VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

OBJECT ORIENTED JAVA PROGRAMMING

Submitted by

UMESHA H N (1BM24CS428)

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 Sep

2024-Jan 2025

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 **UMESHA H N (1BM24CS428),** 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 2024-25. The Lab report has been approved as it satisfies the academic requirements in respect of **Object-Oriented Java Programming Lab - (23CS3PCOOJ)** work prescribed for the said degree.

Dr. Nandhini Vineeth

Associate Professor,
Department of CSE,
BMSCE, Bengaluru

Dr. Kavitha Sooda

Professor and Head,
Department of CSE
BMSCE, Bengaluru

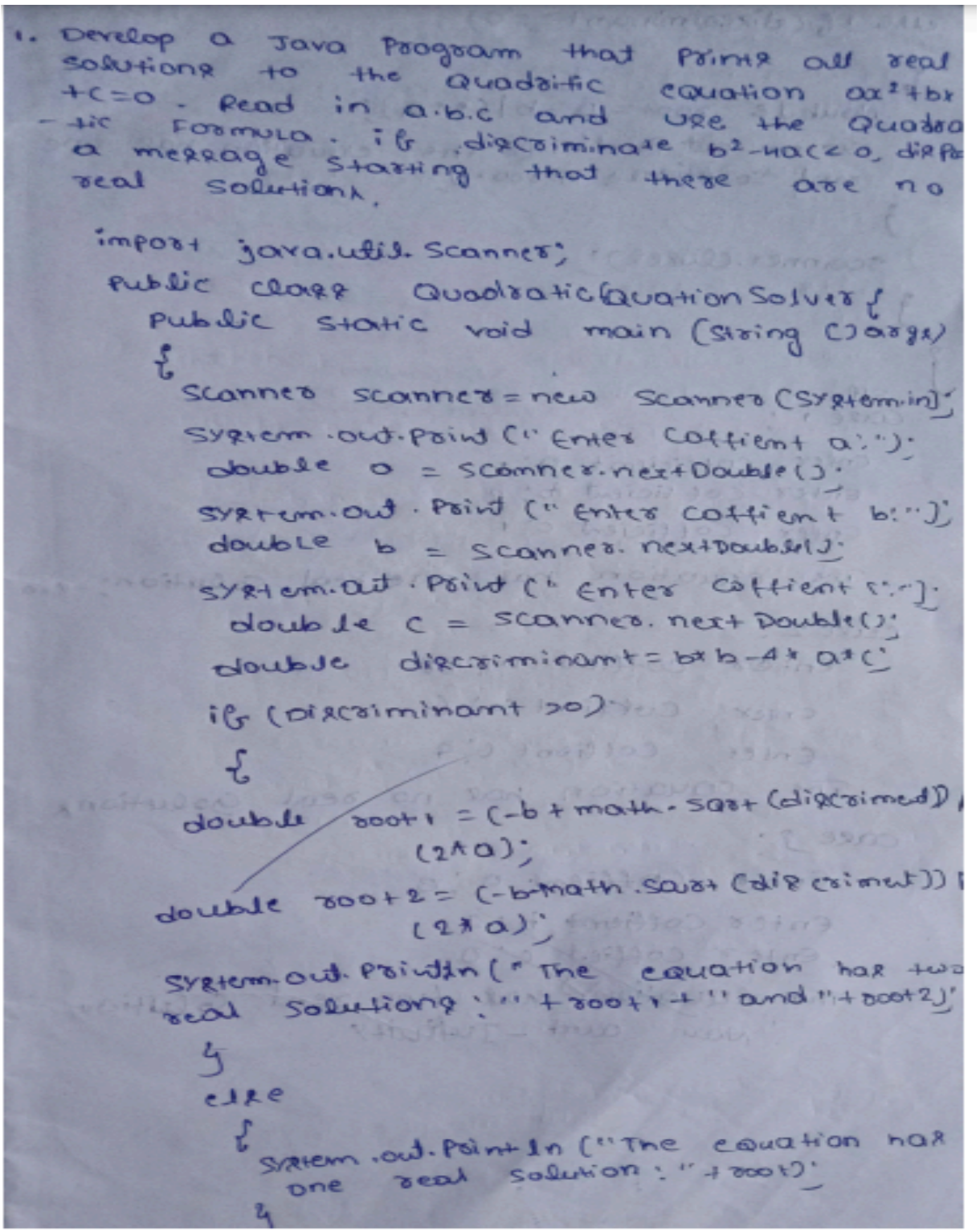
INDEX

Sl. No.	Date	Experiment Title	Page No.
1	26-09-2024	Quadratic equation	1-3
2	19-10-2024	Student Details and SGPA Calculation	4-9
3	3-10-2024	Book Details Management	10 - 16
4	24-10-2024	Abstract Shape Area Calculation	17 - 21
5	07-11-2024	Bank Account Management	22 - 33
6	14-11-2024	Student Marks from CIE and SEE	34 - 42
7	21-11-24	Exception Handling in Inheritance	43 - 48
8	28-11-24	Threading in Java	49 - 52
9	05-12-2024	Integer Division GUI	53 - 59
10	19-12-2024	Inter-Process Communication and Deadlock	60 - 65

PROGRAMS

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 discriminant b2 -4ac is negative, display a message stating that there are no real solutions.

Screen shots



```
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 Quadra
-tic Formula. if discriminate b²-4ac<0, disp
a message stating that there are no
real solutions.

import java.util.Scanner;
Public class QuadraticEquationSolver {
  Public static void main (String [] args)
  {
    Scanner scanner = new Scanner (System.in);
    System.out.Print (" Enter Coffient a: ");
     double a = scanner.nextDouble();
    System.out.Print (" Enter Coffient b: ");
     double b = scanner.nextDouble();
    System.out.Print (" Enter Coffient c: ");
     double c = scanner.nextDouble();
     double discriminant = b*b-4*a*c;
     if (discriminant >0)
     {
     double root1 = (-b + math.sqrt (discrimed))
                    (2*a);
     double root2 = (-b-math.sqrt (discrimet))
                    (2*a);
     System.out.Println (" The equation has two
     real Solutions: " + root1 + " and " + root2);
     }
     else
     {
       System.out.Println (" The equation has
         one real solution: " + root);
     }
```

```
else if(discriminant==0)
{
    double root= a-b/(2*a);
    System.out.println("The equation has one
    real solution:"+root);
}
scanner.close();
}
}
```

or

```
case 1:
    Enter coefficient a: 1
    Enter coefficient b: 2
    Enter coefficient c: 1
    The equation has one real solution: -1.0
case 2:
    Enter coefficient a: 88
    Enter coefficient b: 6
    Enter coefficient c: 9
    The equation has no real solutions
case 3:
    Enter coefficient a: 0
    Enter coefficient b: 7
    Enter coefficient c: 0
    The equation has two real solutions:
    Nan and -Infinity
```

Code

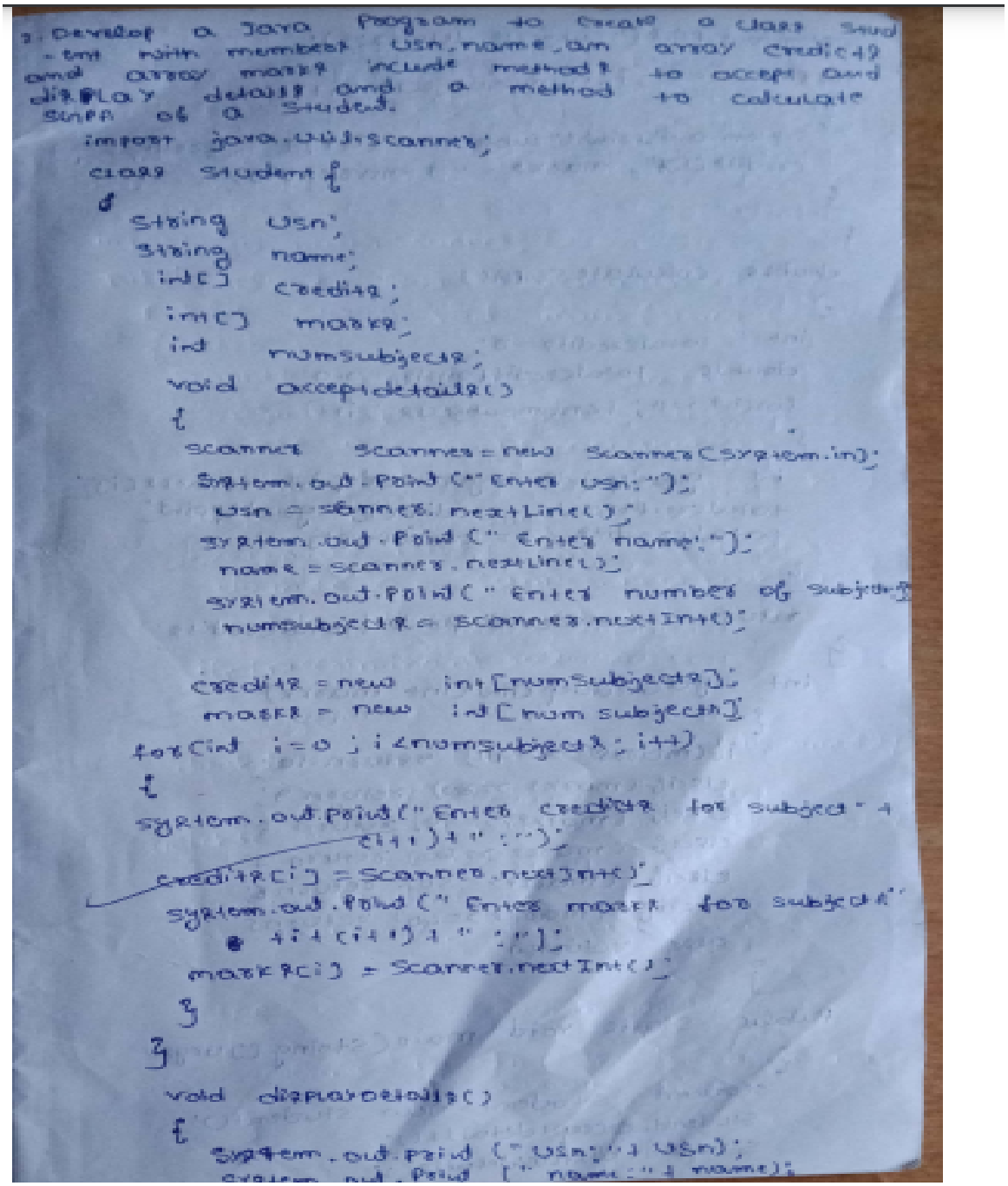
```
import java.util.Scanner;
class QuadraticEquation {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter value of a: ");
        double a = scanner.nextDouble();

        System.out.print("Enter value of b: ");
        double b = scanner.nextDouble();

        System.out.print("Enter value of c: ");
        double c = scanner.nextDouble();

        double discriminant = b * b - 4 * a * c;

        if (discriminant < 0) {
            System.out.println("No real solutions");
        } else {
            double root1 = (-b + Math.sqrt(discriminant)) / (2 * a);
            double root2 = (-b - Math.sqrt(discriminant)) / (2 * a);
            System.out.println("Root 1: " + root1);
            System.out.println("Root 2: " + root2);
        }
    }
}
```

OUTPUT

```
PS C:\Users\User\Desktop\RECORE_JAVA> javac java1.java
PS C:\Users\User\Desktop\RECORE_JAVA> java QuadraticEquation
Enter value of a: 1
Enter value of b: -3
Enter value of c: 2
Root 1: 2.0
Root 2: 1.0
PS C:\Users\User\Desktop\RECORE_JAVA> java QuadraticEquation
Enter value of a: 1
Enter value of b: 22
Enter value of c: 5
Root 1: -0.22967038573099252
Root 2: -21.77032961426901
```

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.

Screen shots



```
System.out.Println("Subjects details:");
for(int i=0 ; i<numSubjects ; i++)
  {
    System.out.println("Subject " + (i+1) + " : Credits=" +
    Credits[i]", marks = " + marks[i]);
  }
}
  double calculateSGPA()
  {
    int totalCredits = 0;
    double totalCreditPoints = 0.0;
    for(int i=0; i<numSubjects ; i++)
    {
      int gradepoint = getGradepoint(marks[i]);
      totalCreditPoints = credits[i] * gradepoint;
      totalcredits += credits[i];
    }
    return totalcreditPoints/totalCredits;
  }
  int getGradepoint(int marks)
    {
      if(marks >= 90) return 10;
      else if(marks >= 80) return 9;
      else if(marks >= 70) return 8;
      else if(marks >= 60) return 7;
      else if(marks >= 50) return 6;
      else if(marks >= 40) return 5;
      else return 0;
    }
  Public static void main(String [] args)
    {
      Student Student = new Student();
      Student.acceptdetails();
      Student.displaydetails();
```

```
double sgpa = Student.calculateSGPA();
System.out.println("SGPA:" + sgpa);
}
}
```

o/p

```
Enter number of subjects : 2
Student name: umesh
USN : 1VBECS424
Enter max credits
  sub1 : 4
  sub 2 : 4
Enter marks for each subject:
  marks sub1 : 75
  marks sub2 : 67
  marks for sub3 : 79
sub
Student details:
  name : uday
  USN : 1VBECS402
  SGPA : 7.66
```

Code

```
import java.util.Scanner;

class Student {
    String usn;
    String name;
    int[] credits;
    int[] marks;

    Student(int numSubjects) {
        credits = new int[numSubjects];
        marks = new int[numSubjects];
    }

    void acceptDetails() {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter USN: ");
        usn = scanner.nextLine();

        System.out.print("Enter Name: ");
        name = scanner.nextLine();

        for (int i = 0; i < credits.length; i++) {
            System.out.print("Enter credits for subject " + (i + 1) + ": ");
            credits[i] = scanner.nextInt();

            System.out.print("Enter marks for subject " + (i + 1) + ": ");
            marks[i] = scanner.nextInt();
```

```
        }
    }

    void displayDetails() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);

        for (int i = 0; i < credits.length; i++) {
            System.out.println("Subject " + (i + 1) + ": Marks = " + marks[i] + ", Credits = " + credits[i]);
        }
    }

    double calculateSGPA() {
        double totalCredits = 0;
        double weightedMarks = 0;

        for (int i = 0; i < credits.length; i++) {
            totalCredits += credits[i];
            weightedMarks += (marks[i] * credits[i]);
        }

        return weightedMarks / totalCredits;
    }
}

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

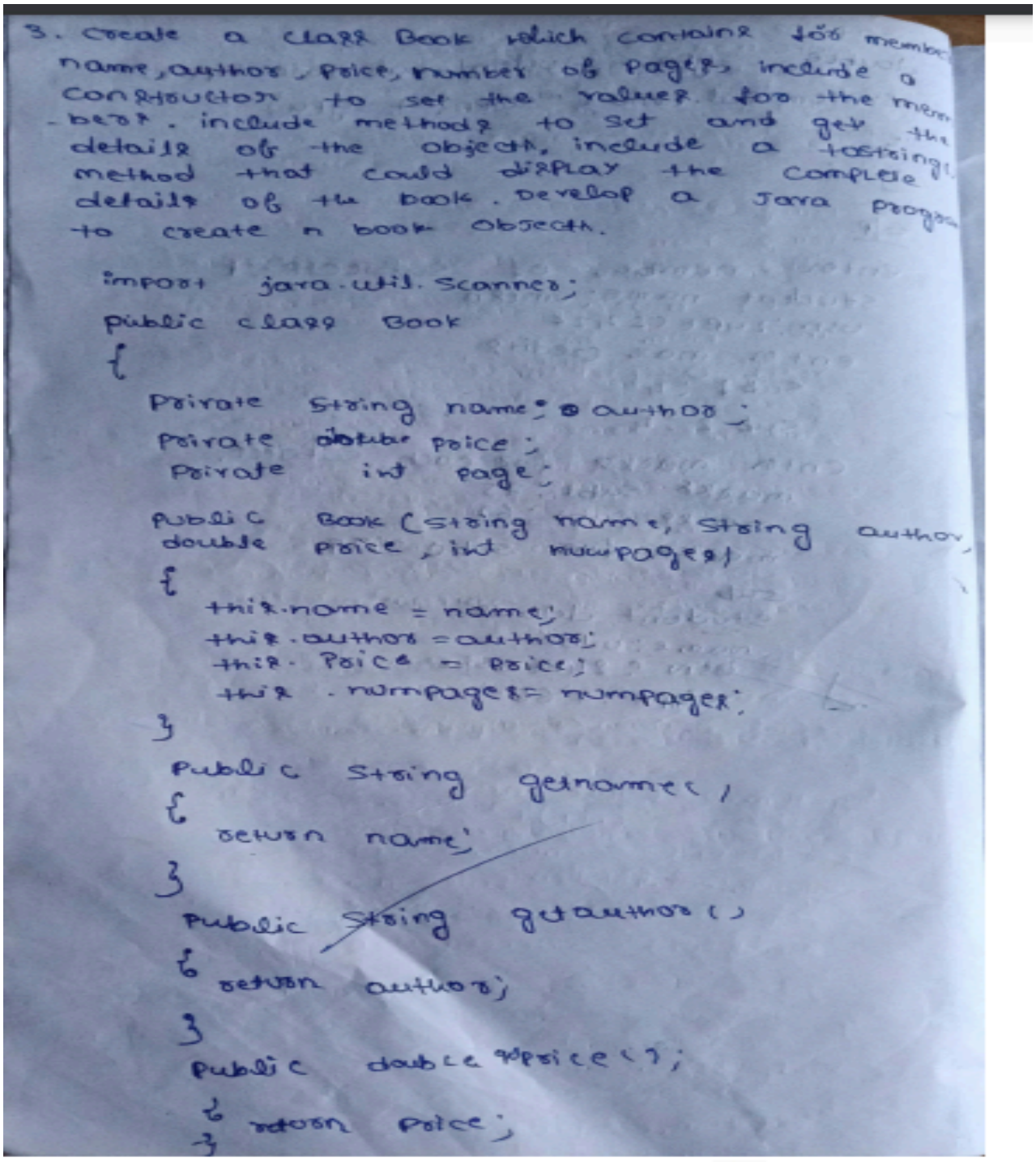
```
        System.out.print("Enter number of subjects: ");
        int numSubjects = scanner.nextInt();

        Student student = new Student(numSubjects);

        student.acceptDetails();
        student.displayDetails();

        double sgpa = student.calculateSGPA();
        System.out.println("SGPA: " + sgpa);
    }
}
```

Output

```
PS C:\Users\User\Desktop\RECORE_JAVA> javac java2.java
PS C:\Users\User\Desktop\RECORE_JAVA> java Main
Enter number of subjects: 3
Enter USN: 24BE24cs428
Enter Name: Umesha H N
Enter credits for subject 1: 3
Enter marks for subject 1: 98
Enter credits for subject 2: 2
Enter marks for subject 2: 91
Enter credits for subject 3: 4
Enter marks for subject 3: 89
USN: 24BE24cs428
Name: Umesha H N
Subject 1: Marks = 98, Credits = 3
Subject 2: Marks = 91, Credits = 2
Subject 3: Marks = 89, Credits = 4
SGPA: 92.44444444444444
```

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

Screen shots



```
3. Create a class Book which contains four members
name, author, price, number of pages. include a
constructor to set the values for the mem
-bers. 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 Progra
to create n book objects.

import java.util.Scanner;
public class Book
{
   Private String name, author;
   Peivate double price;
    Private int page;

   Public Book (String name, String author,
   double price, int numpages)
   {
      this.name = name;
      this.author = author;
      this. Price = price;
      this .numpages= numpages;
   }
    Public String getname()
    {
      return name;
    }
    Public String getauthor()
    {
      return author;
    }
    Public double getprice();
     {
      return price;
    }
```

```
public int getPage()
{
    return page;
}

public void setName(String name)
{
    this.name = name;
}
public void setAuthor(String author)
{
    this.author = author;
}
public void setPrice(double price)
{
    this.price = price;
}
public void setNumPages(int numPages)
{
    this.numPages = numPages;
}
@Override
public String toString()
{
    return "Book{" +
            "name='" + name + '\'' +
            ", author='" + author + '\'' +
            ", price=" + price +
            ", number of pages=" + numPages + '}';
}
}
```

```
public class BookProgram {
    public static void main (String [] args)
    { Scanner scanner= new Scanner(System.in);
      System.out.print("Enter the number of books:")
      int n = scanner.nextInt();
      scanner.nextLine();
      Book[] books = new Book[n];
    for(int i=0 ; i<n; i++)
    {
      System.out.println("Enter details of b
      System.out.print("name:");
      String name = Scanner.nextLine();
      System.out.print("Author:");
      String author = Scanner.nextLine();
      System.out.print("Price:");
      double price = Scanner.nextDouble();
      System.out.print("number of pages:");
      int numpages = scanner.nextInt();
       Scanner.nextLine();
    books[i] =new Book(name,author,price,num
                -pages);
      System.out.println("\n Book Details:
       for(Book book: books)
        {
          System.out.println(" Book details
        }
      Scanner.close();
    }
}
```

o/p

```
number of books : 1
Enter details of Book 1
Enter book name : java
author name : ZXC
enter price = 560
number of pages : 200
```

Code

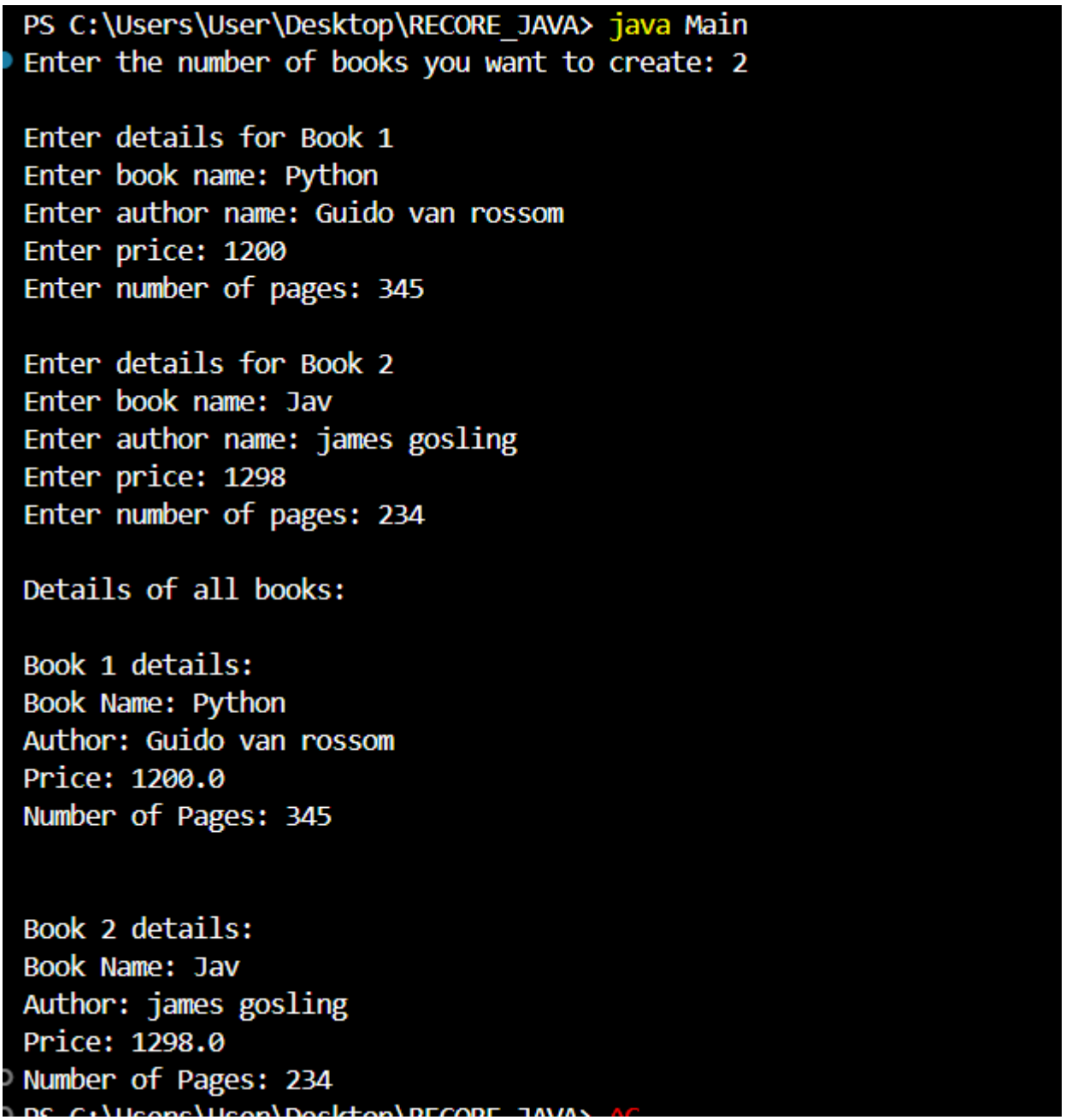
```
import java.util.Scanner;

class Book {
    String name;
    String author;
    double price;
    int num_pages;

    public Book(String name, String author, double price, int num_pages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.num_pages = num_pages;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setAuthor(String author) {
        this.author = author;
    }

    public String getAuthor() {
```

```
        return author;
    }

    public void setPrice(double price) {
        this.price = price;
    }

    public double getPrice() {
        return price;
    }

    public void setNumPages(int num_pages) {
        this.num_pages = num_pages;
    }

    public int getNumPages() {
        return num_pages;
    }

    @Override
    public String toString() {
        return "Book Name: " + name + "\n" +
                "Author: " + author + "\n" +
                "Price: " + price + "\n" +
                "Number of Pages: " + num_pages;
    }
}

class Main {
```

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

    System.out.print("Enter the number of books you want to create: ");
    int n = scanner.nextInt();
    scanner.nextLine();

    Book[] books = new Book[n];

    for (int i = 0; i < n; i++) {
        System.out.println("\nEnter details for Book " + (i + 1));

        System.out.print("Enter book name: ");
        String name = scanner.nextLine();

        System.out.print("Enter author name: ");
        String author = scanner.nextLine();

        System.out.print("Enter price: ");
        double price = scanner.nextDouble();

        System.out.print("Enter number of pages: ");
        int num_pages = scanner.nextInt();
        scanner.nextLine();

        books[i] = new Book(name, author, price, num_pages);
    }

    System.out.println("\nDetails of all books:");
```

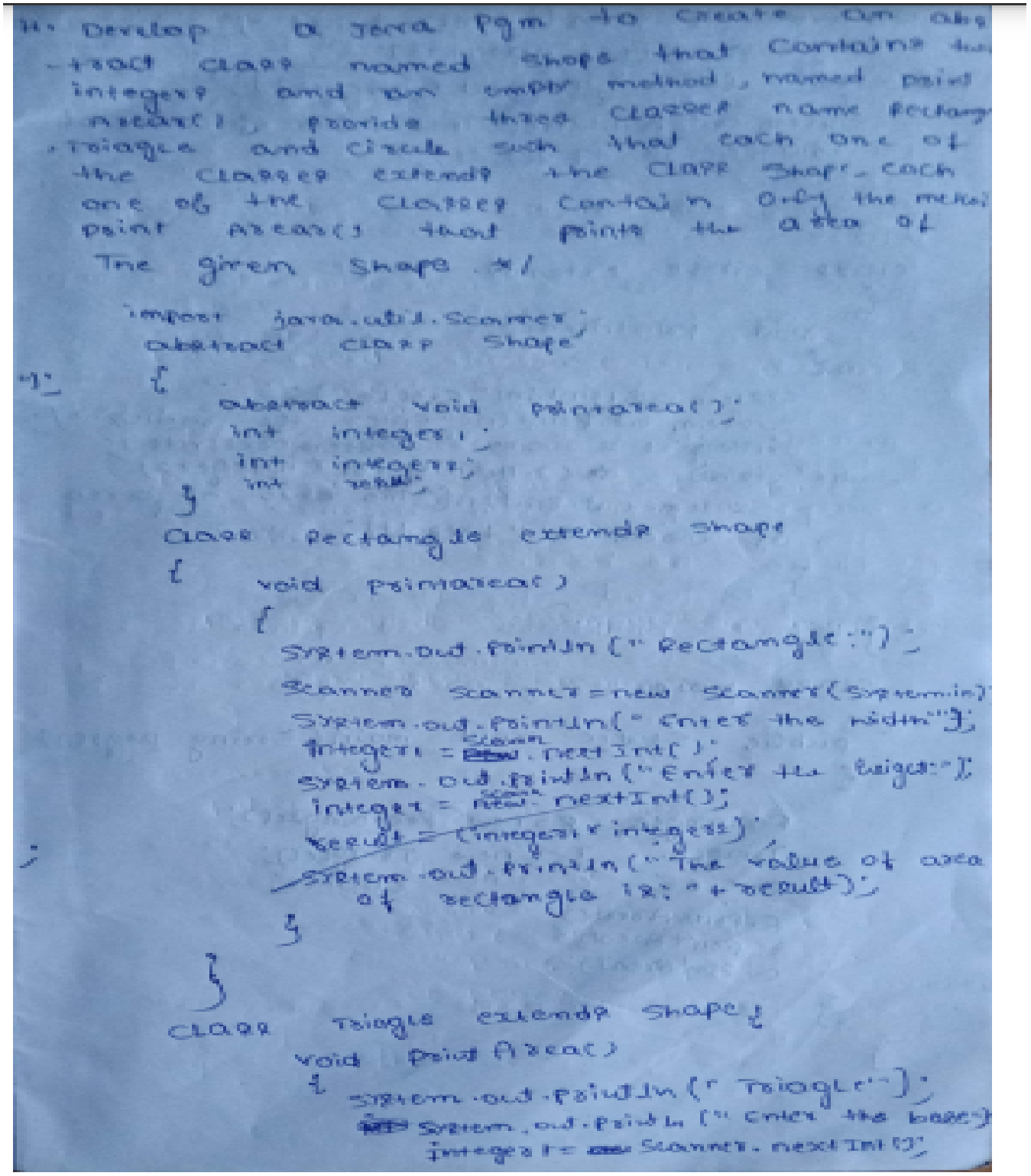
```
        for (int i = 0; i < n; i++) {
            System.out.println("\nBook " + (i + 1) + " details:");
            System.out.println(books[i].toString());
        }
    }
}
```

Output



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.

Screen shots



```
        System.out.println("Enter height:");
        integer2 = scanner.nextInt();
double result = (0.5 * integer1 * integer2);
        System.out.println("The value of area of
        Triagle is:" + result);
    }
}

class Circle extends Shape
{
    void printArea()
    {
        System.out.println("Circle");
        System.out.println("Enter the radius:");
        integer1 = scanner.nextInt();
double result = (3.14*integer1 * integer2)
        System.out.println("The value of area
        of Circle is:" +result);
    }
}

class main
{
    public static void main(String args[])
    {
        Rectangle R = new Rectangle();
        Triagle T = new Triagle();
        Circle C = new Circle();
        R.printArea();
        T.printArea();
        C.printArea();
    }
}
```

```
abstract class Shape {
    int dimension1;
    int dimension2;

    abstract void printArea();
}

class Rectangle extends Shape {
    public Rectangle(int length, int width) {
        dimension1 = length;
        dimension2 = width;
    }

    void printArea() {
        int area = dimension1 * dimension2;
        System.out.println("Area of Rectangle: " + area);
    }
}

class Triangle extends Shape {
    public Triangle(int base, int height) {
        dimension1 = base;
        dimension2 = height;
    }

    void printArea() {
        double area = 0.5 * dimension1 * dimension2;
        System.out.println("Area of Triangle: " + area);
    }
```

```
}

class Circle extends Shape {
    public Circle(int radius) {
        dimension1 = radius;
        dimension2 = 0;
    }

    void printArea() {
        double area = Math.PI * dimension1 * dimension1;
        System.out.println("Area of Circle: " + area);
    }
}

class Main {
    public static void main(String[] args) {
        Shape rectangle = new Rectangle(5, 3);
        Shape triangle = new Triangle(4, 6);
        Shape circle = new Circle(7);

        rectangle.printArea();
        triangle.printArea();
        circle.printArea();
    }
}
```

Output

```
PS C:\Users\User\Desktop\RECORE_JAVA> javac java4.java
PS C:\Users\User\Desktop\RECORE_JAVA> java Main
Area of Rectangle: 15
Area of Triangle: 12.0
Area of Circle: 153.93804002589985
```

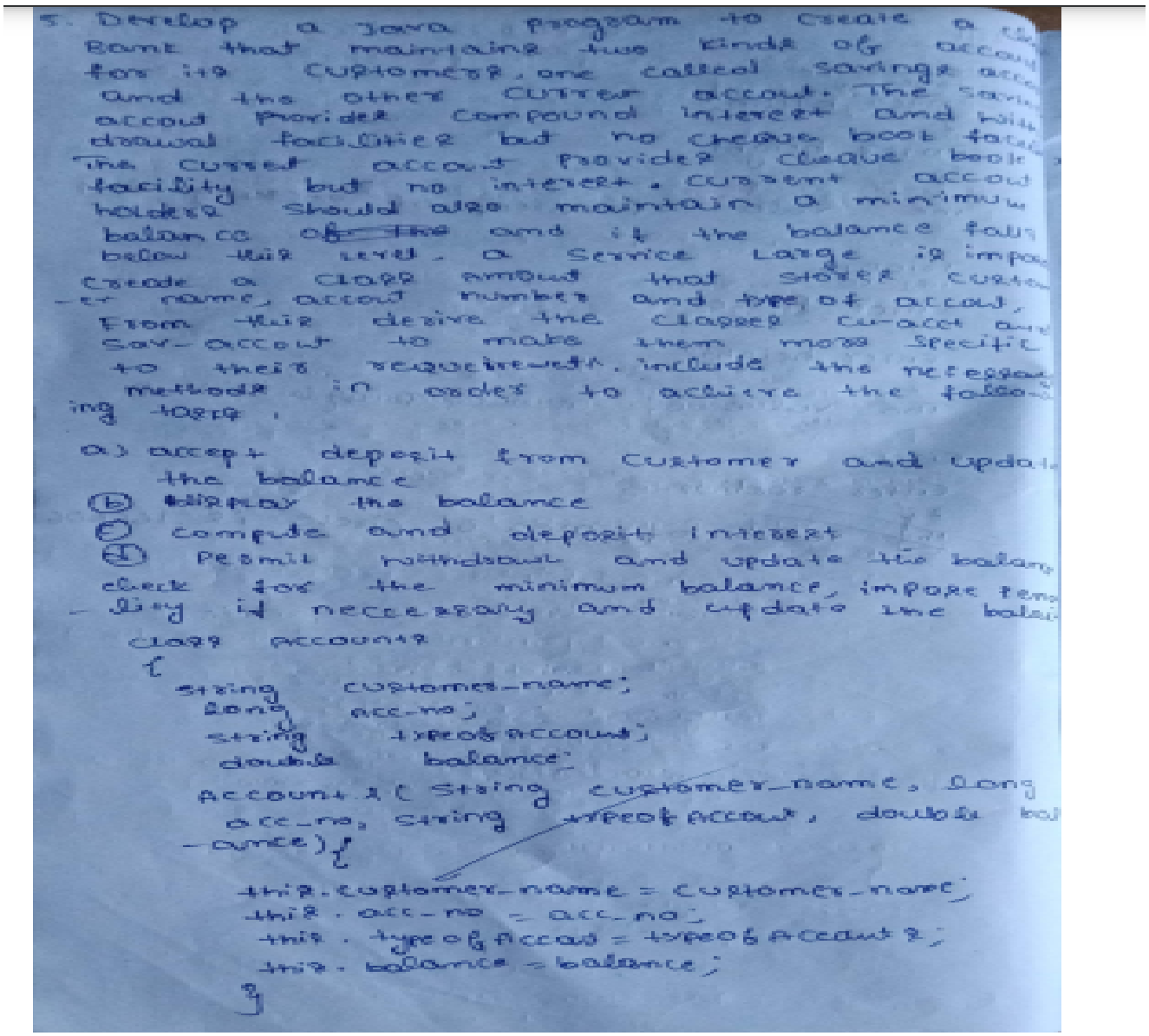
5. Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book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



```
void displaybalance()
{
System.out.println("The balance of the
accno" + account_no + "and name" + customer
name +" is :" + balance()';
}
void deposit(int amount)
{
balance += amount;
System.out.println("The amount of" + amo
-unt + " has been debited");
}
void withdraw(int amount){
if(balance > amount)
{
System.out.println("The Insufficient
Balance");
}
else{
this.balance -= amount;
System.out.println("Amount of" +
amount + " has been succesfully withdraw");
}
}

class Saving_Account extends Account {
double compounded_Interest = 0.04;
savingAccount(String customerName, long
acc_no)
{
super(customername, accno, "saving"
,0);
}
```

 import

```
void compound_interest()
{
  double interest_amount = balance * Comp
  Interest;
   balance += interest amount;
   system.out.println(" Interest deposited:");
  }
}
 class CurrentAccount extends Account
  {
    boolean checkbookIs = True;
    double minimum_balance = 5000;
    double Service_charges = 50;

  currentAccount (String customerName, long
   accno) {
    super(customer name, accno, "current", 5000
    }
  void withdraw(int amount)
   {
     if(balance > amount)
     {
       this.balance -= amount;
       system.out.println ("The amount of
     amount = " withdraw Successfully");
     }
     if (balance <= minimum_balance)
     { 'impose Penality();
     }
  else {
       System.out.Println(" Insuffient Balan
```

```
void imposePenalty(){
    balance -= serviceCharge;
    System.out.println("Penalty added");
    }
}
}

public class Bank{
  public static void main(String args[])
 {
  SavingsAccount sa = new
   savingAccount("Umesh", 12345678);
    sa.displaybalance();
    sa.withdraw();
    sa.deposit(1000);
    sa.deposit(1000);
    sa.compoundInterest();
    sa.withdraw(1000);
    sa.displaybalance();
    System.out.println();

  CurrentAccount CA = new CurrentAccount("
    Nitesh", 98765432);
    CA.displaybalance();
    CA.withdraw(500);
    CA.deposit(1000);
    CA.deposit(1000);
    CA.withdraw(1000);
    CA.displayBalance();

  }
}
```

o/p

The Balance of the 12345678 A/c umesh is 0.0

Insufficient balance

Amount of 1000.0 has been debited
Amount of 1000.0 has been debited

Interest deposited

Amount of 1000.0 Successfully drawn
The balance of the 12345678 and umesh is 1080.6

The balance of the 98765432 and nitish is 5000.00

Amount of 500 withdrawn successfully
Penalty added.
Amount of 1000.00 has been debited
Amount of 1000.00 has been debited
Amount of 1000 withdrawn successfully
The balance of the 98765321 A/c name nitish is 3450.0

Code

```
java.util.Scanner;


class Account {
    String customerName;
    int accountNumber;
    double balance;
    String accountType;


    public Account(String customerName, int accountNumber, String accountType, double
initialBalance) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = initialBalance;
    }


    public void deposit(double amount) {
        balance += amount;
    }


    public void displayBalance() {
        System.out.println("Current Balance: " + balance);
    }


    public void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
            System.out.println("Withdrawal successful. New balance: " + balance);
```

```
        } else {
            System.out.println("Insufficient balance.");
        }
    }

    public String getAccountType() {
        return accountType;
    }
}

class SavAcct extends Account {
    double interestRate;

    public SavAcct(String customerName, int accountNumber, double initialBalance, double
interestRate) {
        super(customerName, accountNumber, "Savings", initialBalance);
        this.interestRate = interestRate;
    }

    public void computeAndDepositInterest() {
        double interest = balance * interestRate / 100;
        balance += interest;
        System.out.println("Interest of " + interest + " has been added to the balance.");
    }
}

class CurAcct extends Account {
    double minimumBalance;
    double serviceCharge;
```

```
    public CurAcct(String customerName, int accountNumber, double initialBalance, double
minimumBalance, double serviceCharge) {
        super(customerName, accountNumber, "Current", initialBalance);
        this.minimumBalance = minimumBalance;
        this.serviceCharge = serviceCharge;
    }

    @Override
    public void withdraw(double amount) {
        if (balance - amount >= minimumBalance) {
            balance -= amount;
            System.out.println("Withdrawal successful. New balance: " + balance);
        } else {
            System.out.println("Balance falls below minimum required balance. Service charge applied.");
            balance -= serviceCharge;
            System.out.println("Service charge of " + serviceCharge + " has been deducted.");
        }
    }
}

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

        Account account = null;

        System.out.println("Enter customer name: ");
        String name = scanner.nextLine();
```

```
        System.out.println("Enter account number: ");
        int accountNumber = scanner.nextInt();

        System.out.println("Enter account type (Savings/Current): ");
        String accountType = scanner.next();

        if (accountType.equalsIgnoreCase("Savings")) {
            System.out.println("Enter initial deposit: ");
            double initialBalance = scanner.nextDouble();
            System.out.println("Enter interest rate: ");
            double interestRate = scanner.nextDouble();
            account = new SavAcct(name, accountNumber, initialBalance, interestRate);
        } else if (accountType.equalsIgnoreCase("Current")) {
            System.out.println("Enter initial deposit: ");
            double initialBalance = scanner.nextDouble();
            System.out.println("Enter minimum balance: ");
            double minimumBalance = scanner.nextDouble();
            System.out.println("Enter service charge: ");
            double serviceCharge = scanner.nextDouble();
            account = new CurAcct(name, accountNumber, initialBalance, minimumBalance,
serviceCharge);
        } else {
            System.out.println("Invalid account type.");
            return;
        }

        while (true) {
            System.out.println("\nChoose an action:");
```

```
System.out.println("1. Deposit");
System.out.println("2. Withdraw");
System.out.println("3. Display Balance");
System.out.println("4. Compute and Deposit Interest (Savings only)");
System.out.println("5. Exit");
int choice = scanner.nextInt();

switch (choice) {
    case 1:
        System.out.println("Enter deposit amount: ");
        double depositAmount = scanner.nextDouble();
        account.deposit(depositAmount);
        break;

    case 2:
        System.out.println("Enter withdrawal amount: ");
        double withdrawalAmount = scanner.nextDouble();
        account.withdraw(withdrawalAmount);
        break;

    case 3:
        account.displayBalance();
        break;

    case 4:
        if (account instanceof SavAcct) {
            ((SavAcct) account).computeAndDepositInterest();
        } else {
            System.out.println("Interest can only be computed for Savings accounts.");
```

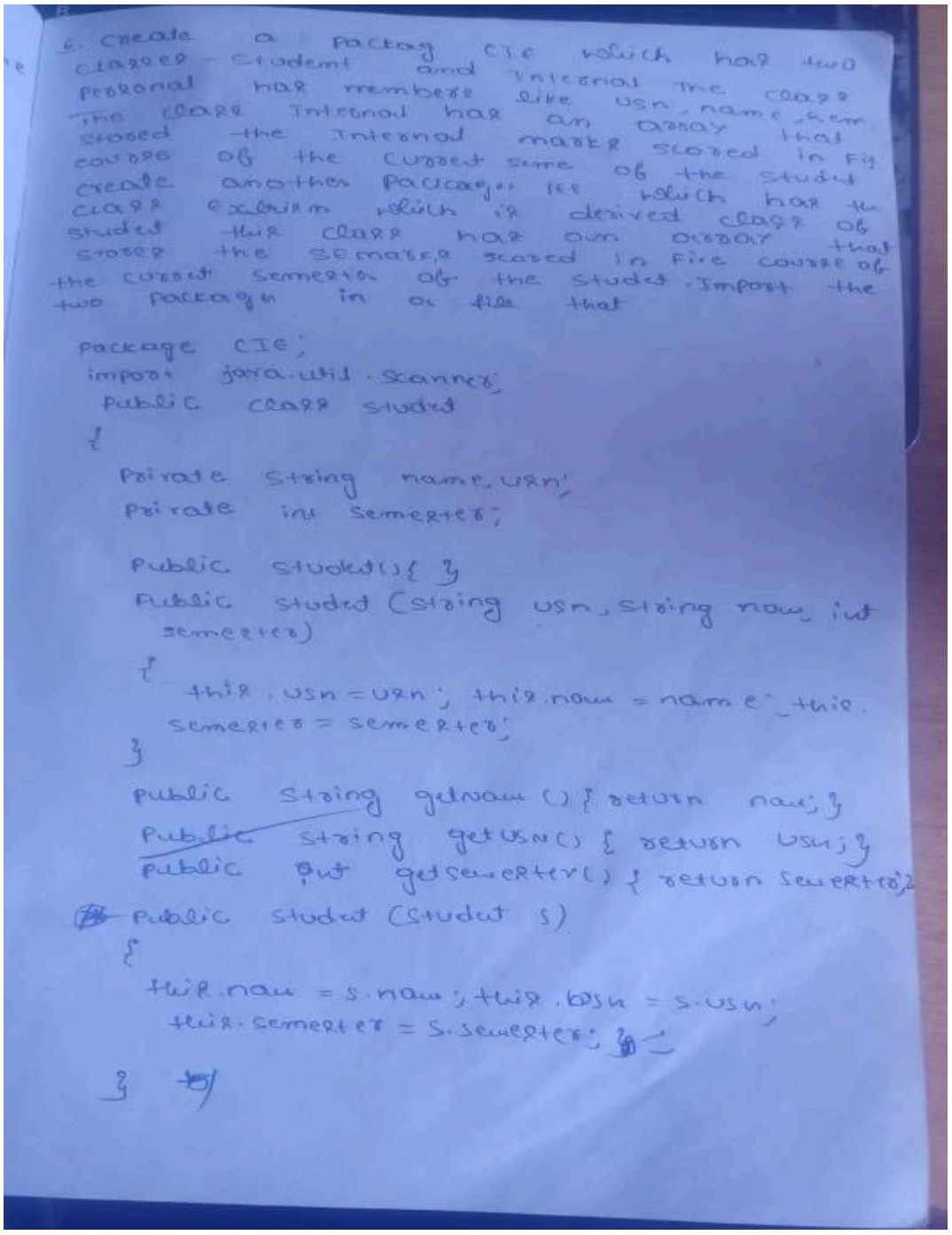
```
                }
                break;

            case 5:
                System.out.println("Exiting program.");
                return;

            default:
                System.out.println("Invalid choice. Please try again.");
        }
    }
  }
}
```

Output

```
PS C:\Users\User\Desktop\RECORE_JAVA> javac java5.java
PS C:\Users\User\Desktop\RECORE_JAVA> java Bank
Enter customer name:
Umesha H N
Enter account number:
65432456
Enter account type (Savings/Current):
Savings
Enter initial deposit:
10000
Enter interest rate:
2

Choose an action:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest (Savings only)
5. Exit
3
Current Balance: 10000.0

Choose an action:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest (Savings only)
5. Exit
1
Enter deposit amount:
14332

Choose an action:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest (Savings only)
5. Exit
3
Current Balance: 24332.0
```

6. Create a package CIE which has two classes - Personal and Internals. The class Personal has members like usn, name, sem. The class Internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Personal. 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.

Screen shots



6. Create a Package CIE which has two classes - Student and Internal. The class Personal has members like usn, name, sem. The class Internal has an array that stored the Internal marks scored in five courses of the current sem. of the student. Create another Package SEE which has the class External which is 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

```
package CIE;
import java.util.Scanner;
Public class student
{
  Private String name, USN;
  Private int Semester;

  Public student(){ }
  Public student (String usn, String naw, int
    semester)
  {
    this.usn = USN; this.naw = name; this.
    Semester = Semester;
  }
  Public String getnaw () { return naw; }
  Public String getUSN() { return usn; }
  Public int getSemester() { return Semester; }
  Public student (student s)
  {
    this.naw = s.naw; this.USN = s.usn;
    this.Semester = s.Semester;
  }
```

```
package CIE;
import java.util.Scanner;
public class Internals extends Student
{
  private int[] marks;
  private Internals(String usn, String name,
    int semester, int marks[])
  {
    super(usn, name, semester);
    this.marks = marks;
  }
  public static Internals getNewInstanceOf()
  {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter Student details
    & array of CIE marks");
    String usn = sc.next().strip().toUpperCase();
    String name = sc.next();
    int semester = sc.nextInt();
    int[] marks = new int[5];
    for(int i=0; i<marks.length; i++){
      marks[i] = sc.nextInt();
    }
    sc.nextLine();
    return new Internals(usn, name, semester,
      marks);
  }
  Public Static Internals getNewInstanceOf
  (Student s)
  {
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter CIE marks details");
    int marks[] = new int[5];
    for(int i=0; i<marks.length; i++){
      marks[i]= sc.nextInt();
    }
    sc.nextLine();
    return new Internals(s, marks);
  }
  public int getMarks(int i) { return
    marks[i];}
```

```
private Internals(Student s, int[] marks)
{
    super(s);
    this.marks= marks;
}
Public Internals (int[] marks)
{
    this.marks = marks;
}
```

main class:

// External

```
Package see;
import java.util.Scanner;
import CIE.*;
Public class Externals extends Student {
    private int[] marks;
    private Externals (String usn, String name, int
semester, int[] marks) {
        super(s);
        this.marks = marks;
    }
Public Static Externals getNewInstanceOf() {
    Scanner sc = new Scanner (System.in);
    System.out.println("Enter Student details and
        array of SEE marks");
    String usn = sc.next().strip().toUpperCase();
    String name= sc.next();
    int semester = sc.next();
    int[] marks = new int[5];
    for(int i=0; i< marks.length; i++)
    { marks[i]=sc.nextInt();
    }
    sc.nextLine();
    return new External (usn, name, semester, ma
                -rks);
}
Public Static Externals getNewInstanceOfStud
                ){
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter SEE marks details");
    int[] marks = new int[5];
    for(int i=0; i<marks.length; i++) {
        marks[i]= sc.nextInt();
    }
    sc.nextLine();
    return new Externals(s,marks)
}
Public int getMarks(int i) { return marks[i];
```

```
main class
import CIE.*;
import SEE.*;
import java.util.Scanner;
class main{
 public static void main(String args[])
  Scanner sc = new Scanner(System.in);
  S.O.P ("Enter the number of students:");
  int n = sc.nextInt();
  Internals[] ai = new Internals[n];
  Externals[] ae = new Externals[n];
   int[] result = new int[5];
  for(int i=0 ; i<n; i++) {
      ai[i] = Internals.getNewInstance();
      ae[i] = Externals.getNewInstanceOf(ai[i]);
  }
for(int i=0; i<n ;i++){
   S.O.P ("USN:" + ai[i].getUSN() + "\t Name:"+
  ai[i].getName() + "\t Semester:" + ai[i].getSemester());
  S.O.P("Internals:")
  for(int j=0 ;j<5 ;j++){
     S.O.P (ai[i].getMarks()[j] + "\t");
    result[j] = ai[i].getMarks()[j];
  }
  sc.close();
}
}
```

o/p

Enter the number of students: 1
Enter Student details and marks of CIE marks:
Enter USN: 4BEC5415
Enter Name: Sushanth
Enter Semester: 5
Enter CIE marks: 20 18 15 22 19
Enter 5 See marks: 60 70 65 55 75

Student marks details

USN: 4BEC5415 Name: John Semester: 5

CIE marks:	20	18	15	22	19
SEE marks:	60	70	65	55	75
Final marks:	50	53	58	49	56

Code

Package CIE

```
package CIE;

public class Personal {
    public String usn;
    public String name;
    public int sem;

    public void setDetails(String usn, String name, int sem) {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
}
```

```
package CIE;

public class Internals {
    public int[] internalMarks = new int[5];

    public void setInternalMarks(int[] marks) {
        for (int i = 0; i < 5; i++) {
            internalMarks[i] = marks[i];
        }
    }
}
```

Package SEE

```
package SEE;

import CIE.Personal;

public class External extends Personal {
    public int[] externalMarks = new int[5];

    public void setExternalMarks(int[] marks) {
        for (int i = 0; i < 5; i++) {
            externalMarks[i] = marks[i];
        }
    }
}
```

Main Program

```
import CIE.*;
import SEE.*;
import java.util.Scanner;

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

        System.out.print("Enter the number of students: ");
        int n = scanner.nextInt();
        scanner.nextLine();

        Personal[] personal = new Personal[n];
        Internals[] internals = new Internals[n];
        External[] externals = new External[n];

        for (int i = 0; i < n; i++) {
            personal[i] = new Personal();
            internals[i] = new Internals();
            externals[i] = new External();

            System.out.println("Enter details for Student " + (i + 1));
            System.out.print("USN: ");
            String usn = scanner.nextLine();
            System.out.print("Name: ");
```

```
        String name = scanner.nextLine();
        System.out.print("Semester: ");
        int sem = scanner.nextInt();
        personal[i].setDetails(usn, name, sem);

        System.out.println("Enter internal marks for 5 courses: ");
        int[] internalMarks = new int[5];
        for (int j = 0; j < 5; j++) {
            internalMarks[j] = scanner.nextInt();
        }
        internals[i].setInternalMarks(internalMarks);

        System.out.println("Enter SEE marks for 5 courses: ");
        int[] externalMarks = new int[5];
        for (int j = 0; j < 5; j++) {
            externalMarks[j] = scanner.nextInt();
        }
        externals[i].setExternalMarks(externalMarks);
        scanner.nextLine();
    }

    System.out.println("\nFinal Marks of Students:");
    for (int i = 0; i < n; i++) {
        System.out.println("Student " + (i + 1) + ":");
        System.out.println("USN: " + personal[i].usn);
        System.out.println("Name: " + personal[i].name);
```

```
            System.out.println("Semester: " + personal[i].sem);


            System.out.println("Final Marks for 5 courses:");
            for (int j = 0; j < 5; j++) {
                int finalMarks = internals[i].internalMarks[j] + (externals[i].externalMarks[j]
/ 2);
                System.out.println("Course " + (j + 1) + ": " + finalMarks);
            }
            System.out.println();
        }


        scanner.close();
    }
}
```

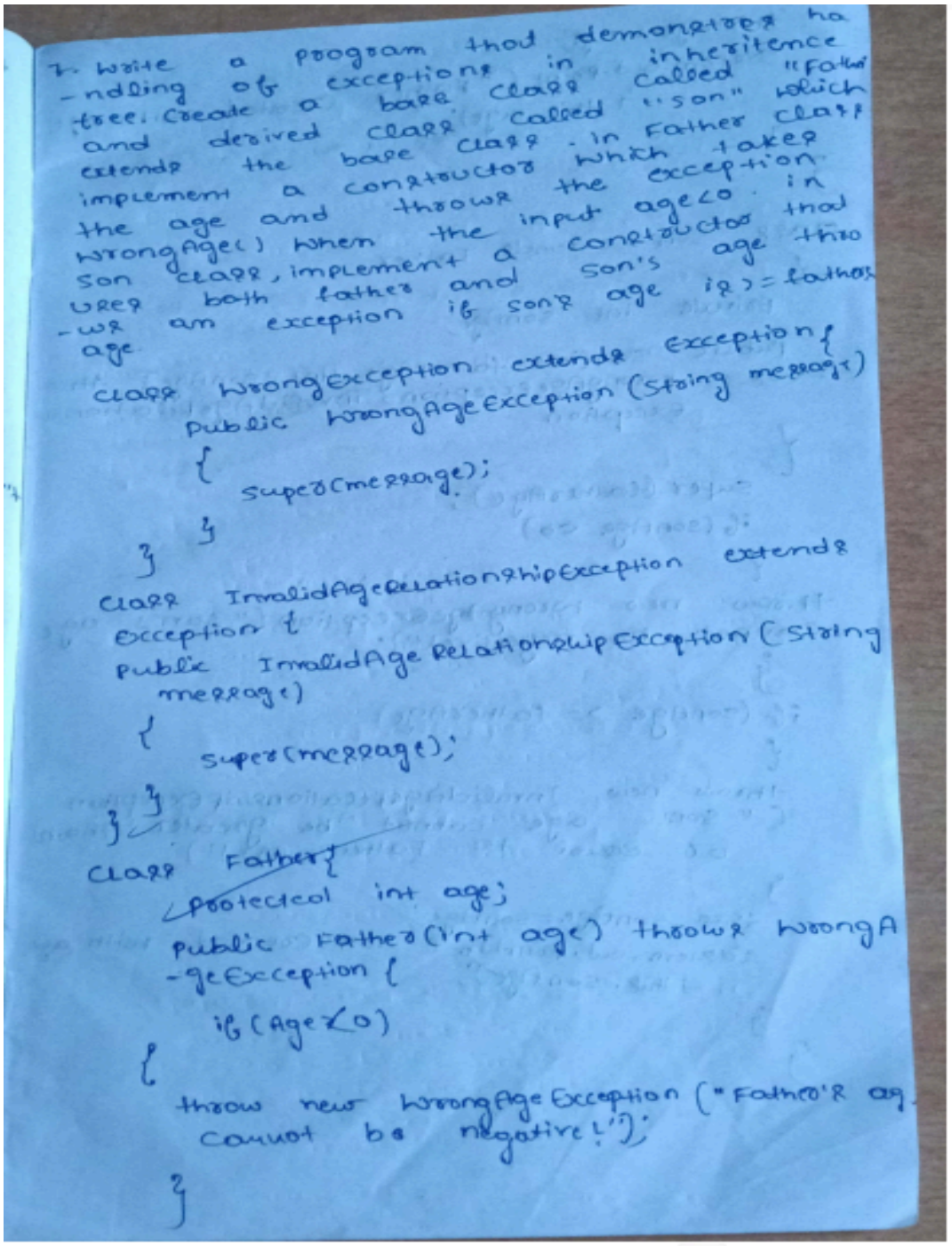
Output

```
PS C:\Users\Shashank U\Desktop\OOJ\Lab programs> javac CIE/Internals.java CIE/Student.java
PS C:\Users\Shashank U\Desktop\OOJ\Lab programs> javac SEE/Externals.java
PS C:\Users\Shashank U\Desktop\OOJ\Lab programs> javac Main.java
PS C:\Users\Shashank U\Desktop\OOJ\Lab programs> java -cp . Main
>>
Enter the number of students whose details you want to enter
1
Student 1 details:
Enter student USN
1BM22IS007
Enter student name
Dikshith
Enter semester
2
Enter CIE marks in subject 1
90
Enter CIE marks in subject 2
85
Enter CIE marks in subject 3
83
Enter CIE marks in subject 4
90
Enter CIE marks in subject 5
86
Enter SEE marks in subject 1
98
```

```
Enter SEE marks in subject 1
98
Enter SEE marks in subject 2
96
Enter SEE marks in subject 1
98
Enter SEE marks in subject 2
96
Enter SEE marks in subject 3
Enter SEE marks in subject 2
96
Enter SEE marks in subject 3
Enter SEE marks in subject 3
86
Enter SEE marks in subject 4
86
Enter SEE marks in subject 4
80
Enter SEE marks in subject 5
84

Student USN: 1BM22IS007
Student name: Dikshith
Semester: 2

Subject 1: 139
Subject 2: 133
Subject 3: 126
Subject 4: 130
Subject 5: 128
```

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

Screen shots



```
        this.age = age;
        System.out.Print ("Father Created with age:"
                + this.age);
    }
}

class Son extends Father
{
    Pairate int sonAge;

    Public Son(int fatherAge, int sonAge) throw
- wr     WrongAgeException, InvalidAgeRelationship
            Exception
    {
        super (fatherAge);
        if (sonAge <0)
        {
throw new WrongAgeException ("son's age
 cannot be, negative!");
        }
        if (sonAge >= fatherAge)
        {
        throw new InvalidAgeRelationshipException
        ("son's age cannot be greater than
            or equal to Father's age!");
        }
          this.sonAge = sonAge;
          System.out.Print("son Created with age:
              + this.sonAge);
        }
    }
```

```
class ExceptionDemo
{
  public static void main(String args[])
  {
    try
    {
      Father father = new Father(40);
      Son son = new Son(40,20);
    }
    catch(WrongAgeException | InvalidRelationsh
-Exception e)
    {
      System.out.println("Exception:" + e.getMessa
    }
  }
}
```

O/P

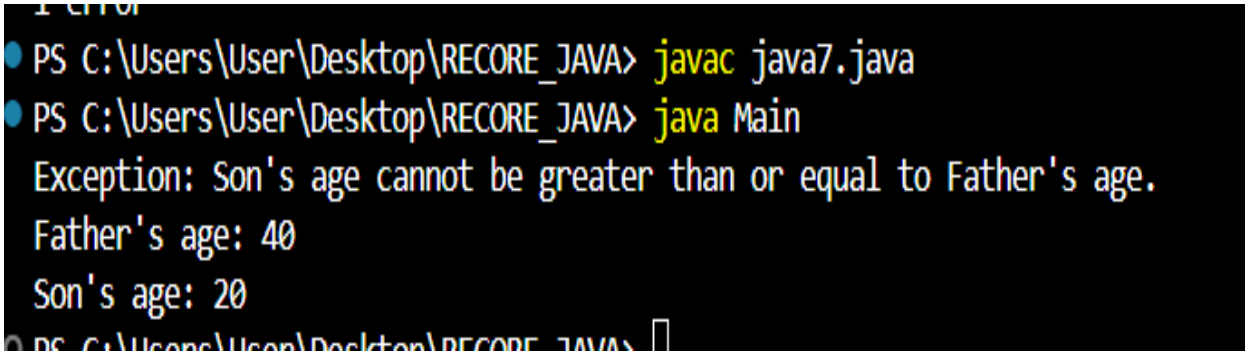
Enter Father's age: 16
Enter Son's age: 18
Error!: Son's age >= Father's age
Enter Father's age: 24
Enter Son's age: 29
Enter Son's age: 18
Father's age: 29
Son's age: 18

Code

```
class WrongAgeException extends Exception {
    public WrongAgeException(String message) {
        super(message);
    }
}

class Father {
    int fatherAge;

    public Father(int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException("Father's age cannot be negative.");
        }
        this.fatherAge = age;
    }
}

class Son extends Father {
    int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAgeException {
        super(fatherAge);
        if (sonAge < 0) {
            throw new WrongAgeException("Son's age cannot be negative.");
        }
```

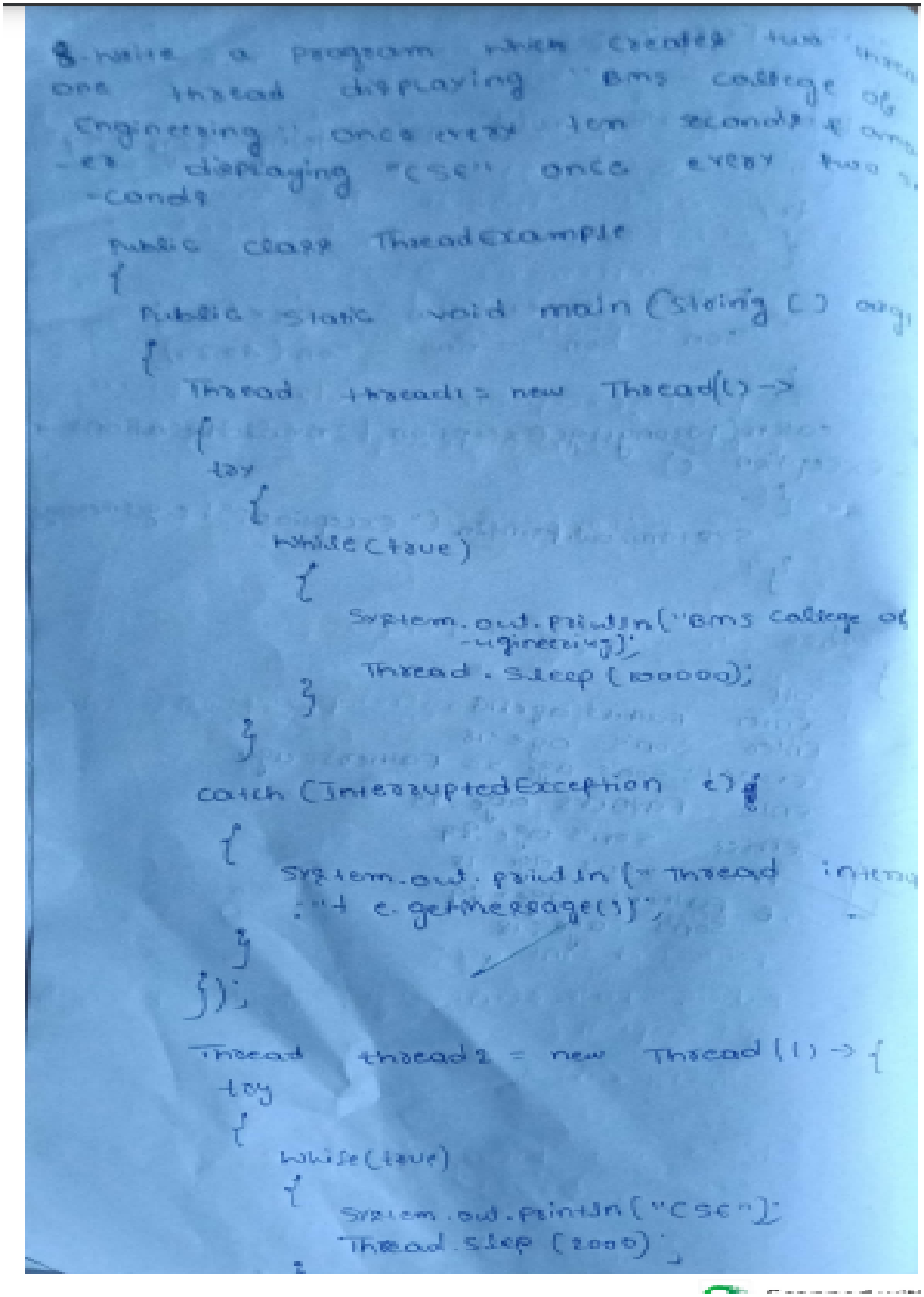
```
        if (sonAge >= fatherAge) {
            throw new WrongAgeException("Son's age cannot be greater than or equal to Father's age.");
        }
        this.sonAge = sonAge;
    }
}

public class Main {
    public static void main(String[] args) {
        try {
            Father father = new Father(45);
            Son son = new Son(45, 50);
        } catch (WrongAgeException e) {
            System.out.println("Exception: " + e.getMessage());
        }

        try {
            Father father2 = new Father(40);
            Son son2 = new Son(40, 20);
            System.out.println("Father's age: " + father2.fatherAge);
            System.out.println("Son's age: " + son2.sonAge);
        } catch (WrongAgeException e) {
            System.out.println("Exception: " + e.getMessage());
        }
    }
}
```

Output


PS C:\Users\User\Desktop\RECORE_JAVA> javac java7.java
PS C:\Users\User\Desktop\RECORE_JAVA> java Main
Exception: Son's age cannot be greater than or equal to Father's age.
Father's age: 40
Son's age: 20

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

Screen Shots



```
}
catch(InterruptedException e)
{
    System.out.println("Thread interrupted:" + e.get
    Message());
}
});
thread1.start();
thread2.start();
}
}
```

o/p

Bms college of Engineering
CSE
CSE
CSE
CSE
CSE

Bms college of engineering
CSE
CSE
CSE
CSE
CSE

Bms college of Engineering
CSE
CSE
CSE
CSE
CSE

Code

```
class ThreadBMS extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000);
            }
        } catch (InterruptedException e) {
            System.out.println(e);
        }
    }
}

class ThreadCSE extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        } catch (InterruptedException e) {
            System.out.println(e);
        }
    }
}
```

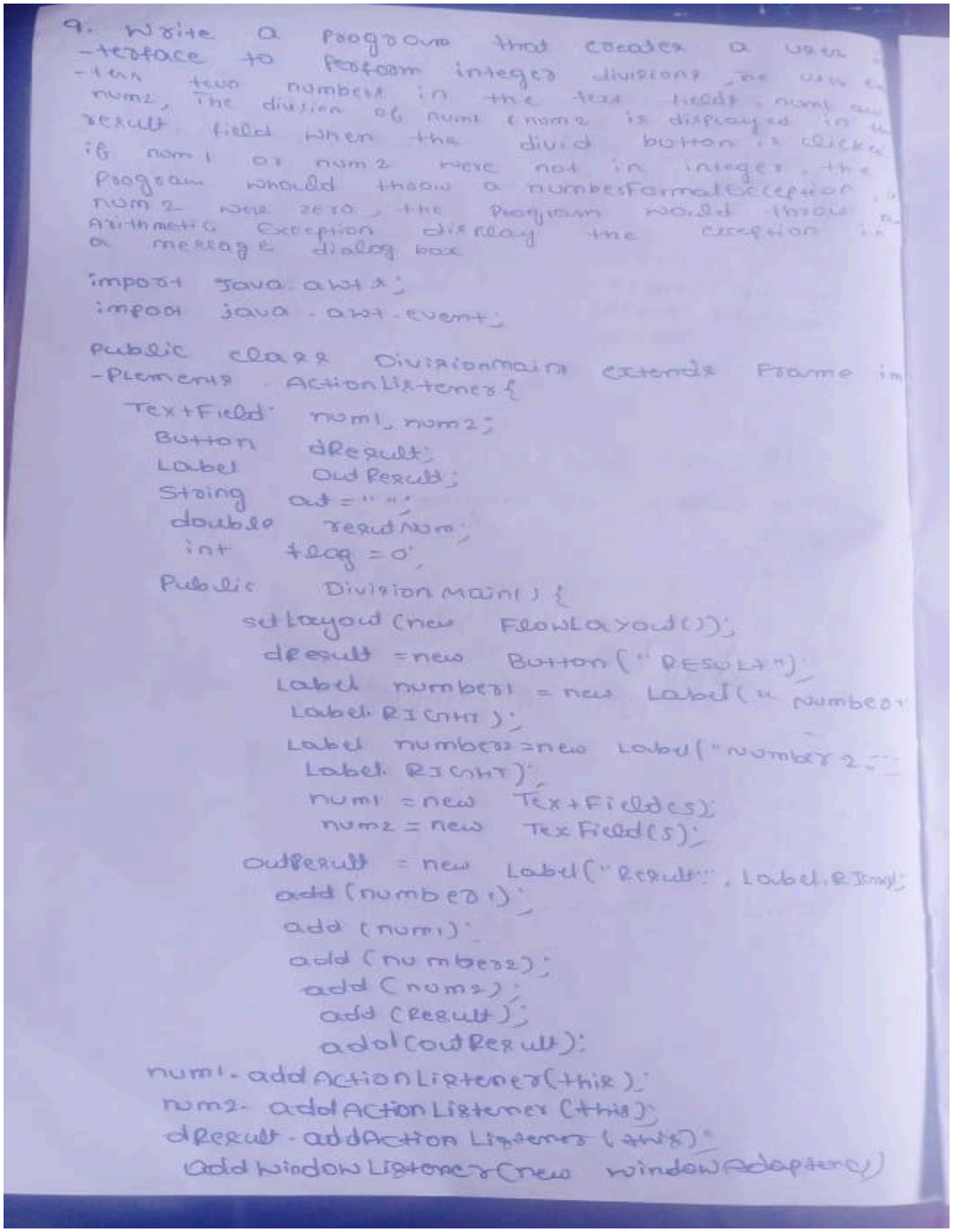
```
public class Main {
    public static void main(String[] args) {
        ThreadBMS thread1 = new ThreadBMS();
        ThreadCSE thread2 = new ThreadCSE();


        thread1.start();
        thread2.start();
    }
}
```

Output

```
PS C:\Users\User\Desktop\RECORE_JAVA> javac java8.java
PS C:\Users\User\Desktop\RECORE_JAVA> java Main
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
```

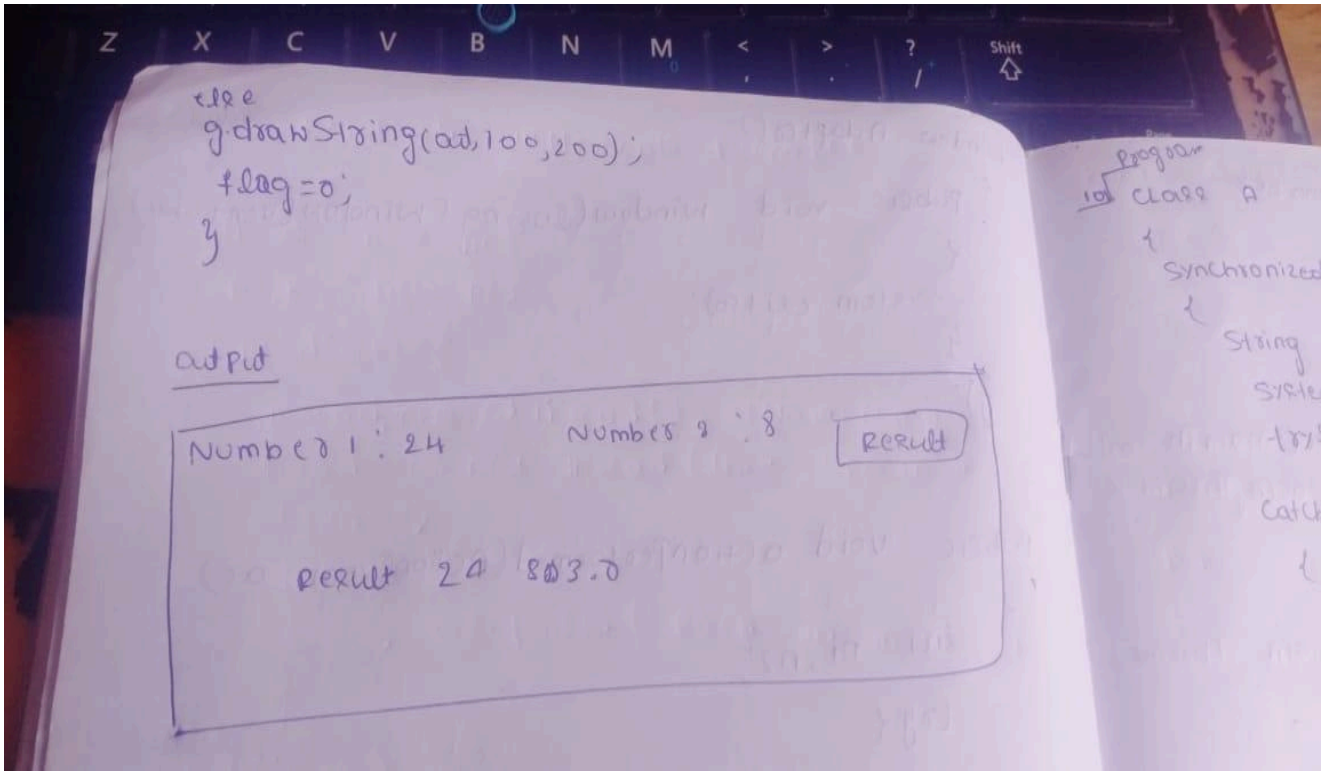
9.Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a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

Screen shots



```
9. Write a Program that creates a user
-terface to Perform integer divisions, The user e
-ters two numbers in the text fields, Num1 and
Num2, The division of Num1 and Num2 is displayed in t
Result field when the divid button is clicke
if Num1 or Num2 were not in integer, the
Program should throw a NumberFormatException,
Num2 were zero, the Program would throw a
Arithmetic Exception display the exception in
a message dialog box

import java.awt.*;
import java.awt.event;

Public class DivisionMain extends Frame im
-Plements ActionListener{
   TextField num1, num2;
   Button dResult;
   Label outResult;
   String out = "";
   double resultNum;
   int flag = 0;
   Public DivisionMain() {
      setLayout(new FlowLayout());
      dResult = new Button("RESULT");
      Label number1 = new Label("Number1
      Label.RIGHT);
      Label number2 =new Label("Number2
      Label.RIGHT);
      num1 = new TextField(5);
      num2 = new TextField(5);
      outResult = new Label("Result:", Label.RIGHT);
      add(number1);
      add(num1);
      add(number2);
      add(num2);
      add(Result);
      add(outResult);
   num1.addActionListener(this);
   num2.addActionListener(this);
   dResult.addAction Listener(this);
   Add WindowListener(new WindowAdapter())
```

```
Window Adapter()
{
  Public void window(closing(WindowEvent w)
  {
    System.exit(0);
  }
});
}
Public void actionPerformed(ActionEvent ae)
{
  int n1, n2;
  try{
    if(ae.getSource() == dResult)
    {
      n1 = Integer.parseInt(num1.getText());
      n2 = Integer.parseInt(num2.getText());
      out = n1 + " " + n2 + " ";
      resultNum = n1/n2;
      out = String.valueOf(resultNum);
      repaint();
    }
  }
catch(NumberFormatException e1)
  {
    flag=1;
    out= "Number format Exception!" +e1;
    repaint();
  }
catch(ArithmeticException e2)
  {
    flag = 1;
    out = "Divide by 0 Exception!" +e2;
    repaint();
  }
Public void paint(Graphics g)
{
  if(flag==0)
  g.drawString(out, outResult.getX() + outResult.get
    -width(), outResult.getY() + outResult.getHeight
    -ght()-7);
```


else
g.drawString(ad,100,200);
flag=0;
}
adpid
Number 1 : 24
Number 2 : 8
Result
Result 24 803.0
Program
10 class A
{
Synchronized
{
String
Syste
try
Catch
{

Code

```
import
java.awt.event.*;

public class DivisionMain1 extends Frame implements ActionListener
{
        TextField
        num1,num2;
        Button dResult;
        Label
        outResult;
        String
        out="";
        double
        resultNum;
        int flag=0;

        public DivisionMain1()
        {
                setLayout(new FlowLayout());

                dResult = new Button("RESULT");
                Label number1 = new Label("Number
                1:",Label.RIGHT); Label number2 = new
                Label("Number          2:",Label.RIGHT);
                num1=new TextField(5);
                num2=new TextField(5);
                outResult = new Label("Result:",Label.RIGHT);

                add(number
                1);
                add(num1);
                add(number
                2);
                add(num2);
                add(dResult)
                ;
                add(outResul
                t);
```

```
        num1.addActionListener(this);
        num2.addActionListener(this);
        dResult.addActionListener(this);
        addWindowListener(new
        WindowAdapter()
        {
                public void windowClosing(WindowEvent we)
                {
                        System.exit(0);
                }
        });
}
public void actionPerformed(ActionEvent ae)
{
        int
        n1,n2;
        try
        {
                if (ae.getSource() == dResult)
                {
                        n1=Integer.parseInt(num1.getText());
                        n2=Integer.parseInt(num2.getText());

                        /*if(n2==0)
                                throw new
                        ArithmeticException();*/ out=n1+"
                        "+n2+" ";
                        resultNum=n1/n2;
                        out+=String.valueOf(result
                        Num); repaint();

                }
        }
        catch(NumberFormatException e1)
        {
                flag=1;
                out="Number Format Exception!
                "+e1; repaint();
        }
        catch(ArithmeticException e2)
```

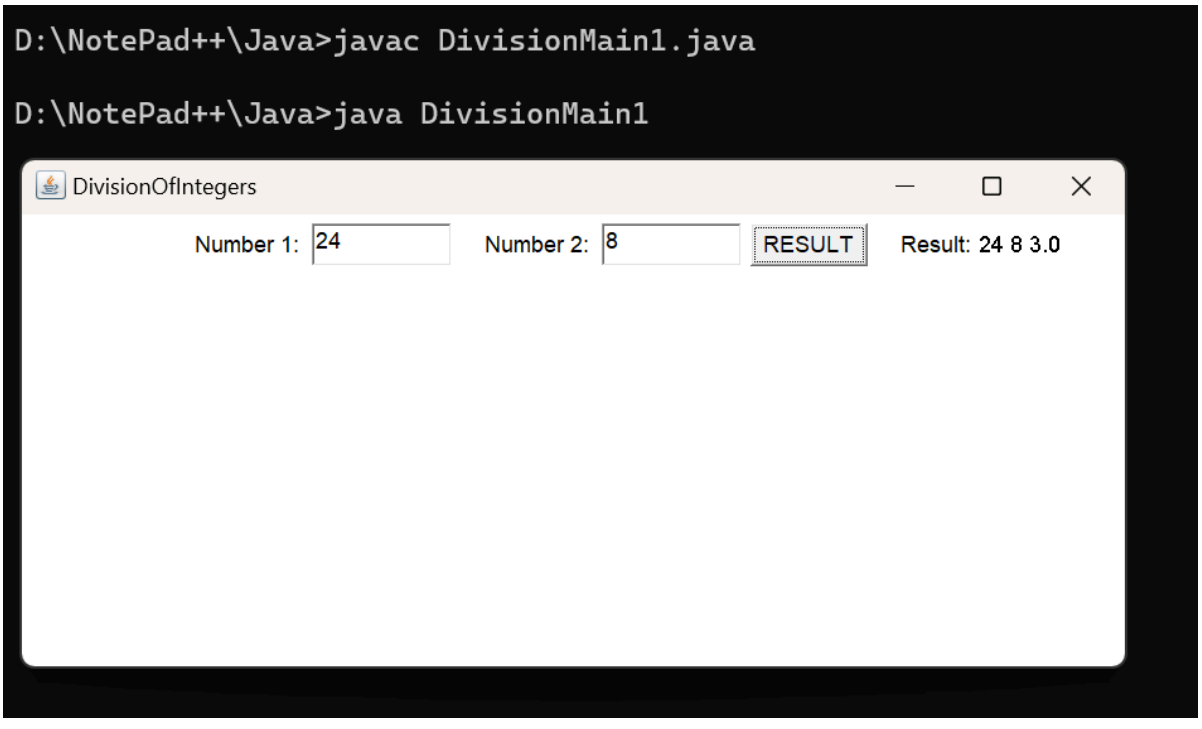
```
        {
                flag=1;
                out="Divide by 0 Exception!
                "+e2; repaint();
        }

}
public void paint(Graphics g)
{
        if(flag==0)
        g.drawString(out,outResult.getX()+outResult.getWidth(),outR
        esult.getY()+outResult. getHeight()-8);
        else
        g.drawString(out,1
        00,200); flag=0;
}
```

Demonstrate Interprocess communication and deadlock

```
class Q {
int n;
boolean valueSet = false;

synchronized int get() {
while(!valueSet)
try {
System.out.println("\nConsumer waiting\n");
wait();
} catch(InterruptedException e) {
System.out.println("InterruptedException caught");
}
System.out.println("Got: " + n);
valueSet = false;
System.out.println("\nIntimate Producer\n");
notify();
return n;
}

synchronized void put(int n) {
while(valueSet)
try {
System.out.println("\nProducer waiting\n");
wait();
} catch(InterruptedException e) {
System.out.println("InterruptedException caught");
}
this.n = n;
valueSet = true;
System.out.println("Put: " + n);
System.out.println("\nIntimate Consumer\n");
notify();
}
```

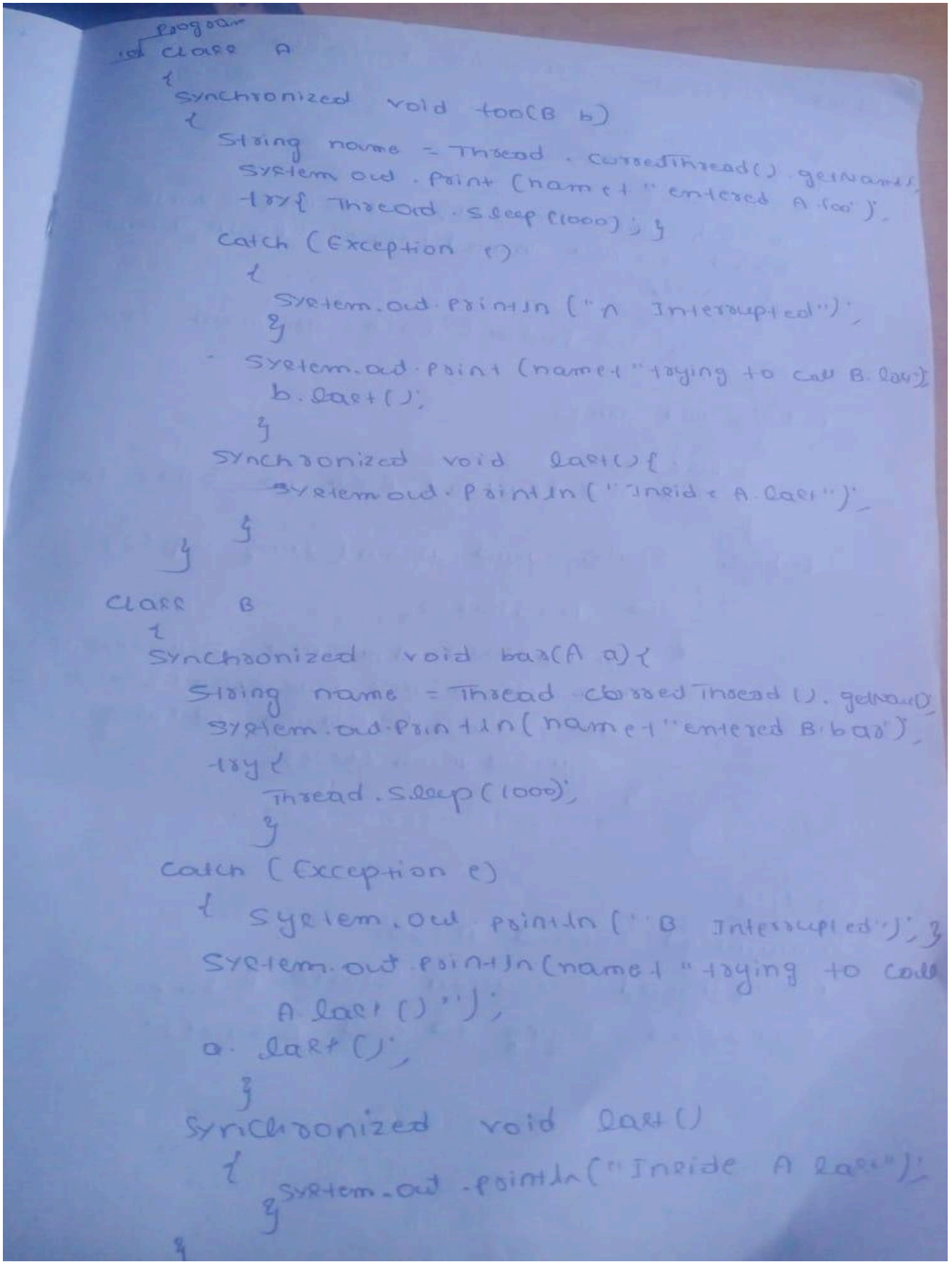
```
}

class Producer implements Runnable {
Q q;
Producer(Q q) {
this.q = q;
new Thread(this, "Producer").start();
}
public void run() {
int i = 0;
while(i<15) {
q.put(i++);
}
}
}

class Consumer implements Runnable {
Q q;
Consumer(Q q) {
this.q = q;
new Thread(this, "Consumer").start();
}
public void run() {
        int i=0;
while(i<15) {
int r=q.get();
System.out.println("consumed:"+r);
i++;
}
}
}

class PCFixed {
public static void main(String args[]) {
Q q = new Q();
new Producer(q);
new Consumer(q);
System.out.println("Press Control-C to stop.");
}
}
```

OUTPUT



10.Demonstrate Inter process Communication and deadlock

Screen shots



```
Program
class A
{
 synchronized void foo(B b)
 {
  String name = Thread.currentThread().getName();
  System.out.Print(name+" entered A.foo");
  try{ Thread.sleep(1000); }
  catch (Exception e)
  {
   System.out.println(" A Interrupted");
  }
  System.out.Print(name+" trying to call B.last()");
  b.last();
 }
 synchronized void last(){
  System.out.Println("Inside A.last");
 }
}
class B
{
 synchronized void bar(A a){
  String name = Thread.currentThread().getName();
  System.out.println(name+" entered B.bar");
  try{
   Thread.sleep(1000);
  }
 catch (Exception e)
  { System.out.println(" B Interrupted"); }
  System.out.println(name+" trying to call
   A.last()");
  a.last();
 }
 synchronized void last()
 {
  System.out.println("Inside A.last");
 }
}
```

```
class Deadlock implements Runnable
{
  A a = new A();
  B b = new B();
  Deadlock() {
   Thread.currentThread().setName("mainThread");
   Thread t = new Thread(this, "RacingThread");
   t.start();
   b.foo(a);
   System.out.println("Back in main thread");
  }
 public void run()
  {
   b.bar(a);
   System.out.println("Back in other thread");
  }
 public static void main(String args[]){
    new Deadlock();
   }
     DivisionMain dm = new DivisionMain();
      dm.setSize(new Dimension(800,400));
      dm.setTitle("Division of Integers");
       dm.setVisible(true);
    }
  }
```

Output

mainThread entered A.foo.

RacingThread Execution Started B.bar.

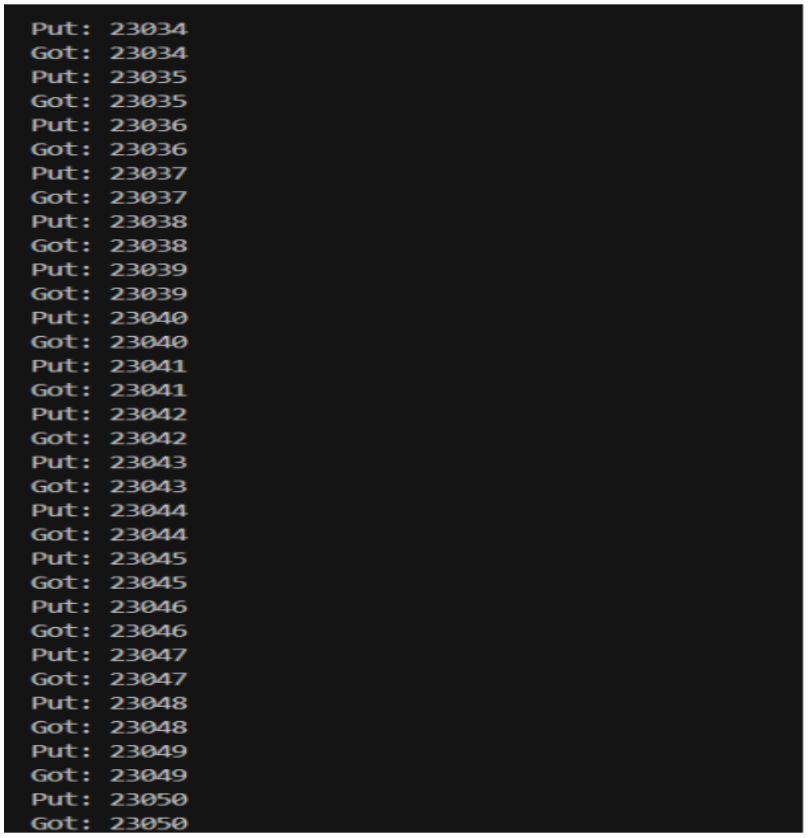
mainThread trying to call B.last();

Code

```
class Q {
 int n;
 boolean valueSet = false;
 synchronized int get() {
 while (!valueSet) {
 try {
 wait();
 } catch (InterruptedException e) {
 System.out.println("InterruptedException caught");
 }
 }
 System.out.println("Got: " + n);
 valueSet = false;
 notify();
 return n;
 }
 synchronized void put(int n) {
 while (valueSet) {
 try {
 wait();
 } catch (InterruptedException e) {
 System.out.println("InterruptedException caught");
 }
 }
 this.n = n;
 valueSet = true;
 System.out.println("Put: " + n);
 notify();
 }
}
class Producer implements Runnable {
 Q q;
 Thread t;
 Producer(Q q) {
 this.q = q;
 t = new Thread(this, "Producer");
 }
 public void run() {
 int i = 0;
 while (true) {
 q.put(i++);
 }
 }
}
class Consumer implements Runnable {
 Q q;
 Thread t;
 Consumer(Q q) {
 this.q = q;
 t = new Thread(this, "Consumer");
 }
```

```
 public void run() {
 while (true) {
 q.get();
 }
 }
}
class PCFixed {
 public static void main(String args[]) {
 Q q = new Q();
 Producer p = new Producer(q);
 Consumer c = new Consumer(q);
 // Start the threads.
 p.t.start();
 c.t.start();
 System.out.println("Press Control-C to stop.");
 }
}
```

Output



Code

```
class A {
 synchronized void foo(B b) {
 String name = Thread.currentThread().getName();
 System.out.println(name + " entered A.foo");
 try {
 Thread.sleep(1000);
 } catch (Exception e) {
 System.out.println("A Interrupted");
 }
 System.out.println(name + " trying to call B.last()");
 b.last();
 }
 synchronized void last() {
 System.out.println("Inside A.last");
 }
}
class B {
 synchronized void bar(A a) {
 String name = Thread.currentThread().getName();
 System.out.println(name + " entered B.bar");
 try {
 Thread.sleep(1000);
 } catch (Exception e) {
 System.out.println("B Interrupted");
 }
 System.out.println(name + " trying to call A.last()");
 a.last();
 }
 synchronized void last() {
 System.out.println("Inside B.last");
 }
}
class Deadlock implements Runnable {
 A a = new A();
 B b = new B();
 Thread t;
 Deadlock() {
 Thread.currentThread().setName("MainThread");
 t = new Thread(this, "RacingThread");
 }
 void deadlockStart() {
 t.start();
 a.foo(b); // get lock on a in this thread
 System.out.println("Back in main thread");
 }
 public void run() {
 b.bar(a); // get lock on b in other thread
 System.out.println("Back in other thread");
 }
 public static void main(String args[]) {
```

```
Deadlock dl = new Deadlock();
dl.deadlockStart();
 }
}
```

Output

```
PS C:\Users\Shashank U\Desktop\NOTES\Object oriented java programming> javac Deadlock.java
PS C:\Users\Shashank U\Desktop\NOTES\Object oriented java programming> java Deadlock
RacingThread entered B.bar
MainThread entered A.foo
RacingThread trying to call A.last()
MainThread trying to call B.last()
```