ISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

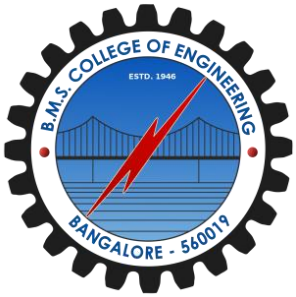
LAB REPORT
on

Object Oriented Java Programming

Submitted by:

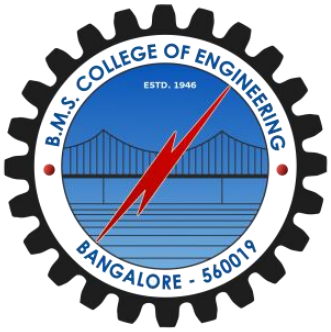
Sanchay Agrawal (1BM21CS186)

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
Oct 2022-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 "**Object Oriented Java Programming**" carried out by **Sanchay Agrawal (1BM21CS186),** 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 2022-23. The Lab report has been approved as it satisfies the academic requirements in respect of **Object Oriented Java Programming-(21CS3PCOOJ)** work prescribed for the said degree.

Prameetha Pai
Assistant Professor
Department of CSE
BMSCE, Bengaluru

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

Table Of Contents

1. Course Outcomes

CO1: Apply the knowledge of Java concepts to find the solution for a given problem.

CO2: Analyse the given Java application for correctness/functionalities.

CO3: Develop Java programs / applications for a given requirement.

CO4: Conduct practical experiments for demonstrating features of Java.

2. Experiments

2.1 Experiment - 1

2.1.1 Question:

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.

2.1.2 Code:

```
import java.util.*;

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

	Scanner sc = new Scanner(System.in);

		System.out.println("Enter value of a: ");
		double a = sc.nextDouble();

		System.out.println("Enter value of b: ");
		double b = sc.nextDouble();

		System.out.println("Enter value of c: ");
		double c = sc.nextDouble();

		double d = (b*b)-(4*a*c);

		if (d>0)
		{
			double r1 = (-b+Math.sqrt(d))/(2*a);
			double r2 = (-b-Math.sqrt(d))/(2*a);
			System.out.println("\nRoots are Real and Unequal\n");
			System.out.format("Root 1: %.2f\n", r1);
			System.out.format("Root 2: %.2f\n", r2);
```

```
            }

            else if (d==0)
            {
                  double r1,r2;
                  r1= r2 = -b/(2*a);
                  System.out.println("\nRoots are Real and Equal\n");
                  System.out.format("Root 1 = Root 2 = %.2f\n", r1, r2);
            }

            else
            {
                  double real = -b / (2 * a);
                  double imaginary = Math.sqrt(-d) / (2 * a);
                  System.out.println("\nRoots are Imaginary\n");
            System.out.format("Root1 = %.2f+%.2fi\n", real, imaginary);
                  System.out.format("Root2 = %.2f-%.2fi\n", real, imaginary);

            }
            sc.close();
      }
}
```

2.1.3 Observation Notebook Pictures

classmate
Date 18/11/2022
Page

Experiment - 1

Aim:
To find roots of a quadratic equation.

Code:

```
import java.util.*;

Class QuadraticEquation {
    public static void main(String args[]) {

    Scanner sc = new Scanner(System.in);

    System.out.println("Enter Value of a: ");
    double a = sc.nextDouble();

    System.out.println("Enter value of b: ");
    double b = sc.nextDouble();

    System.out.println("Enter value of c: ");
    double c = sc.nextDouble();

    double d= (b*b) - (4*a*c);

    if (d>0)
    {
        double x1 = (-b + Math.sqrt(d)) /(2*a);
        double x2 = (-b - Math.sqrt(d)) /(2*a);
        System.out.format("Root 1: %.2f", x1);
        System.out.format("Root 2: %.2f", x2);
    }
```

```
else if (d == 0)
{
    double x1, x2;
    x1 = x2 = -b/(2*a);
    System.out.format("Root 1 = Root 2 = %.2f", x1, x2);
}

else
{
    double real = -b/(2*a);
    double imaginary = Math.sqrt(-d)/(2*a);
    System.out.format("Root 1 = %.2f + %.2f i", real, imaginary);

    System.out.format("Root 2 = %.2f - %.2f i", real, imaginary);
}
    sc.close();
}
}
```

Output:

Enter value of a = 2	Enter value of a = 8	Enter value of a = 1
Enter value of b = 7	Enter value of b = 9	Enter value of b = 2
Enter value of c = 5	Enter value of c = 3	Enter value of c = 1
Root 1 = -1.00	Root 1 = -0.56 + 0.24i	Root 1 = -1.00
Root 2 = -2.50	Root 2 = -0.56 - 0.24i	Root 2 = -1.00

2.1.4 Output:

//Roots are Real and Unequal

```
Microsoft Windows [Version 10.0.22621.1105]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Acer\Desktop\Notes (2nd Year)\java practice programs\
Java Programs\Quadratic Equation>javac QuadraticEquation.java

C:\Users\Acer\Desktop\Notes (2nd Year)\java practice programs\
Java Programs\Quadratic Equation>java QuadraticEquation
Enter value of a:
2
Enter value of b:
7
Enter value of c:
5

Roots are Real and Unequal

Root 1: -1.00
Root 2: -2.50
```

//Roots are Imaginary

```
C:\Users\Acer\Desktop\Notes (2nd Year)\java practice programs\
Java Programs\Quadratic Equation>java QuadraticEquation
Enter value of a:
8
Enter value of b:
9
Enter value of c:
3

Roots are Imaginary

Root1 = -0.56+0.24i
Root2 = -0.56-0.24i
```

//Roots are Real and Equal

```
C:\Users\Acer\Desktop\Notes (2nd Year)\java practice programs\
Java Programs\Quadratic Equation>java QuadraticEquation
Enter value of a:
1
Enter value of b:
2
Enter value of c:
1

Roots are Real and Equal

Root 1 = Root 2 = -1.00
```

2.2 Experiment - 2

2.2.1 Question:

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.

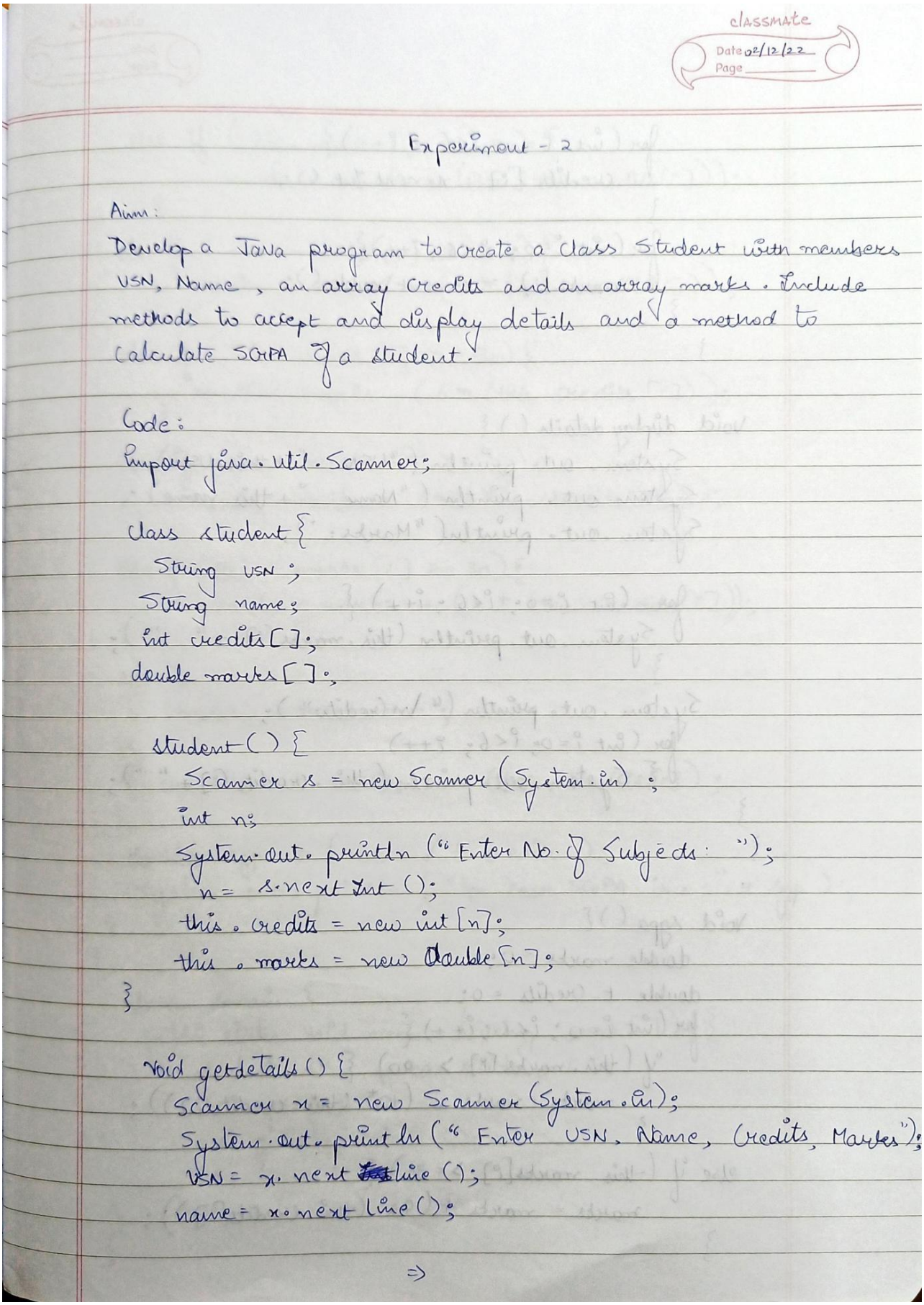
2.2.2 Code:

```
import java.util.Scanner;

class student {
	String usn;
	String name;
	int credits[];
	double marks[];

	student() {
		Scanner s=new Scanner(System.in);
		int n;
		System.out.println("Enter no. of subjects: ");
		n=s.nextInt();
		this.credits=new int[n];
		this.marks=new double[n];

	}

	void getsd() {
		Scanner x=new Scanner(System.in);
		System.out.println("Enter USN, Name, Credits, Marks for subjects");
		usn=x.nextLine();
		name=x.nextLine();
		for(int i=0;i<6;i++) {
			credits[i]=x.nextInt();
		}
		for(int i=0;i<6;i++) {
			marks[i]=x.nextDouble();
		}
	}

	void putsd() {
		System.out.println("USN: "+this.usn);
		System.out.println("Name: "+this.name);
		System.out.println("Marks: ");
		for(int i=0;i<6;i++) {
			System.out.print(this.marks[i]+" ");
		}
		System.out.println("\nCredits: ");
```

```
		for(int i=0;i<6;i++) {
			System.out.print(this.credits[i]+" ");
		}
	}

	void sgpa() {
		double marks=0;
		double t_credits=0;
		for(int i=0;i<6;i++) {
			if(this.marks[i]>=90) {
				marks=marks+(10*(this.credits[i]));
			}
			else if(this.marks[i]>=80) {
				marks=marks+(9*(this.credits[i]));
			}
			else if(this.marks[i]>=70) {
				marks=marks+(8*(this.credits[i]));
			}
			else if(this.marks[i]>=60) {
				marks=marks+(7*(this.credits[i]));
			}
			else if(this.marks[i]>=50) {
				marks=marks+(6*(this.credits[i]));
			}
			else if(this.marks[i]>=40) {
				marks=marks+(5*(this.credits[i]));
			}
			else if(this.marks[i]>=30) {
				marks=marks+(4*(this.credits[i]));
			}
			else {
				marks=marks+0;
			}
			t_credits=t_credits+(this.credits[i]);
		}
		double sgp=(marks/t_credits);
		System.out.println("\nYour SGPA is: "+sgp);
	}
}

class smain {
	public static void main(String xx[]) {
		student s1=new student();
		s1.getsd();
		s1.putsd();
		s1.sgpa();
```

```
    }
}
```

2.2.3 Observation Notebook Pictures:



Experiment - 2

Aim:
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.

Code:

```
Import java.util.Scanner;

class student{
  String USN;
  String name;
  int credits[];
  double marks[];

  student(){
    Scanner s = new Scanner(System.in);
    int n;
    System.out.println("Enter No. of Subjects: ");
     n = s.nextInt();
    this.credits = new int[n];
    this.marks = new double[n];
}

  void getdetails(){
    Scanner x = new Scanner(System.in);
    System.out.println("Enter USN, Name, Credits, Marks");
    USN = x.nextLine();
    name = x.nextLine();

⇒
```

```
        for(int i=0; i<6 ; i++){
            credits[i] = x.nextInt();
        }
        for(int i=0; i<6; i++){
            marks[i] = x.nextDouble();
        }
    }

void displaydetails(){
    System.out.println("USN: " + this.USN);
    System.out.println("Name: " + this.name);
    System.out.println("Marks:");

    for(int i=0; i<6; i++){
        System.out.println(this.marks[i] + "  ");
    }
    System.out.println("\nCredits:");
    for(int i=0; i<6; i++)
    { System.out.println(this.credits[i] + " ");
}
}

void sgpa(){
    double marks=0;
    double t_Credits=0;
    for(int i=0; i<6; i++){
        if(this.marks[i] >= 90){
            marks = marks + (10*(this.credits[i]));
        }
else if(this.marks[i] >= 80){
        marks = marks + (9*(this.credits[i]));
    }
```

```
else if (this.marks[i] >= 70) {
        marks = marks + (8 * (this.credits[i]));
    }
else if (this.marks[i] >= 60) {
        marks = marks + (7 * (this.credits[i]));
    }
else if (this.marks[i] >= 50) {
        marks = marks + (6 * (this.credits[i]));
    }
else if (this.marks[i] >= 40) {
        marks = marks + (5 * (this.credits[i]));
    }
else if (this.marks[i] >= 30) {
        marks = marks + (4 * (this.credits[i]));
}
else {
        marks = marks + 0;
    }
t_credits = t_credits + (this.credits[i]);
}
double sgp = (marks / t_credits);
System.out.println("\n Your SGPA is: " + sgp);
}
}

class Smain {
public static void main (String args[]) {
    student s1 = new student();
    s1.getdetails();
    s1.display details();
    s1.sgpa();
}
}
```

Output:

Enter no. of Subjects: 6

Enter USN, Name, Marks, Credits:

1BM21CS186

Sanchay Agrawal

1 2 2 3 3 4

95

98

96

85

75

97

USN: 1BM21CS186

Name: Sanchay Agrawal

Marks: 95.0, 98.0, 96.0, 85.0, 75.0, 97.0

Credits: 1 2 2 3 3 4

Your SGPA is: 9.4

2.2.4 Output:

```
Microsoft Windows [Version 10.0.22621.819]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Acer>cd C:\Users\Acer\Documents\Java Programs\Student

C:\Users\Acer\Documents\Java Programs\Student>javac Student.java

C:\Users\Acer\Documents\Java Programs\Student>java smain
Enter no. of subjects:
6
Enter USN, Name, Credits, Marks for subjects
1BM21CS186
Sanchay Agrawal
1 2 2 3 3 4
95
98
96
85
75
97
USN: 1BM21CS186
Name: Sanchay Agrawal
Marks:
95.0 98.0 96.0 85.0 75.0 97.0
Credits:
1 2 2 3 3 4
Your SGPA is: 9.4
```

```
C:\Users\Acer\Desktop\Notes (2nd Year)\java practice programs\Java Programs\Student>javac Student.java


C:\Users\Acer\Desktop\Notes (2nd Year)\java practice programs\Java Programs\Student>java smain
Enter no. of subjects:
6
Enter USN, Name, Credits, Marks for subjects
1BM21CS172
Rushil Bindroo
2 1 3 2 3 4
98
91
89
80
92
81
USN: 1BM21CS172
Name: Rushil Bindroo
Marks:
98.0 91.0 89.0 80.0 92.0 81.0
Credits:
2 1 3 2 3 4
Your SGPA is: 9.4
```

2.3 Experiment - 3

2.3.1 Question:

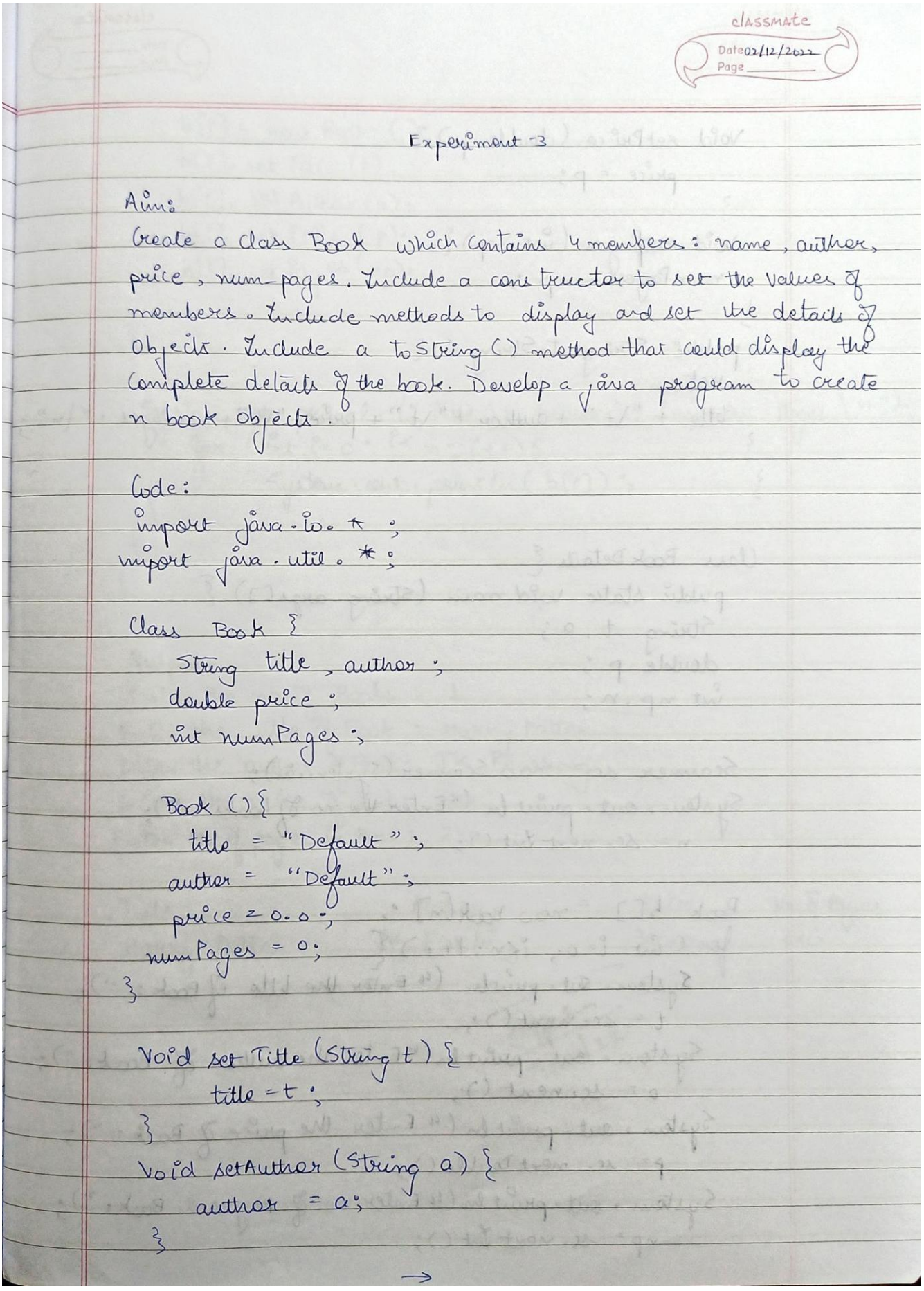
Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString( ) method that could display the complete details of the book. Develop a Java program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

2.3.2 Code:

```
import java.io.*;
import java.util.*;

class Book {

	String title, author;
	double price;
	int numPages;

	Book() {

		title="Default";
		author="Default";
		price=0.0;
		numPages=0;
	}
	void setTitle(String t) {

		title=t;

	}
	void setAuthor(String a) {

		author=a;

	}
	void setPrice(double p) {

		price=p;

	}
	void setPages(int np) {

		numPages=np;
	}

	public String toString() {
```

```
        return title+"\t"+author+"\t"+price+"\t"+numPages+"\n";
    }
}

class BookDetails {

    public static void main(String args[]) {
        String t, a;
        double p;
        int np,n;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number of Books: ");
        n = sc.nextInt();
        Book b[]= new Book[n];
        for(int i=0; i<n;i++) {
            System.out.println("Enter the Title of  the Book: ");
            t= sc.next();
            System.out.println("Enter the Author of  the Book: ");
            a= sc.next();
            System.out.println("Enter the Price of  the Book: ");

            p= sc.nextDouble();
            System.out.println("Enter the Number of pages of the Book: ");
            np= sc.nextInt();

            b[i] = new Book();
            b[i].setTitle(t);
            b[i].setAuthor(a);
            b[i].setPrice(p);
            b[i].setPages(np);
        }

        System.out.println("Title\tAuthor\tPrice\tPages\n");
        for(int i=0; i<n;i++) {
            System.out.println(b[i]);
        }

    }
}
```

2.3.3 Observation Notebook Pictures:



classmate
Date 02/12/2022
Page

Experiment -3

Aim:
Create a class Book which contains 4 members: name, author, price, num_pages. Include a constructor to set the values of members. Include methods to display and set the details of objects. Include a toString() method that could display the complete details of the book. Develop a java program to create n book objects.

Code:

```
import java.io.*;
import java.util.*;

Class Book {
    String title, author;
    double price;
    int numPages;

    Book (){
      title = "Default";
      author = "Default";
      price = 0.0;
     numPages = 0;
}

Void setTitle (String t) {
      title = t;
}
Void setAuthor (String a) {
     author = a;
}
```

→

```
Void setPrice (double p) {
    price = p;
}
Void setPages (int np) {
  numPages = np;
}
public String toString () {
  return
  title + "\t" + author + "\t" + price + "\t" + numPages + "\n";
}
}

Class BookDetails {
  public static void main (String args[]) {
   String t, a;
   double p;
   int np, n;

Scanner sc = new Scanner (System.in);
System.out.println ("Enter the no of books :");
    n = sc.nextInt();

Book b[] = new Book[n];
for (int i=0; i<n; i++) {
   System.out.println ("Enter the title of Book :");
     t = sc.next();
   System.out.println ("Enter the author of Book:");
     a = sc.next();
   System.out.println ("Enter the price of Book:");
     p = sc.nextDouble();
   System.out.println ("Enter no. of pages in Book:");
     np = sc.nextInt();
```

```
b[i] = new Book ();
b[i].setTitle (t);
b[i].setAuthor (a);
b[i].setPrice (p);
b[i].setPages (np);

}

System.out.println ("Title \t Author \t Price \t Pages \n");
    for (int i=0; i<n; i++){
        System.out.println (b[i]);
    }
}
}
```

Output:

```
Enter the no. of Books : 1
Enter the title of Book : Harrey Potter
Enter the author of Book : JK Rowling
Enter the price of Book : 300
Enter No. of pages in Book : 240

Title          Author          Price      No. of Pages
Harrey Potter  JK Rowling      300.0      240
```

O/P
9/12/22

2.3.4 Output:

```
C:\Users\Acer\Desktop\Notes (2nd Year)\java practice programs\Java Programs
\Book>javac BookDetails.java

C:\Users\Acer\Desktop\Notes (2nd Year)\java practice programs\Java Programs
\Book>java BookDetails
Enter the number of Books:
2
Enter the Title of  the Book:
Harry
Enter the Author of  the Book:
Rowling
Enter the Price of  the Book:
200
Enter the Number of pages of the Book:
350
Enter the Title of  the Book:
Captain
Enter the Author of  the Book:
Pilkey
Enter the Price of  the Book:
150
Enter the Number of pages of the Book:
200
Title   Author  Price   Pages

Harry   Rowling 200.0   350

Captain Pilkey  150.0   200
```

2.4 Experiment - 4

2.4.1 Question:

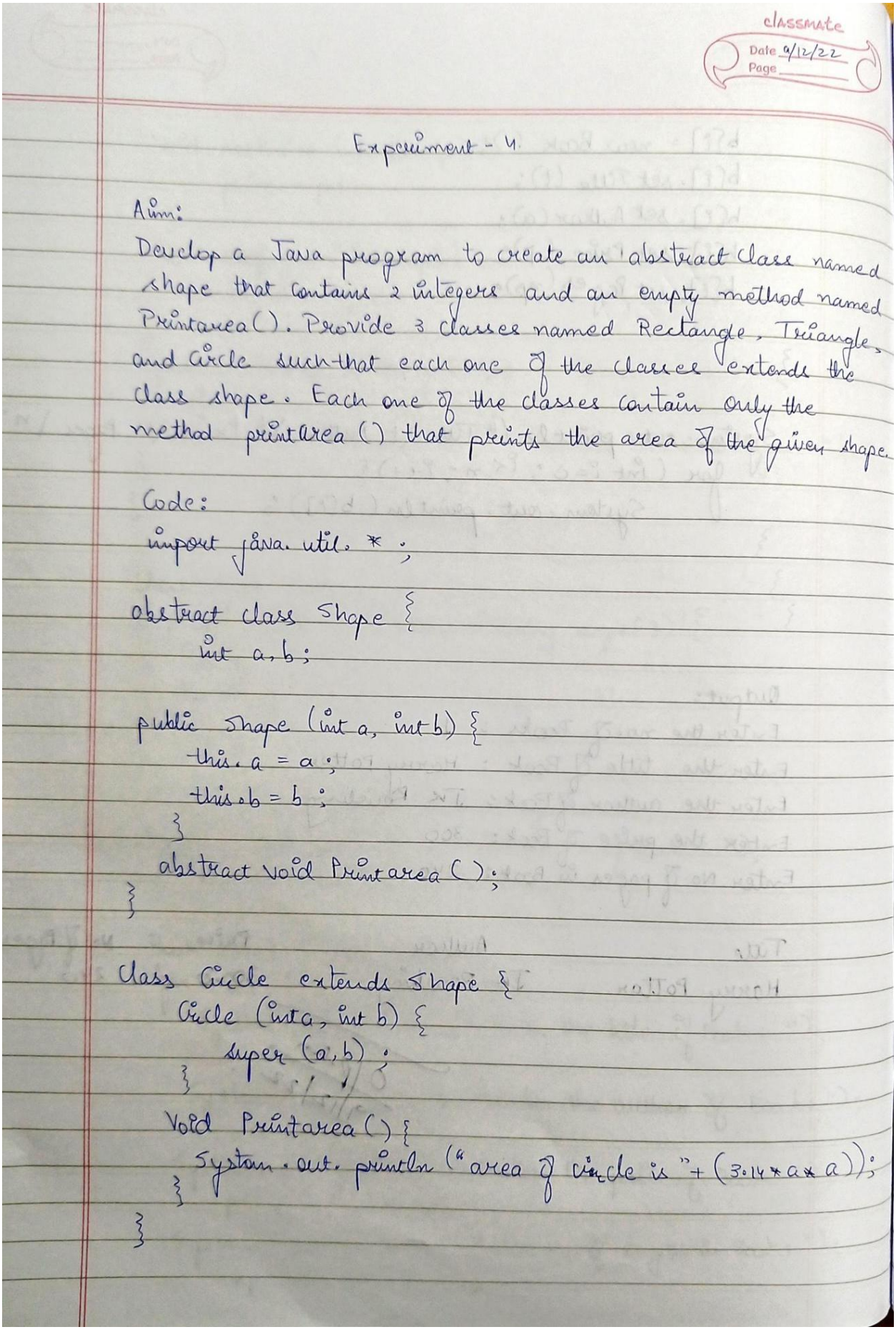
Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea(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

2.4.2 Code:

```
import java.util.*;

abstract class Shape {

int a, b;

public Shape(int a, int b) {
   this.a = a;
   this.b = b;
        }
        abstract void Printarea();
        }

class Circle extends Shape {
   Circle(int a, int b) {
      super(a, b);
   }

   void Printarea()
   {
      System.out.println("Area of Circle is: " + (3.14 * a * a));
   }
}

class Rectangle extends Shape {
 public Rectangle(int a, int b) {
   super(a, b);
}
        void Printarea()
        {
           System.out.println("Area of Rectangle is: " + (a * b));
        }
}

class Triangle extends Shape {
public Triangle(int a, int b) {
   super(a, b);
```

```
}
        void Printarea()
        {
                System.out.println("Area of Triangle is: " + (0.5 * a * b));
        }
}

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

                Scanner in = new Scanner(System.in);

                System.out.println("Enter dimension 1: ");
                int x = in.nextInt();
                System.out.println("\nEnter dimension 2: ");
                int y = in.nextInt();

                Shape b;

                b = new Circle(x, y);
                b.Printarea();

                b = new Rectangle(x, y);
                b.Printarea();

                b = new Triangle(x, y);
                b.Printarea();
        }
}
```

2.4.3 Observation Book Pictures:



Experiment - 4

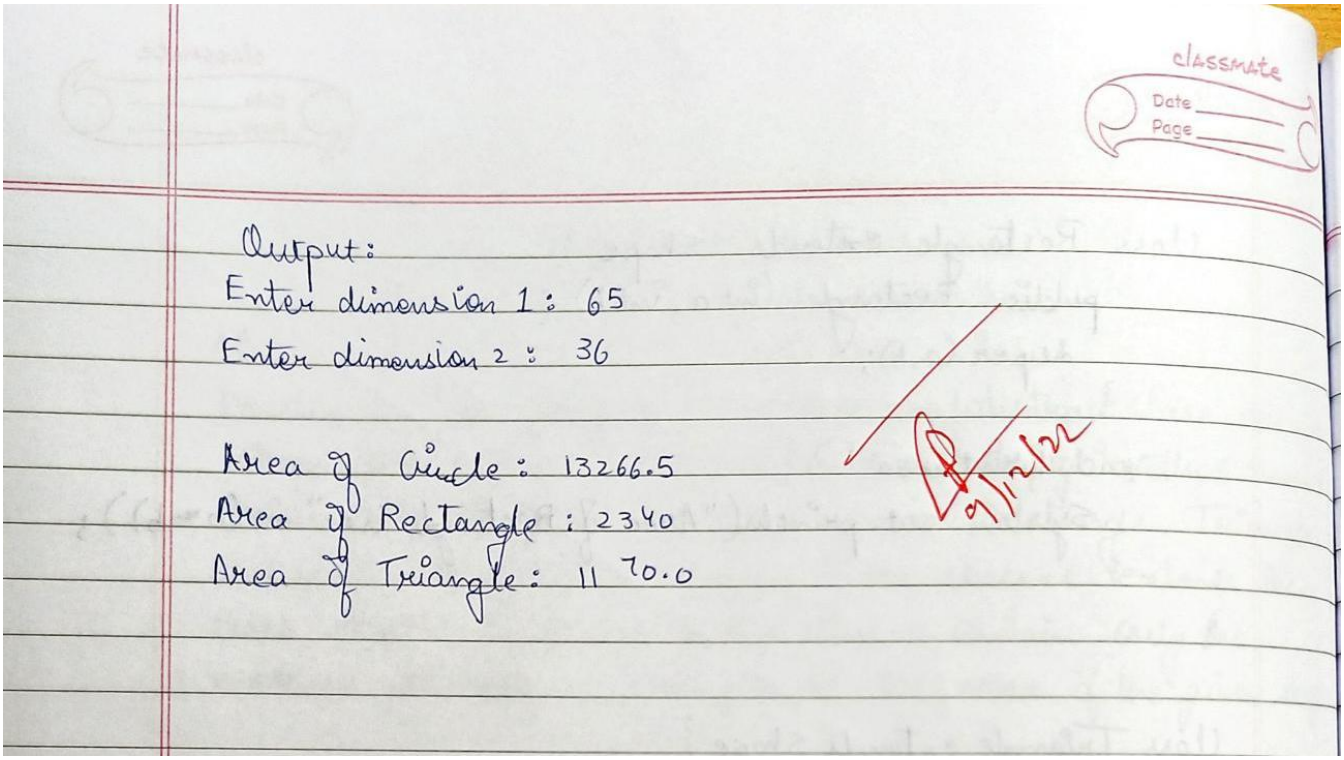
Aim:

Develop a Java program to create an abstract class named Shape that contains 2 integers and an empty method named PrintArea(). Provide 3 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.

Code:

```
import java.util.*;

abstract class Shape {
    int a, b;

public Shape (int a, int b) {
    this.a = a;
    this.b = b;
    }
  abstract void PrintArea();
}

class Circle extends Shape {
    Circle (int a, int b) {
        super (a,b);
    }
    Void PrintArea() {
      System.out.println("area of circle is "+(3.14*a*a));
    }
}
```

```
class Rectangle extends Shape {
    public Rectangle (int a, int b) {
        super (a,b);
    }
    void Printarea () {
        System.out.println("Area of Rectangle is: " + (a*b));
    }
}

class Triangle extends Shape {
    public Triangle (int a, int b) {
        super (a,b);
    }
    void Printarea () {
    System.out.println ("Area of Triangle is: " + (0.5*a*b));
    }
}

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

    System.out.println ("Enter dimension 1: ");
    int x = sc.nextInt();
    System.out.println ("Enter dimension 2: ");
    int y = sc.nextInt();
    Shape b;
    b = new Circle(x,y);
    b.Printarea();
    b = new Rectangle (x,y);
    b.Printarea();
    b = new Triangle (x,y);
    b.Printarea();
}}
```



2.4.4 Output:

```
Microsoft Windows [Version 10.0.22621.1105]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Acer\Desktop\Notes (2nd Year)\java practice programs
\Java Programs\Shape>javac Shape.java

C:\Users\Acer\Desktop\Notes (2nd Year)\java practice programs
\Java Programs\Shape>java Main
Enter dimension 1:
26

Enter dimension 2:
5
Area of Circle is: 2122.64
Area of Rectangle is: 130
Area of Triangle is: 65.0
```

2.5 Experiment - 5

2.5.1 Question:

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.

2.5.2 Code:

```
import java.util.Scanner;

class Account
{
	String name;
	int type;
	long accno;
	double balance;
	void setA()
	{
		Scanner s=new Scanner(System.in);
		System.out.print("Enter Customer Name: ");
		name=s.nextLine();

		System.out.print("Enter Account number: ");
		accno=s.nextLong();
		System.out.println("Account Balance Should Not Be Less than 5000!!!!");
		System.out.print("Enter Bank Balance: ");
		balance=s.nextDouble();
	}

	void display()
	{
		System.out.println("Customer Name is: "+name);
```

```
            if(type==1) {
                System.out.println("Customer Account Type is: Savings");
            }
            else
            {
                System.out.println("Customer Account Type is: Current");
            }
            System.out.println("Customer Account Number is: "+accno);
            System.out.println("Current Balance is: "+balance);
        }

        void deposit()
        {
            System.out.print("Enter the Amount to be Deposited: ");
            Scanner x=new Scanner(System.in);
            double amt=x.nextDouble();
            balance+=amt;
            System.out.println("Updated Balance: "+balance);
        }
}


class Sav_acct extends Account
{
        double interest;
        Scanner s=new Scanner(System.in);

        Sav_acct() {
            type=1;
        }
        void cinterest()
        {
            int timey;
            float irate;
            int times;
            System.out.println("Compound Interest details:");

            System.out.println("Enter Time in Years: ");
            timey=s.nextInt();
            System.out.println("Enter Rate of Interest: ");
            irate=s.nextFloat();
            System.out.println("Enter Number of Times: ");
            times=s.nextInt();
            System.out.println("Interest will be Compunded "+times+" times a year");
            interest=balance*(Math.pow((1+irate/times),(times*timey)));
            balance+=interest;
```

```
            System.out.println("Balance:"+balance);
        }
        void withdraw()
        {
            System.out.println("Enter the Amount to be Withdrawn: ");
            double amt=s.nextDouble();

            if(balance>amt)
            {
                balance-=amt;
                System.out.println("Updated Balance: "+balance);
            }
            else
            {       System.out.println("Amount to be withdrawn greater than balance!!!");
                balance=balance-(balance/10);
                System.out.println("10% penalty has been charged!!!");
                System.out.println("Updated Balance: "+balance);
            }
        }
}

class Curr_acct extends Account
{
        double check_amt;

        Curr_acct() {
            type=2;
        }

        void cheque()
        {
            System.out.print("Enter the cheque amount: ");
            Scanner s=new Scanner(System.in);
            check_amt = s.nextDouble();
            if(check_amt>balance)
            {
                System.out.println("Rs. 500 penalty imposed...Is it ok to proceed? Enter y
                                    for yes and n for no");
                String option=s.next();
                if(option.equals("y"))
                    {balance=balance-500;}
                else {System.out.println("no Check debited");}
            }
            else
            {
                System.out.println("Rupees "+check_amt+" debited");
```

```
                balance-=check_amt;
                System.out.println("Updated Balance: "+balance);
            }
        }
        void withdraw()
        {
            System.out.println("Enter the amount to be withdrawn: ");
            Scanner s=new Scanner(System.in);
            double amt=s.nextDouble();

            if(balance>amt)
            {
                balance-=amt;
                System.out.println("Updated Balance: "+balance);
            }

            else
            {   System.out.println("Amount to be withdrawn greater than balance!!!");
                balance=balance-(balance/10);
                System.out.println("10% penalty has been charged!!!");
                System.out.println("Updated Balance: "+balance);}
        }
}

class Bank {
    public static void main(String ss[]) {
        String op1,op2;
        Scanner s=new Scanner(System.in);
        System.out.println("1. Savings or   2. Current");
        System.out.println("Enter your Choice");
        int q;
        q=s.nextInt();
        if(q==1) {
            Sav_acct s1 = new Sav_acct();
            while(true) {
            System.out.print("\n1. Set the values for Savings Account
                                \n2. Display
                                \n3. Deposit
                                \n4. Interest
                                \n5. Withdraw
                                \n6. Exit
                                \n Cheque Facility Not Available!!\n");
            System.out.println("Enter the choice: ");
            op1=s.next();
            switch(op1)
            {
```

```
                case "1":s1.setA();
                        break;
                case "2":s1.display();
                        break;
                case "3":s1.deposit();
                        break;
                case "4":s1.cinterest();
                        break;
                case "5":s1.withdraw();
                        break;
                case "6":System.exit(0);
                }
                }
        }

        else if(q==2) {
                Curr_acct c1 = new Curr_acct();
                while(true) {
                System.out.print("\n1. Set the values for current account
                                \n2. Display
                                \n3. Deposit
                                \n4. Cheque Facility
                                \n5. Withdraw
                                \n6. Exit\n");
                System.out.println("Enter your Choice: ");
                op2=s.next();
                switch(op2)
                {
                case "1":c1.setA();
                        break;
                case "2":c1.display();
                        break;
                case "3":c1.deposit();
                        break;
                case "4":c1.cheque();
                        break;
                case "5":c1.withdraw();
                        break;
                case "6":System.exit(0);
                }
                }
        }
    }
}
```

2.5.3 Observation Book Pictures:

classmate
Date 9/12/22
Page

Experiment - 5

Aim:
Develop a Java program to create a class Bank that maintains 2 kinds of account for its customers, one called savings accounts and the other current account. The savings account provide compound interest and withdrawal facilities but no cheque book facility. Current account holders should also maintain a minimum balance and if the balance falls below this level a service charge is imposed. The current account provides cheque book facility but no interest.

Create a class Account that stores customer name, account number and type of account. From this derive 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:

1. Accept deposit from customer and update the balance.
2. Display the balance.
3. Compute and deposite interest.
4. Permit withdrawal and update the balance.

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

* Keep fixed minimum balance of 5000 (during account creation, please display the info that the min balance is 5000) and if the balance is less than the min balance then levy/impose 10% of penalty.
* Rate of interest per year must be 6%.

Code:

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

class Bank {
 Scanner sc = new Scanner(System.in);
 String name;
int acc_no;
float bal, si;

 void accept (){
System.out.println("Enter your name:");
 name = sc.nextline();
 System.out.println(" Enter the balance amount: ");
 bal = sc.nextFloat();
}

void display (){
  System.out.println(" Name: " + name);
}

void deposit (){
 float amount;
int choice;
System.out.println(" Deposit (1 for yes, 2 for no)");
choice = sc.nextInt();
 if( choice == 1){
System.out.println(" Enter the amount to be deposited: ");
 amount = sc.nextFloat();
 if (amount > bal){
System.out.println("Amount is bank insufficient ");
```

```
else {
    bal = bal + amount;
}
System.out.println("Current Balance : " + bal);
    }
  }
}

class current extends bank {
  int service_fee = 0.1;

  void cheque_book () {
  System.out.println("Cheque service available");

  void withdrawal (){
    float amount amt;
  System.out.println("Enter the amount to be withdrawn");
  amt = sc.nextFloat();
  if (amt > bal)
    System.out.println("Balance in sufficient");

  else {
    bal = bal - amt;
    if (bal < 1000) {
      bal = bal - service_fee
    System.out.println("10% penalty is taken as service
      charge");
    System.out.println("Withdrawn: " + amt);
    System.out.println("Balance : " + bal);
  }
}
```

```
Class Savings extends Bank {
  Void cheque(){
  System.out.println ("Cheque Service not available ");
  }

  Void withdrawal (){
    float amt ;
  System.out.println ("Enter amount to be withdrawn:" );
    amt = sc.nextFloat ();
   if (amt > bal)
     { System.out.println (" Insufficient balance ");
     bal = bal - 0.1 ;
       System.out.println (" 10% penalty has been charged
                            due to insufficient balance ");
       }
    else
    { bal = bal - amt;
       System.out.println (" Withdrawn: "+amt);
       System.out.println (" Balance : " +bal);
    }
}

Void interest (){
  System.out.println (" Enter the rate of interest : ");
  int r = sc.nextInt ();
  System.out.println (" Enter the no of times interest
                        applied per time period");
  int n = sc.nextInt ();
  System.out.println (" Enter the time elapsed in years ");
  int t = sc.nextInt();
  si = bal * (1 + (r/n)) ;
```

```
System.out.println("Compound Interest is "+(Math.pow(si,n*t)));
    }
  }
}

public class account(){
  public static void main(String args[]){
    Scanner sc = new Scanner(System.in);
    Savings obj1 = new savings();
    Current obj2 = new current();

    System.out.println("1. Savings \n 2. Current");
    int choice = sc.nextInt();
     switch(choice)
      { case 1:
            obj1.accept();
            obj1.display();
            obj1.cheque();
            obj1.deposit();
            obj1.interest();
            obj2.withdrawl();
           break;

case 2:
        obj2.accept();
        obj2.display();
        obj2.cheque();
        obj2.deposit();
        obj2.withdrawal();
       break;
default: System.out.println("Invalid choice");
    }
  }
}
```

Output:

1. Savings Account
2. Current Account

Enter your choice : 1

Enter your Name :
Sanchay
Enter balance amount :
10000

Name :
Sanchay

Cheque Service Not Available

Enter the amount to be withdrawn : 1000
Withdrawn : 1000
Balance : 9000

Deposit (1 for yes, 2 for no)
1
Enter amount to be deposited : 1000
Current Balance : 11000

Enter the rate of interest = 5
Enter the no of times interest applied per time period : 2
Enter time elapsed in years = 2
Compound interest :

Enter the amount to be withdrawn : 1000
Withdrawn : 1000
Balance : 10000

2.5.4 Output:

//Savings Account Operations

//Inserting Account Details

```
Microsoft Windows [Version 10.0.22621.1105]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Acer\Desktop\Notes (2nd Year)\java practice programs
\Java Programs\Bank>javac Bank.java

C:\Users\Acer\Desktop\Notes (2nd Year)\java practice programs
\Java Programs\Bank>java Bank
1. Savings or   2. Current
1

1. Set the values for Savings Account
2. Display
3. Deposit
4. Interest
5. Withdraw
6. Exit
 Cheque Facility Not Available!!
Enter the choice:
1
Enter Customer Name: Sanchay
Enter Account number: 123456789
Account Balance Should Not Be Less than 5000!!!!
Enter Bank Balance: 6000
```

//Displaying Savings Account Details

```
1. Set the values for Savings Account
2. Display
3. Deposit
4. Interest
5. Withdraw
6. Exit
 Cheque Facility Not Available!!
Enter the choice:
2
Customer Name is: Sanchay
Customer Account Type is: Savings
Customer Account Number is: 123456789
Current Balance is: 6000.0
```

//Depositing in Savings Account

```
1. Set the values for Savings Account
2. Display
3. Deposit
4. Interest
5. Withdraw
6. Exit
 Cheque Facility Not Available!!
Enter the choice:
3
Enter the Amount to be Deposited: 2000
Updated Balance: 8000.0
```

//Computing and Depositing Interest in Savings Account

```
1. Set the values for Savings Account
2. Display
3. Deposit
4. Interest
5. Withdraw
6. Exit
 Cheque Facility Not Available!!
Enter the choice:
4
Compound Interest details:
Enter Time in Years:
2
Enter Rate of Interest:
3.5
Enter Number of Times:
2
Interest will be Compunded 2 times a year
Balance:465531.25
```

//Withdraw Amount From Savings Account
//Withdraw Amount Greater than Account Balance

```
1. Set the values for Savings Account
2. Display
3. Deposit
4. Interest
5. Withdraw
6. Exit
 Cheque Facility Not Available!!
Enter the choice:
5
Enter the Amount to be Withdrawn:
500000
Amount to be withdrawn greater than balance!!!
10% penalty has been charged!!!
Updated Balance: 418978.125
```

//Withdraw Amount Less than Account Balance

```
1. Set the values for Savings Account
2. Display
3. Deposit
4. Interest
5. Withdraw
6. Exit
 Cheque Facility Not Available!!
Enter the choice:
5
Enter the Amount to be Withdrawn:
200000
Updated Balance: 218978.125
```

//Current Account Operations

//Inserting Current Account Details

```
C:\Users\Acer\Desktop\Notes (2nd Year)\java practice programs\J
ava Programs\Bank>javac Bank.java

C:\Users\Acer\Desktop\Notes (2nd Year)\java practice programs\J
ava Programs\Bank>java Bank
1. Savings or   2. Current
2

1. Set the values for current account
2. Display
3. Deposit
4. Cheque Facility
5. Withdraw
6. Exit
Enter your Choice:
1
Enter Customer Name: Sanchay
Enter Account number: 789456123
Account Balance Should Not Be Less than 5000!!!!
Enter Bank Balance: 10000
```

//Displaying Current Account Details

```
1. Set the values for current account
2. Display
3. Deposit
4. Cheque Facility
5. Withdraw
6. Exit
Enter your Choice:
2
Customer Name is: Sanchay
Customer Account Type is: Current
Customer Account Number is: 789456123
Current Balance is: 10000.0
```

//Depositing in Current Account

```
1. Set the values for current account
2. Display
3. Deposit
4. Cheque Facility
5. Withdraw
6. Exit
Enter your Choice:
3
Enter the Amount to be Deposited: 1500
Updated Balance: 11500.0
```

//Withdrawing From Current Account
//Withdraw Amount Less than Account Balance

```
1. Set the values for current account
2. Display
3. Deposit
4. Cheque Facility
5. Withdraw
6. Exit
Enter your Choice:
5
Enter the amount to be withdrawn:
2000
Updated Balance: 9500.0
```

//Withdraw Amount Greater than Account Balance

```
1. Set the values for current account
2. Display
3. Deposit
4. Cheque Facility
5. Withdraw
6. Exit
Enter your Choice:
5
Enter the amount to be withdrawn:
8000
Amount to be withdrawn greater than balance!!!
10% penalty has been charged!!!
Updated Balance: 7020.0
```

//Cheque Withdraw

//Cheque Amount Less than Account Balance

```
1. Set the values for current account
2. Display
3. Deposit
4. Cheque Facility
5. Withdraw
6. Exit
Enter your Choice:
4
Enter the cheque amount: 1200
Rupees 1200.0 debited
Updated Balance: 8300.0
```

//Cheque Amount Greater than Account Balance

```
1. Set the values for current account
2. Display
3. Deposit
4. Cheque Facility
5. Withdraw
6. Exit
Enter your Choice:
4
Enter the cheque amount: 8500
Rs. 500 penalty imposed...Is it ok to proceed? Enter y for yes
and n for no
y

1. Set the values for current account
2. Display
3. Deposit
4. Cheque Facility
5. Withdraw
6. Exit
Enter your Choice:
2
Customer Name is: Sanchay
Customer Account Type is: Current
Customer Account Number is: 789456123
Current Balance is: 7800.0
```

2.6 Experiment - 6

2.6.1 Question:

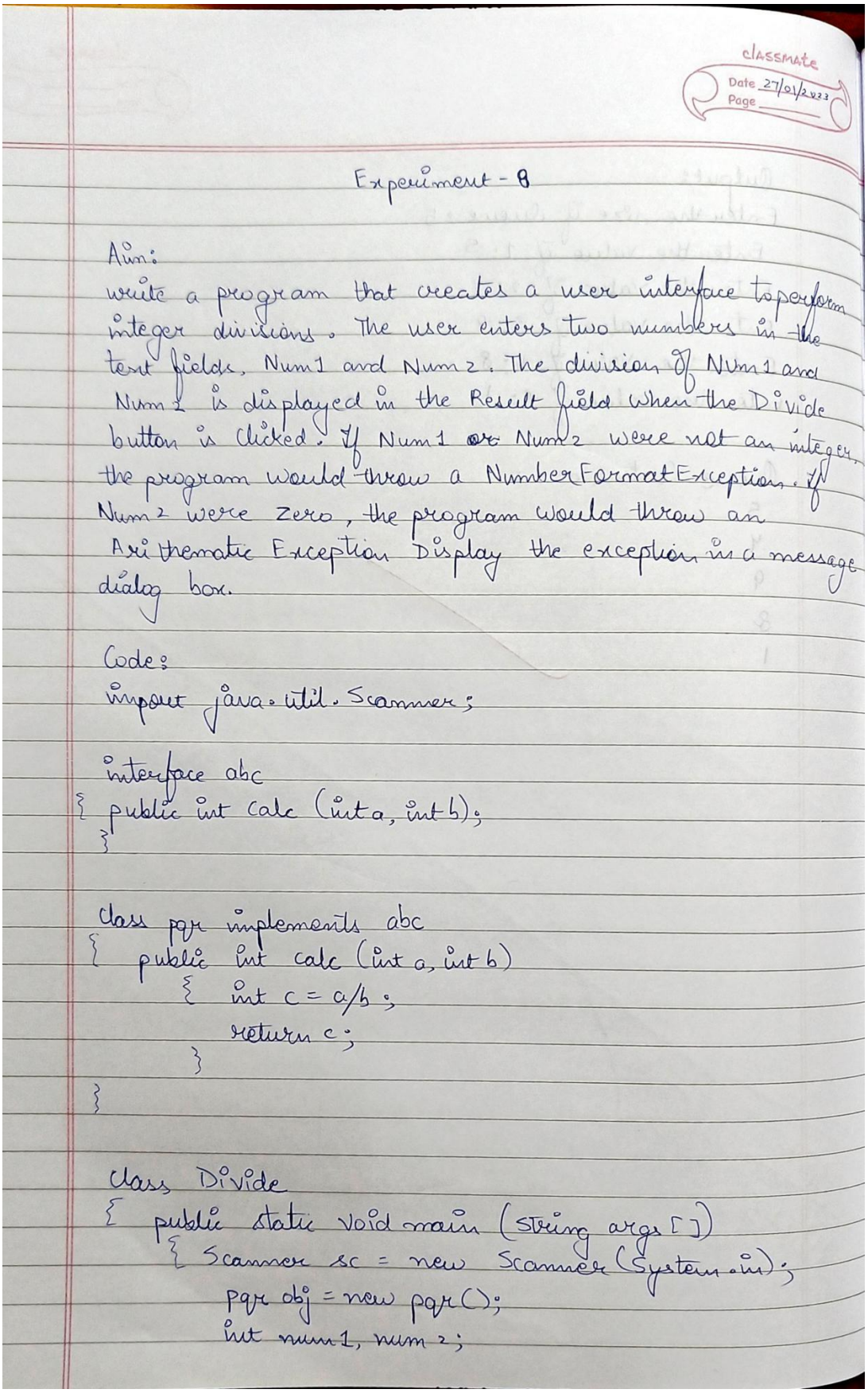
Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception. Display the exception in a message dialog box.

2.6.2 Code:

```
import java.util.Scanner;

interface abc
{
   public int calc(int a,int b);
}

class pqr implements abc
{
   public int calc(int a, int b)
   {
      int c = a/b;
      return c;
   }
}

class Divide
{
   public static void main(String[] args)
   {
      Scanner sc = new Scanner(System.in);
      pqr obj = new pqr();
      int num1,num2;

      try
      {
         System.out.println("Enter the two numbers: ");
         num1 = sc.nextInt();
         num2 = sc.nextInt();
         int c = obj.calc(num1,num2);
         System.out.println("Division Result: "+c);
      }

      catch(ArithmeticException | NumberFormatException e)
      {
```

```
            System.out.println("Exception: "+e);
        }
    }
}
```

2.6.3 Observation Book Pictures:



Experiment - 8

Aim:
write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were zero, the program would throw an ArithmeticException Display the exception in a message dialog box.

Code:

```
import java.util.Scanner;

interface abc
{ public int calc (int a, int b);
}

class pqr implements abc
{  public int calc (int a, int b)
      { int c = a/b;
        return c;
      }
}

class Divide
{  public static void main (String args[])
    { Scanner sc = new Scanner(System.in);
      pqr obj = new pqr();
      int num1, num2;
```

```
try
{ System.out.println("Enter 2 numbers: ");
  num1 = sc.nextInt();
  num2 = sc.nextInt();
  int c = obj.calc(num1, num2);
  System.out.println("Division result: " +c);
}

Catch(ArithmeticException | NumberFormatException e)
{ System.out.println("Exception: " +e);
}
  }
}
```

Output:

```
Enter the 2 numbers:
18
9
Division Result: 2

Enter the 2 numbers:
18
0
Exception: java.lang.ArithmeticException: / by zero

Enter the 2 numbers:
0.15
Exception in thread "main" java.util.InputMismatchException
    at java.base/java.util.Scanner.throwFor(Scanner.java:939)
    at java.base/java.util.Scanner.next(Scanner.java:1594)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2258)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2212)
    at Divide.main(Inter_Division.java:33)
```

2.6.4 Output:

```
Microsoft Windows [Version 10.0.22621.1105]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Acer\Desktop\Notes (2nd Year)\java practice programs>javac Integer_Division.java


C:\Users\Acer\Desktop\Notes (2nd Year)\java practice programs>java Divide
Enter the two numbers:
18
9
Division Result: 2

C:\Users\Acer\Desktop\Notes (2nd Year)\java practice programs>java Divide
Enter the two numbers:
18
0
Exception: java.lang.ArithmeticException: / by zero

C:\Users\Acer\Desktop\Notes (2nd Year)\java practice programs>java Divide
Enter the two numbers:
0.15
Exception in thread "main" java.util.InputMismatchException
        at java.base/java.util.Scanner.throwFor(Scanner.java:939)
        at java.base/java.util.Scanner.next(Scanner.java:1594)
        at java.base/java.util.Scanner.nextInt(Scanner.java:2258)
        at java.base/java.util.Scanner.nextInt(Scanner.java:2212)
        at Divide.main(Integer_Division.java:33)
```

2.7 Experiment - 7

2.7.1 Question:

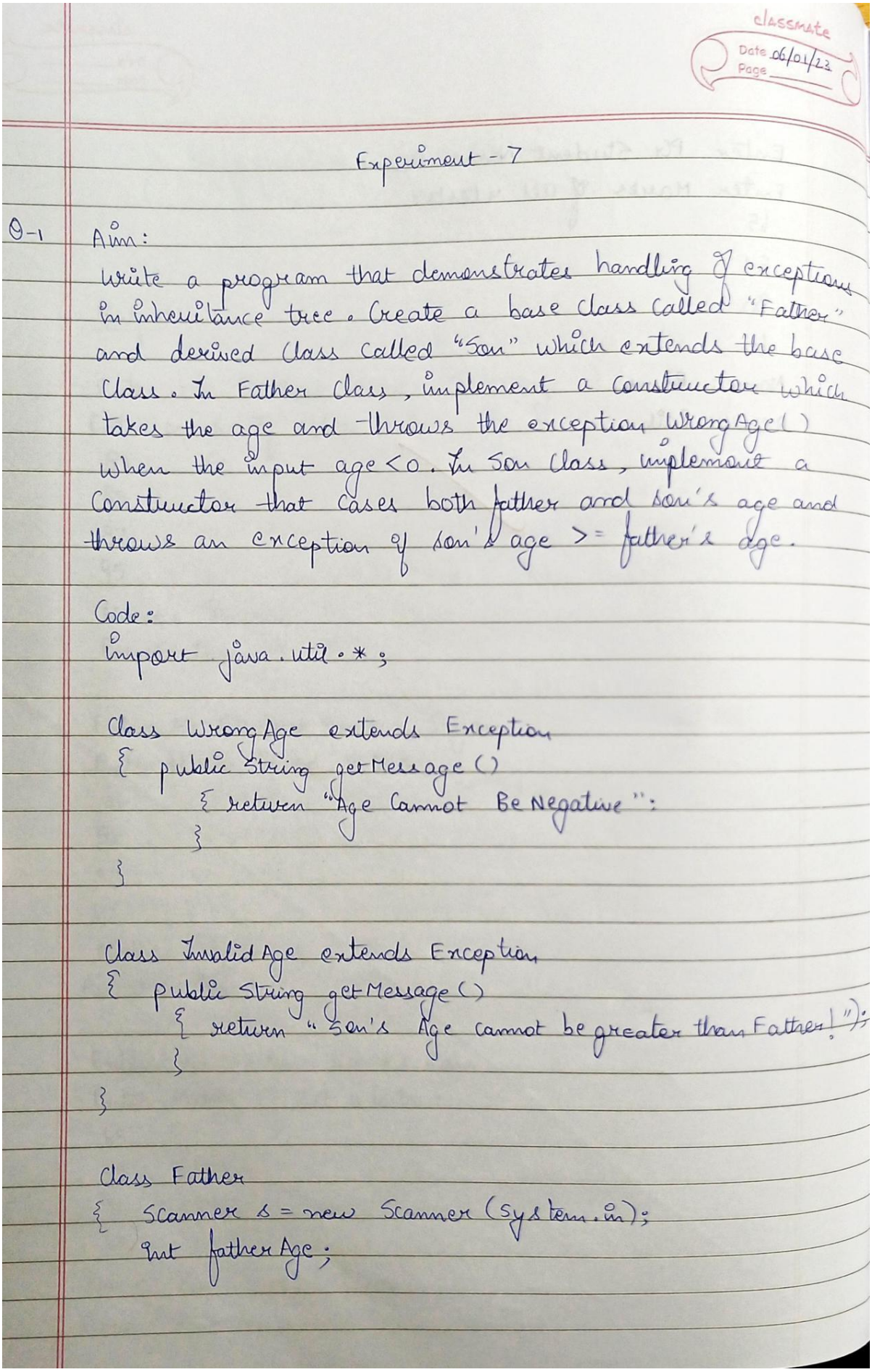
Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age<0. In Son class, implement a constructor that cases both father and son’s age and throws an exception if son’s age is >=father’s age.

2.7.2 Code:

```
import java.util.*;

class WrongAge extends Exception
{
   public String getMessage()
   {
      return "Age Cannot Be Negative";
   }
}

class InvalidAge extends Exception
{
   public String getMessage()
   {
      return "Son's Age cannot be greater than Father's!";
   }
}

class Father
{
   Scanner s = new Scanner(System.in);
   int fatherAge;
   Father() throws WrongAge
   {
      System.out.print("Enter the Father's Age: ");
      fatherAge = s.nextInt();

      try
      {
               if(fatherAge<0)
         throw new WrongAge();
      }

      catch(WrongAge e1)
      {
         System.out.println(e1.getMessage());
         System.exit(0);
```

```
        }
    }
}

class Son extends Father
{
    int sonAge;
    Son() throws WrongAge,InvalidAge
    {
        super();
        System.out.print("Enter the Son's Age: ");
        sonAge = s.nextInt();
        try
        {
          if(sonAge<0)
            throw new WrongAge();
        }

        catch(WrongAge e2)
        {
            System.out.println(e2.getMessage());
        }

        try
        {
          if(sonAge>fatherAge)
            throw new InvalidAge();
        }

        catch(InvalidAge e3)
        {
            System.out.println(e3.getMessage());
        }
    }
}

class Agecheck
{
    public static void main(String[] args) throws WrongAge,InvalidAge
    {
        new Son();
    }
}
```

2.7.3 Observation Book Pictures:



classmate
Date 06/01/22
Page

Experiment - 7

Q-1 Aim:
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 of son's age >= father's age.

Code:

```
import java.util.*;

Class WrongAge extends Exception
{ public String getMessage()
    { return "Age Cannot Be Negative";
    }
}

Class InvalidAge extends Exception
{ public String getMessage()
    { return "Son's Age cannot be greater than Father!");
    }
}

Class Father
{ Scanner s = new Scanner(System.in);
  int fatherAge;
```

```
Father () throws WrongAge
{ System.out.print ("Enter the Father's Age: ");
  fatherAge = s.nextInt();

  try
  { if (fatherAge<0)
      throw new WrongAge();
  }

  catch (WrongAge e1)
  { System.out.println(e1.getMessage());
    System.exit(0);
  }
}
}

Class Son extends Father
{ int sonAge;
  Son() throws WrongAge, InvalidAge
  { super();
    System.out.print ("Enter the Son's Age: ");
    sonAge = s.nextInt();
    try
    { if (sonAge<0)
        throw new WrongAge();
    }
    Catch (WrongAge e2)
    { System.out.println(e2.getMessage());
    }
```

→

```
        try
        { if (sonAge > fatherAge)
            throw new InvalidAge();
        }
        catch (InvalidAge e3)
        { System.out.println(e3.getMessage());
        }
    }
}

class AgeCheck
{ public static void main (String args[]) throws
    WrongAge, InvalidAge
  { new Son();
  }
}
```

Output:

```
Enter the Father's Age: 25
Enter the Son's Age: 98
Son's Age cannot be greater than Father!

Enter the Father's Age: -85
Age cannot be Negative.
```

2.7.4 Output:

```
Microsoft Windows [Version 10.0.22621.1105]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Acer\Desktop\Notes (2nd Year)\java practice programs\Lab
Programs (06-01-2023)\Father and Son>javac FatherAndSon.java

C:\Users\Acer\Desktop\Notes (2nd Year)\java practice programs\Lab
Programs (06-01-2023)\Father and Son>java Agecheck
Enter the Father's Age: 52
Enter the Son's Age: 20

C:\Users\Acer\Desktop\Notes (2nd Year)\java practice programs\Lab
Programs (06-01-2023)\Father and Son>java Agecheck
Enter the Father's Age: 12
Enter the Son's Age: 40
Son's Age cannot be greater than Father's!

C:\Users\Acer\Desktop\Notes (2nd Year)\java practice programs\Lab
Programs (06-01-2023)\Father and Son>java Agecheck
Enter the Father's Age: -10
Age Cannot Be Negative

C:\Users\Acer\Desktop\Notes (2nd Year)\java practice programs\Lab
Programs (06-01-2023)\Father and Son>java Agecheck
Enter the Father's Age: 0.15
Exception in thread "main" java.util.InputMismatchException
        at java.base/java.util.Scanner.throwFor(Scanner.java:939)
        at java.base/java.util.Scanner.next(Scanner.java:1594)
        at java.base/java.util.Scanner.nextInt(Scanner.java:2258)
        at java.base/java.util.Scanner.nextInt(Scanner.java:2212)
        at Father.<init>(FatherAndSon.java:26)
        at Son.<init>(FatherAndSon.java:47)
        at Agecheck.main(FatherAndSon.java:78)
```

2.8 Experiment - 8

2.8.1 Question:

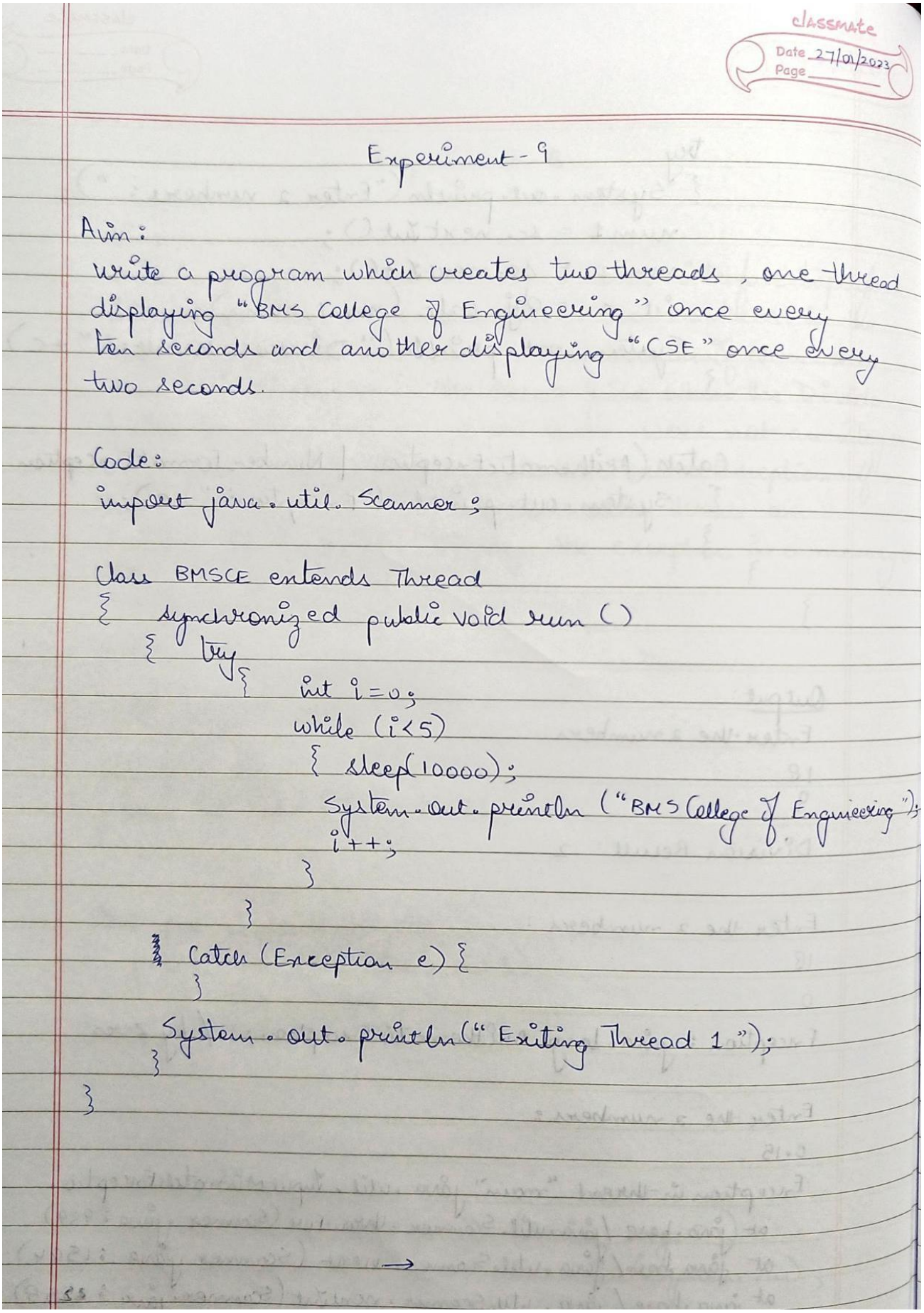
Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds.

2.8.2 Code:

```
import java.util.Scanner;

class BMSCE extends Thread {
	synchronized public void run()
	{
		try
		{
			int i=0;
			while (i<5)
			{
				sleep(10000);
				System.out.println("BMS College of Engineering ");
				i++;
			}
		}

		catch (Exception e) {
		}
	System.out.println("Exiting Thread 1");
	}
}

class CSE extends Thread
	{
	synchronized public void run()
	{
		try
		{
			int i=0;
			while (i<5)
			{
				sleep(2000);
				System.out.println("CSE");
				i++;
			}
		}

		catch (Exception e) {
		}
```

```
        System.out.println("Exiting Thread 2");

        }
}

class Multithreading
{
        public static void main(String args[])
        {
                BMSCE t1 = new BMSCE();
                CSE t2 = new CSE();
                t1.start();
                t2.start();
        }
}
```

2.8.3 Observation Book Pictures:



Experiment - 9

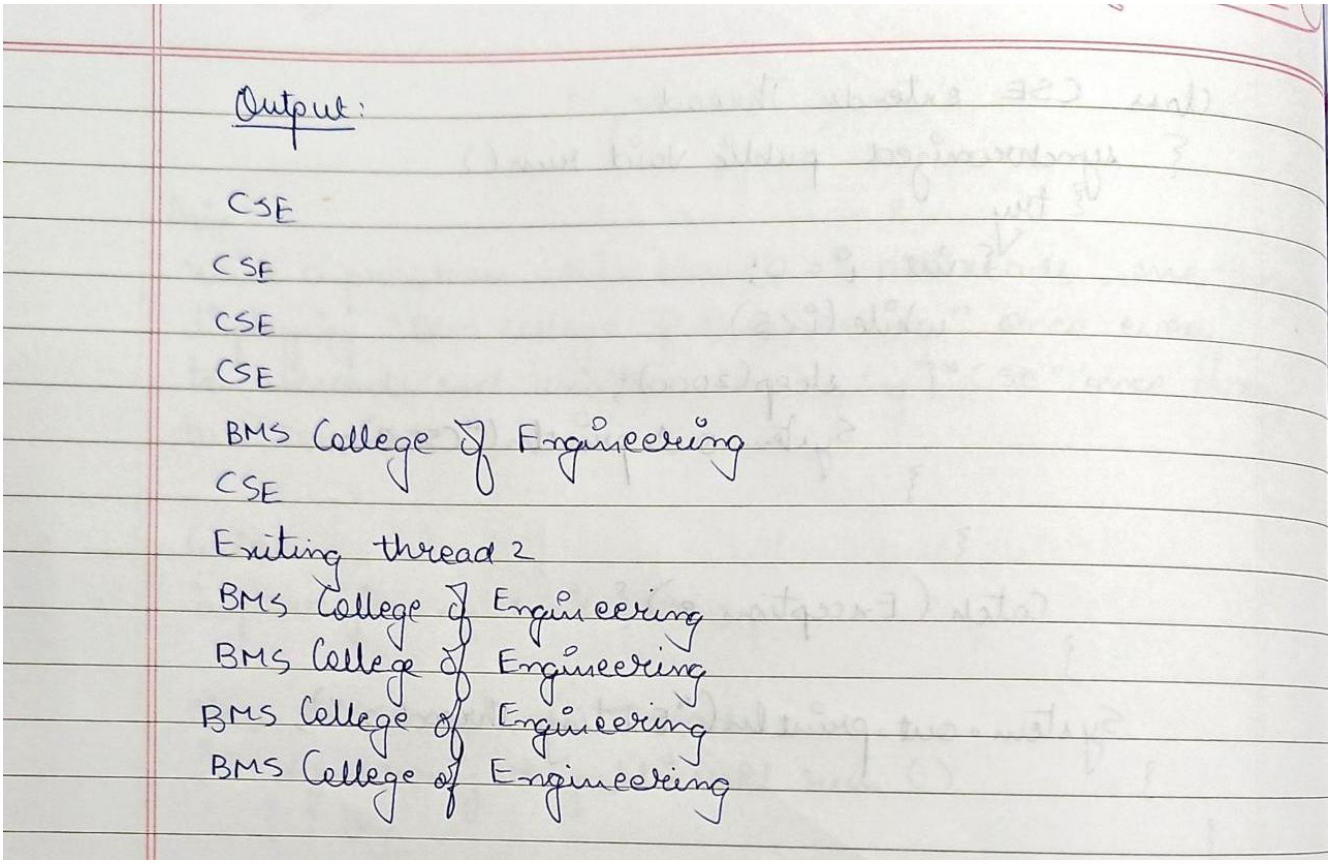
Aim:
Write a program which creates two threads, one thread displaying "BMS College of Engineering" once every ten seconds and another displaying "CSE" once every two seconds.

Code:

```
import java.util.Scanner;

class BMSCE extends Thread
{  synchronized public void run()
   {  try{  int i=0;
            while(i<5)
            {  sleep(10000);
               System.out.println("BMS College of Engineering");
               i++;
            }
         }
      catch(Exception e){
      }
      System.out.println("Exiting Thread 1");
   }
}
```

```
class CSE extends Thread
{ synchronized public void run()
  { try
    { int i = 0;
      while (i<5)
      { sleep(2000);
        System.out.println("CSE");
      }
    }
    catch (Exception e){
    }
  System.out.println("Exiting Thread 2");
  }
}

class Multithreading
{ public static void main(String args[])
  { BMSCE t1 = new BMSCE();
    CSE t2 = new CSE();
    t1.start()
    t2.start()
  }
}
```

→



2.8.4 Output:

```
Microsoft Windows [Version 10.0.22621.1105]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Acer\Desktop\Notes (2nd Year)\java practice
programs\OOJ Lab Programs\Threads>javac Threads.java

C:\Users\Acer\Desktop\Notes (2nd Year)\java practice
programs\OOJ Lab Programs\Threads>java Multithreading

CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
Exiting Thread 2
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
Exiting Thread 1
```