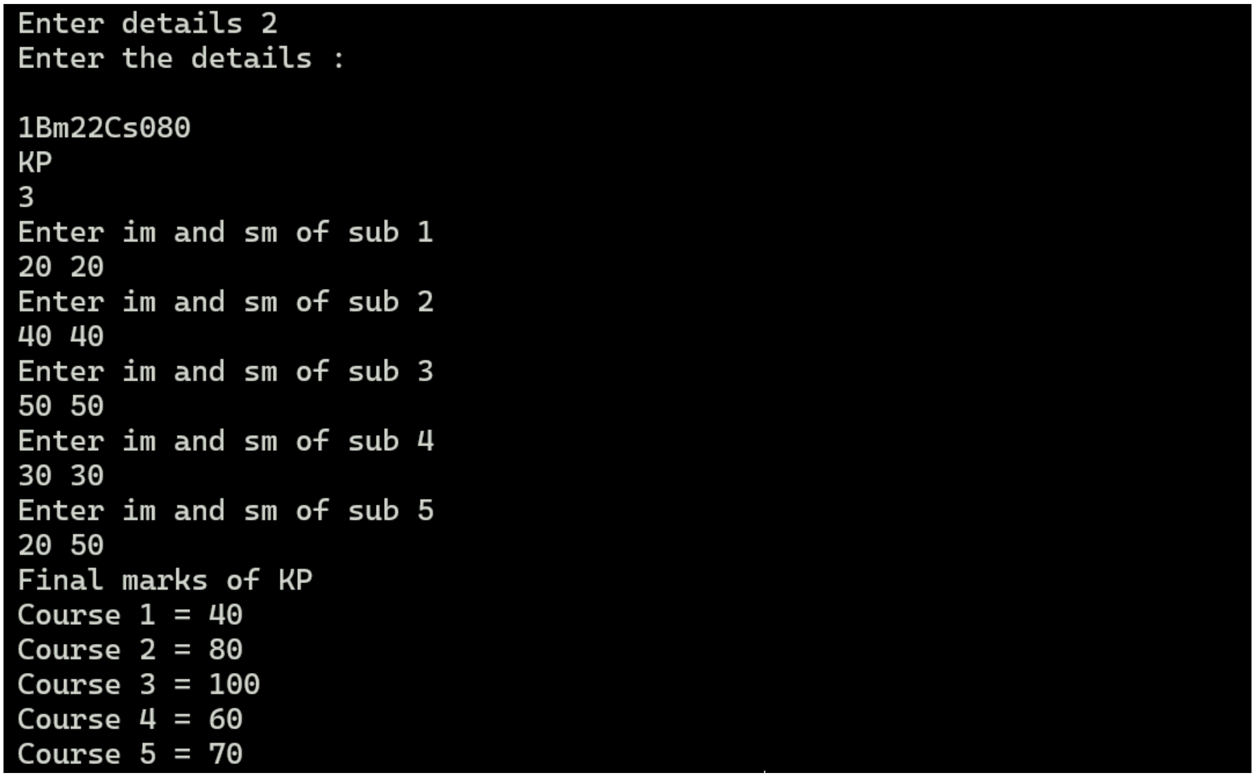
```
package CIE;
import java.util.*;
public class Student
{
        public int sem;
        public String usn;
        public String name;
        public void accept()
     {
      Scanner s = new Scanner(System.in);
     system.out.println("Enter USN: \n");
                usn = s.nextLine()
                name = s.nextLine();
                sem = s.nextLine();
     }
  }

Package CIE;
public class Internals
{
     public int im[] = new int[5];
}

package SEE;
import CIE.Student;
public class External extends Student
{
          lic int s[] = new int[5].
```

```
Import javd.util.*;
import SEE.*;
Import CIE.*;
public ~~class~~ static void main (String args[])
{
    int fm[] = new int[5];
    Scanner s = new Scanner (System.in);
    System.out.println ("Enter n: ");
    int n = s.nextInt();
    SEE.Externals st[] = new SEE.Extend[n];
    CEE.Intenals s[] = new CIE.Intenals[n];

  for(int i=0; i<n; i++)
  {
      st[i] = new SEE.Externals();
      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] = s.nextInt();
     st[i].sm[j] = s.nextInt();
    fm[j] = s[i].im[j] + st[i].sm[j];
  }
  System.out.println(" ~~Enter fin~~ final marks
        of " + st[i].name);
  for(int k=0; k<5; k++)
  {
    System.out.println("Course" + (k+1) + " = " +
                  fm[k]);
  }
  }
}
```

OUTPUT:



```
PS C:\Users\anees\Desktop\coding\Java\6th> java FinalMarks
Aneesh K P
1BM22CS040
Enter n:
2
Enter details 1
Enter the details :

1Bm22Cs040
Aneesh
3
Enter im and sm of sub 1
30 40
Enter im and sm of sub 2
40 50
Enter im and sm of sub 3
50 05
Enter im and sm of sub 4
30 30
Enter im and sm of sub 5
40 40
Final marks of Aneesh
Course 1 = 70
Course 2 = 90
Course 3 = 55
Course 4 = 60
Course 5 = 80
```



```
Enter details 2
Enter the details :

1Bm22Cs080
KP
3
Enter im and sm of sub 1
20 20
Enter im and sm of sub 2
40 40
Enter im and sm of sub 3
50 50
Enter im and sm of sub 4
30 30
Enter im and sm of sub 5
20 50
Final marks of KP
Course 1 = 40
Course 2 = 80
Course 3 = 100
Course 4 = 60
Course 5 = 70
```

LAB-7

7.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 and throws the exception WrongAge( ) when the input age<0. In Son class, implement a constructor that cases both father and son’s age and throws an exception if son’s age is >=father’s age.

7. WAP that demonstrate handling of exceptions. Create a base class called Father and derived class called 'SON' which extends the base class. In Father class, implement a constructor which takes age and throws WrongAge() when input age<0.

```
class WrongAgeException extends Exception {
      public WrongAgeException(String message) {
                super(message);
      }
 }
class father {
        int d_age;
    public father(int a) throws WrongAgeException {
        if (a<0) {
                  throw new WrongAgeException ("Age
    is less than zero!");
          }
              d_age =a;
       }
 }
class son extends father {
      int s_age;
    public son(int d_age, int s) throws
   WrongAgeException {
        super(d_age);
      if(d_age <s)
     {
       throw new WrongAgeException ("father age c
   be less than son!");
     }
       s_age = s;
 }
3
```

```
public class Main {
  public static void main(String[] args) {
  ~~System.out.println("Hello World");~~
try {
      son s1 = new son (-1, 20);
   }
catch (WrongAge Exception e ) {
      System.out.println(" Exception: " +
      e.getMessage());
   }
 }
}
```

Algorithm:

Step 1:- Start

ep 2:- Create a class WrongAgeException extending exception class which prints the message

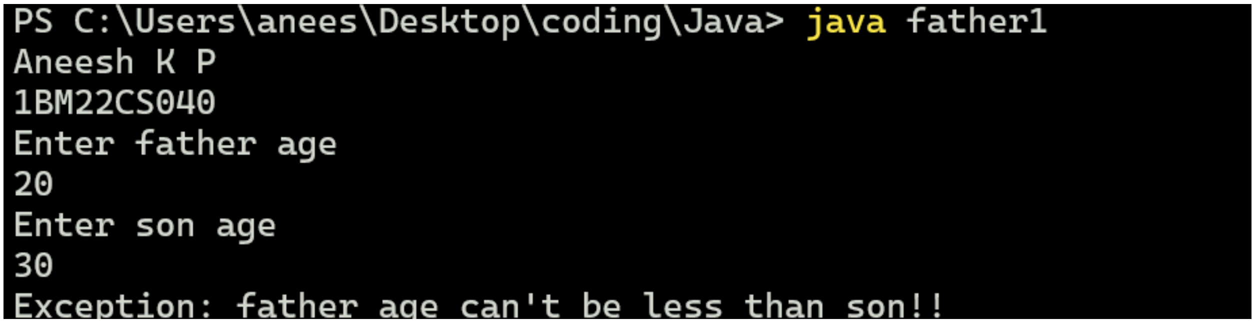
p 3: Create class father which throws WrongAge Exception

p 4: Create a class son which extends father which throws wrongageexception when age of father is less than that of son.

5: In the Main Class, Give a try block where we create object of the class son and check if it throws the WrongAge Exception if depending on the age input of father and son. Catch the exception if needed

6: Stop

OUTPUT;



```
PS C:\Users\anees\Desktop\coding\Java> java father1
Aneesh K P
1BM22CS040
Enter father age
20
Enter son age
30
Exception: father age can't be less than son!!
```

LAB-8

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.

8. WAP which create two threads, one thread displaying BMSCE once every 10 sec and another displaying CSE every 2 sec.

```
class ps1 implements Runnable{
    public void run()
    {
      for (int i=0; i<5; i++){
      System.out.println("BMS College of
           Engineering");
      try {
          Thread.sleep(10000);
      }
      catch (Exception e)
      {   e.printStackTrace(); }
    }
   }
}
 class ps2 implements Runnable{
  public void run()
   {
     for(int i=0; i<5; i++)
     {
      System.out.println("CSE");
     try {
         Thread.sleep(2000);
     }
    catch(Exception e){ e.printStackTrace();}
   }
  }
}
```

```
public class Main{
  public static void main(String[] args){
     ps1 p1 = new ps1();
     ps2 p2= new ps2();
     Thread t1= new Thread(p1);
     Thread t2= new Thread(p2);
       t1.start();   t2.start();
   }
}
```

Algorithm:

Step1: Start

Step 2: Create a class ps1 which implements the Runnable interface

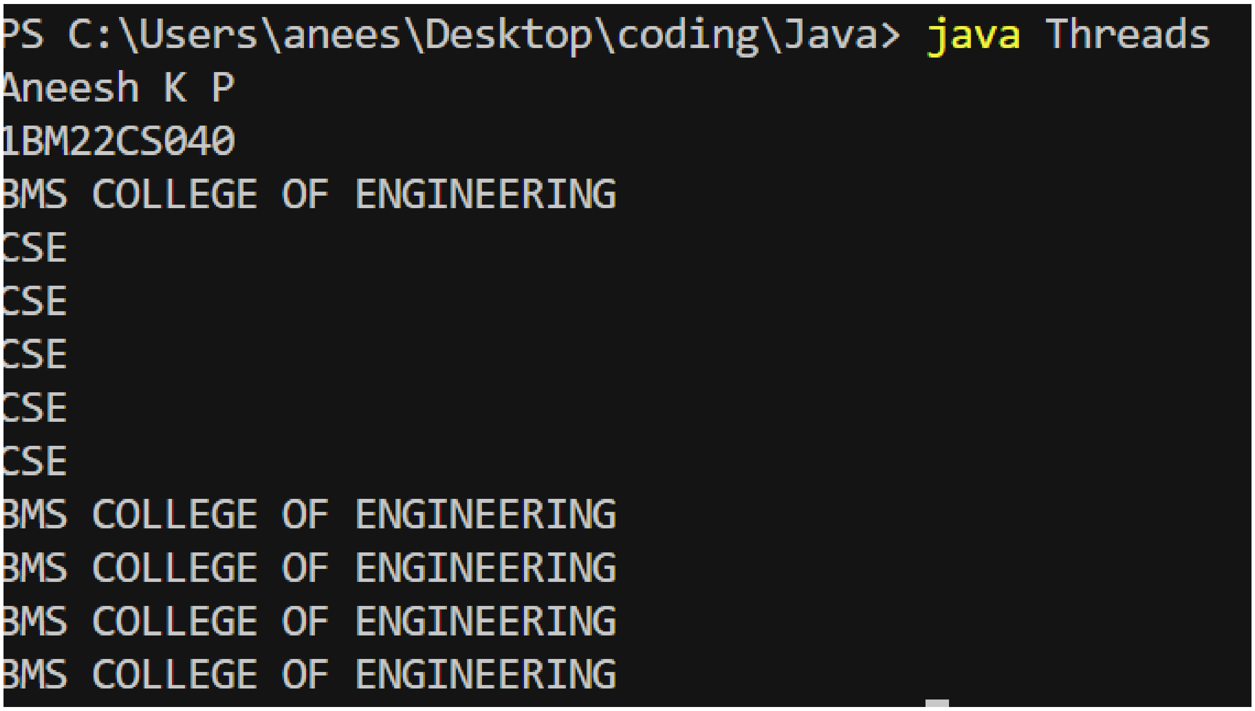
Step 3: Create a run() function which prints "BMS College of Engineering" and let the thread suspend for 10 secs using sleep() function.
Keep Thread.sleep() inside try block and catch the exception and print it using printStackTrace()

Step4: Create class ps2 which implements the Runnab interface

Step5: Create a run() function which prints "CSE" and the thread suspend for 2 secs using sleep() functio
Keep Thread.sleep() in try block and catch th exception and print it using printStackTrace()

Step6: In the Main public class, create objects of the class ps1 and ps2 as p1 and p2 respectively.

OUTPUT:



```
PS C:\Users\anees\Desktop\coding\Java> java Threads
Aneesh K P
1BM22CS040
BMS COLLEGE OF ENGINEERING
CSE
CSE
CSE
CSE
CSE
BMS COLLEGE OF ENGINEERING
BMS COLLEGE OF ENGINEERING
BMS COLLEGE OF ENGINEERING
BMS COLLEGE OF ENGINEERING
```