**OOJ LAB RECORD**

**LAB PROGRAM 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.**

import java.util.Scanner;

class quadratic

{

public static void main(String args[])

{

double a, b, c;

double r1, r2;

double determinant;

Scanner input = new Scanner(System.in);

System.out.println("Enter the values of a, b, c : ");

a = input.nextDouble();

b = input.nextDouble();

c = input.nextDouble();

determinant = (b\*b)-(4\*a\*c);

if(determinant > 0)

{

r1 = (-b + Math.sqrt(determinant))/(2\*a);

r2 = (-b - Math.sqrt(determinant))/(2\*a);

System.out.println("Real roots of the quadratic equation are : " + r1 + " and " + r2 );

}

else if(determinant == 0)

{

r1 = (-b + Math.sqrt(determinant))/(2\*a);

System.out.println("They have equal real roots : " + r1 + " and " + r1 );

}

else

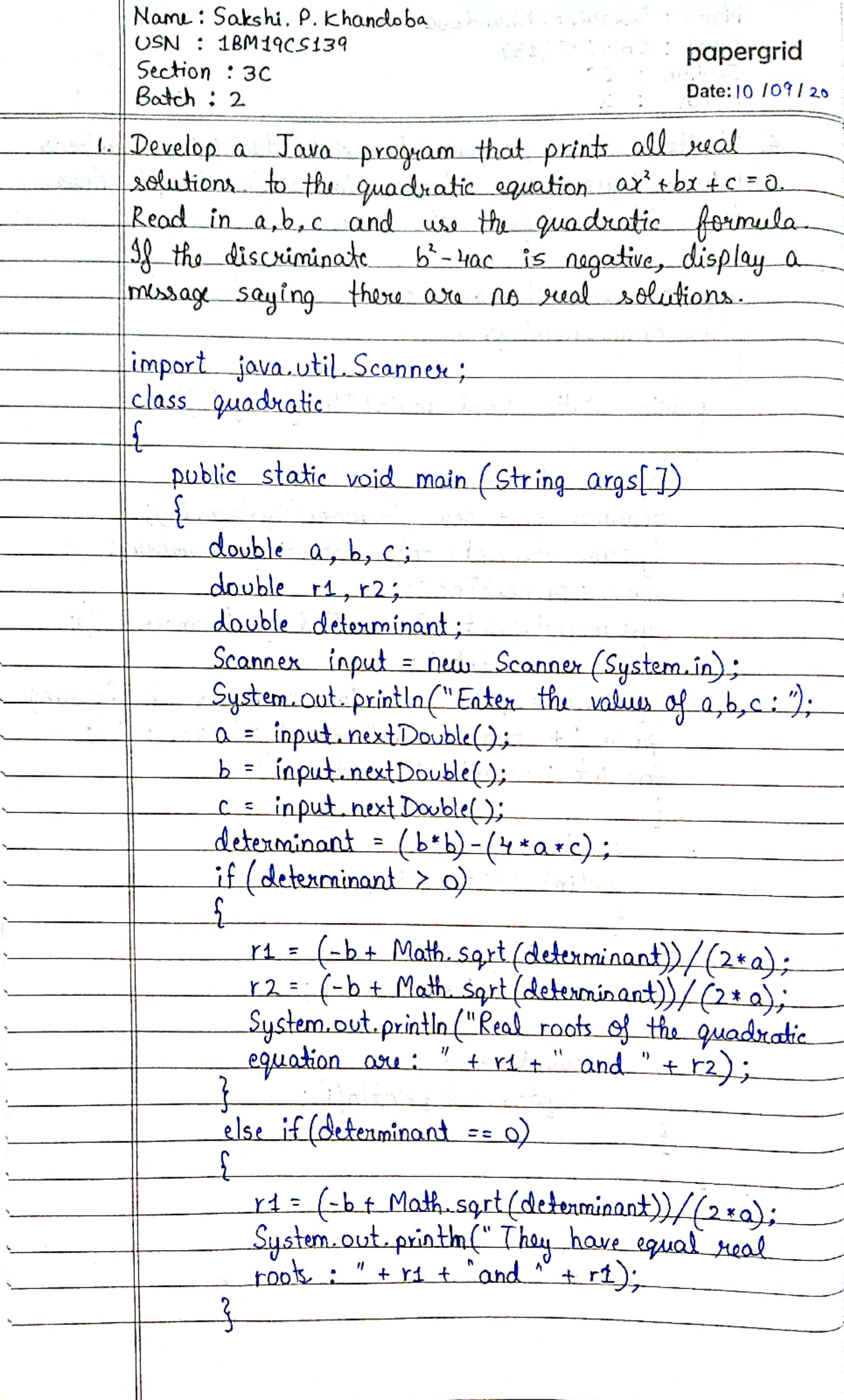
{

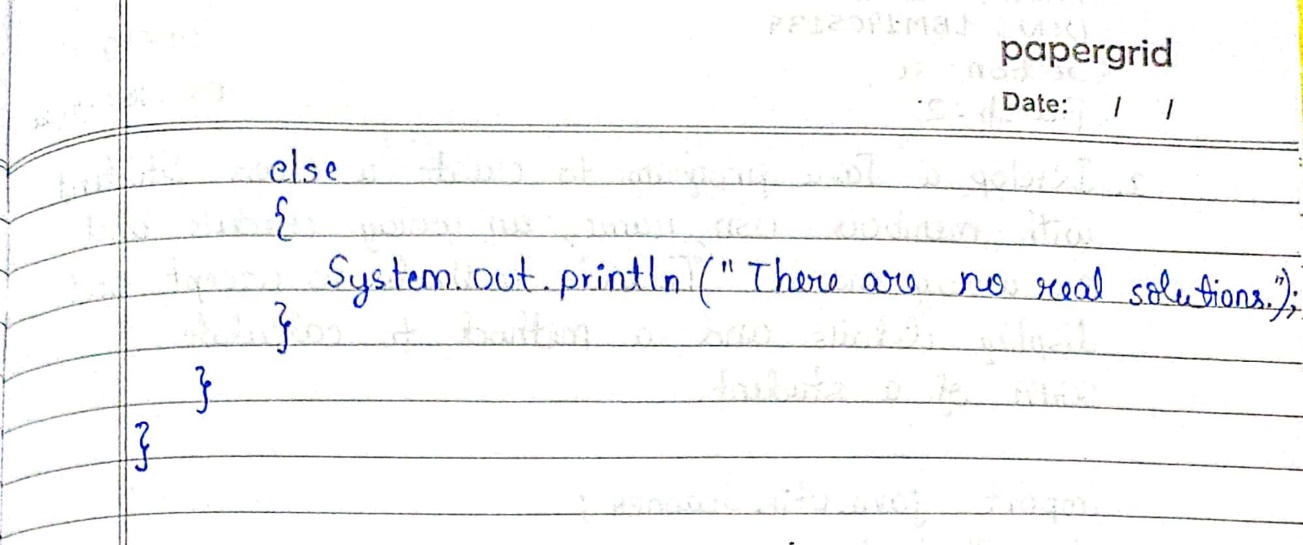
System.out.println("There are no real solutions.");

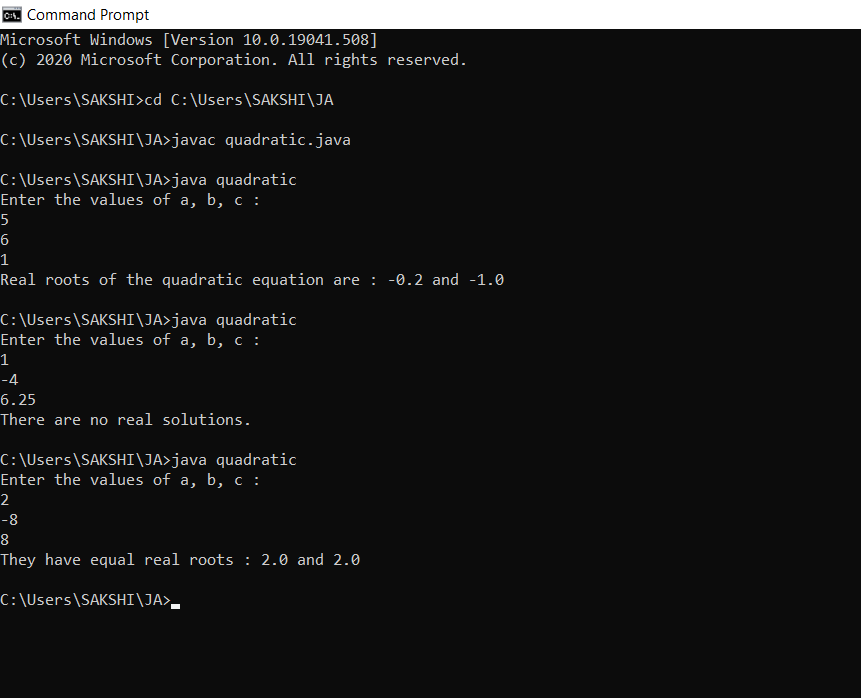
}

}

}







**LAB PROGRAM 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.**

import java.util.Scanner;

class Student

{

String name;

String usn;

int marks[] = new int[5];

int credits[] = new int[5];

int i, n;

int grade=0;

double total=0;

void get\_data()

{

Scanner in = new Scanner(System.in);

System.out.println("Enter Student Name : ");

name = in.next();

System.out.println("Enter Student USN : ");

usn = in.next();

System.out.println("Enter the number of subjects : ");

n = in.nextInt();

System.out.println("Enter subject credits and subject marks : ");

for(i=0;i<n;i++)

{

System.out.println("Credits for subject " + (i+1) + " : ");

credits[i] = in.nextInt();

System.out.println("Marks in subject " + (i+1) + " : ");

marks[i] = in.nextInt();

}

}

void calculate\_sgpa()

{

for(i=0;i<n;i++)

{

if(marks[i] >= 90 && marks[i] <=100)

grade=10;

else if(marks[i] >= 80 && marks[i] < 90)

grade=9;

else if(marks[i] >= 70 && marks[i] < 80)

grade=8;

else if(marks[i] >= 60 && marks[i] < 70)

grade=7;

else if(marks[i] >= 50 && marks[i] < 60)

grade=6;

else if(marks[i] >= 40 && marks[i] < 50)

grade=5;

else if(marks[i] >= 0 && marks[i] < 40)

grade=0;

else

System.out.println("Invalid marks entered");

total = total + (grade\*credits[i]);

}

total = total/20;

System.out.println("SGPA = " + total);

}

void stud\_details()

{

System.out.println("Name : " +name);

System.out.println("USN : " + usn);

System.out.println("Marks and Credits of all subjects : ");

for(i=0;i<n;i++)

{

System.out.println("Subject " + (i+1) + " : ");

System.out.println("Marks : " + marks[i]);

System.out.println("Credits : " + credits[i]);

}

calculate\_sgpa();

}

public static void main(String args[])

{

Student obj = new Student();

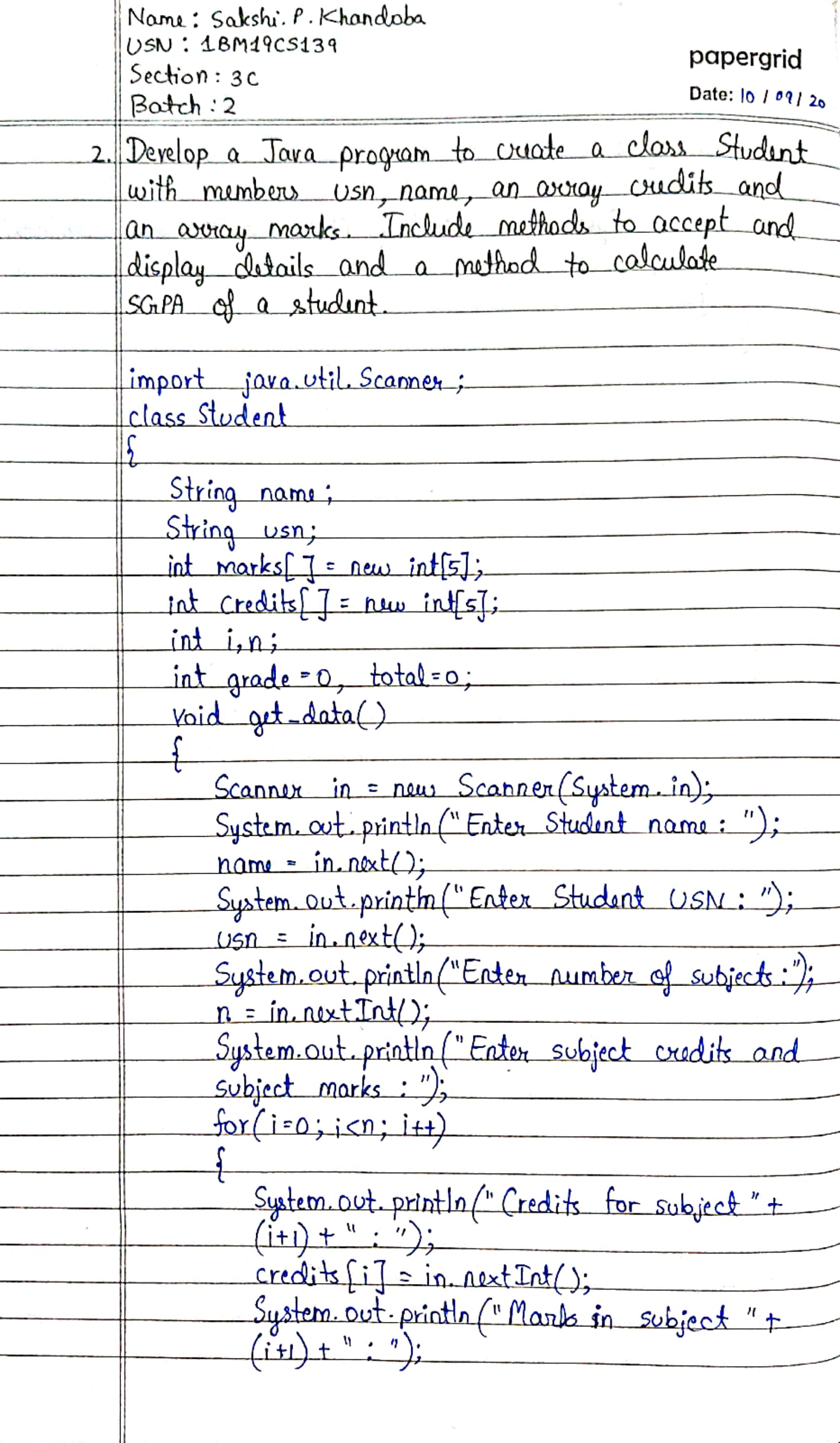
obj.get\_data();

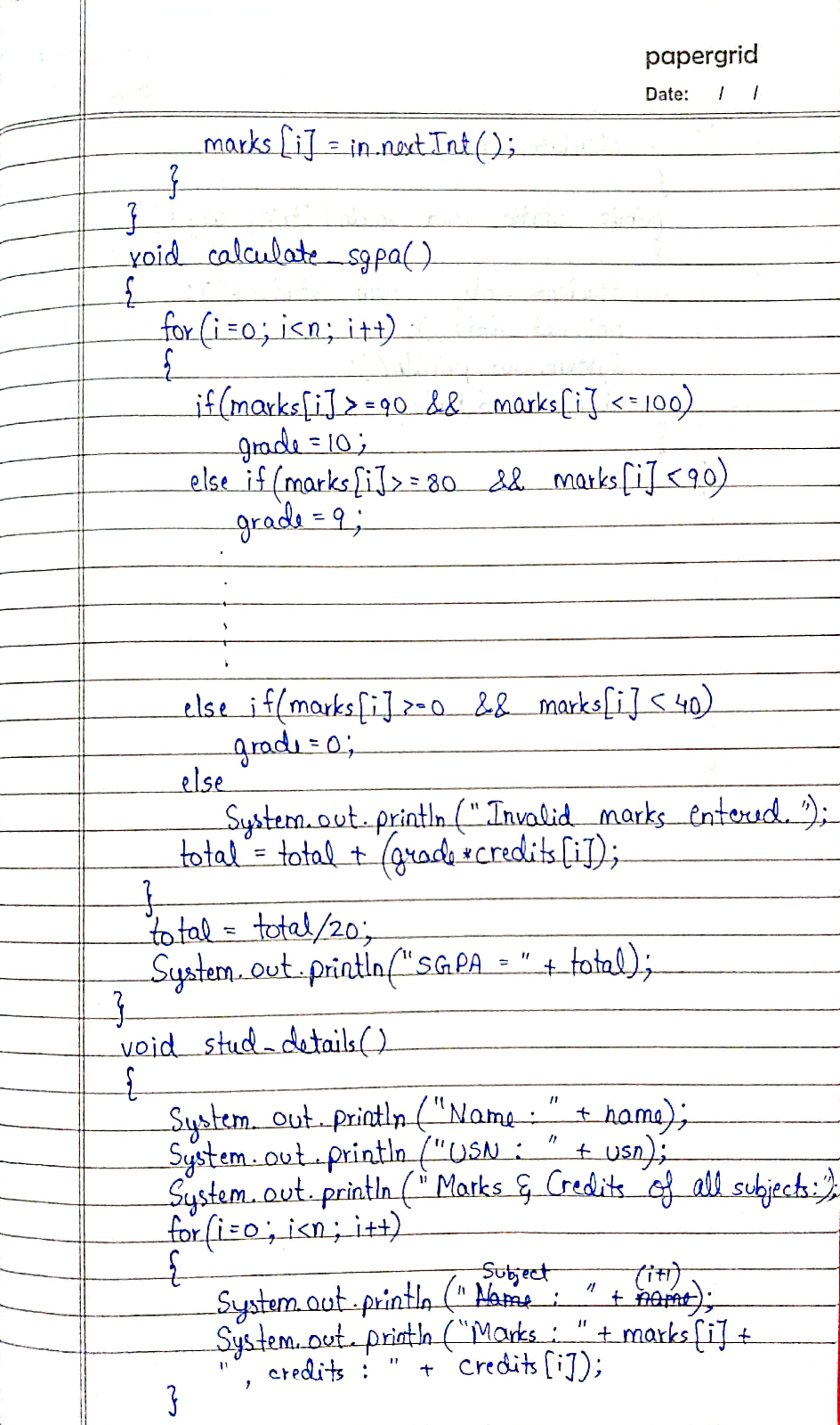
obj.calculate\_sgpa();

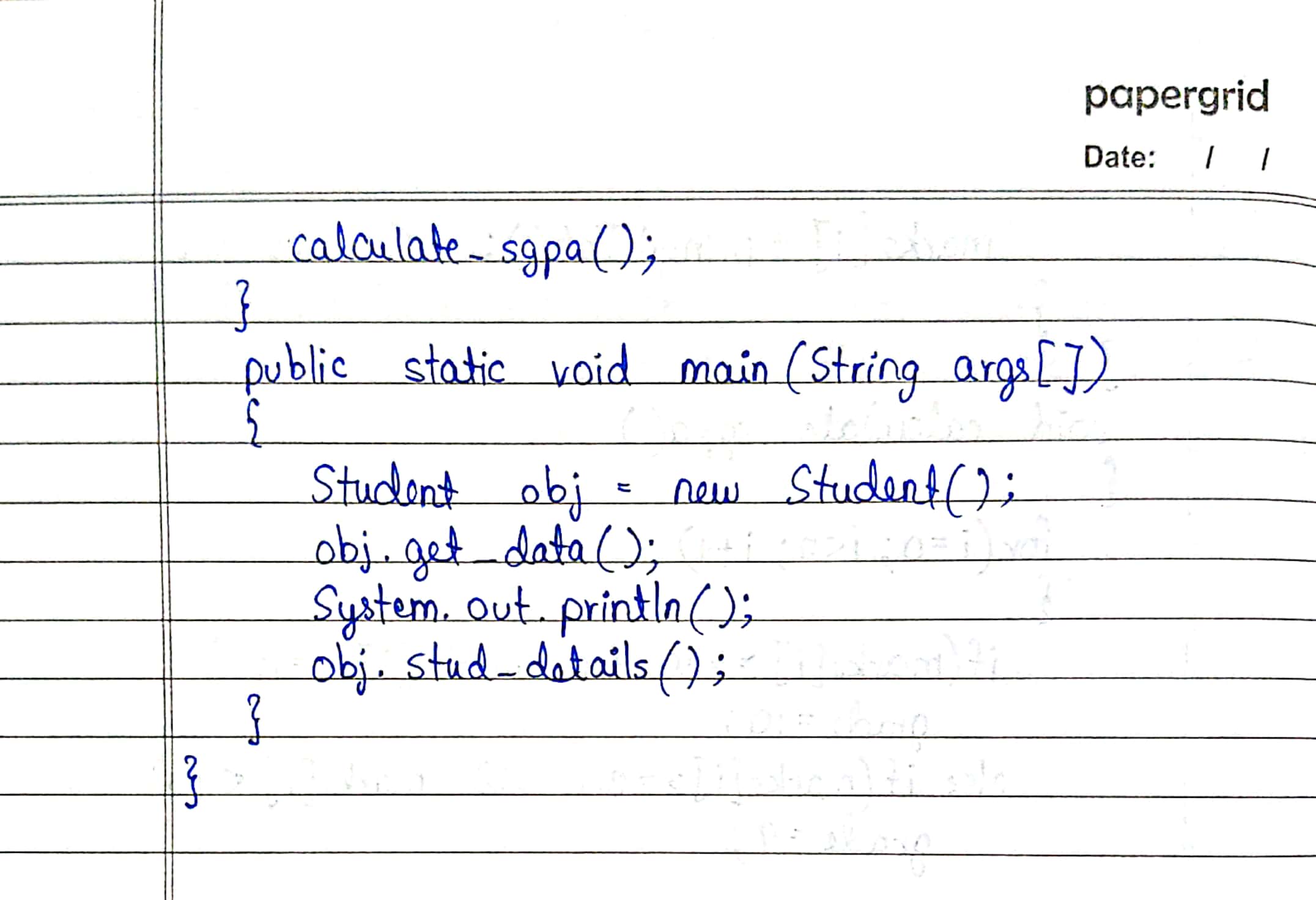
obj.stud\_details();

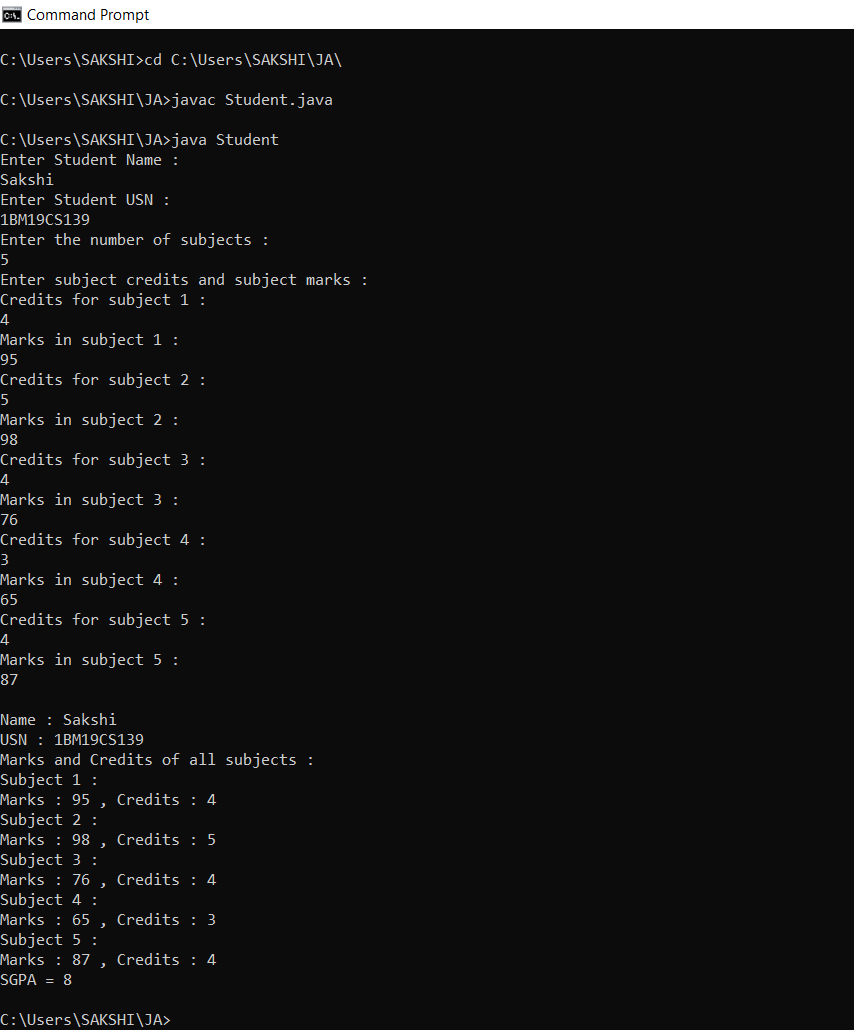
}

}









**LAB 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 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 Book

{

String name;

String author;

int price;

int num\_pages;

Book()

{

Scanner in = new Scanner(System.in);

System.out.println("Enter Book name : ");

name=in.nextLine();

System.out.println("Enter Author name : ");

author=in.nextLine();

System.out.println("Enter Price : ");

price=in.nextInt();

System.out.println("Enter number of pages ; ");

num\_pages=in.nextInt();

}

public String toString()

{

return("Name: "+name+"\nAuthor: "+author+"\nPrice: "+price+"\nNumber 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();

Book b[] = new Book[n];

for(int i=0;i<n;i++)

{

System.out.println("Enter the details of Book "+(i+1)+" : ");

b[i]=new Book();

}

for(int i=0;i<n;i++)

{

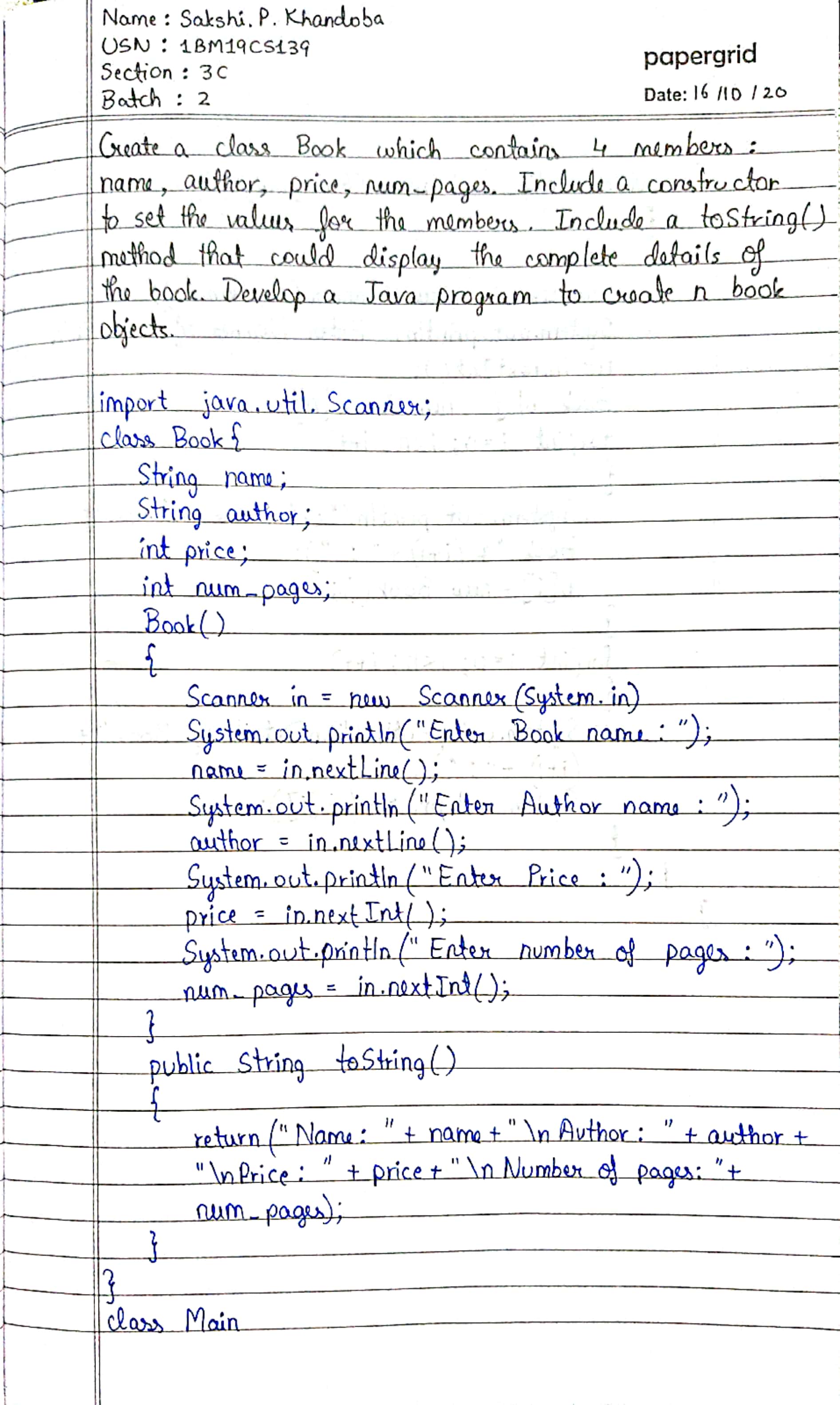
System.out.println("\nDetails of Book "+(i+1)+" : ");

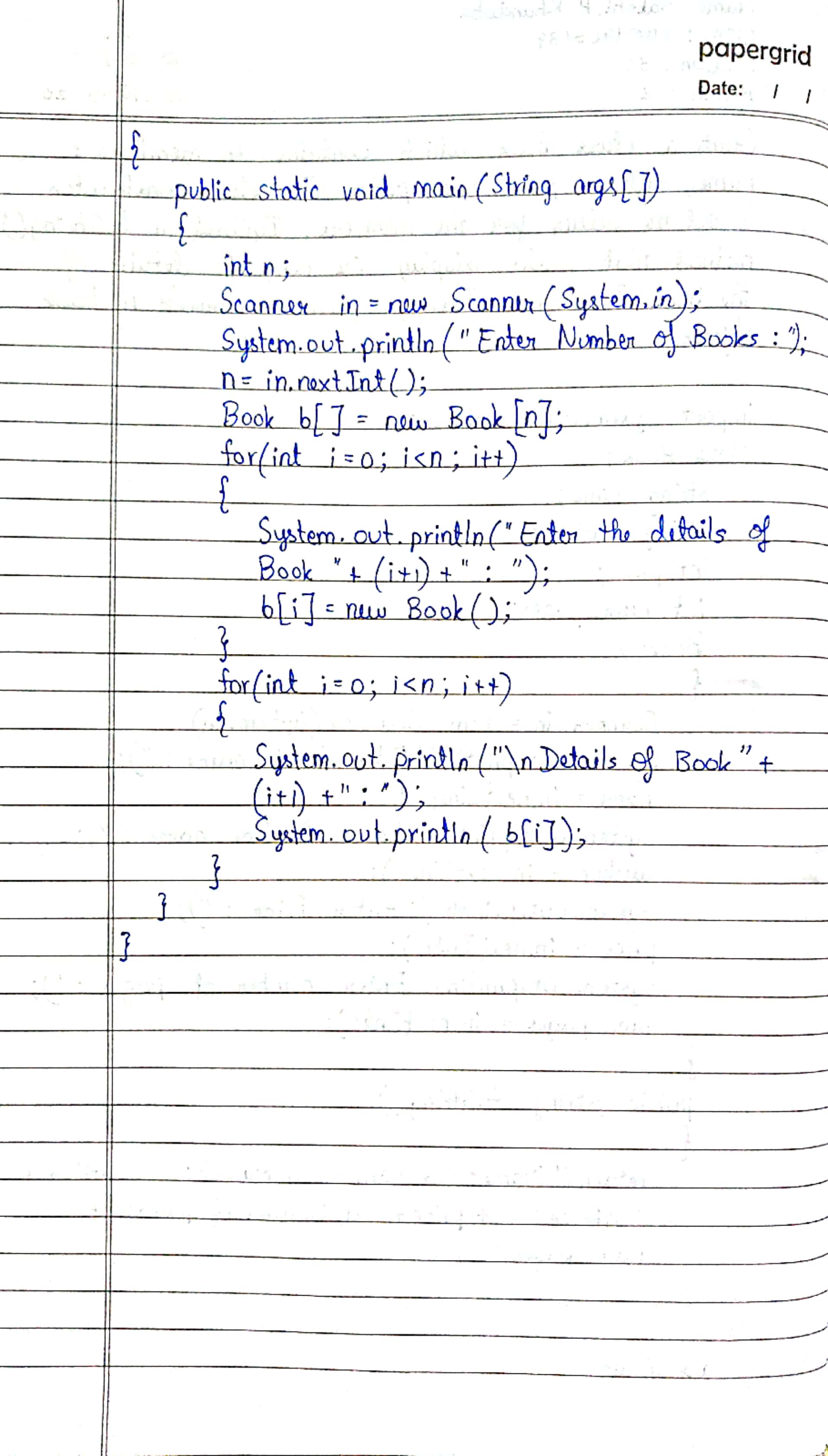
System.out.println(b[i]);

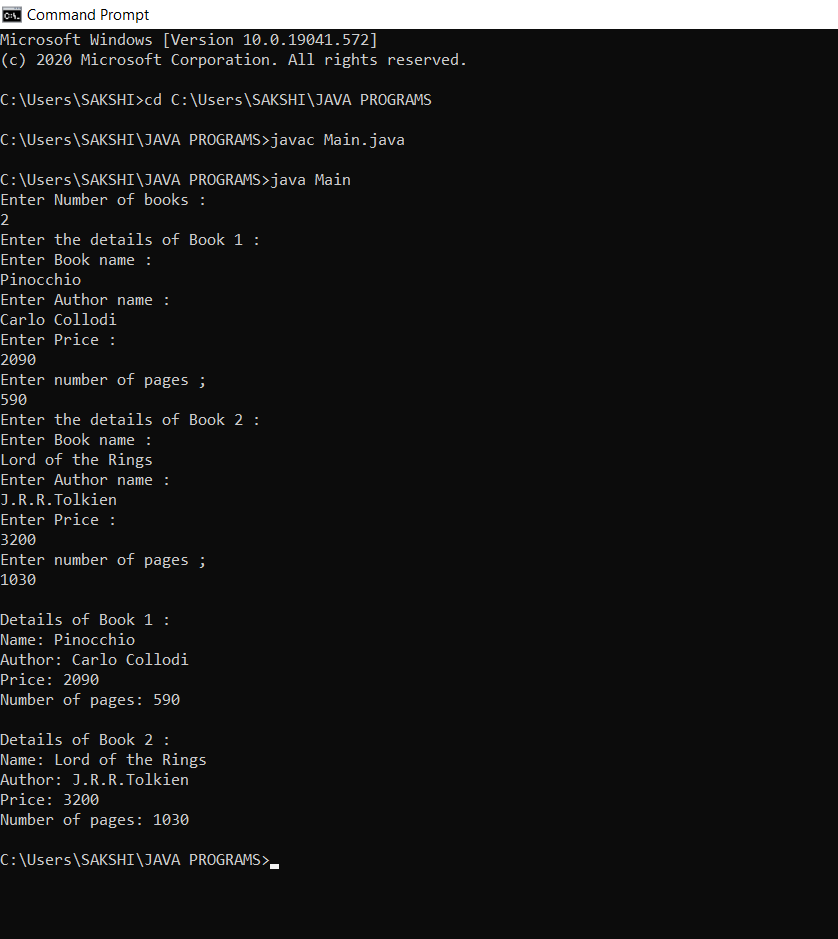
}

}

}







**LAB 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.**

abstract class Shape

{

int a=8, b=6;

abstract void printArea();

}

class Rectangle extends Shape

{

int area\_rectangle;

void printArea()

{

area\_rectangle = a\*b;

System.out.println("Area of rectangle = " + area\_rectangle);

}

}

class Triangle extends Shape

{

float area\_triangle;

void printArea()

{

area\_triangle = (float)(0.5\*a\*b);

System.out.println("Area of triangle = " + area\_triangle);

}

}

class Circle extends Shape

{

float area\_circle\_1, area\_circle\_2;

void printArea()

{

area\_circle\_1 = (float)(3.14\*a\*a);

area\_circle\_2 = (float)(3.14\*b\*b);

System.out.println("Area of circle 1 = " + area\_circle\_1);

System.out.println("Area of circle 2 = " + area\_circle\_2);

}

}

class abstract\_areas

{

public static void main(String args[])

{

Rectangle r = new Rectangle();

r.printArea();

Triangle t = new Triangle();

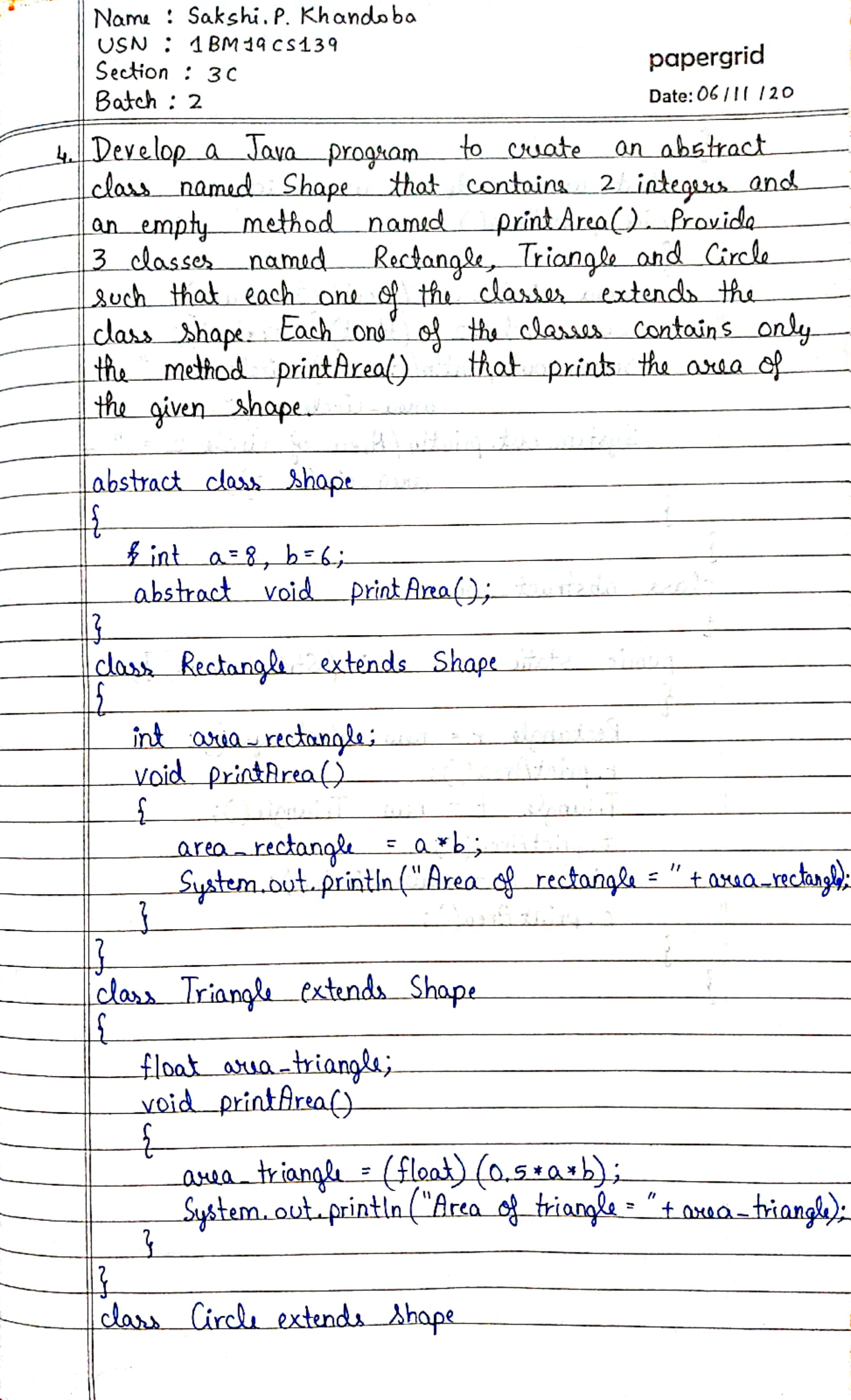
t.printArea();

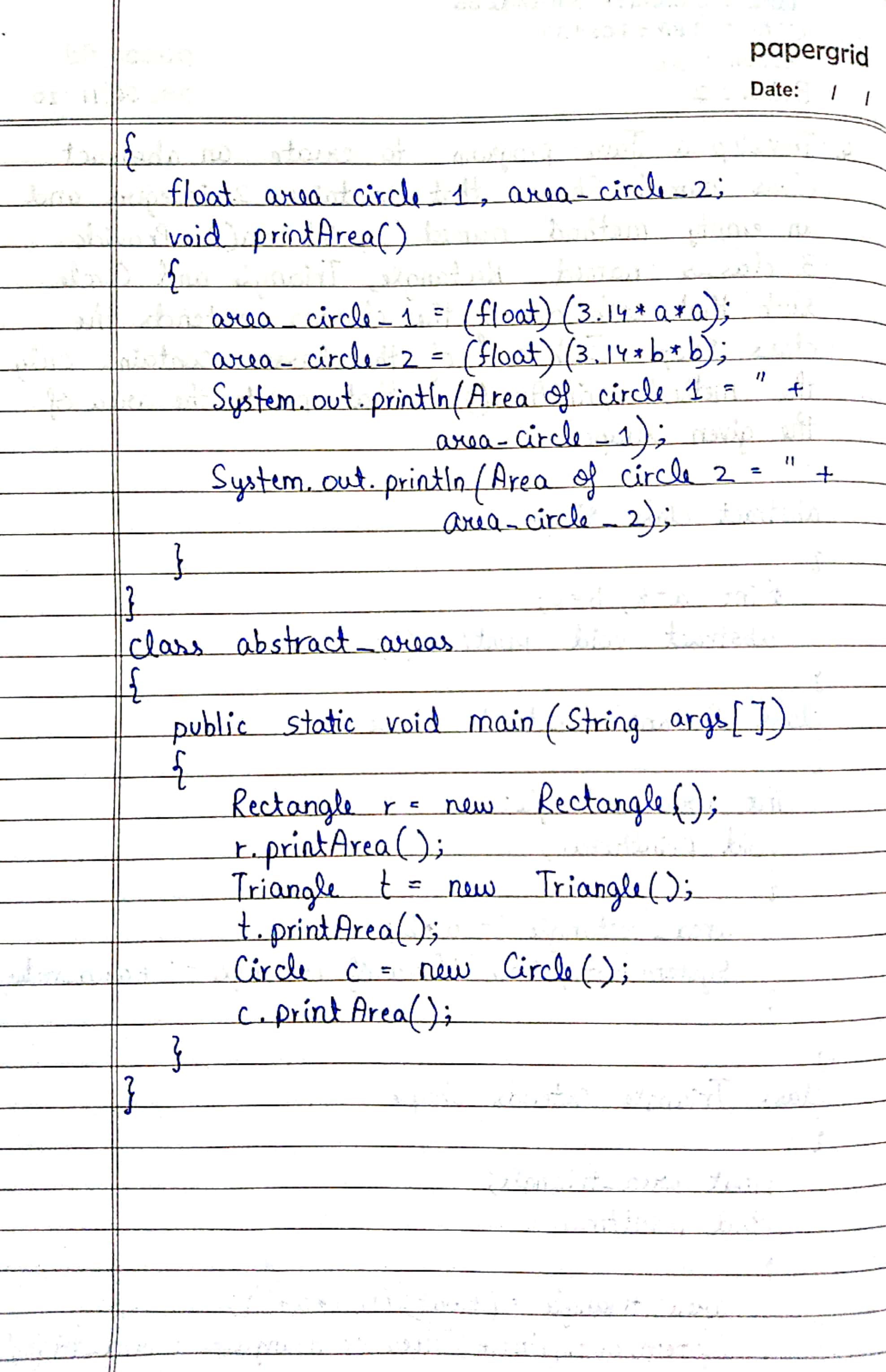
Circle c = new Circle();

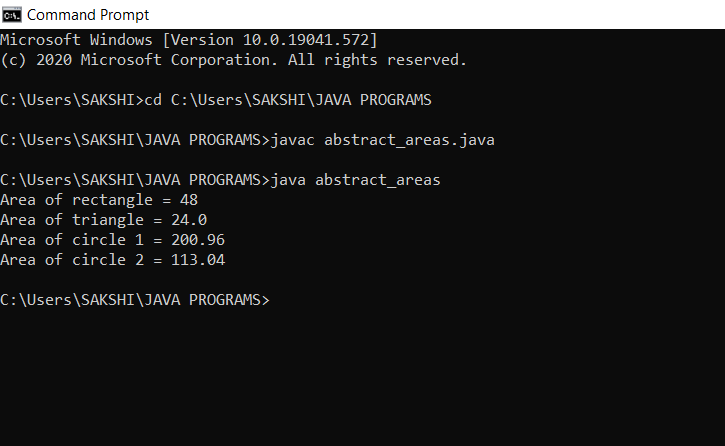
c.printArea();

}

}







**LAB PROGRAM 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 Curr-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks: • Accept deposit from customer and update the balance. • Display the balance. • Compute and deposit interest • Permit 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 cust\_name;

int acc\_num;

String acc\_type;

double balance;

Scanner in = new Scanner(System.in);

Account(String cust\_name,int acc\_num,String acc\_type,double balance)

{

this.cust\_name = cust\_name;

this.acc\_num = acc\_num;

this.acc\_type = acc\_type;

this.balance = balance;

}

void Customer()

{

System.out.println("The " + this.acc\_type + " status is:");

System.out.println("Customer Name: "+this.cust\_name);

System.out.println("Account number: "+this.acc\_num);

System.out.println("Account Type: "+this.acc\_type);

}

void Balance\_Status()

{

System.out.println("Balance Amount: "+this.balance);

}

void Deposit()

{

System.out.println("Enter deposit amount : ");

double deposit= in.nextDouble();

balance+=deposit;

}

}

class Savings extends Account

{

double withdraw;

double deposit;

int rate, time;

double bal, cinterest;

Savings(String cust\_name,int acc\_num,String acc\_type,double balance)

{

super(cust\_name,acc\_num,acc\_type,balance);

}

Scanner in = new Scanner(System.in);

void Compound\_Interest()

{

System.out.println("Compound interest : ");

System.out.println("Enter rate of interest : ");

rate = in.nextInt();

System.out.println("Enter time in years : ");

time = in.nextInt();

bal = balance\*Math.pow(1+(rate\*0.01),time);

cinterest = bal - balance;

System.out.println("Compound interest is : " + cinterest);

balance = bal;

}

void Withdraw()

{

System.out.println("Enter the amount to be withdrawn : ");

withdraw = in.nextDouble();

if(balance<withdraw)

{

System.out.println("Not enough balance. Cannot withdraw.");

withdraw=0.0;

}

else

{

balance -= withdraw;

}

System.out.println("Amount withdrawn = " + withdraw);

}

}

class Current extends Account

{

double withdraw;

double deposit;

double min\_balance = 5000;

Scanner in = new Scanner(System.in);

Current(String cust\_name,int acc\_num,String acc\_type,double balance)

{

super(cust\_name,acc\_num,acc\_type,balance);

}

void Withdraw()

{

System.out.println("Enter the amount to be withdrawn : ");

withdraw = in.nextDouble();

if(balance<withdraw)

{

System.out.println("Not enough balance. Cannot withdraw.");

withdraw=0.0;

}

else

{

balance -= withdraw;

}

System.out.println("Amount withdrawn = " + withdraw);

}

void Minimum\_balance()

{

if(balance<min\_balance)

{

System.out.println("Since balance amount is less than the minimum balance, service charge of 500 is imposed.");

balance = balance - 500;

}

}

}

public class Bank

{

public static void main(String args[])

{

String cust\_name;

int acc\_num, type;

double balance;

Scanner xx = new Scanner(System.in);

System.out.println("Enter Customer Name : ");

cust\_name = xx.next();

System.out.println("Enter Account Number : ");

acc\_num = xx.nextInt();

System.out.println("Enter Account Type : ");

System.out.println("1. Savings Account");

System.out.println("2. Current Account");

type = xx.nextInt();

if(type==1)

{

System.out.println("Enter the balance amount : ");

balance = xx.nextInt();

Savings s = new Savings(cust\_name,acc\_num,"Savings",balance);

s.Customer();

s.Balance\_Status();

s.Deposit();

s.Balance\_Status();

s.Withdraw();

s.Balance\_Status();

s.Compound\_Interest();

s.Customer();

s.Balance\_Status();

}

else if(type==2)

{

System.out.println("Enter the balance amount : ");

balance = xx.nextInt();

Current c = new Current(cust\_name,acc\_num,"Current",balance);

c.Customer();

c.Balance\_Status();

c.Deposit();

c.Balance\_Status();

c.Withdraw();

c.Balance\_Status();

c.Minimum\_balance();

c.Customer();

c.Balance\_Status();

}

else

System.out.println("Invalid choice");

}

}

