VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

Submitted by

Shana Diya Sujit(1BM21CS196)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
October-2022 to Feb-2023

B. M. S. College of Engineering,

Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

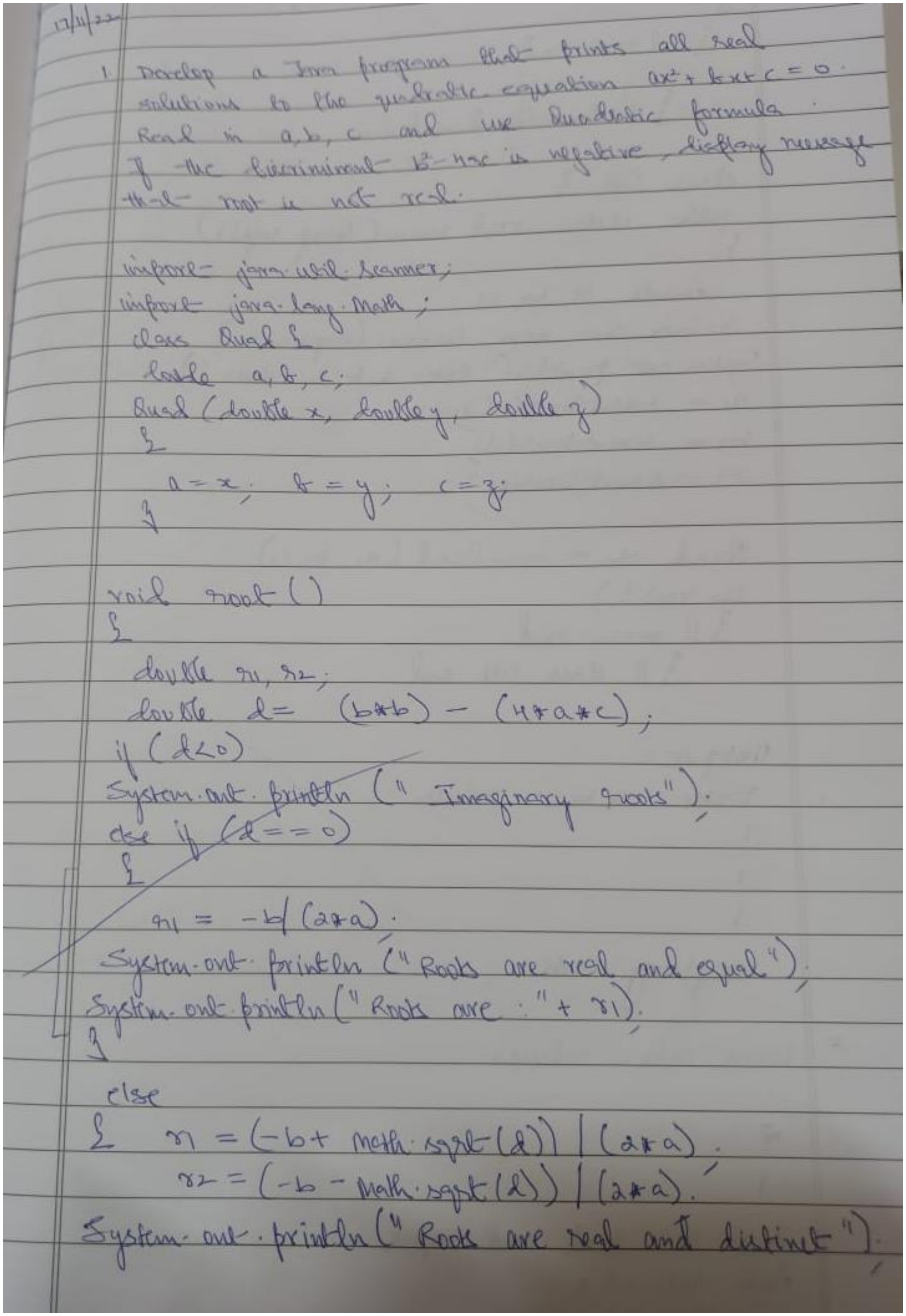
This is to certify that the Lab work entitled “Database Management Systems (22CS3PCDBM)” carried out by **Shana Diya Sujit (1BM21CS196)**, who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022. The Lab report has been approved as it satisfies the academic requirements in respect of a **Course Object Oriented Java Programming - (21CS3PCOOJ)** workprescribed for the said degree.

Vikranth Bm
Name of the Lab-Incharge
Designation
Department of CSE
BMSCE, Bengaluru

Dr. Jyothi S Nayak
Professor and Head
Department of CSE
BMSCE, Bengaluru

Index

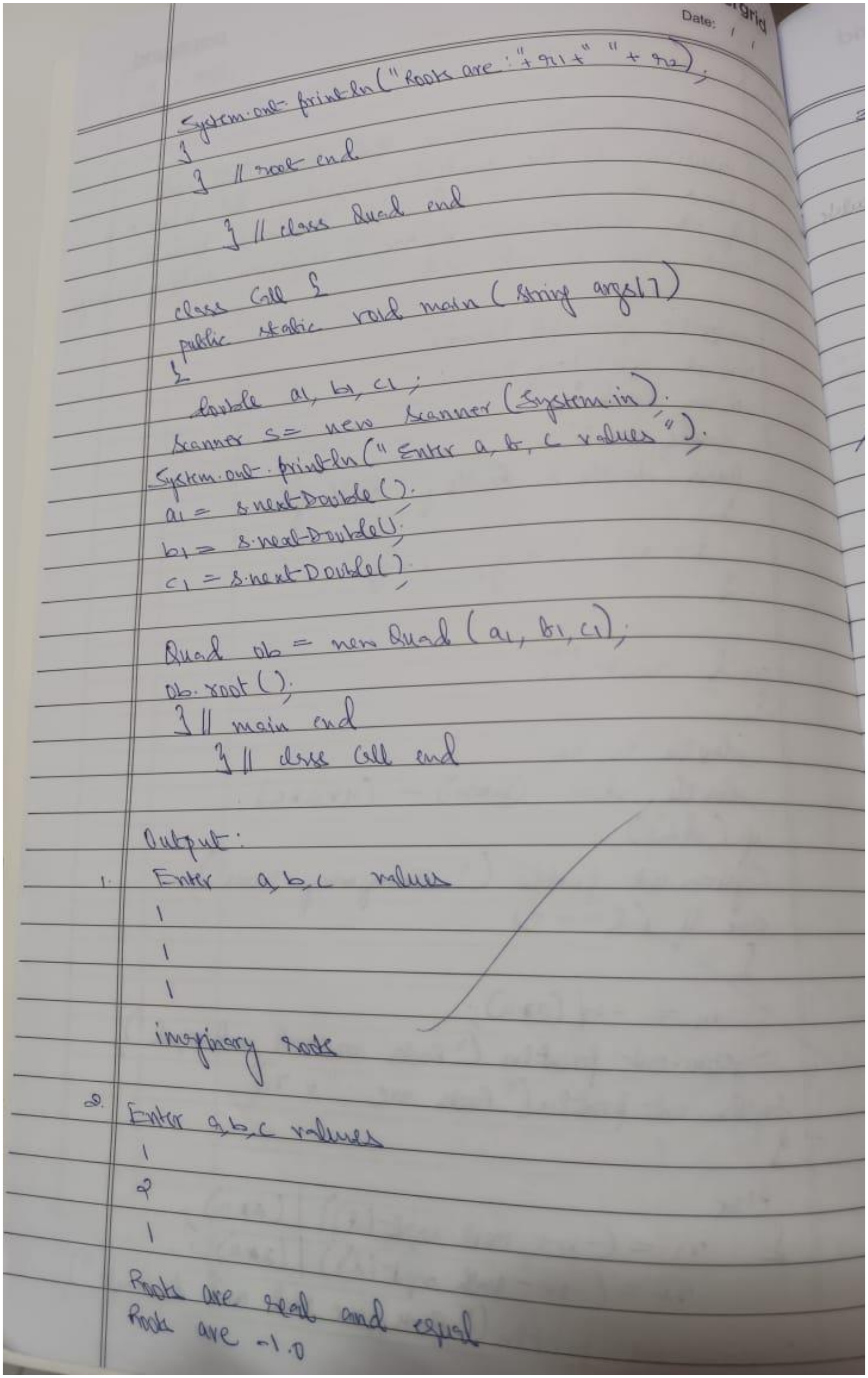
1.Develop a Java program that prints all real solutions to the quadratic equation ax2+bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b2 -4ac is negative, display a message stating that there are no real solutions.



17/11/22

1. 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 Quadratic formula. If the discriminant b^2-4ac is negative, display message that root is not real.

```
import java.util.Scanner;
import java.lang.Math;
class Quad {
 double a, b, c;
Quad (double x, double y, double z)
{
   a=x; b=y; c=z;
}

void root ()
{
 double r1, r2;
 double d= (b*b) - (4*a*c);
if (d<0)
System.out.println ("Imaginary roots");
else if (d==0)
{
   r1 = -b/(2*a);
System.out.println ("Roots are real and equal");
System.out.println ("Roots are :" + r1);
}
else
{  r1 =(-b+ math.sqrt(d)) / (2*a);
    r2 =(-b - Math.sqrt(d)) / (2*a);
System.out.println ("Roots are real and distinct");
```

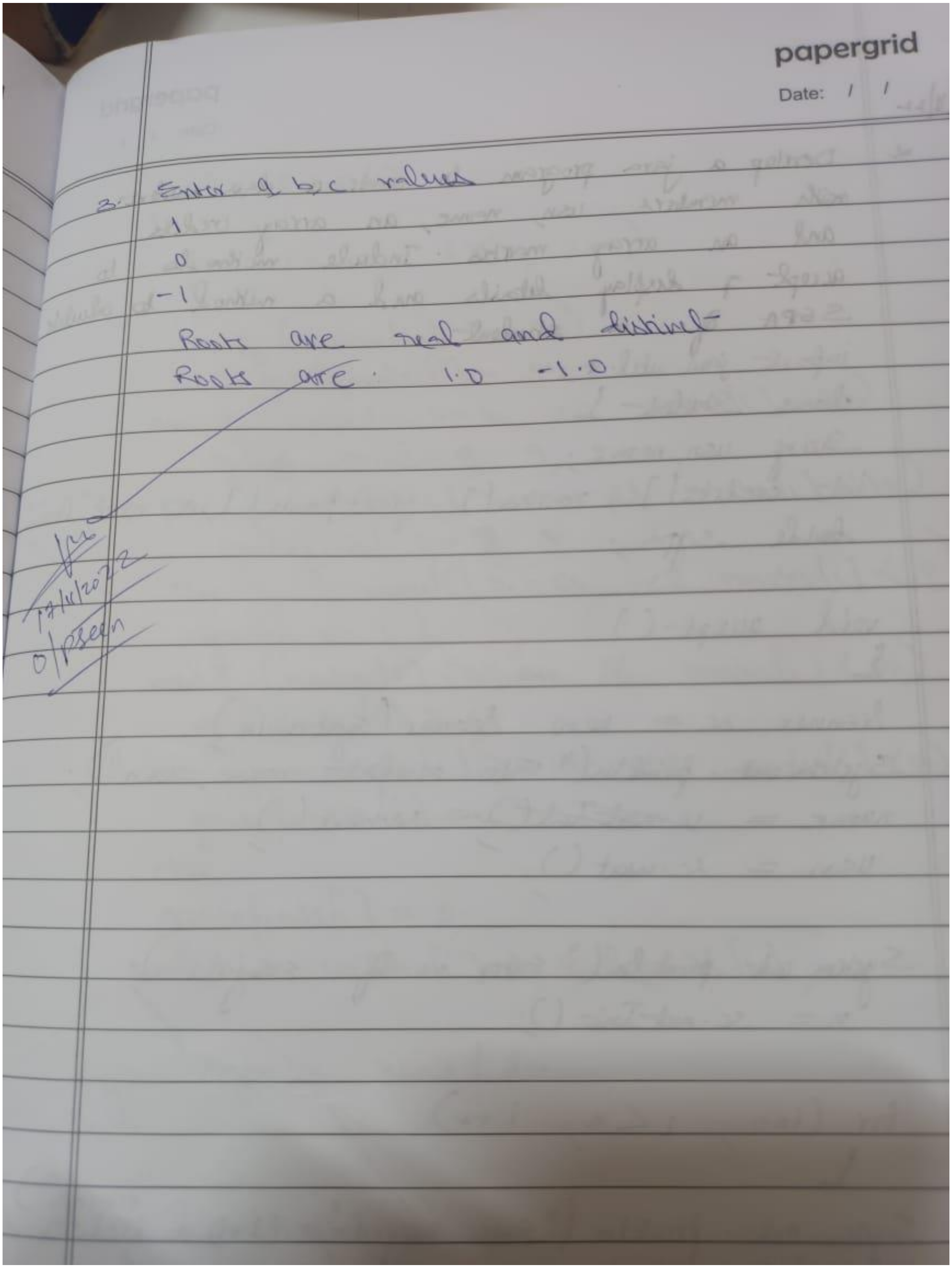


Date: / /

```
System.out.println("Roots are :"+r1+" "+r2);
}
} // root end
    } // class Quad end

class Call {
public static void main ( String args[])
{
  double a1, b1, c1;
Scanner s= new Scanner (System.in).
System.out.println(" Enter a, b, c values ").
a1 = s.nextDouble().
b1 = s.nextDouble();
c1 = s.nextDouble()

Quad ob = new Quad (a1, b1, c1);
ob.root();
} // main end
   } // class Call end
```

Output:

```
1. Enter a,b,c values
   1
   1
   1
   imaginary roots
2. Enter a,b,c values
   1
   2
   1
Roots are real and equal
Roots are -1.0
```



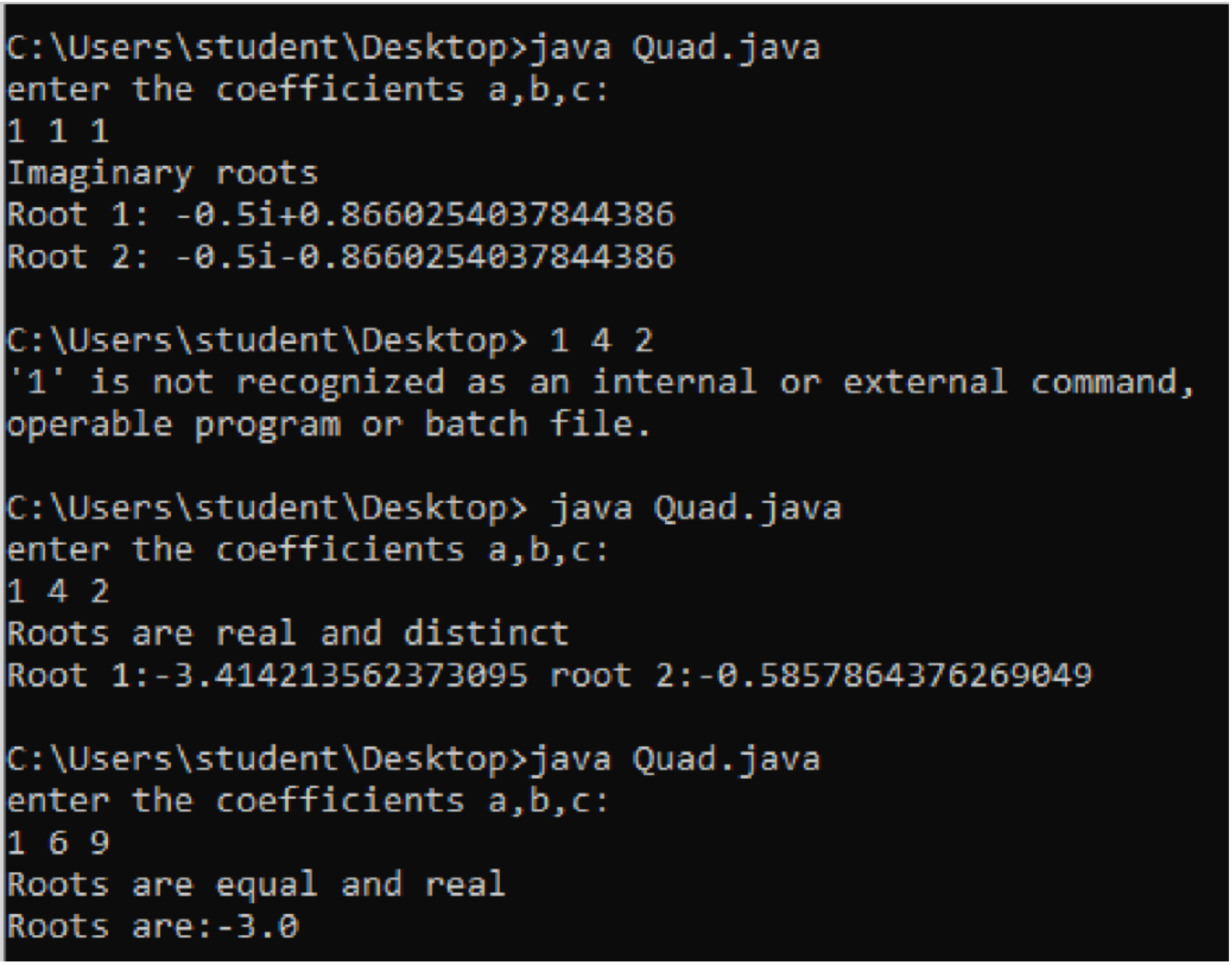
papergrid

Date: / /

3.
```
Enter a b,c values
1
0
-1
Roots are real and distinct
Roots are . 1.0 -1.0
```

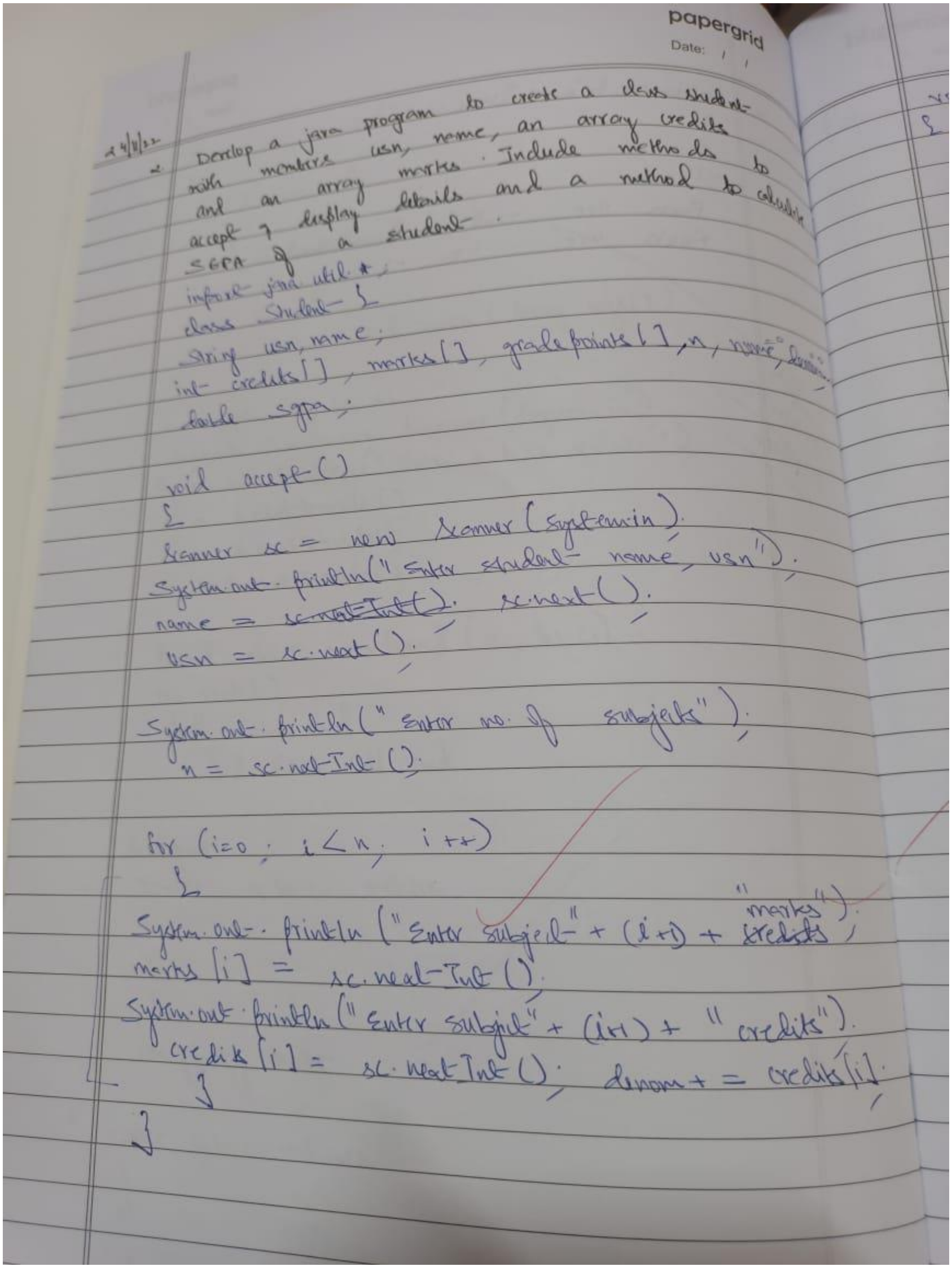
17/11/2022
O/P seen

OUTPUT:



```
C:\Users\student\Desktop>java Quad.java
enter the coefficients a,b,c:
1 1 1
Imaginary roots
Root 1: -0.5i+0.8660254037844386
Root 2: -0.5i-0.8660254037844386

C:\Users\student\Desktop> 1 4 2
'1' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\student\Desktop> java Quad.java
enter the coefficients a,b,c:
1 4 2
Roots are real and distinct
Root 1:-3.414213562373095 root 2:-0.5857864376269049

C:\Users\student\Desktop>java Quad.java
enter the coefficients a,b,c:
1 6 9
Roots are equal and real
Roots are:-3.0
```

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.



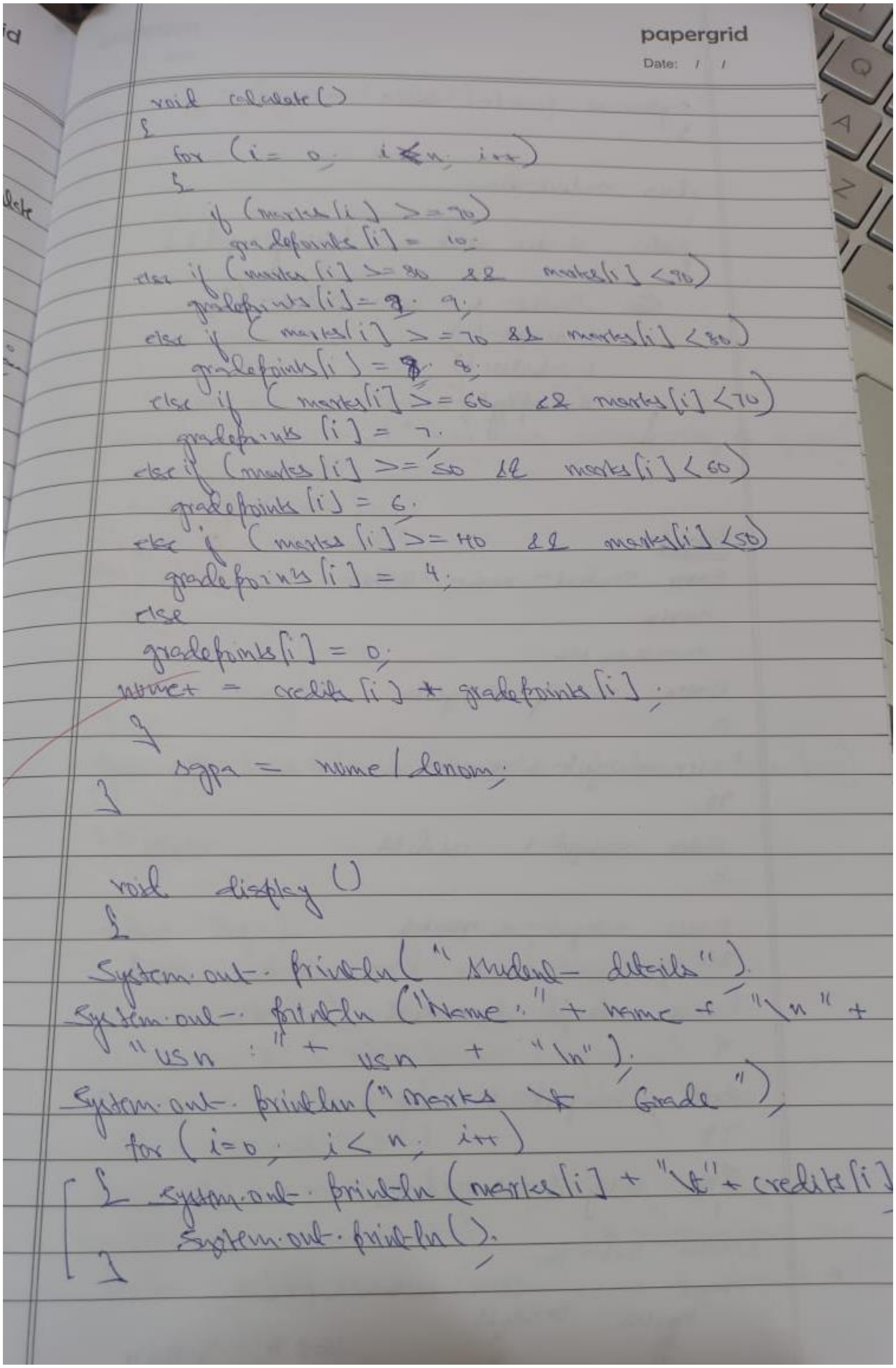
papergrid

Date: / /

24/11/22

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

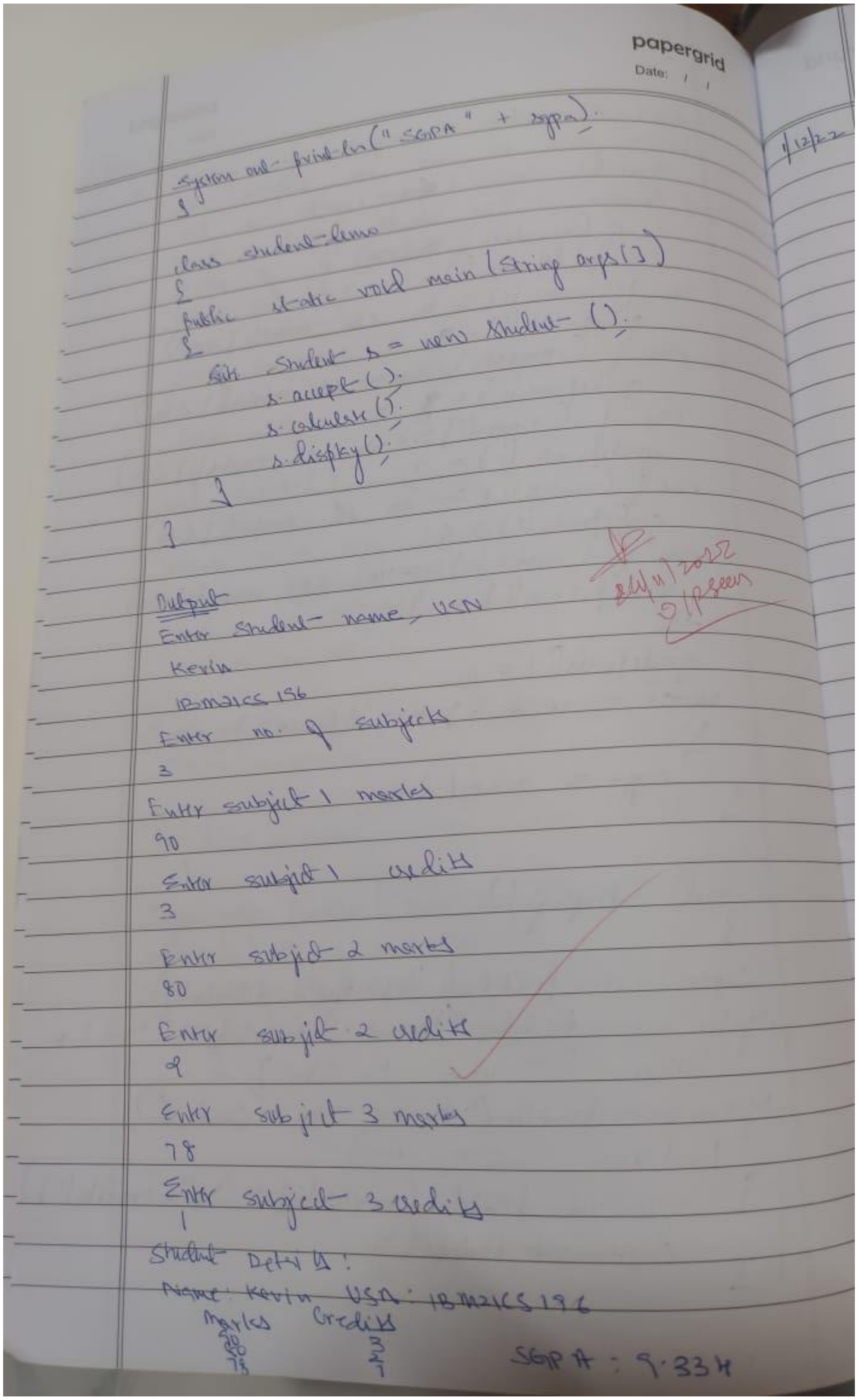
```
import java.util.*;
class Student {
String usn, name;
int credits[], marks[], gradepoints[], n, nume, denom;
double sgpa;

void accept()
{
Scanner sc = new Scanner(system.in);
System.out.println(" Enter student name, usn");
name = ~~sc.nextInt();~~ sc.next();
usn = sc.next();

System.out.println(" Enter no. of subjects");
 n = sc.nextInt();

for (i=0; i<n; i++)
 {
System.out.println("Enter subject" + (i+1) + "marks" ~~credits~~);
marks[i] = sc.nextInt();
System.out.println("Enter subject" + (i+1) + "credits");
  credits[i] = sc.nextInt(); denom+ = credits[i];
   }
}
```



papergrid

Date: / /

```
void calculate()
{
  for (i= 0; i~~<~~<n; i++)
  {
      if (marks[i] >=90)
        gradepoints[i] = 10;
else if (marks[i] >= 80  &&  marks[i] <90)
     gradepoints[i]= ~~9~~. 9;
 else if ( marks[i] > =70 && marks[i] <80)
     gradepoints[i] = ~~8~~. 8;
  else if ( marks[i] >= 60   && marks[i] <70)
    gradepoints [i] = 7.
 else if (marks[i] >= 50  &&  marks[i] < 60)
   gradepoints [i] = 6.
 else if ( marks [i] >= 40  &&  marks[i] <50)
   gradepoints [i] =  4;
 else
  gradepoints[i] = 0;
 nume+ = credits [i] * gradepoints [i] ;
  }
    sgpa = nume / denom;
}


 void  display ()
 {
 System.out.println( " student- details" ).
System.out.println ("Name :" + name + "\n" +
  "usn : " + usn + "\n" ).
System.out.println(" marks \t Grade "),
   for ( i=0; i< n; i++)
[ { System.out.println (marks[i] + "\t"+ credits[i]
      System.out.println().
[ }
```



papergrid
Date: / /

1/12/22

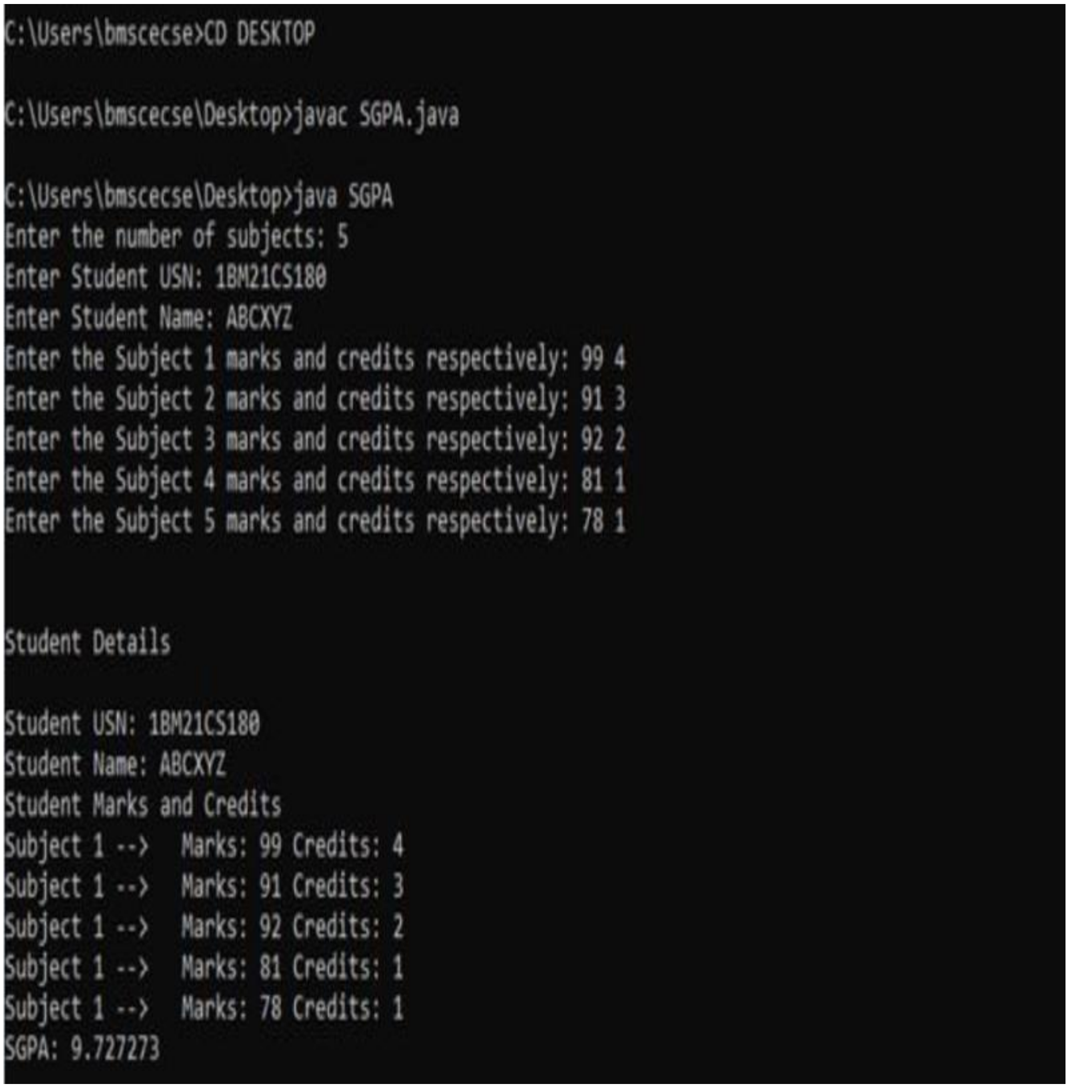
```
System out println (" SGPA " + sgpa);
}

class student_demo
{
public static void main (String args[])
{
    Student s = new Student ();
        s. accept ();
        s. calculate ();
        s. display ();
    }
}
```

8/11/2022

Output

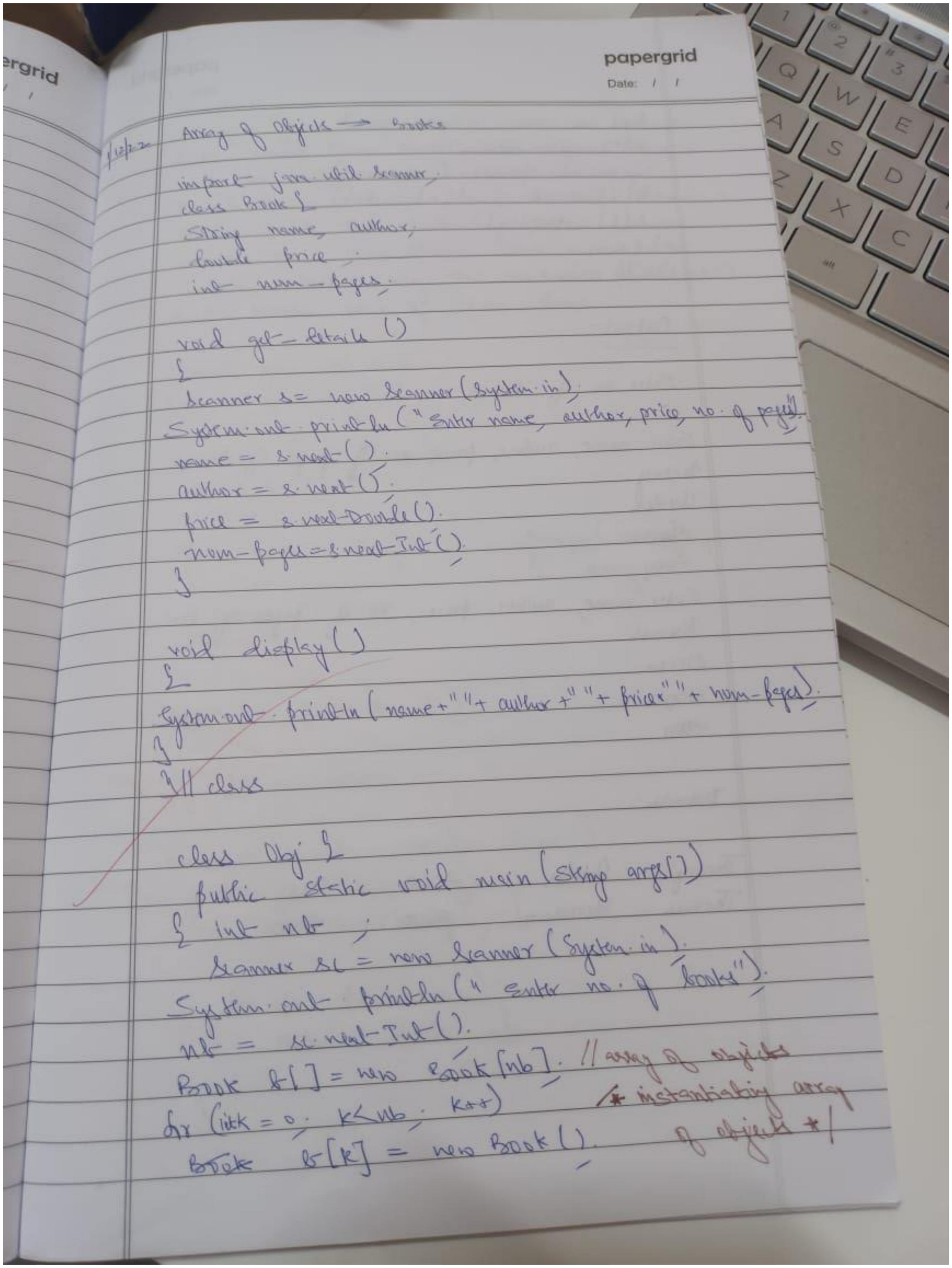
```
Enter student name, USN
 Kevin
 1BM21CS 156
Enter no. of subjects
3
Enter subject 1 marks
90
Enter subject 1 credits
3
Enter subject 2 marks
80
Enter subject 2 credits
2
Enter subject 3 marks
78
Enter subject 3 credits
 1
Student Details :
Name: Kevin  USN : 1BM21CS196
   Marks   Credits
    90        3
    80        2
    78        1
                          SGPA : 9.334
```

OUTPUT:



```
C:\Users\bmscecse>CD DESKTOP

C:\Users\bmscecse\Desktop>javac SGPA.java

C:\Users\bmscecse\Desktop>java SGPA
Enter the number of subjects: 5
Enter Student USN: 1BM21CS180
Enter Student Name: ABCXYZ
Enter the Subject 1 marks and credits respectively: 99 4
Enter the Subject 2 marks and credits respectively: 91 3
Enter the Subject 3 marks and credits respectively: 92 2
Enter the Subject 4 marks and credits respectively: 81 1
Enter the Subject 5 marks and credits respectively: 78 1


Student Details

Student USN: 1BM21CS180
Student Name: ABCXYZ
Student Marks and Credits
Subject 1 -->   Marks: 99 Credits: 4
Subject 1 -->   Marks: 91 Credits: 3
Subject 1 -->   Marks: 92 Credits: 2
Subject 1 -->   Marks: 81 Credits: 1
Subject 1 -->   Marks: 78 Credits: 1
SGPA: 9.727273
```

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.

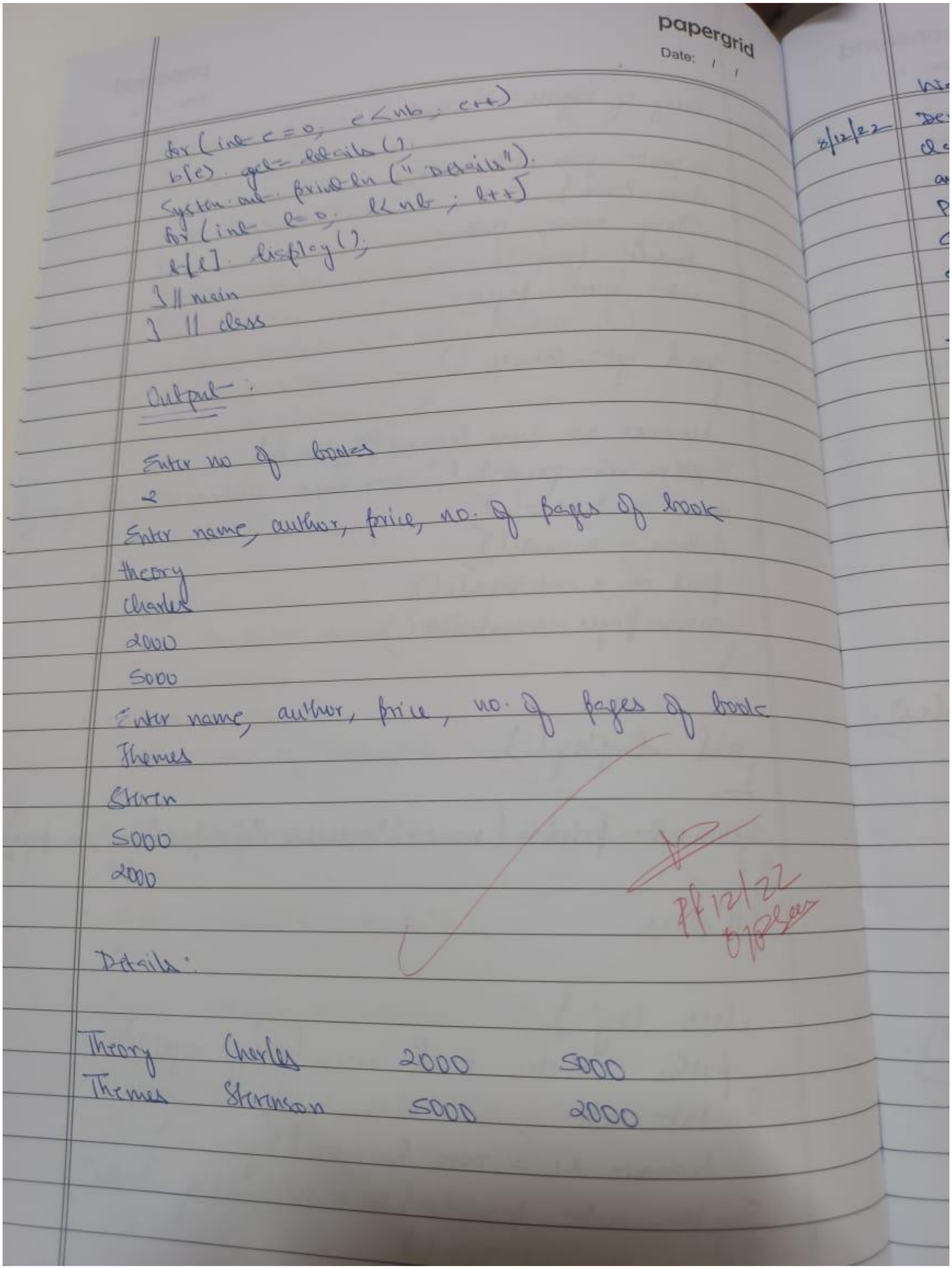


papergrid

Date: / /

1/12/22 Array of Objects → Books

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

void get_details ()
{
 Scanner s= new Scanner(System.in);
System.out.println ("Enter name, author, price, no. of pages");
name = s.next();
author = s.next();
 price = s.nextDouble();
 num_pages=s.nextInt();
}

void display()
{
System.out.println(name+" "+ author +" "+ price+" "+ num_pages);
}
}// class

 class Obj {
  public static void main(String args[])
 { int nb ;
    Scanner sc = new Scanner (System.in);
 System.out.println(" Enter no. of books");
 nb = sc.nextInt();
 Book b[] = new Book[nb]; // array of objects
for (int k = 0; k<nb; k++)        /* instantiating array
 Book b[k] = new Book();                  of objects */
```



papergrid

Date: / /

```
for (int c=0; c<nb; c++)
b[c].getdetails();
System.out.println("Details");
for (int l=0; l<nb; l++)
b[l].display();
}// main
} // class
```

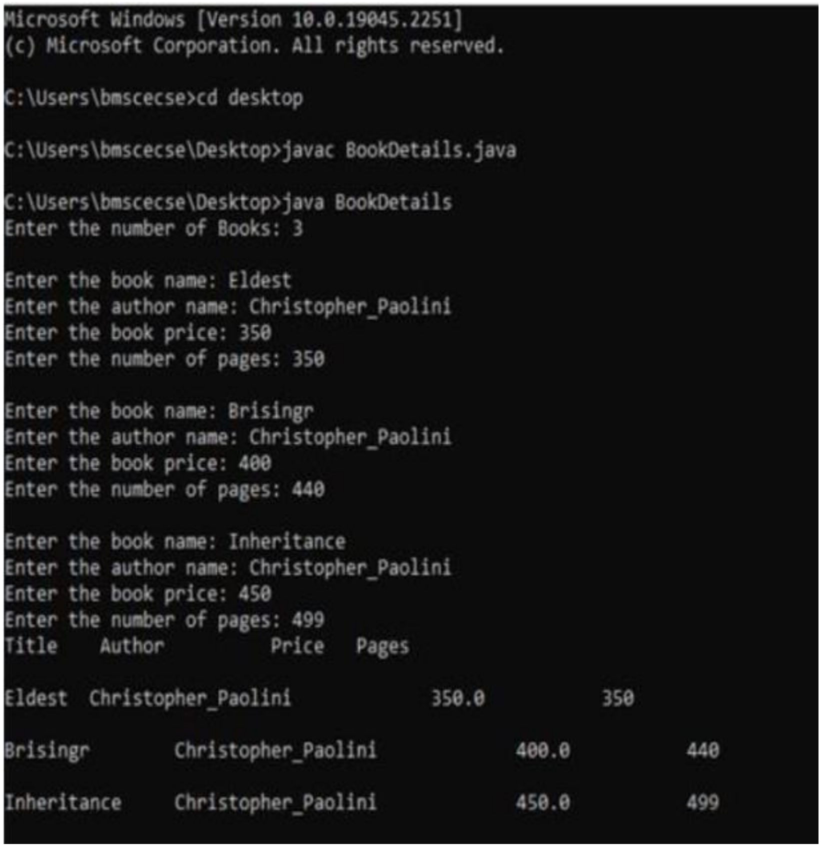
Output:

```
Enter no of books
2
Enter name, author, price, no. of pages of book
theory
charles
2000
5000
Enter name, author, price, no. of pages of book
Themes
Steven
5000
2000

Details:

Theory   Charles     2000   5000
Themes   Stevenson   5000   2000
```

8/12/22

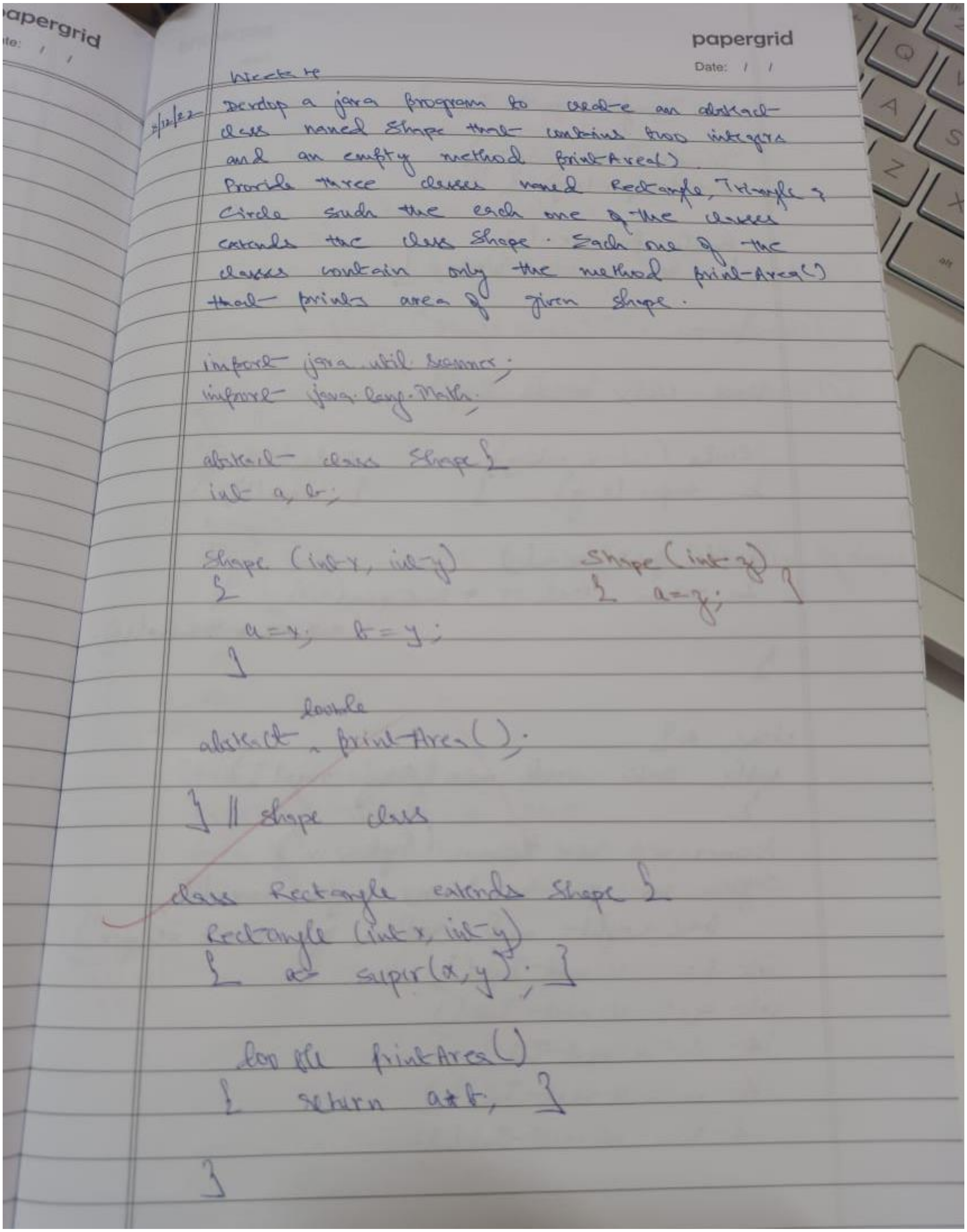
8/12/22

OUTPUT:



```
Microsoft Windows [Version 10.0.19045.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bmscecse>cd desktop

C:\Users\bmscecse\Desktop>javac BookDetails.java

C:\Users\bmscecse\Desktop>java BookDetails
Enter the number of Books: 3

Enter the book name: Eldest
Enter the author name: Christopher_Paolini
Enter the book price: 350
Enter the number of pages: 350

Enter the book name: Brisingr
Enter the author name: Christopher_Paolini
Enter the book price: 400
Enter the number of pages: 440

Enter the book name: Inheritance
Enter the author name: Christopher_Paolini
Enter the book price: 450
Enter the number of pages: 499
Title    Author          Price   Pages

Eldest  Christopher_Paolini             350.0           350

Brisingr        Christopher_Paolini             400.0           440

Inheritance     Christopher_Paolini             450.0           499
```

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.

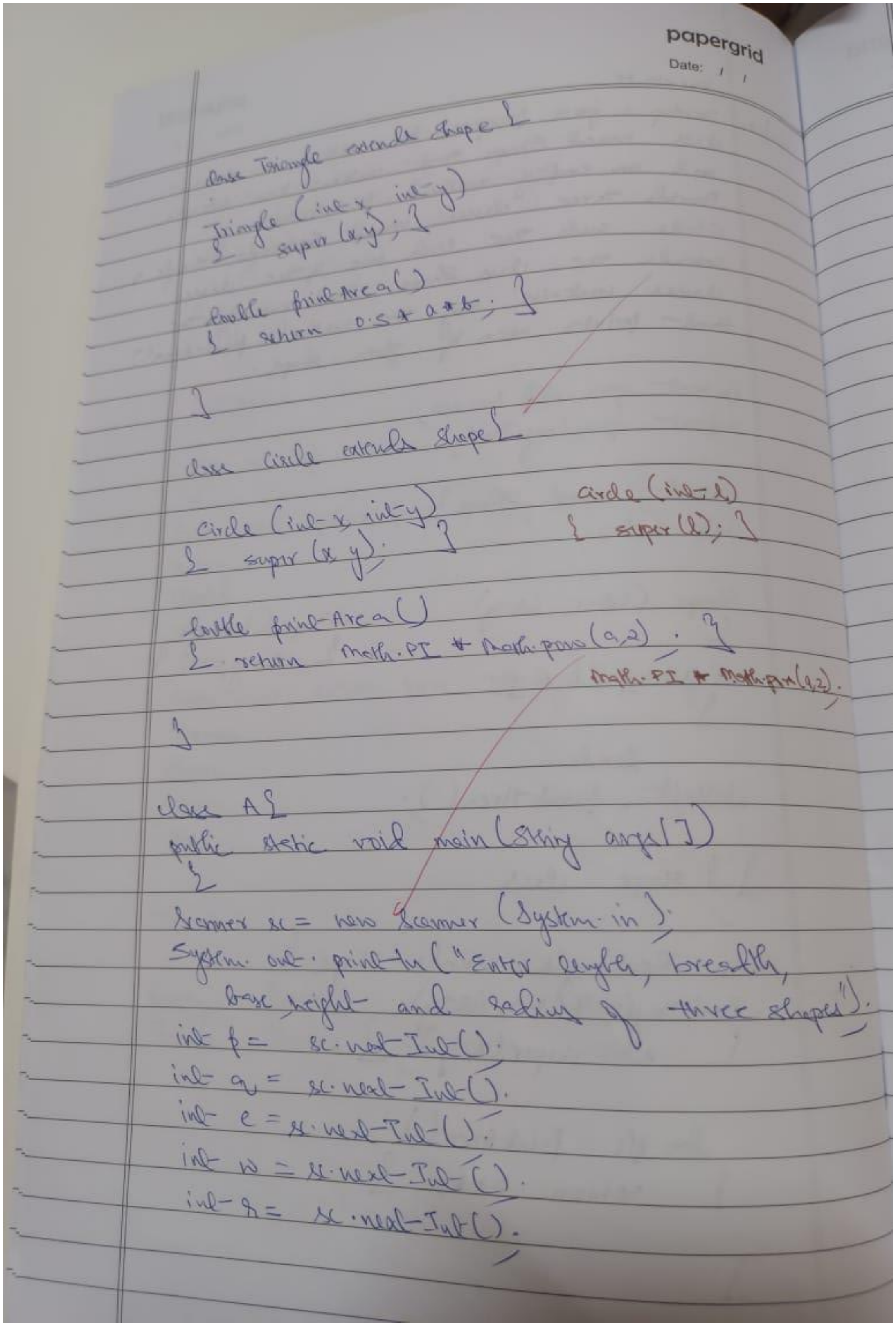


papergrid
Date: / /

Week 4

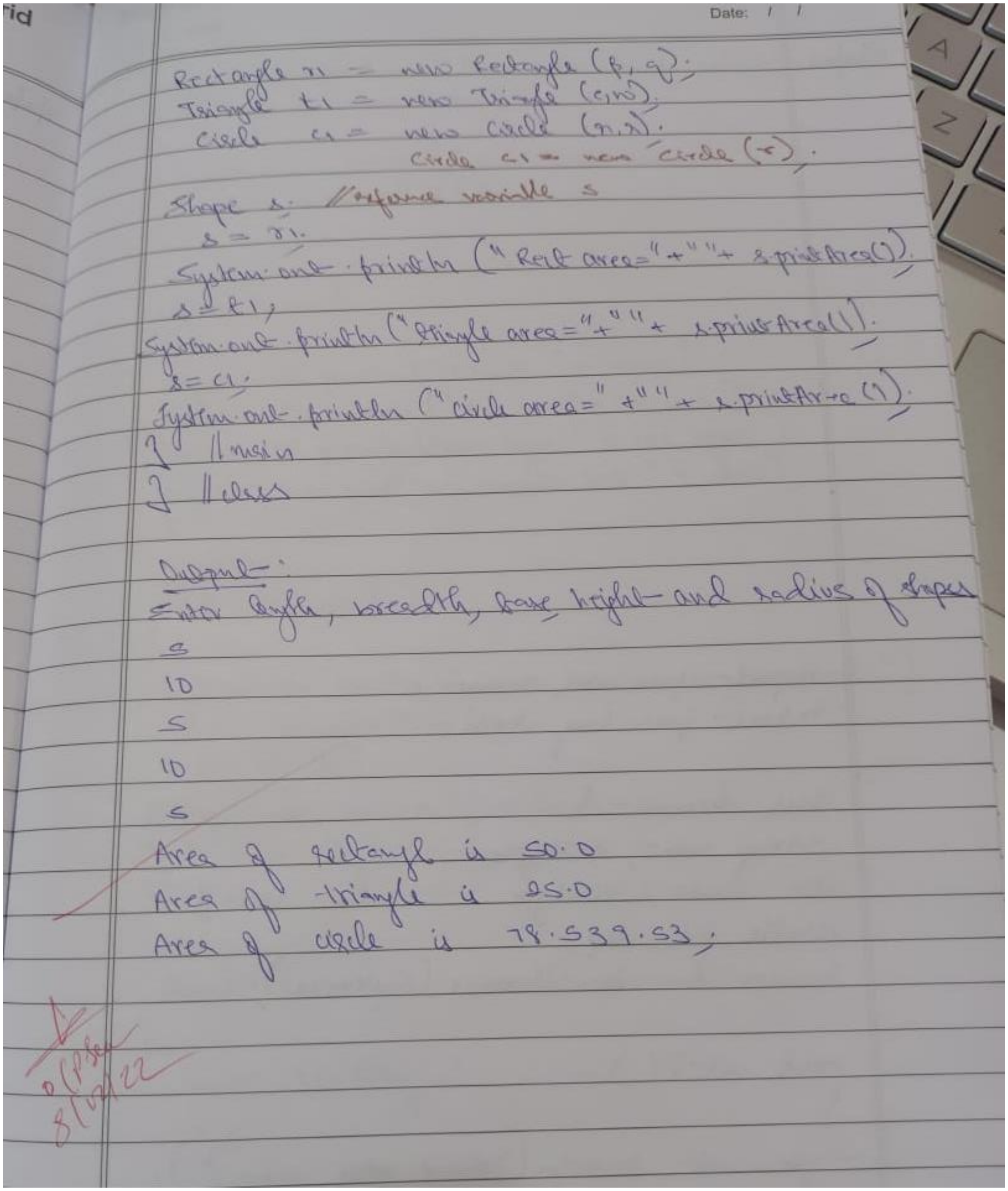
5/12/22 Develop a java program to create an abstract class named Shape that contains two integers and an empty method printArea() . Provide three classes named Rectangle, Triangle & Circle such the each one of the classes extends the class Shape . Each one of the classes contain only the method print-Area() that prints area of given shape .

```
import java.util.Scanner;
import java.lang.Math;

abstract class Shape {
int a, b;

Shape (int x, int y)                  Shape (int z)
{                                     {  a=z;   }
   a=x;   b=y;
}
           double
abstract ^ printArea();

} // shape class

class Rectangle extends Shape {
  Rectangle (int x, int y)
  {  ~~a=~~  super(x,y);  }

   double printArea()
  {   return a*b;  }

}
```



papergrid
Date: / /

```
class Triangle extends Shape {
Triangle (int x, int y)
{   super (x,y); }

double print Area()
{  return  0.5 * a * b; }

}

class Circle extends Shape{

Circle (int x, int y)                     circle (int l)
{   super (x y).    }                     {  super (l); }

double print-Area()
{  return  Math.PI * Math.pow (a,2) . }
                                          Math.PI * Math.pow(a,2).

}

class A{
public static void main(String args[])
 {
Scanner sc= new Scanner (System.in).
System.out.println("Enter length, breadth,
    base height and radius of three shapes").
int p=  sc.next Int().
int q =  sc.next- Int().
int e = sc.next-Int-().
int w = sc.next-Int- ().
int r=  sc.next-Int().
```



Date: / /

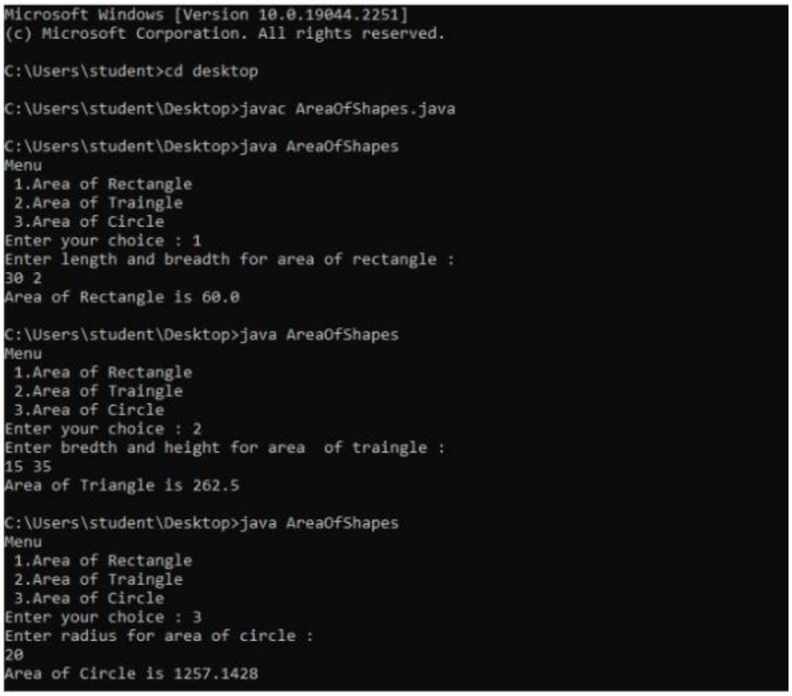
```
Rectangle r1 = new Rectangle (p, q);
Triangle t1 = new Triangle (c,w);
 Circle c1 = new Circle (n,x).
             Circle c1 = new Circle (r);

Shape s; //reference variable s
  s = r1.
System.out.println ("Rect area=" + " " + s.printArea());
s = t1;
System.out.println ("triangle area=" + " " + s.printArea()).
 s = c1;
System.out.println ("circle area=" + " " + s.printArea ()).
} //main
} //class
```

Output:

```
Enter length, breadth, base, height and radius of shapes
5
10
5
10
5
Area of rectangle is 50.0
Area of triangle is 25.0
Area of circle is 78.539.53;
```

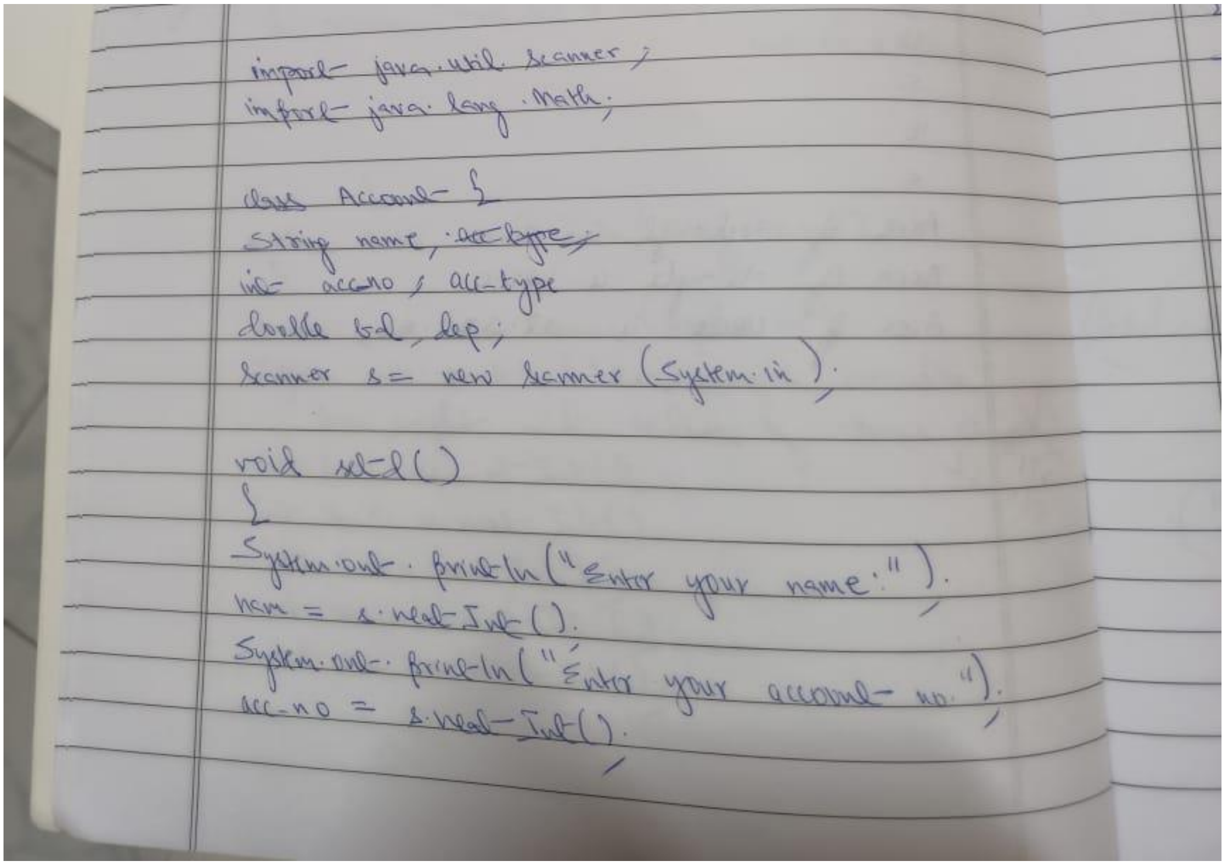
8/09/22

OUTPUT:

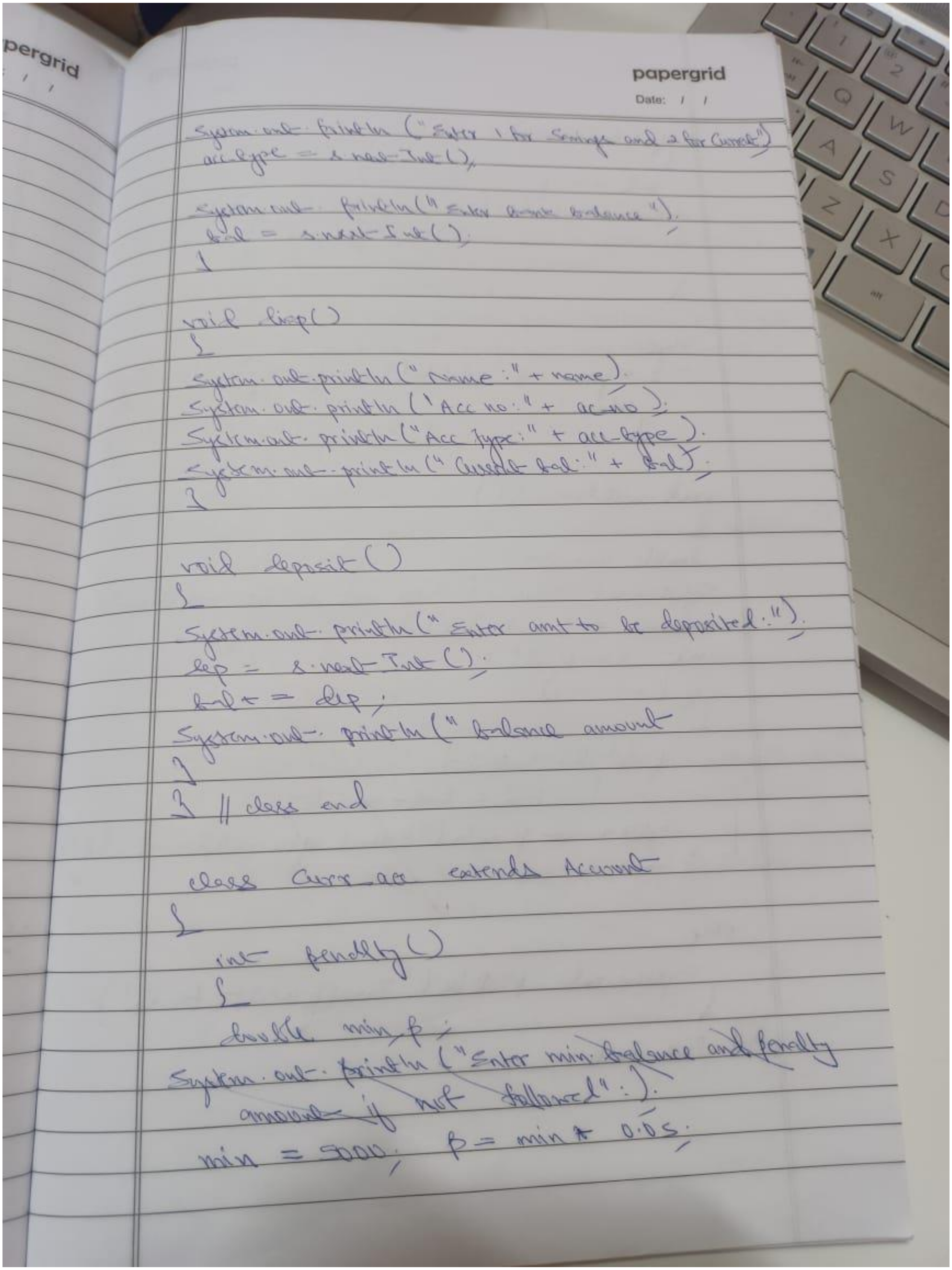


```
Microsoft Windows [Version 10.0.19044.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\student>cd desktop

C:\Users\student\Desktop>javac AreaOfShapes.java

C:\Users\student\Desktop>java AreaOfShapes
Menu
 1.Area of Rectangle
 2.Area of Traingle
 3.Area of Circle
Enter your choice : 1
Enter length and breadth for area of rectangle :
30 2
Area of Rectangle is 60.0

C:\Users\student\Desktop>java AreaOfShapes
Menu
 1.Area of Rectangle
 2.Area of Traingle
 3.Area of Circle
Enter your choice : 2
Enter bredth and height for area  of traingle :
15 35
Area of Triangle is 262.5

C:\Users\student\Desktop>java AreaOfShapes
Menu
 1.Area of Rectangle
 2.Area of Traingle
 3.Area of Circle
Enter your choice : 3
Enter radius for area of circle :
20
Area of Circle is 1257.1428
```

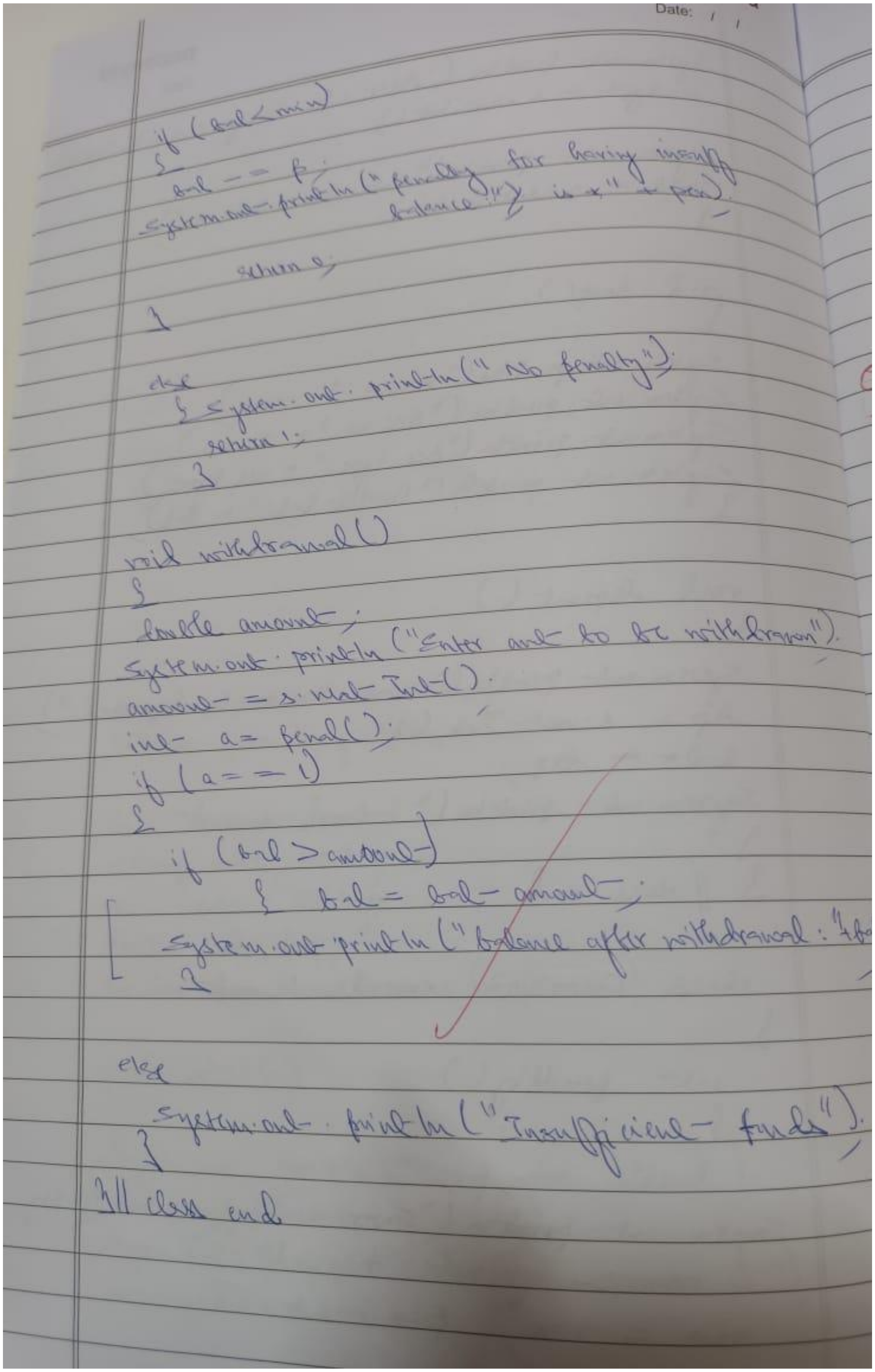
5. Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book 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;
import java.lang.Math;

class Account {
String name, ~~acc type;~~
int accno, acc-type
double bal, dep;
Scanner s= new Scanner (System.in);

void set_d()
{
System.out.println("Enter your name:");
nam = s.next_Int();
System.out.println("Enter your account no.");
acc-no = s.next_Int();
```



```
System.out.println ("Enter 1 for Savings and 2 for Current");
acc_type = s.nextInt();

System.out.println("Enter Bank balance");
bal = s.nextInt();
}

void disp()
{
System.out.println(" name :" + name);
System.out.println ("Acc no :" + ac_no);
System.out.println("Acc type :" + acc_type);
System.out.println(" Current bal :" + bal);
}

void deposit ()
{
System.out.println(" Enter amt to be deposited :");
dep = s.nextInt();
bal += dep;
System.out.println(" balance amount
}
} // class end

class Curr_acc extends Account
{
  int penalty()
  {
   double min, p;
System.out.println ("Enter min balance and penalty
     amount if not followed" :);
  min = 5000;  p = min * 0.05;
```



Date: / /

```
if (bal < min)
{
  bal -= p;
System.out.println ("penalty for having insuff
                    balance is x" + pen);

        return 0;

}

else
 { System.out.println(" No penalty");
   return 1;
   }

void withdrawal()
{
 double amount;
System.out.println ("Enter amt to be withdrawn").
amount = s.nextInt().
int  a = penal();
if (a == 1)
{
    if (bal > amount)
        { bal = bal - amount;
   System.out.println ("balance after withdrawal : "+ba
   }

else
  System.out.println ("Insufficient funds").
  }
}// class end
```

21

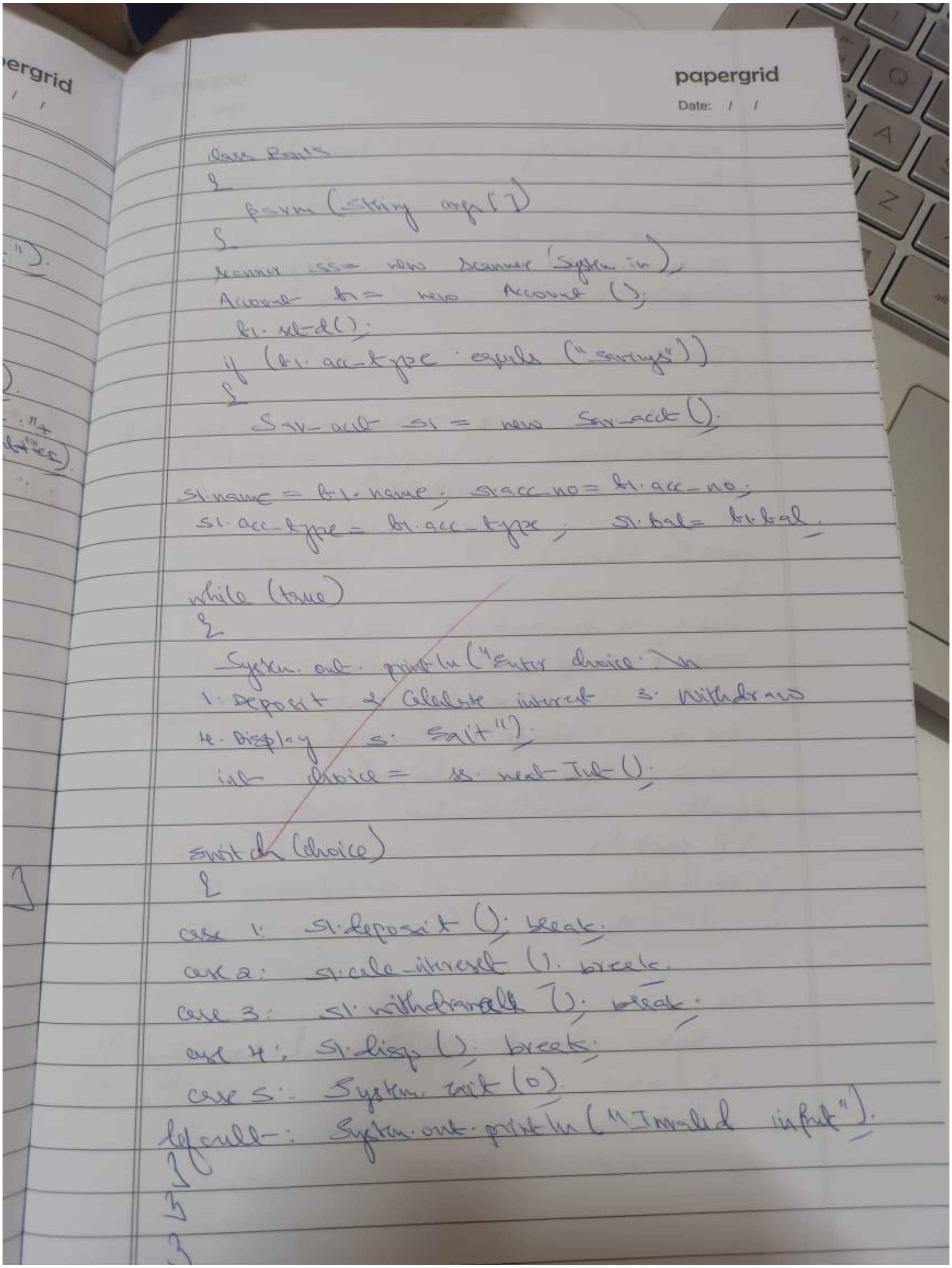
Date: / /

```
class Sav-acc extends Account
{
 void calc-interest ()
 {
System.out.println ("Enter time in years and ROI");
double t = s.nextDouble ();
double r = s.nextDouble ();
double ci = (bal * math.pow((1+r/100), t)) - bal;
System.out.println ("compound interest =" + ci);
}
```

O/p ho.

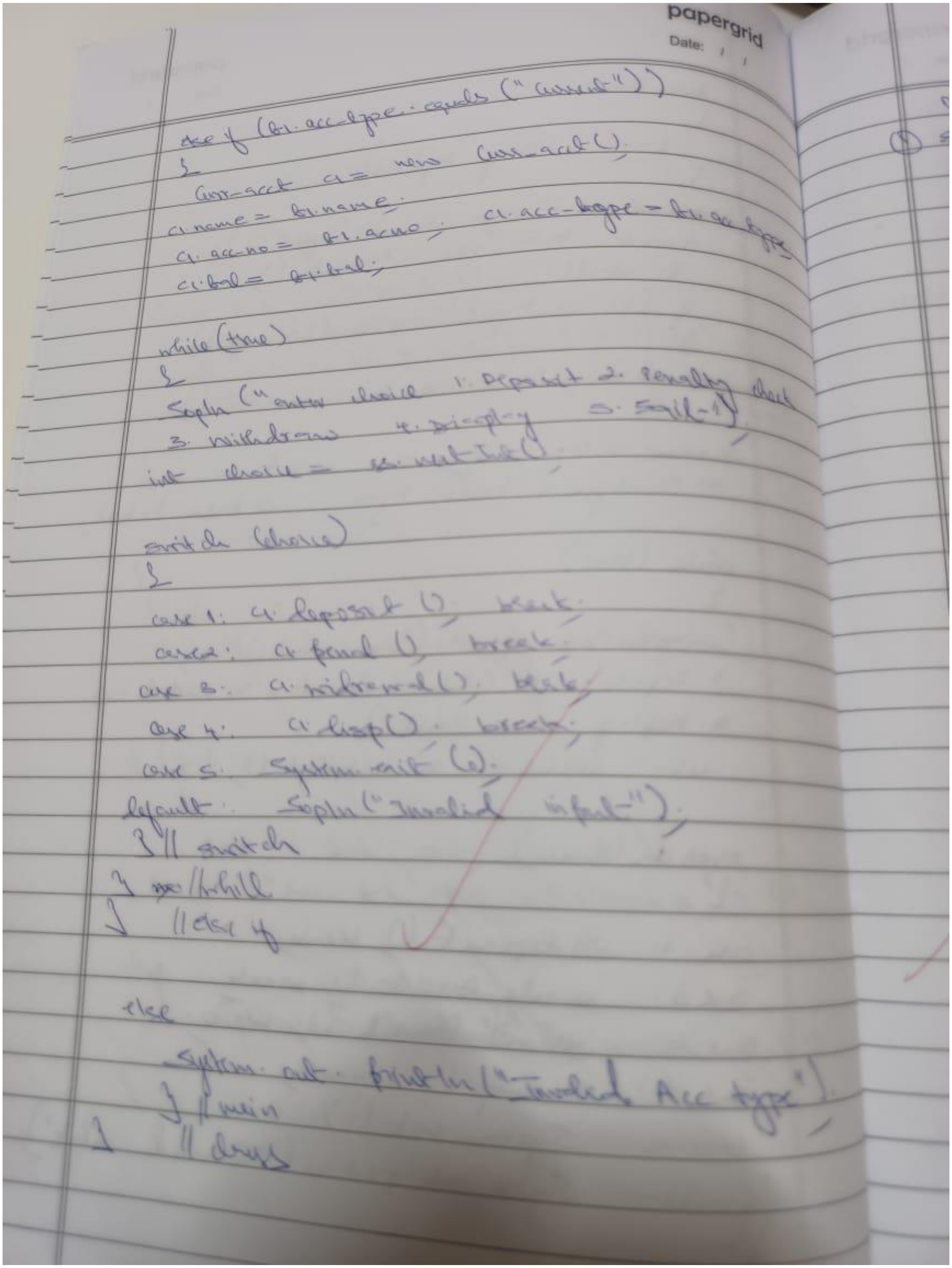
3/12/22

```
void withdrawal ()
{
 double amount;
 System.out.println ("Enter amount to be withdrawn");
 amount = ~~sessa.~~ s.nextInt();
   int a = penal ();

 if (a == 1)
 { if (bal >= amt)
        { bal = bal - amt;
System.out.println (" Account balance after withdrawal is
                                      + bal); }

  else
     { System.out.println (" Amt cant be withdrawn"
 }
```



papergrid
Date: / /

```
class Bank
{
   psvm (String args[])
{
Scanner ss= new Scanner (System.in);
Account b1= new Account ();
 b1.set_d();
 if (b1.acc_type.equals ("savings"))
 {
   Sav_acct s1 = new Sav_acct ().

s1.name = b1.name; s1.acc_no= b1.acc_no;
 s1.acc_type= b1.acc_type; s1.bal= b1.bal.

while (true)
 {
  System.out.println ("Enter choice: \n
 1. Deposit 2. Calculate interest 3. Withdraw
 4. Display  5. Exit");
  int choice = ss.nextInt();

switch (choice)
 {
case 1: s1.deposit (); break;
case 2: s1.calc_interest (). break.
case 3: s1.withdrawal (); break.
case 4: s1.disp (). break.
case 5: System.exit (0).
default: System.out.println ("Invalid input").
}
}
}
```

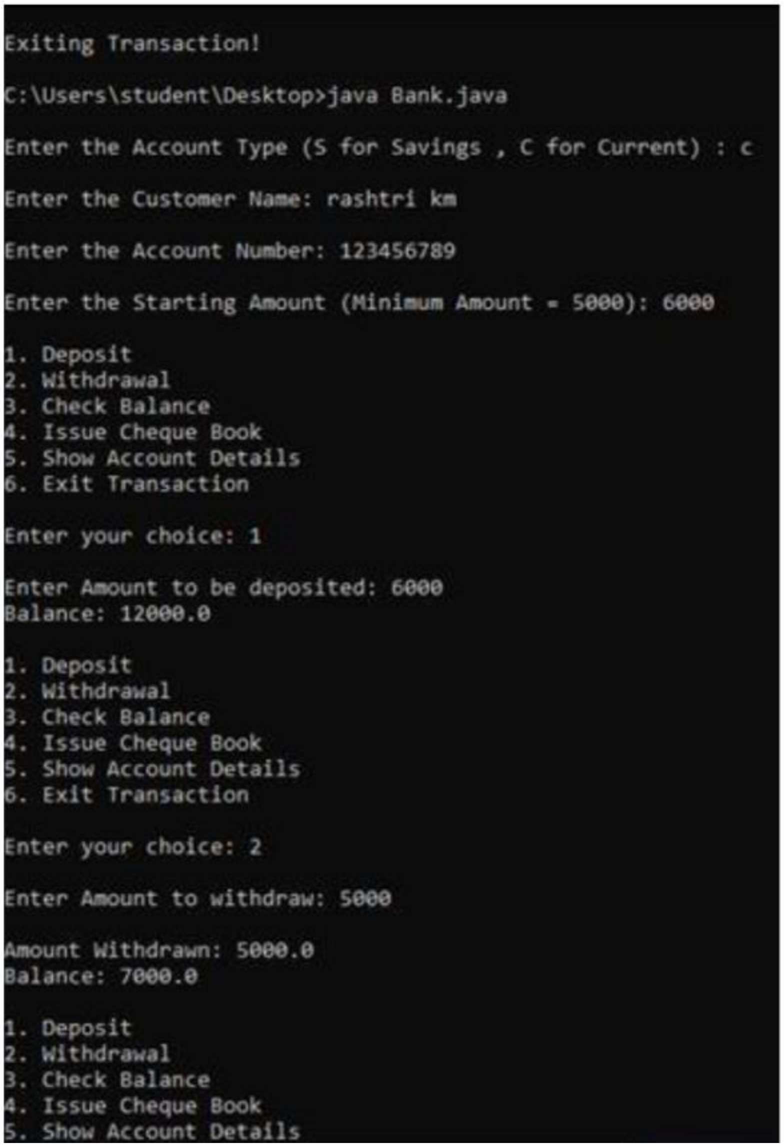


papergrid

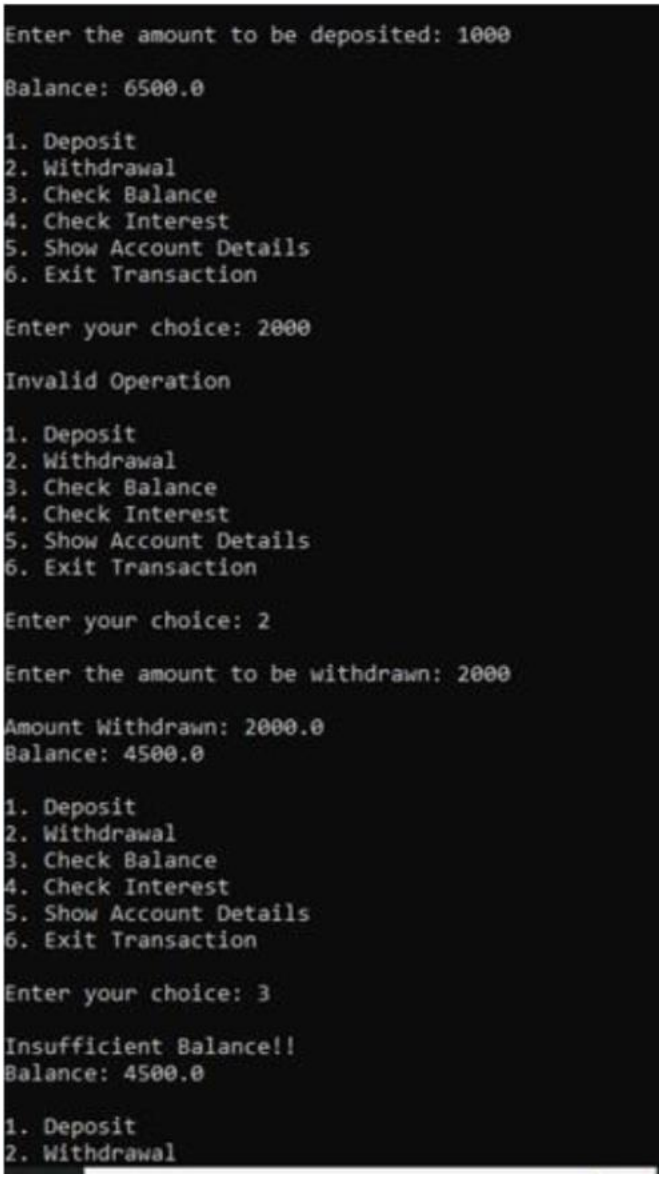
Date: / /

```
else if (b1.acc_type.equals("current"))
{
  Curr_acct c1 = new Curr_acct();
c1.name = b1.name;
c1.acc_no = b1.acno;    c1.acc_type = b1.acc_type;
c1.bal = b1.bal;

while (true)
{
 Sopln ("enter choice  1. Deposit 2. Penalty check
  3. withdraw   4. Display   5. Exit-1");
int choice = sc.nextInt();

switch (choice)
{
 case 1: c1.deposit(); break;
 case 2: c1.penal(); break;
case 3: c1.withdrawl(); break;
 case 4: c1.disp(); break;
 case 5: System.exit(0);
default: Sopln("Invalid input");
 }// switch
} //while
} //else if

else
  System.out.println("Invalid Acc type");
  } // main
}   // class
```

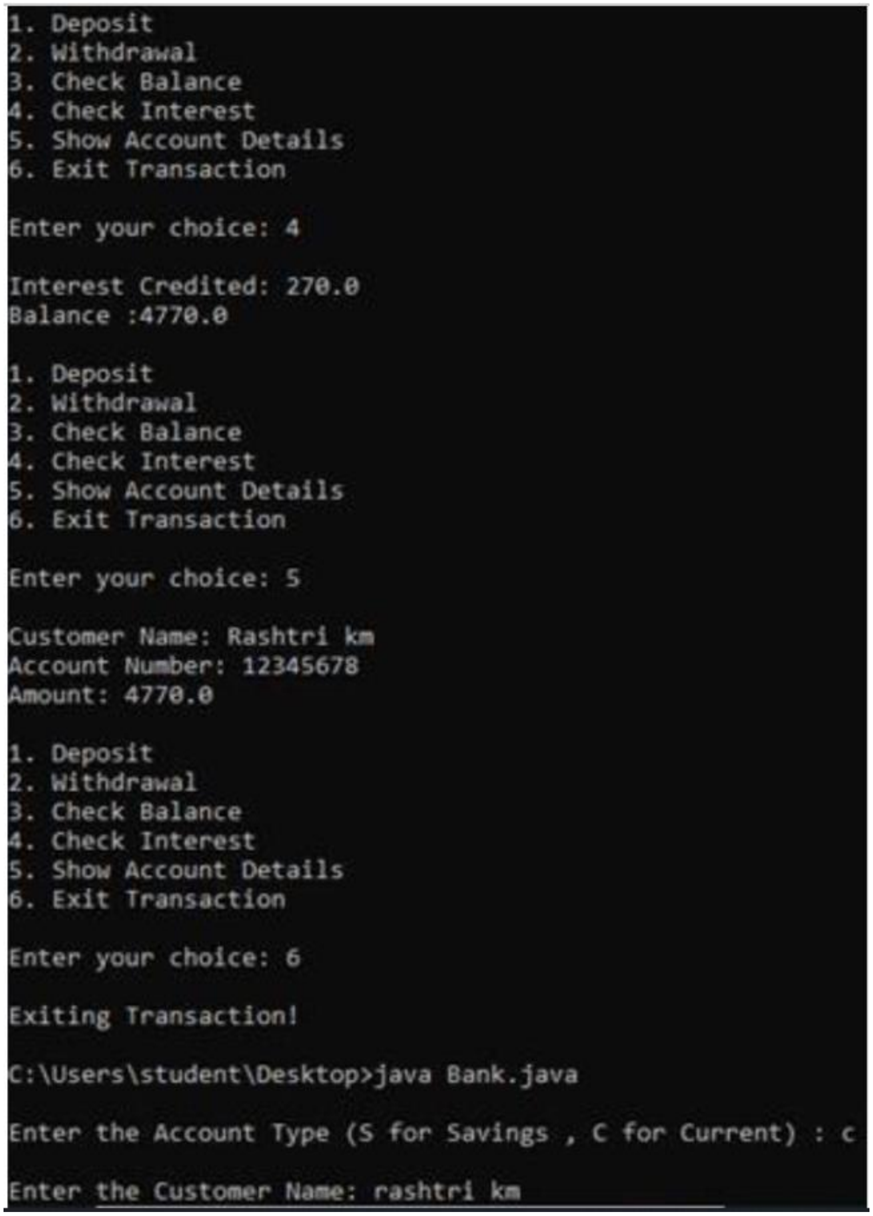
OUTPUT:



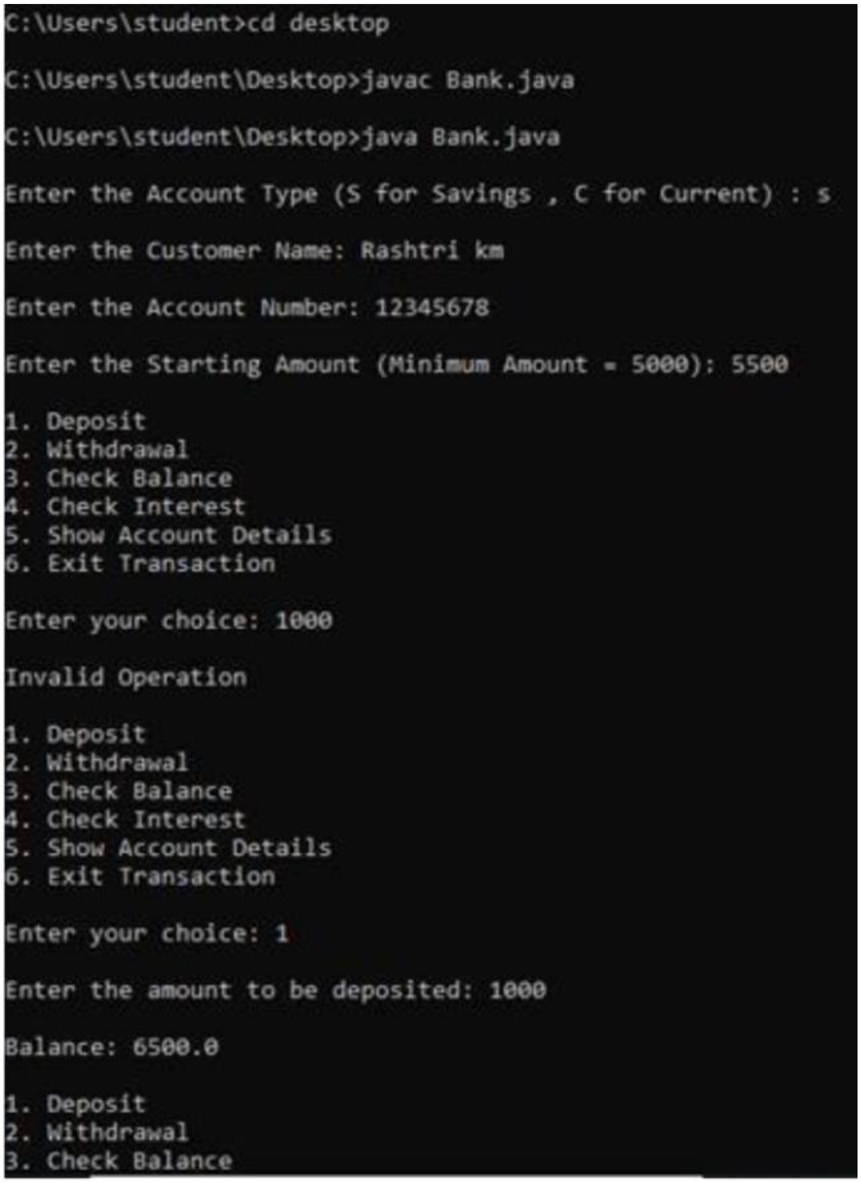
```
Exiting Transaction!

C:\Users\student\Desktop>java Bank.java

Enter the Account Type (S for Savings , C for Current) : c

Enter the Customer Name: rashtri km

Enter the Account Number: 123456789

Enter the Starting Amount (Minimum Amount = 5000): 6000

1. Deposit
2. Withdrawal
3. Check Balance
4. Issue Cheque Book
5. Show Account Details
6. Exit Transaction

Enter your choice: 1

Enter Amount to be deposited: 6000
Balance: 12000.0

1. Deposit
2. Withdrawal
3. Check Balance
4. Issue Cheque Book
5. Show Account Details
6. Exit Transaction

Enter your choice: 2

Enter Amount to withdraw: 5000

Amount Withdrawn: 5000.0
Balance: 7000.0

1. Deposit
2. Withdrawal
3. Check Balance
4. Issue Cheque Book
5. Show Account Details
```



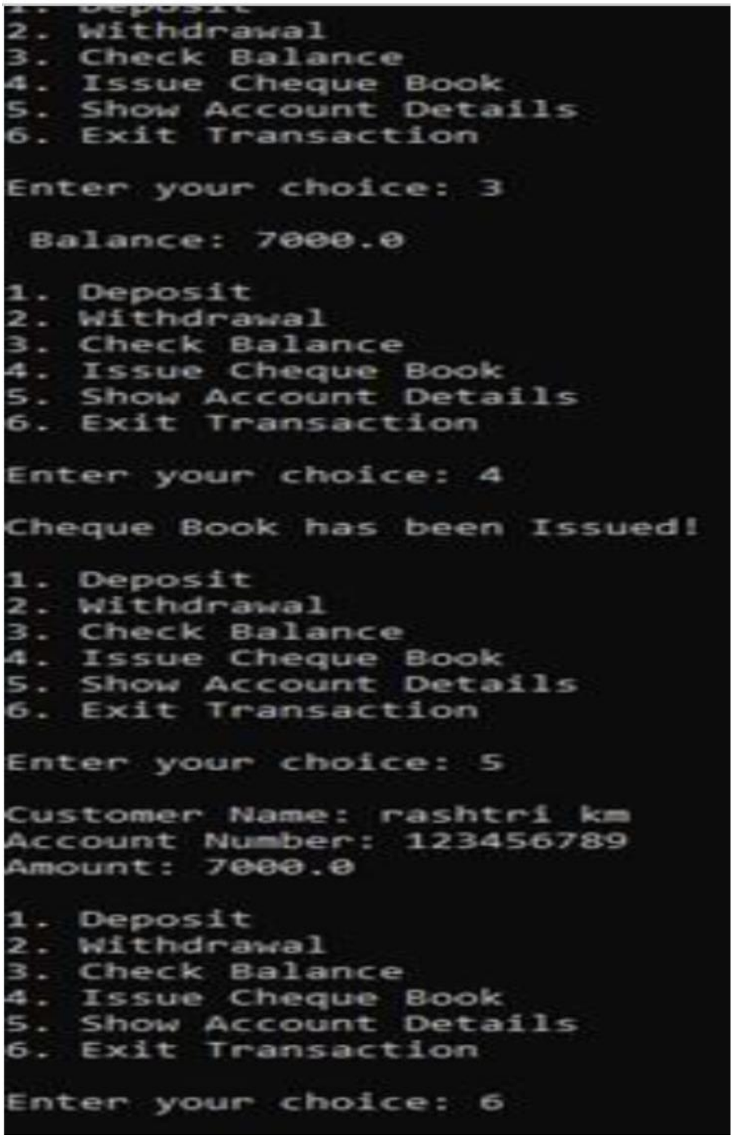
```
Enter the amount to be deposited: 1000

Balance: 6500.0

1. Deposit
2. Withdrawal
3. Check Balance
4. Check Interest
5. Show Account Details
6. Exit Transaction

Enter your choice: 2000

Invalid Operation

1. Deposit
2. Withdrawal
3. Check Balance
4. Check Interest
5. Show Account Details
6. Exit Transaction

Enter your choice: 2

Enter the amount to be withdrawn: 2000

Amount Withdrawn: 2000.0
Balance: 4500.0

1. Deposit
2. Withdrawal
3. Check Balance
4. Check Interest
5. Show Account Details
6. Exit Transaction

Enter your choice: 3

Insufficient Balance!!
Balance: 4500.0

1. Deposit
2. Withdrawal
```



```
1. Deposit
2. Withdrawal
3. Check Balance
4. Check Interest
5. Show Account Details
6. Exit Transaction

Enter your choice: 4

Interest Credited: 270.0
Balance :4770.0

1. Deposit
2. Withdrawal
3. Check Balance
4. Check Interest
5. Show Account Details
6. Exit Transaction

Enter your choice: 5

Customer Name: Rashtri km
Account Number: 12345678
Amount: 4770.0

1. Deposit
2. Withdrawal
3. Check Balance
4. Check Interest
5. Show Account Details
6. Exit Transaction

Enter your choice: 6

Exiting Transaction!

C:\Users\student\Desktop>java Bank.java

Enter the Account Type (S for Savings , C for Current) : c

Enter the Customer Name: rashtri km
```

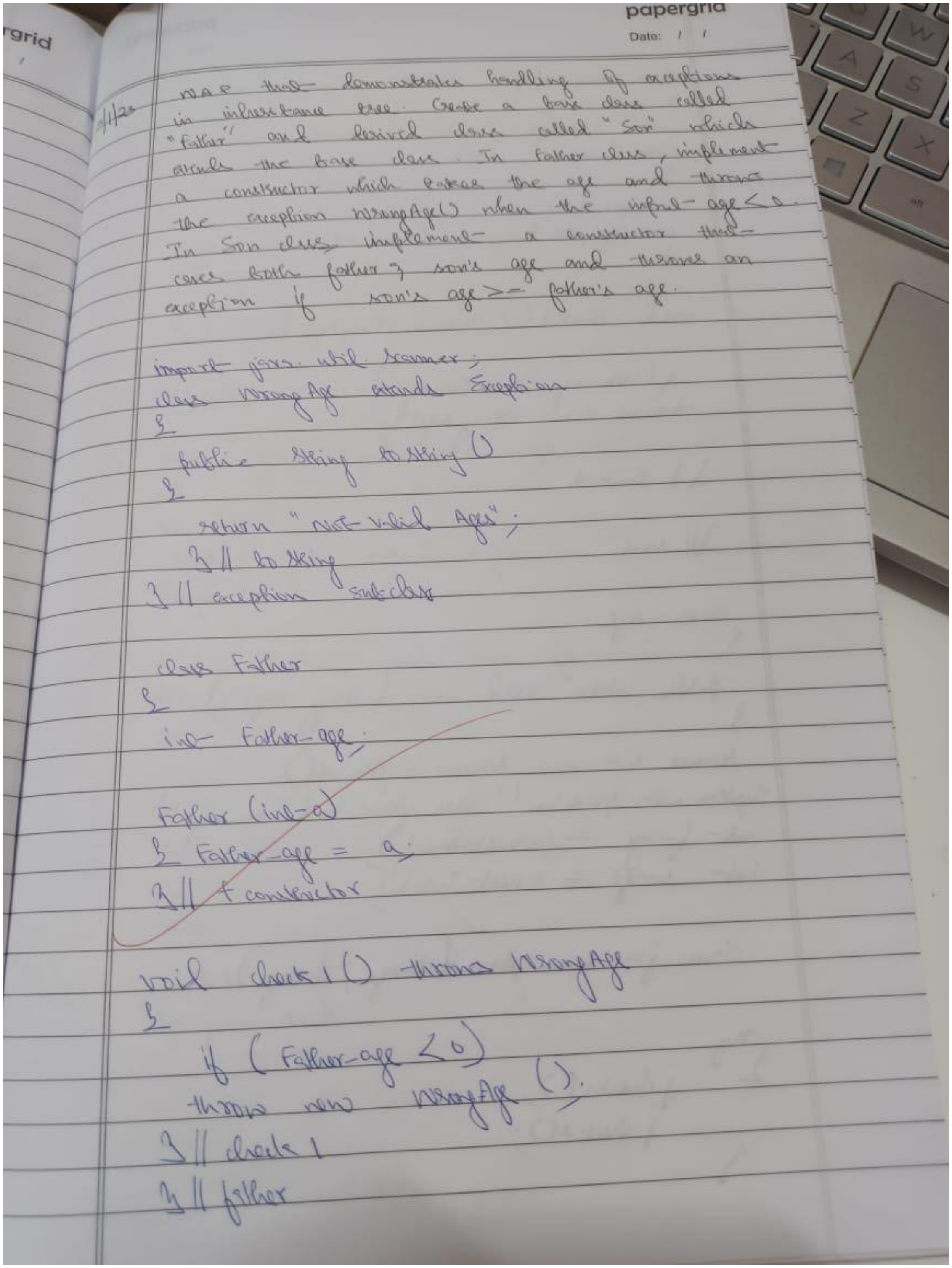


```
C:\Users\student>cd desktop

C:\Users\student\Desktop>javac Bank.java

C:\Users\student\Desktop>java Bank.java

Enter the Account Type (S for Savings , C for Current) : s

Enter the Customer Name: Rashtri km

Enter the Account Number: 12345678

Enter the Starting Amount (Minimum Amount = 5000): 5500

1. Deposit
2. Withdrawal
3. Check Balance
4. Check Interest
5. Show Account Details
6. Exit Transaction

Enter your choice: 1000

Invalid Operation

1. Deposit
2. Withdrawal
3. Check Balance
4. Check Interest
5. Show Account Details
6. Exit Transaction

Enter your choice: 1

Enter the amount to be deposited: 1000

Balance: 6500.0

1. Deposit
2. Withdrawal
3. Check Balance
```



```
1. Deposit
2. Withdrawal
3. Check Balance
4. Issue Cheque Book
5. Show Account Details
6. Exit Transaction

Enter your choice: 3

 Balance: 7000.0

1. Deposit
2. Withdrawal
3. Check Balance
4. Issue Cheque Book
5. Show Account Details
6. Exit Transaction

Enter your choice: 4

Cheque Book has been Issued!

1. Deposit
2. Withdrawal
3. Check Balance
4. Issue Cheque Book
5. Show Account Details
6. Exit Transaction

Enter your choice: 5

Customer Name: rashtri km
Account Number: 123456789
Amount: 7000.0

1. Deposit
2. Withdrawal
3. Check Balance
4. Issue Cheque Book
5. Show Account Details
6. Exit Transaction

Enter your choice: 6
```

6.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.



papergrid

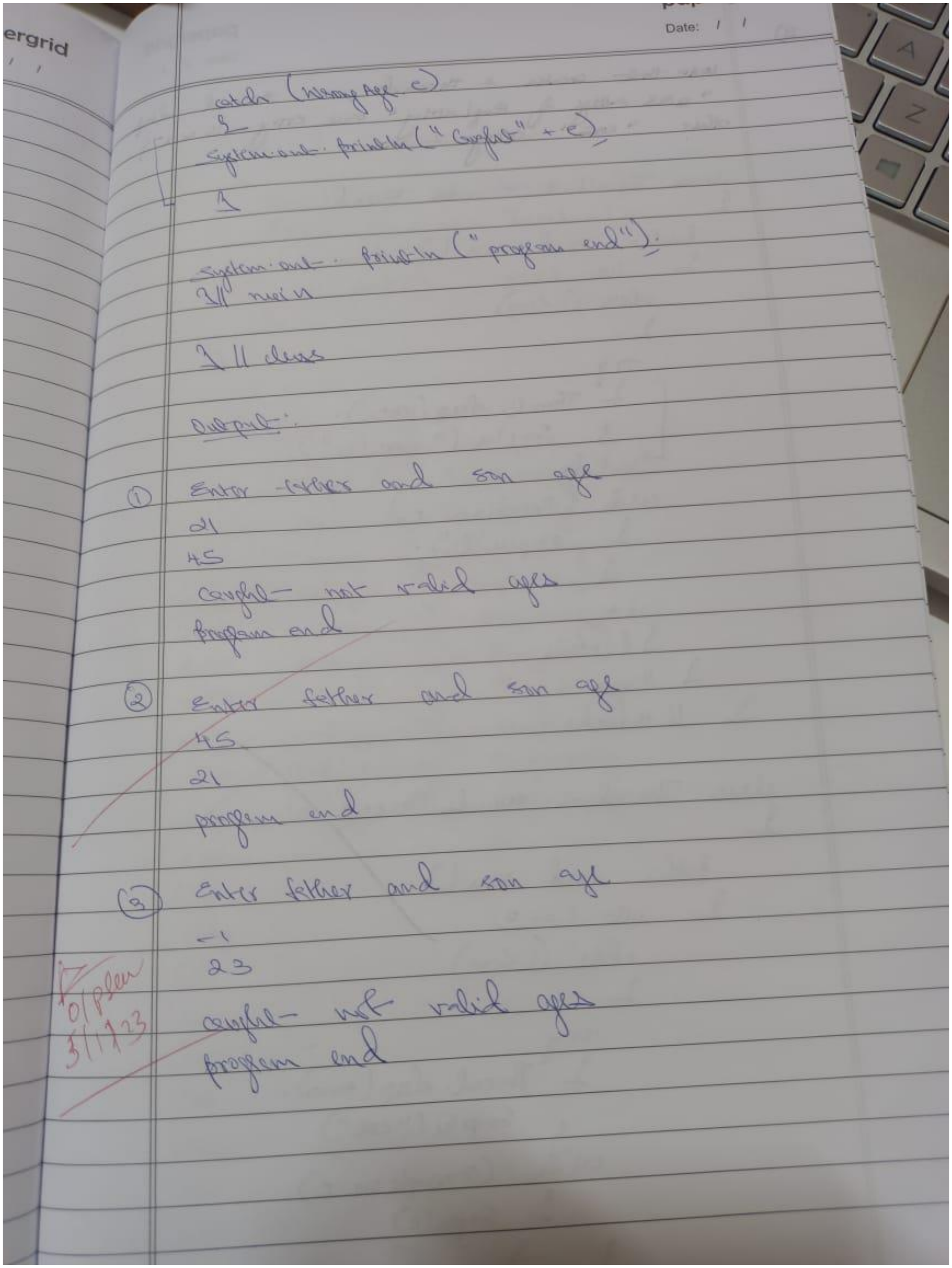
Date: / /

WAP 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 & son's age and throws an exception if son's age >= father's age.

```
import java.util.Scanner;
class WrongAge extends Exception
{
  public String toString()
  {
     return "not valid Age";
   }// toString
}// exception subclass

class Father
{
  int Father-age;

Father (int a)
{ Father-age = a;
}// f constructor

void check1() throws WrongAge
{
    if ( Father-age < 0)
    throw new WrongAge ();
 }// check1
}// father
```

```
class Son extends Father
{
 int son_age ;
 Son (int a, int b)
 {
  super (a);
 son_age = b;
 }

 void check2() throws WrongAge {

  if (son_age > father_age)
 throws new WrongAge();

 }// check 2

}// son

class Age
{
 public static void main (String args[])
 {
 Scanner s = new Scanner (System.in);
System.out.println ("Enter father and son age");
 int f_age = s.nextInt();
 int s_age = s.nextInt();

 Son j= new Son (f_age, s_age);

 try
 {
      j.check1();
      j.check2();
 }
```

O/p seen
3/1/23



ergrid

Date: / /

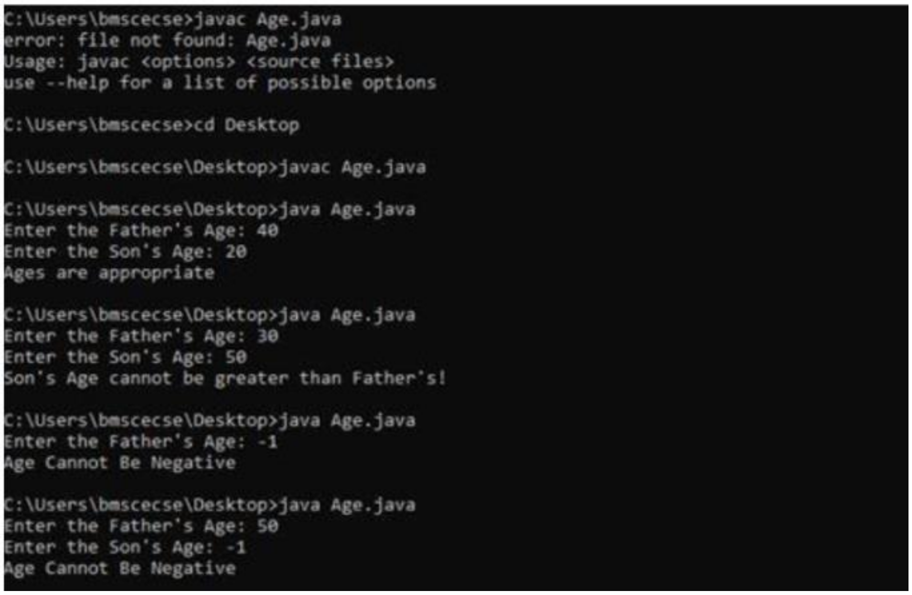
```
catch (wrongAge e)
{
System.out.println("Caught" + e);
}

System.out.println("program end");
}// main

}// class
```

Output:

```
① Enter father and son age
21
45
caught not valid ages
program end

② Enter father and son age
45
21
program end

③ Enter father and son age
-1
23
caught not valid ages
program end
```

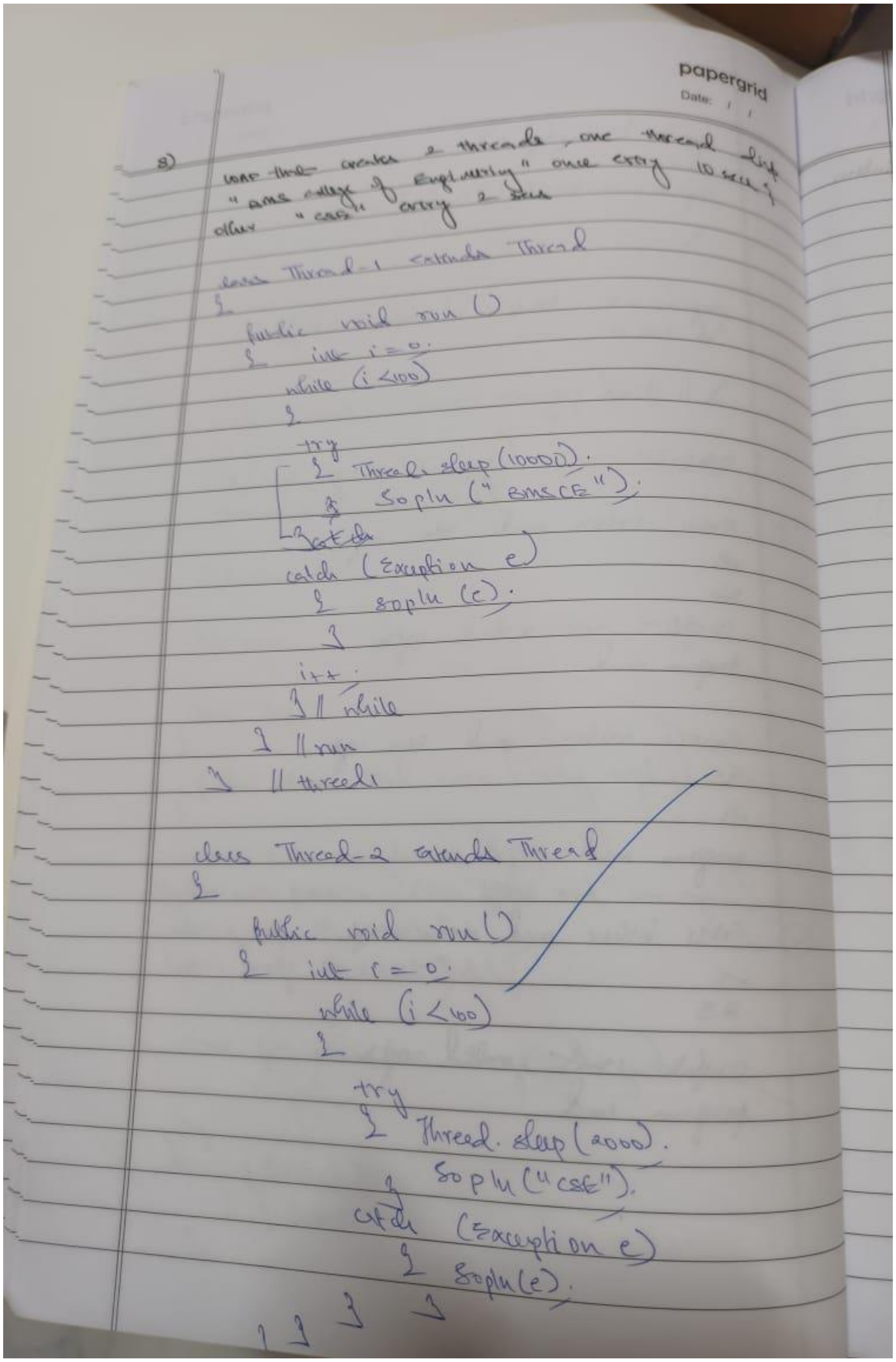
O/p seen
3/1/23

OUTPUT:



```
C:\Users\bmscecse>javac Age.java
error: file not found: Age.java
Usage: javac <options> <source files>
use --help for a list of possible options

C:\Users\bmscecse>cd Desktop

C:\Users\bmscecse\Desktop>javac Age.java

C:\Users\bmscecse\Desktop>java Age.java
Enter the Father's Age: 40
Enter the Son's Age: 20
Ages are appropriate

C:\Users\bmscecse\Desktop>java Age.java
Enter the Father's Age: 30
Enter the Son's Age: 50
Son's Age cannot be greater than Father's!

C:\Users\bmscecse\Desktop>java Age.java
Enter the Father's Age: -1
Age Cannot Be Negative

C:\Users\bmscecse\Desktop>java Age.java
Enter the Father's Age: 50
Enter the Son's Age: -1
Age Cannot Be Negative
```

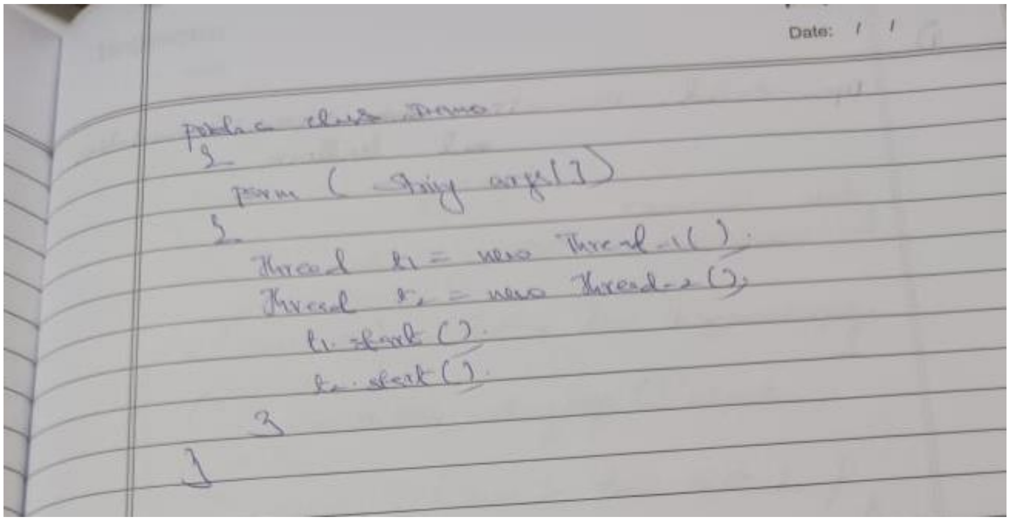
7. Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds.



papergrid
Date: / /

8) WAP that creates 2 threads, one thread disp "BMS college of Engineering" once every 10 secs & other "CSE" every 2 secs

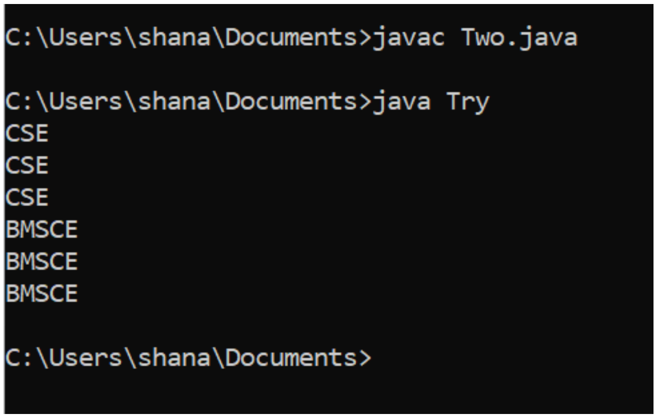
```
class Thread-1 extends Thread
{
    public void run ()
    {   int i = 0;
        while (i<100)
        {
          try
          { Thread.sleep (10000);
            Sopln ("BMSCE");
          }
          }catch
          catch (Exception e)
            { sopln (e);
            }
          i++;
        } // while
    } // run
} // thread1

class Thread-2 extends Thread
{
    public void run()
    {   int i = 0;
          while (i<100)
          {
            try
            { Thread.sleep (2000);
                Sopln("CSE");
            }
            catch (Exception e)
              { Sopln(e);
        } } }   }
```



Date: / /

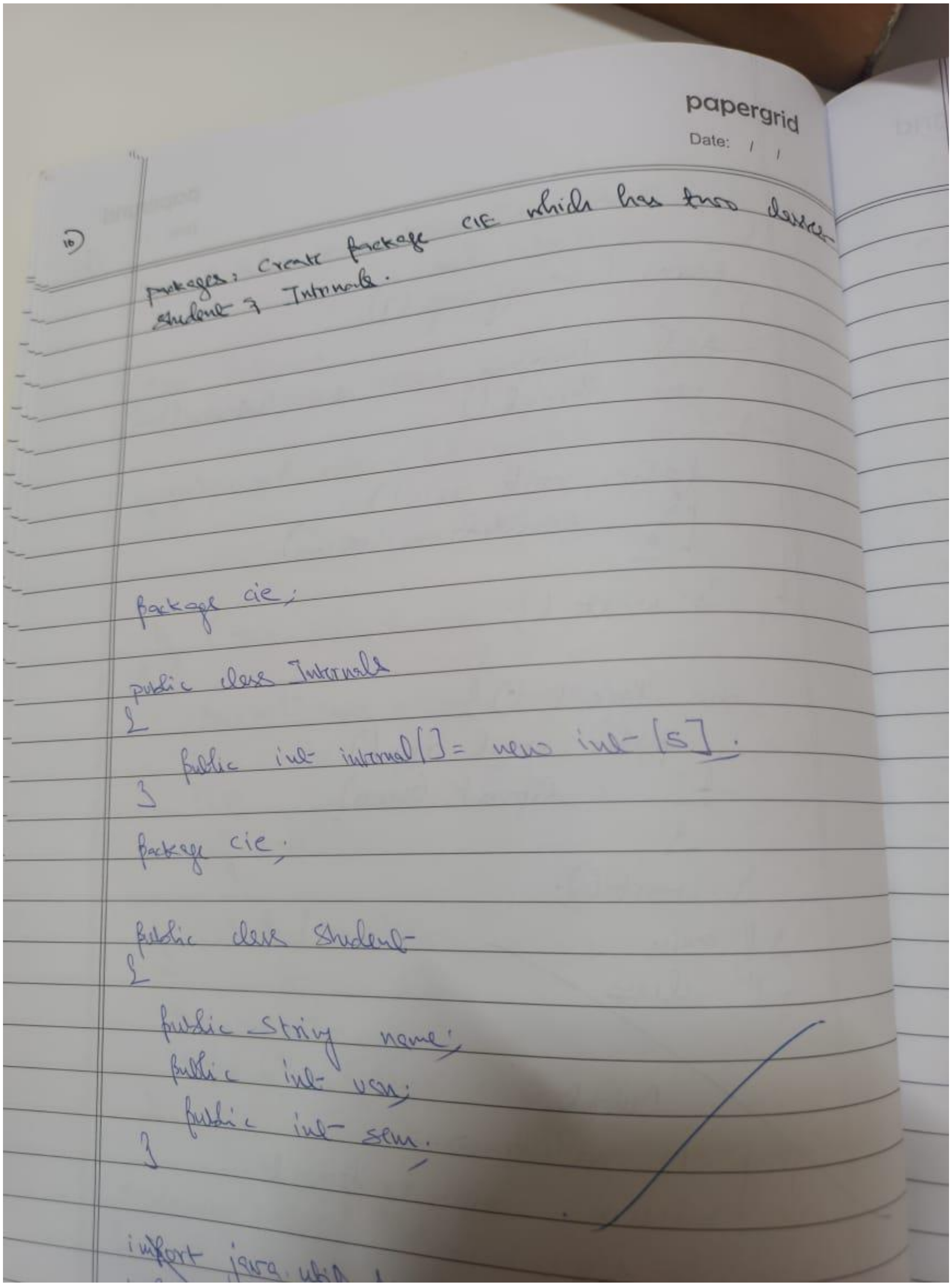
```
public class Demo
{
  psvm ( String args[])
  {
    Thread t1 = new Thread_1();
    Thread t2 = new Thread_2();
      t1.start();
      t2.start();
  }
}
```

OUTPUT:



```
C:\Users\shana\Documents>javac Two.java

C:\Users\shana\Documents>java Try
CSE
CSE
CSE
BMSCE
BMSCE
BMSCE

C:\Users\shana\Documents>
```

8. Create a package CIE which has two classes- Student and Internals. The class Personal has member s 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.

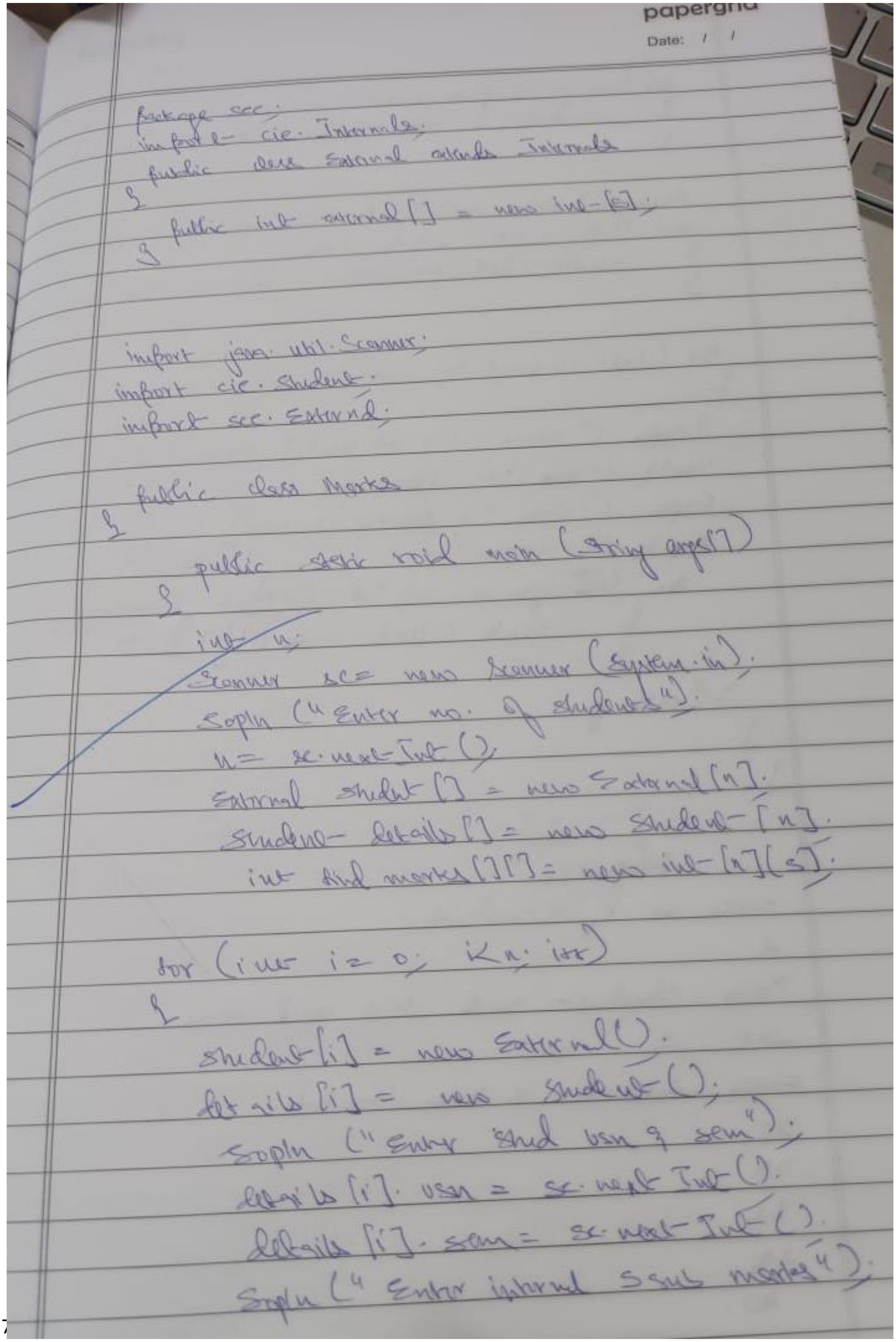


papergrid

Date: / /

10) packages: Create package CIE which has two classes student & Internals.

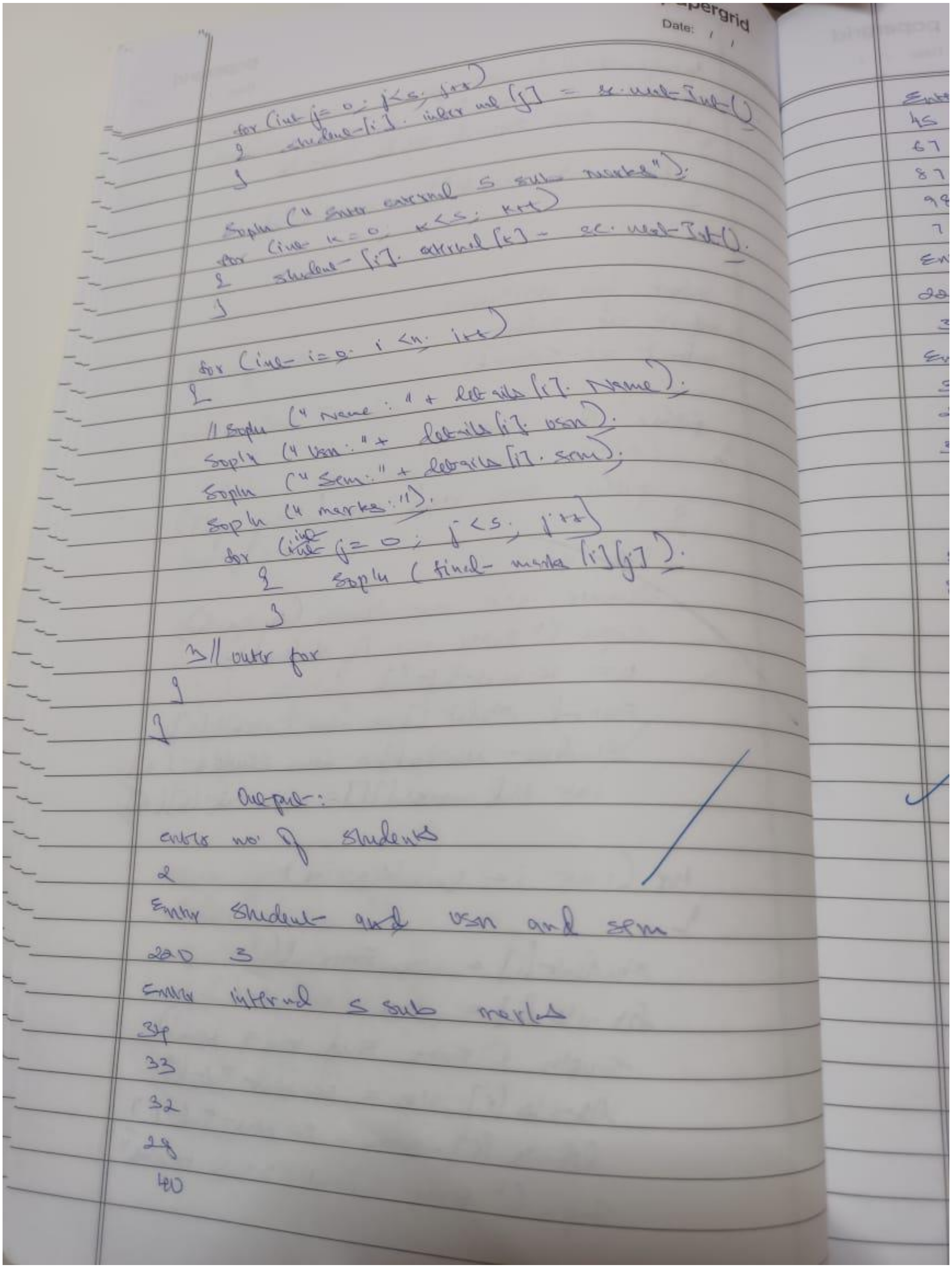
```
package cie;

public class Internals
{
    public int internal[]= new int [5].
}
package cie;

public class Student
{
    public String name;
    public int usn;
     public int sem.
}

import java.util
```



papergrid

Date: / /

```
package sce;
import cie.Internals;
public class External extends Internals
{
  public int external[] = new int[5];
}


import java.util.Scanner;
import cie.Student;
import sce.External;

 public class Marks
{
    public static void main (String args[])
   {
     int n;
     Scanner sc= new Scanner (System.in);
     Sopln ("Enter no. of students");
      n= sc.nextInt ();
      External student [] = new External[n];
       Student details [] = new Student [n];
        int final marks [][] = new int [n][5];

  for (int i= 0; i<n; i++)
  {
     student[i] = new External();
     details [i] = new Student ();
      Sopln ("Enter stud usn & sem");
      details [i]. usn = sc.nextInt ();
      details [i]. sem = sc.nextInt ();
      Sopln ("Enter internal 5 sub marks");
```



Papergrid
Date: / /

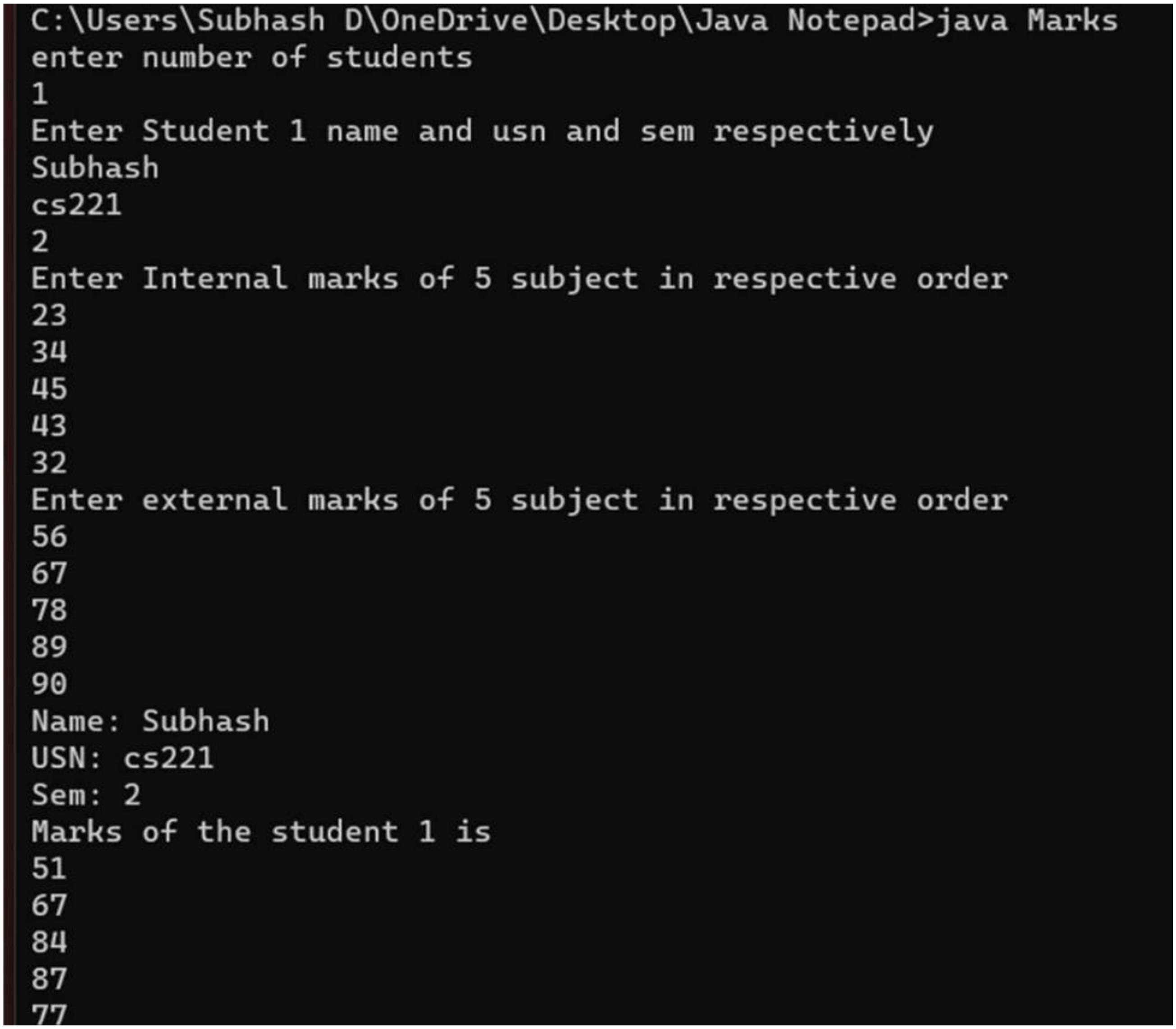
```
for (int j= 0; j<s; j++)
{   student-[i]. internal [j] = sc.nextInt()
}
Sopln ("Enter external 5 sub marks");
for (int k=0; k<s; k++)
{   student-[i]. external [k] = sc.nextInt().
}

for (int i=0; i<n; i++)
{
 Sopln ("Name: " + details [i]. Name);
 // Sopln ("Usn:" + details [i]. usn).
 Sopln ("Sem:" + details [i]. sem);
 Sopln ("marks:").
  for (int j=0; j<s; j++)
    {   Sopln (final-marks [i][j]).
    }
}// outer for
}
}
```

Output:

```
enter no of students
2
Enter student and usn and sem
220 3
Enter internal 5 sub marks
34
33
32
29
40
```

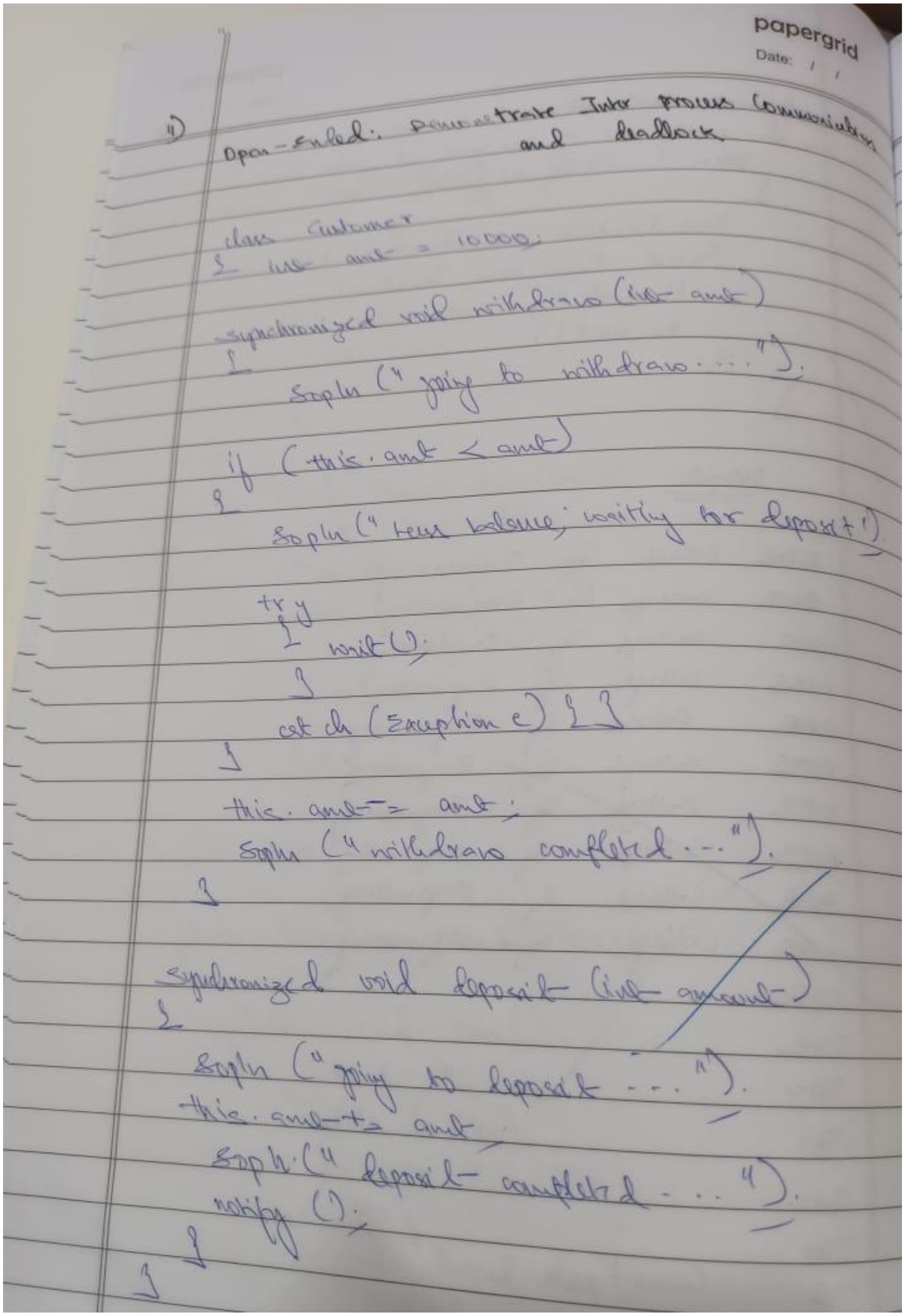
Ent
45
67
87
98
7
En
22
3
Ev

OUTPUT:



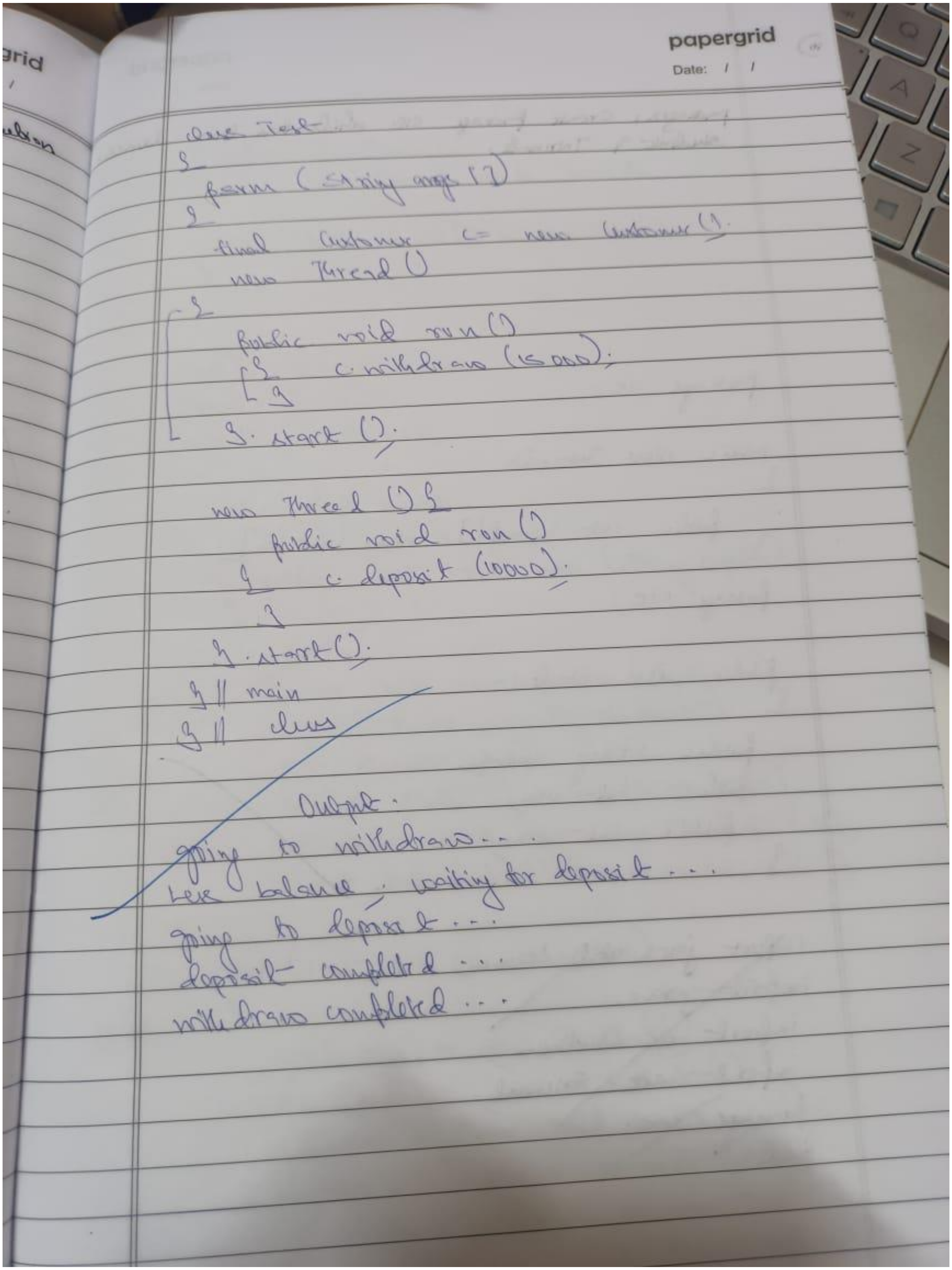
```
C:\Users\Subhash D\OneDrive\Desktop\Java Notepad>java Marks
enter number of students
1
Enter Student 1 name and usn and sem respectively
Subhash
cs221
2
Enter Internal marks of 5 subject in respective order
23
34
45
43
32
Enter external marks of 5 subject in respective order
56
67
78
89
90
Name: Subhash
USN: cs221
Sem: 2
Marks of the student 1 is
51
67
84
87
77
```

9. Demonstrate Inter process Communication and deadlock



11) Open-Ended: Demonstrate Inter process Communication and deadlock

```
class Customer
{ int amt = 10000;

synchronized void withdraw(int amt)
{
      Sopln("going to withdraw....");

if (this.amt < amt)
{
     Sopln("Less balance; waiting for deposit")

     try
     {   wait();
     }
     catch (Exception e) {}
}
this.amt -= amt;
Sopln("withdraw completed...");
}

synchronized void deposit(int amount)
{
  Sopln("going to deposit...");
this.amt += amt
  Sopln("deposit completed...");
  notify();
  }
}
```



papergrid

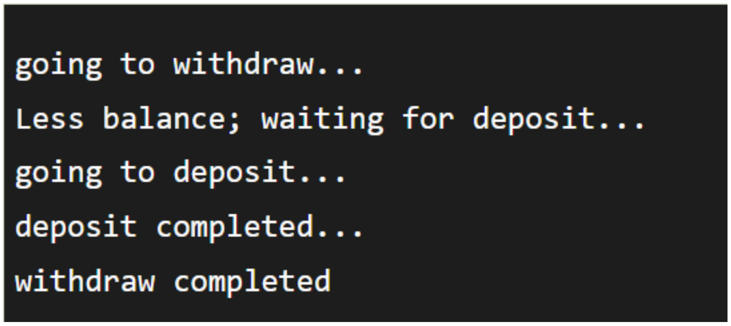
Date: / /

```
class Test
{
 psvm (String args[])
 {
  final Customer c= new Customer();
  new Thread()
 {
    public void run()
    { c.withdraw(15000);
    }
  }.start();

  new Thread(){
     public void run()
    { c.deposit(10000);
    }
  }.start();
 }// main
}// class
```

Output:

```
going to withdraw...
Less balance; waiting for deposit...
going to deposit...
deposit completed...
withdraw completed...
```

OUTPUT:



```
going to withdraw...
Less balance; waiting for deposit...
going to deposit...
deposit completed...
withdraw completed
```