VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Anirudh Raghunand (1BM23CS036)

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 to 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 (23CS3PCOOJ)" carried out by **Anirudh Raghunand (1BM23CS036),** 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. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

Sheetal V A Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE

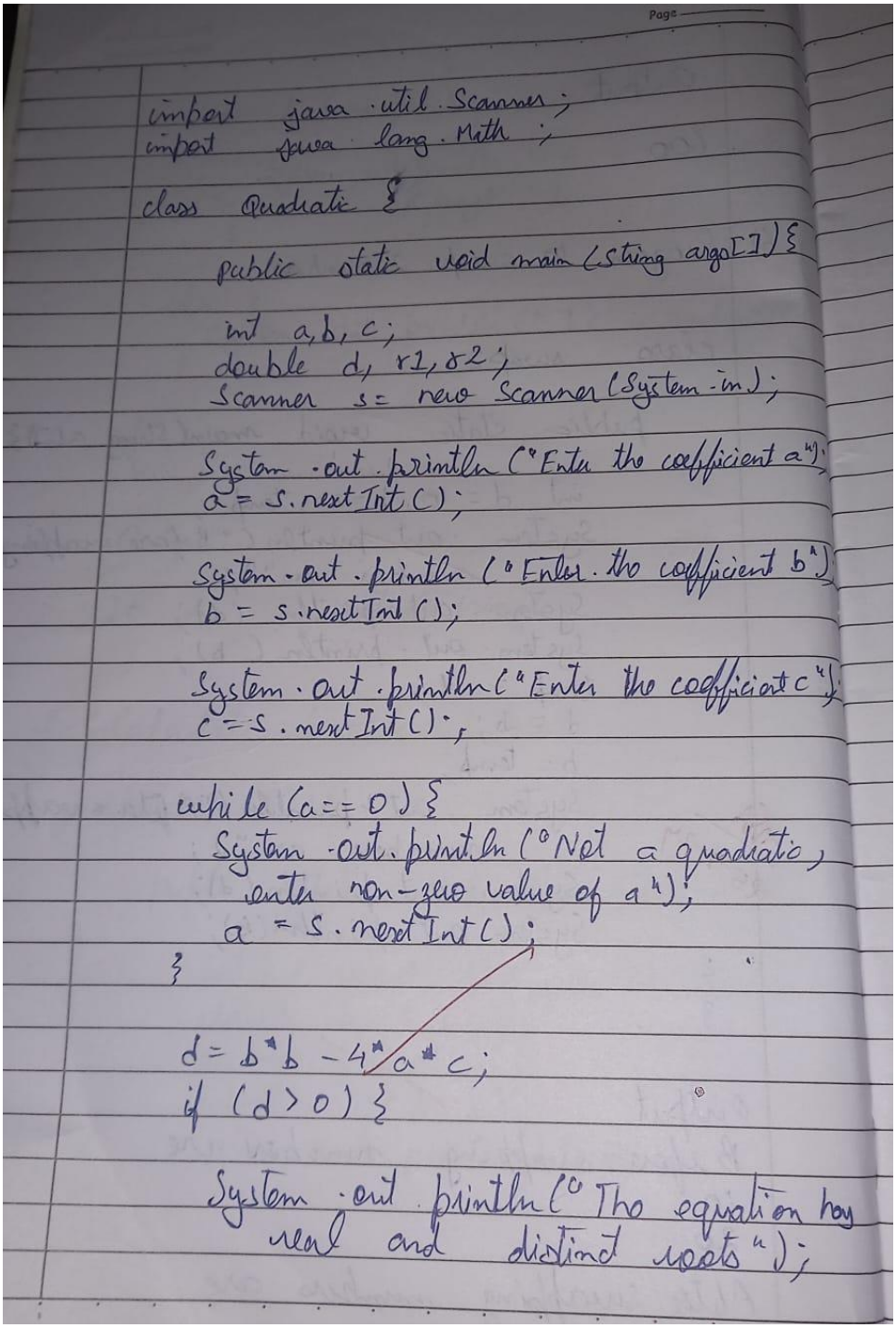
Index

Github Link:

https://github.com/AnirudhR036/JavaLab

Program 1
Implement Quadratic Equation

Algorithm



```
import java.util.Scanner;
import java.lang.Math;
class Quadratic {
    public static void main(String args[]) {
        int a,b,c;
        double d, r1, r2;
        Scanner s= new Scanner(System.in);
        System.out.println("Enter the coefficient a");
        a = s.nextInt();
        System.out.println("Enter the coefficient b");
        b = s.nextInt();
        System.out.println("Enter the coefficient c");
        c = s.nextInt();
        while(a==0){
            System.out.println("Not a quadratic, enter non-zero value of a");
            a = s.nextInt();
        }
        d = b*b - 4*a*c;
        if(d>0){
            System.out.println("The equation has real and distinct roots");
```

```
r1 = ((-b)/(2*a)+(Math.sqrt(d))/(2*a));
r2 = ((-b)/2*a) -(Math.sqrt(d))/(2*a));
System.out.println("The roots are");
System.out.println("r1 = " +r1);
System.out.println("r2 = " +r2);

}

else if (d == 0) {

    System.out.println("The equation has
        real and equal roots");
    r1 = ((-b)/(2*a));
    System.out.println("The root is" +r1);

}

else {
    System.out.println("The equation
        doesn't have real roots");
}
}
```

Output

Enter the coefficient a

1

Enter the coefficient b

0

Enter the coefficient c

-4

The equation has real and distinct roots
the roots are
r1 = 2.0
r2 = -2.0

Enter the coefficient a
1
Enter the coefficient b
2
Enter the coefficient c
1
The equation has real and equal roots
The root is -1.0

Enter the coefficient a
1
Enter the coefficient b
0
Enter the coefficient c
1
The equation doesn't have real roots.

9-10-24

Code:

```
import java.util.Scanner;
import java.lang.Math;
class Quadratic{

 public static void main(String args[]){

    int a,b,c;
    double d,r1,r2;
    Scanner s= new Scanner(System.in);

    System.out.println("Enter the coefficient a");
    a=s.nextInt();

    System.out.println("Enter the coefficient b");
    b=s.nextInt();

    System.out.println("Enter the coefficient c");
    c=s.nextInt();

    while(a==0){

     System.out.println("Not a quadratic , enter non-zero value of a");
     a=s.nextInt();

}
     d=b*b-4*a*c;
     if(d>0){

      System.out.println("The equation has real and distinct roots");
      r1=((-b)/(2*a)+(Math.sqrt(d))/(2*a));
      r2=((-b)/(2*a)-(Math.sqrt(d))/(2*a));
      System.out.println("The roots are");
      System.out.println("r1="+r1);
      System.out.println("r2="+r2);
    }

     else if(d==0){

          System.out.println("The equation has real and equal roots");
          r1=((-b)/(2*a));
          System.out.println("The root is "+r1);

}
```

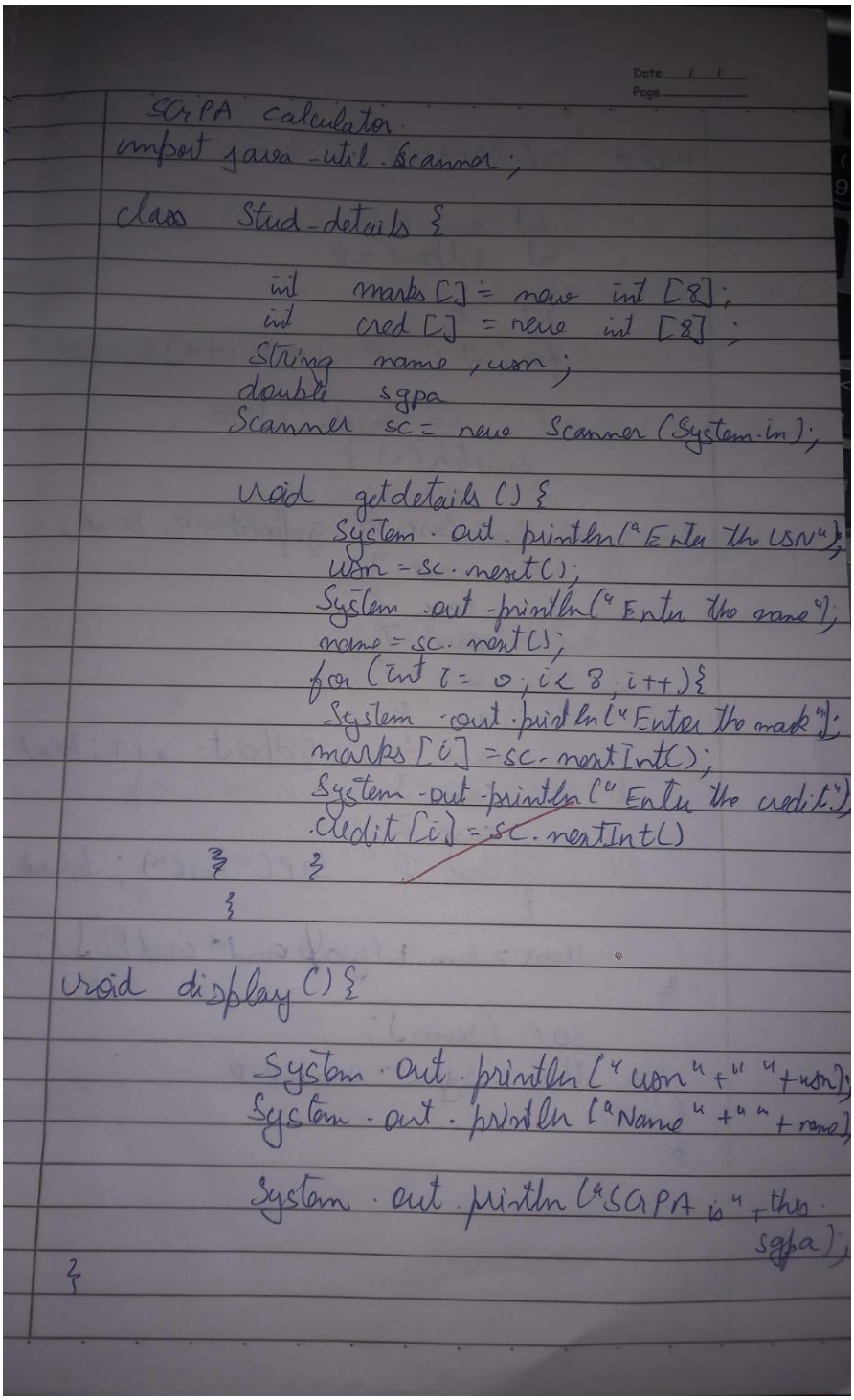
```
else{
    System.out.println("The equation doesn't have real roots");


}

System.out.println("Name:Anirudh");
System.out.println("USN:1BM23CS036");
}

}
```

Output:

```
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>Desktop\1bm23cs036
'Desktop\1bm23cs036' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Admin>cd Desktop\1bm23cs036

C:\Users\Admin\Desktop\1bm23cs036>javac Quadratic.java

C:\Users\Admin\Desktop\1bm23cs036>java Quadratic
Enter the coefficient a
1
Enter the coefficient b
2
Enter the coefficient c
1
The equation has real and equal roots
The root is -1.0
Name:Anirudh
USN:1BM23CS036

C:\Users\Admin\Desktop\1bm23cs036>java Quadratic
Enter the coefficient a
1
Enter the coefficient b
0
Enter the coefficient c
-4
The equation has real and distinct roots
The roots are
r1=2.0
r2=-2.0
Name:Anirudh
USN:1BM23CS036

C:\Users\Admin\Desktop\1bm23cs036>java Quadratic
Enter the coefficient a
1
Enter the coefficient b
0
Enter the coefficient c
1
The equation doesn't have real roots
Name:Anirudh
USN:1BM23CS036
```

Program 2
SGPA Calculator

Algorithm:



```
SGPA calculator
import java.util.Scanner;

class Stud_details {

    int marks[] = new int[8];
    int cred[] = new int[8];
    String name, usn;
    double sgpa
    Scanner sc = new Scanner(System.in);

    void getdetails() {
        System.out.println("Enter the USN");
        usn = sc.next();
        System.out.println("Enter the name");
        name = sc.next();
        for (int i = 0; i < 8; i++){
        System.out.println("Enter the marks");
        marks[i] = sc.nextInt();
        System.out.println("Enter the credit");
        credit[i] = sc.nextInt()
    }
    }

void display() {

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

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

```
void calcSGPA(){

    int sum=0;
    int gradpoint=0;

    int x;
    for(int i=0; i<8; i++){

        x= marks[i]/10;
      switch(x){

        case 10: gradpoint=10; break;
        case 9:
        case 8:
        case 7:
        case 6
        case 5
        case 4 : gradpoint= x+7; break;
        case 3
        case 2:
        case 1: SOP("Fail"); break;
        }

      sum = sum + (gradpoint * cred[i]);
    }

    SOP (sum);
    this.sgpa = sum/20.0
  }
}
```

```
class Student {

  Public static void main(String args[]){

    Stud_details s1[] = new Stud_details[3];

    for(int j=0; j<3; j++){

        s1[j] = new Stud_details();

    for(int j=0; j<3; j++){
        SOP ("Enth the details of"+
            (j+1)+" student");

        s1[j].getdetails();
        s1[j] calcSGPA();
    }

  for(int j=0; j<3; j++){

        s1[j].display();
    }
  }
}
```

Enter the details of 1 Student

Enter the USN
036
Enter the name
aniruddh
Enter the mark
91
Enter the credit
4
Enter the mark
91
Enter the credit
4
Enter the mark
98
Enter the credit
3
Enter the mark
100
Enter the credit
3
Enter the mark
98
Enter the credit
3
Enter the mark
89
Enter the credit
1
Enter the mark
[illegible] 1
Enter the credit

Enter the credit mark
100
Enter the credit
1

USN 036
name aniruddh
SGPA is 9.25

16-10-24

Code:

```
import java.util.Scanner;
class Stud_details{

	int marks[]=new int[8];
	int cred[]= new int[8];
	String name,usn;
	double sgpa;
	Scanner sc =new Scanner(System.in);


	void getdetails(){
		System.out.println(" Enter the USN ");
		usn=sc.next();
		System.out.println(" Enter the name ");
		name=sc.next();
		for(int i=0;i<8;i++){
			System.out.println(" Enter the mark ");
			marks[i]=sc.nextInt();
			System.out.println(" Enter the credit ");
			cred[i]=sc.nextInt();
}
}

void display(){



	System.out.println("usn"+" "+usn);
	System.out.println("name"+" "+name);


	System.out.println("SGPA is"+this.sgpa);


}

void calcSGPA(){
	int sum=0;
	int grdpoint=0;

	int x;
	for(int i=0;i<8;i++){
```

```
                x=marks[i]/10;
                switch(x){

                        case 10: grdpoint=10;break;
                        case 9:
                        case 8:
                        case 7:
                        case 6:
                        case 5:
                        case 4: grdpoint=x+1;break;
                        case 3:
                        case 2:
                        case 1:System.out.println("Fail");break;
                        }
                sum=sum+ (grdpoint*cred[i]);
}
        System.out.println(sum);
        this.sgpa=sum/20.0;

}
}


class Student{
        public static void main(String args[]){
                Stud_details s1[]=new Stud_details[3];
                for(int j=0;j<3;j++){
                        s1[j]=new Stud_details();       }
                for(int j=0;j<3;j++){
                        System.out.println("Enter the details of "+(j+1)+" Student");
                        s1[j].getdetails();
                        s1[j].calcSGPA();
}
for(int j=0;j<3;j++){
        s1[j].display();
}

}

}
```

```
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bmsce>cd Desktop\Anirudh

C:\Users\bmsce\Desktop\Anirudh>javac sgpa.java

C:\Users\bmsce\Desktop\Anirudh>java Student
Enter the details of1Student
Enter the USN
036
Enter the name
anirudh
Enter the mark
91
Enter the credit
4
Enter the mark
91
Enter the credit
4
Enter the mark
98
Enter the credit
3
Enter the mark
98
Enter the credit
3
Enter the mark
100
Enter the credit
3
Enter the mark
89
Enter the credit
1
Enter the mark
1
Enter the credit
1
Enter the mark
89
Enter the credit
1
```

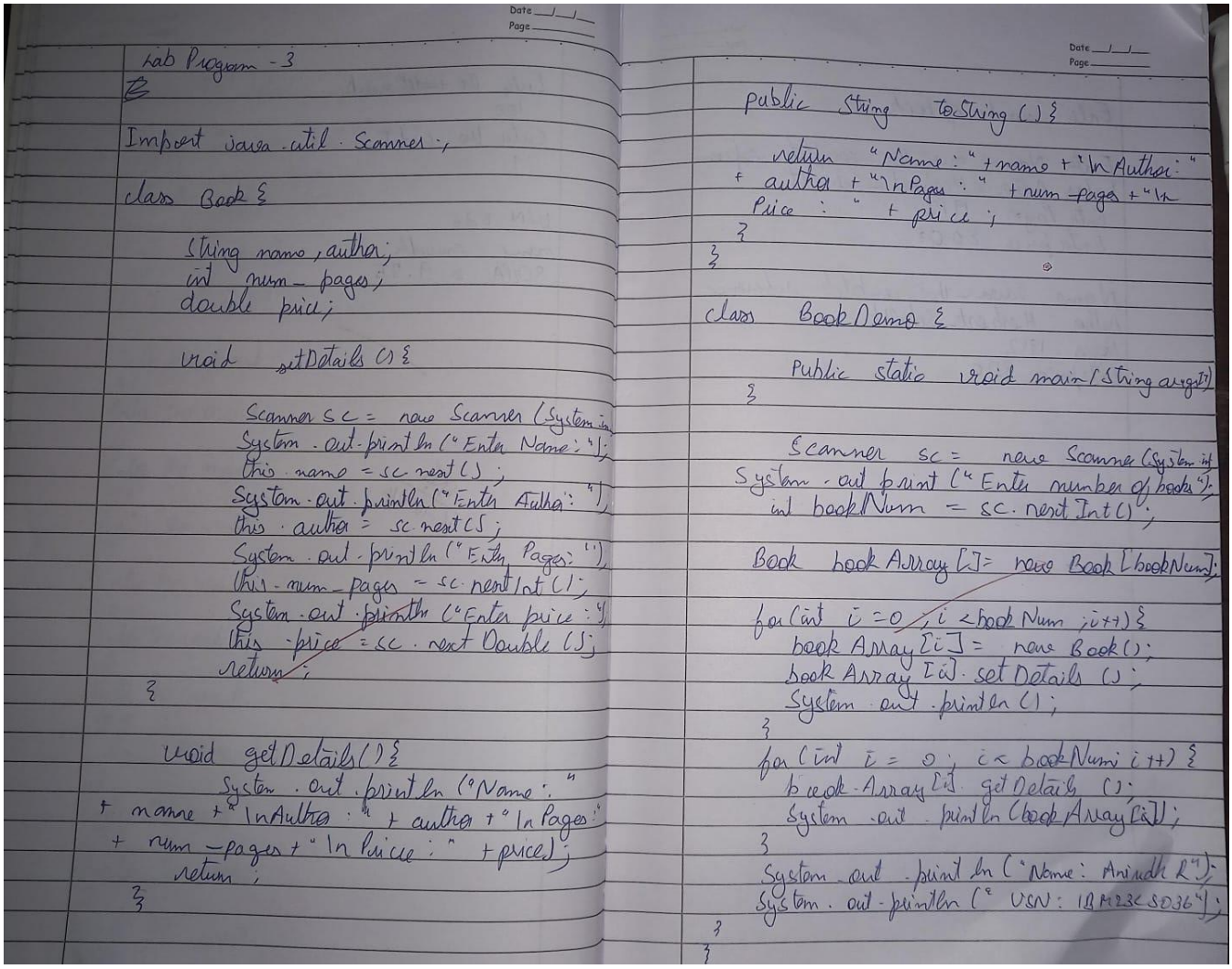
```
Enter the details of3Student
Enter the USN
28
Enter the name
amith
Enter the mark
100
Enter the credit
4
Enter the mark
100
Enter the credit
4
Enter the mark
100
Enter the credit
3
Enter the mark
100
Enter the credit
3
Enter the mark
100
Enter the credit
3
Enter the mark
100
Enter the credit
1
Enter the mark
100
Enter the credit
1
Enter the mark
100
Enter the credit
1
200
usn 036
name anirudh
SGPA is9.85
usn 048
name aprameya
SGPA is9.35
usn 28
name amith
SGPA is10.0
```

```
Enter the details of3Student
Enter the USN
28
Enter the name
amith
Enter the mark
100
Enter the credit
4
Enter the mark
100
Enter the credit
4
Enter the mark
100
Enter the credit
3
Enter the mark
100
Enter the credit
3
Enter the mark
100
Enter the credit
3
Enter the mark
100
Enter the credit
1
Enter the mark
100
Enter the credit
1
Enter the mark
100
Enter the credit
1
200
usn 036
name anirudh
SGPA is9.85
usn 048
name aprameya
SGPA is9.35
usn 28
name amith
SGPA is10.0
```

Program 3

Book Program

Algorithm:



```
Lab Program - 3

Import java.util.Scanner;

class Book {

    String name, autha;
    int num_pages;
    double price;

    void setDetails() {

        Scanner sc = new Scanner(System.in);
        System.out.println("Enter Name: ");
        this.name = sc.next();
        System.out.println("Enter Autha: ");
        this.autha = sc.next();
        System.out.println("Enter Pages: ");
        this.num_pages = sc.nextInt();
        System.out.println("Enter price: ");
        this.price = sc.nextDouble();
        return;
    }

    void getDetails() {
        System.out.println("Name: "
+ name + "\nAuthor: " + autha + "\nPages: "
+ num_pages + "\nPrice: " + price);
        return;
    }

    public String toString() {

        return "Name: " + name + "\nAuthor: "
+ autha + "\nPages: " + num_pages + "\n
Price: " + price;
    }
}

class BookDemo {

    public static void main(String args[])
    {

        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of books: ");
        int bookNum = sc.nextInt();

        Book bookArray[] = new Book[bookNum];

        for(int i=0; i<bookNum; i++) {
            bookArray[i] = new Book();
            bookArray[i].setDetails();
            System.out.println();
        }
        for(int i = 0; i<bookNum; i++) {
            bookArray[i].getDetails();
            System.out.println(bookArray[i]);
        }
        System.out.println("Name: Anirudh R");
        System.out.println("USN: 1BM23CS036");
    }
}
```

Date __/__/__
Page

Enter number of books 1

Enter Name: java_the_complete_reference
Enter Author: Herbert_Schildt
Enter Pages: 1912
Enter price 8003

Name: java_the_complete_reference
Autha: Herbert_Schildt
Pages: 1912
Enterprice: 800

Code:

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

   void setDetails(){
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter Name:");
      this.name = sc.next();
      System.out.print("Enter Author:");
      this.author = sc.next();
      System.out.print("Enter Pages:");
      this.num_pages = sc.nextInt();
      System.out.print("Enter Price:");
      this.price = sc.nextDouble();
      return;
   }

   void getDetails(){
      System.out.println("Name: "+name+"\nAuthor: "+author+"\nPages: "+num_pages+"\nPrice:
"+price);
      return;
   }

   public String toString(){
      return "Name: "+name+"\nAuthor: "+author+"\nPages: "+num_pages+"\nPrice: "+price;
   }
}

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


      Scanner sc = new Scanner(System.in);
      System.out.print("Enter the number of books: ");
      int bookNum = sc.nextInt();

      Book bookArray[] = new Book[bookNum];

      for(int i = 0; i<bookNum; i++){
      bookArray[i] = new Book();
      bookArray[i].setDetails();
      System.out.println();
      }
```

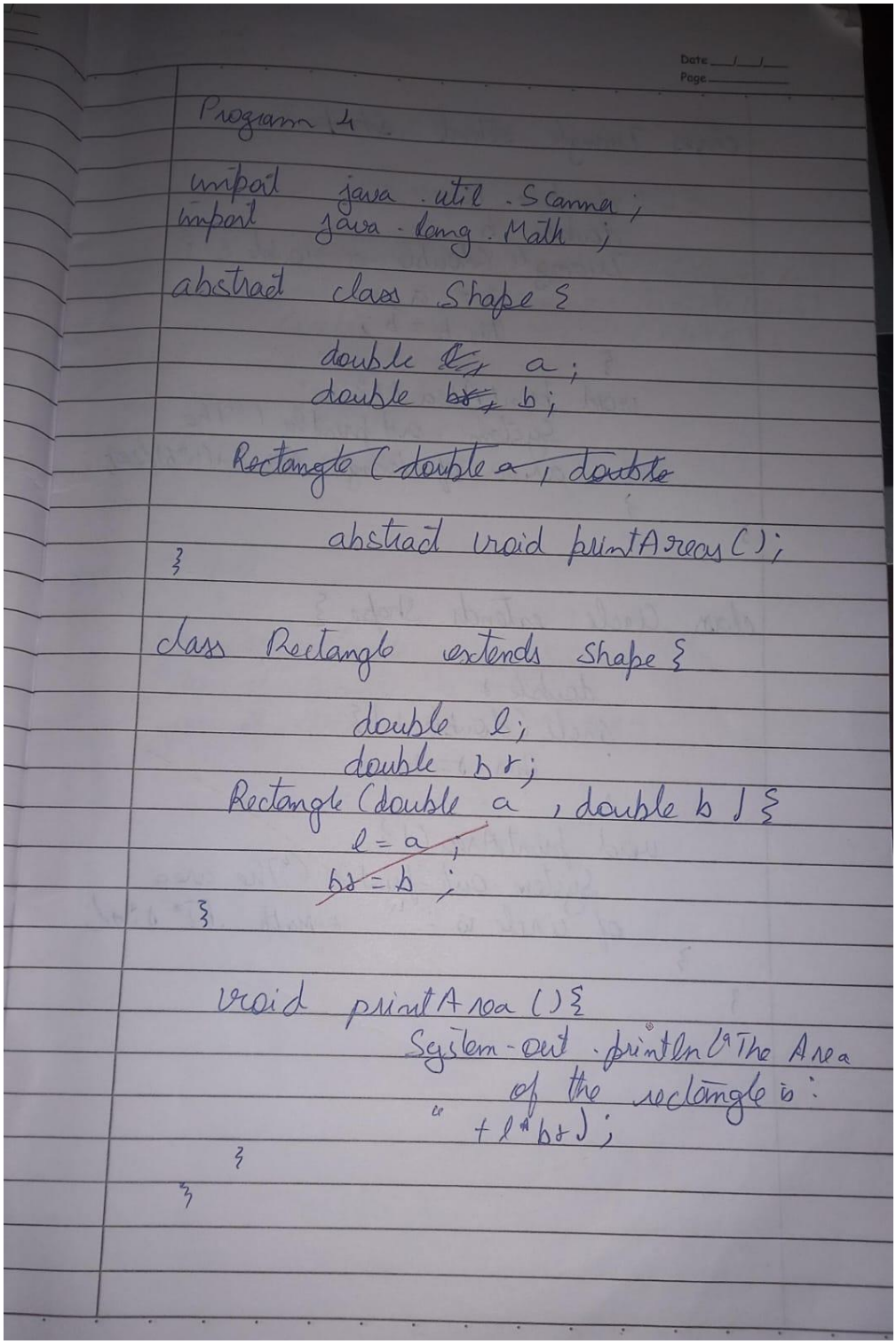
```
        for (int i = 0; i<bookNum; i++){
        bookArray[i].getDetails();
            System.out.println(bookArray[i]);

        }
            System.out.print("Name: Anirudh      R");
            System.out.print("USN: 1BM23CS036");
    }
}
```

```
C:\Users\bmsce\Desktop\1BM23CS036>javac BookDemo.java

C:\Users\bmsce\Desktop\1BM23CS036>java BookDemo
Enter the number of books: 3
Enter Name:java_the_complete_reference
Enter Author:Herbert_Schildt
Enter Pages:1912
Enter Price:800

Enter Name:Computer_Organization_and_Architecture
Enter Author:William_Stallings
Enter Pages:400
Enter Price:700

Enter Name:Introduction_to_Logic_Design
Enter Author:Alan_B_Marcovitz
Enter Pages:300
Enter Price:400

Name: java_the_complete_reference
Author: Herbert_Schildt
Pages: 1912
Price: 800.0
Name: Computer_Organization_and_Architecture
Author: William_Stallings
Pages: 400
Price: 700.0
Name: Introduction_to_Logic_Design
Author: Alan_B_Marcovitz
Pages: 300
Price: 400.0
Name:Anirudh R
USN:1BM23CS036
```

Program 4

Abstract Class shape program

Algorithm



```
class Triangle extends Shape {
    double h;
    double b;
    Triangle(double a, double b) {
        h = a;
        this.b = b;
    }
    void printArea() {
        System.out.println("The
        area of triangle is: "+(h*b)/2.0);
    }
}

class Circle extends Shape {
    double r
    Circle(double r) {
        this.r = r;
    }
    void printArea() {
        System.out.println("The area
        of circle is: " + Math.PI* r*r);
    }
}
```

```
class ShapeDemo {
    Ps vm(String args[]) {
        Rectangle r = new Rectangle(2,5)
        Triangle  t = new Triangle(2,5)
        Circle    c = new Circle(5);

        r.printArea();
        t.printArea();
        c.printArea();
    }
}
```

output

Area of rectangle is 10
Area of triangle is 5
Area of circle is 78.539

Code:

```
import java.util.Scanner;
import java.lang.Math;
abstract class Shape  {
        double a;
        double b;
        abstract void printArea();
}
class Rectangle extends Shape{
        double l;
        double br;
  Rectangle(double a, double b){
        l=a;
        br=b;
}

        void printArea(){
                System.out.println("The Area of the rectangle is: "+l*br);
}
}
class Triangle extends Shape{
        double h;
        double b;
        Triangle(double a, double b){
                h=a;
                this.b=b;
        }
        void printArea(){
                System.out.println("The Area of the Rectangle is: "+(h*b)/2.0);
        }
}
class Circle extends Shape{
        double r;
        Circle(double r){
                this.r=r;
        }
        void printArea(){
                System.out.println("The area of the Circle is: "+ Math.PI*r*r);

        }
}
class ShapeDemo{
        public static void main(String args[]){
                Rectangle r = new Rectangle(2,5);
                Triangle t = new Triangle(2,5);
                Circle c = new Circle(5);
```

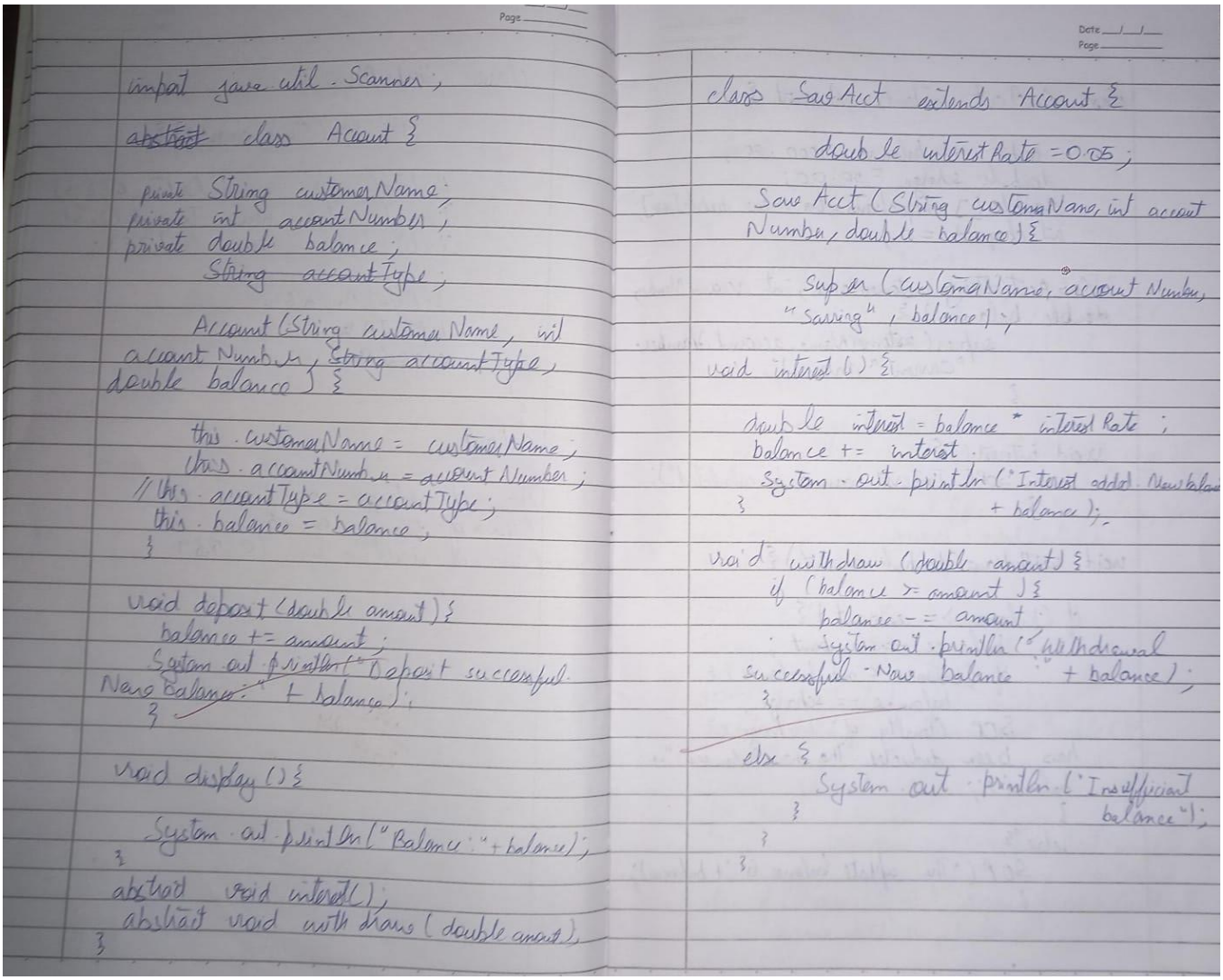
```
                r.printArea();
                t.printArea();
                c.printArea();
                System.out.println("Name:Anirudh R");
                System.out.println("USN:1BM23CS036");
        }
}
```

Output

```
C:\Users\bmsce\Desktop\1BM23CS036>javac ShapeDemo.java

C:\Users\bmsce\Desktop\1BM23CS036>java ShapeDemo
The Area of the rectangle is: 10.0
The Area of the Rectangle is: 5.0
The area of the Circle is: 78.53981633974483
Name:Anirudh R
USN:1BM23CS036
```

Program 5
Bank program

Algorithm



```
import java.util.Scanner;

abstract class Account {

    private String customerName;
    private int accountNumber;
    private double balance;
    String accountType;

    Account(String customerName, int
accountNumber, String accountType,
double balance) {

        this.customerName = customerName;
        this.accountNumber = accountNumber;
        // this.accountType = accountType;
        this.balance = balance;
    }

    void deposit(double amount) {
        balance += amount;
        System.out.println("Deposit successful.
New Balance: " + balance);
    }

    void display() {

        System.out.println("Balance: " + balance);
    }
    abstract void interest();
    abstract void withdraw(double amount);
}
```

```
class SavAcct extends Account {

    double interestRate = 0.05;

    SavAcct(String customerName, int accountNumber, double balance) {

        super(customerName, accountNumber,
        "Saving", balance);

    void interest() {

        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest added. New balance
                                    " + balance);
    }

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

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

```
class CurAcct extends Account {

    double minbalance = 1000.00;
    double charge = 50.00;
    double[] chequetransaction = new double[100];
    int chequid = 0;

    CurAcct(String customerName, int accountNumber,
    double balance) {
        super(customerName, accountNumber,
        "Current", balance);
    }

    void interest() {
        SOP("Interest cannot be calculated");
    }

    void withdraw(double amount) {

        if (balance >= amount) {
            balance -= amount;
            if (balance < minbalance) {
                balance -= charge;
            SOP("Penalty of" + charge + "
            has been deducted. The new balance:" +
            balance);
            }
            else {
                SOP("The updated balance is:" + balance);
            }
```

```
            chequetransaction[chequid++] = amount;
        }
        else {
            SOP("Insufficient balance");
        }
    }
    void displayTransaction() {
        for(int i=0; i< chequid; i++) {
            SOP("Transaction" + (i+1) + ":" +
                chequetransaction[i]); }
    }

public class Bank {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        SOP("Enter account type (1 for savings, 2 for current)
        int choice = scanner.nextInt();
        scanner.nextLine();

        SOP("Enter customer name:");
        String name = scanner.next Int();

        SOP("Enter Initial balance");
        double balance = scanner.nextDouble();

        Account account = null;
        if(choice == 1) {
            account = new SavAcct(name, accNum,
                balance); }

        else if(choice == 2) { account = new CurAcct(
            name, accNum, balance); }
```

```
else {
SOP ("Invalid account type selected.");
return;
}
int exit = 0;
do {
  SOP("Enter the function to be done");
  SOP("1. Deposit 2n2. Display Balance 3n3 Compute
and deposit interest 4n4. Withdraw 5n5. Exit");
int option = scanner.nextInt();
switch (option) {
  case 1: SOP("Enter deposit amount");
  double depositAmount = scanner.nextDouble();
  account.deposit(depositAmount);
  break;
  case 2: account.display();
    if (choice == 2) {
    ((CurrAcct) account).displayTransaction();
    } break;
  case 3: account.interest(); break;
  case 4: System.out.println("Enter withdrawal
  amount:");
  double withdrawAmount = scanner.nextDouble();
  account.withdraw(withdrawAmount);
    break;
  case 5 - System.out.println("Exiting...");
        exit = 1; break;
default: SOP("Invalid"); }
while (exit == 0);
  scanner.close();
}
```

```
Name: Anirudh R
USM : 1BM23CS036
Choose an account type:
1: Savings Account
2. Current Account
1
Choose an Action:
1. Deposit
2. Display balance
3. Compute Interest (Savings only)
4. Withdraw
5. Exit
1
Enter amount to deposit: 1000
Deposit successful! New balance: 2000.0
Choose an action:
1. Deposit
2. Display Balance
3. Compute Interest (Savings only)
4. Withdraw
5. Exit
2
Balance for account number 1001: 2000.0

Choose an action
1. Deposit
2. Display Balance
3. Compute Interest (Savings only)
4: Withdraw
5. Exit
3
Interest computed and added: 80.0
Balance for account number 1001: 2080.0
```

Code:

```
import java.util.Scanner;

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

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

    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposit successful! New balance: " + balance);
    }

    public void displayBalance() {
        System.out.println("Balance for account number " + accountNumber + ": " + balance);
    }
}

class SavAcct extends Account {
    private static final double INTEREST_RATE = 0.04; // Example interest rate of 4%

    public SavAcct(String customerName, int accountNumber, double balance) {
        super(customerName, accountNumber, balance, "Savings");
    }

    public void computeAndDepositInterest() {
        double interest = balance * INTEREST_RATE;
        balance += interest;
        System.out.println("Interest computed and added: " + interest);
        displayBalance();
    }

    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawal successful! New balance: " + balance);
        } else {
            System.out.println("Insufficient balance for withdrawal.");
        }
```

```
    }
}

class CurAcct extends Account {
    private static final double MINIMUM_BALANCE = 500.0;
    private static final double SERVICE_CHARGE = 50.0;

    public CurAcct(String customerName, int accountNumber, double balance) {
        super(customerName, accountNumber, balance, "Current");
    }

    public void checkMinimumBalance() {
        if (balance < MINIMUM_BALANCE) {
            balance -= SERVICE_CHARGE;
            System.out.println("Minimum balance not maintained. Service charge imposed: " +
SERVICE_CHARGE);
            displayBalance();
        }
    }

    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawal successful! New balance: " + balance);
            checkMinimumBalance();
        } else {
            System.out.println("Insufficient balance for withdrawal.");
        }
    }
}

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

        // Creating savings account
        SavAcct savingsAccount = new SavAcct("Alice", 1001, 1000.0);

        // Creating current account
        CurAcct currentAccount = new CurAcct("Bob", 1002, 800.0);

        // Display account options
        System.out.println("Choose an account type:\n1. Savings Account\n2. Current Account");
        int choice = scanner.nextInt();

        Account selectedAccount = (choice == 1) ? savingsAccount : currentAccount;
```

```
        // Menu for actions
        boolean exit = false;
        while (!exit) {
            System.out.println("\nChoose an action:\n1. Deposit\n2. Display Balance\n3. Compute
Interest (Savings only)\n4. Withdraw\n5. Exit");
            int action = scanner.nextInt();
            switch (action) {
                case 1:
                    System.out.print("Enter amount to deposit: ");
                    double depositAmount = scanner.nextDouble();
                    selectedAccount.deposit(depositAmount);
                    break;

                case 2:
                    selectedAccount.displayBalance();
                    break;

                case 3:
                    if (selectedAccount instanceof SavAcct) {
                        ((SavAcct) selectedAccount).computeAndDepositInterest();
                    } else {
                        System.out.println("Interest computation is not applicable for Current Account.");
                    }
                    break;

                case 4:
                    System.out.print("Enter amount to withdraw: ");
                    double withdrawAmount = scanner.nextDouble();
                    if (selectedAccount instanceof SavAcct) {
                        ((SavAcct) selectedAccount).withdraw(withdrawAmount);
                    } else if (selectedAccount instanceof CurAcct) {
                        ((CurAcct) selectedAccount).withdraw(withdrawAmount);
                    }
                    break;

                case 5:
                    exit = true;
                    System.out.println("Thank you for banking with us!");
                    break;

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

        scanner.close();
    }}
```

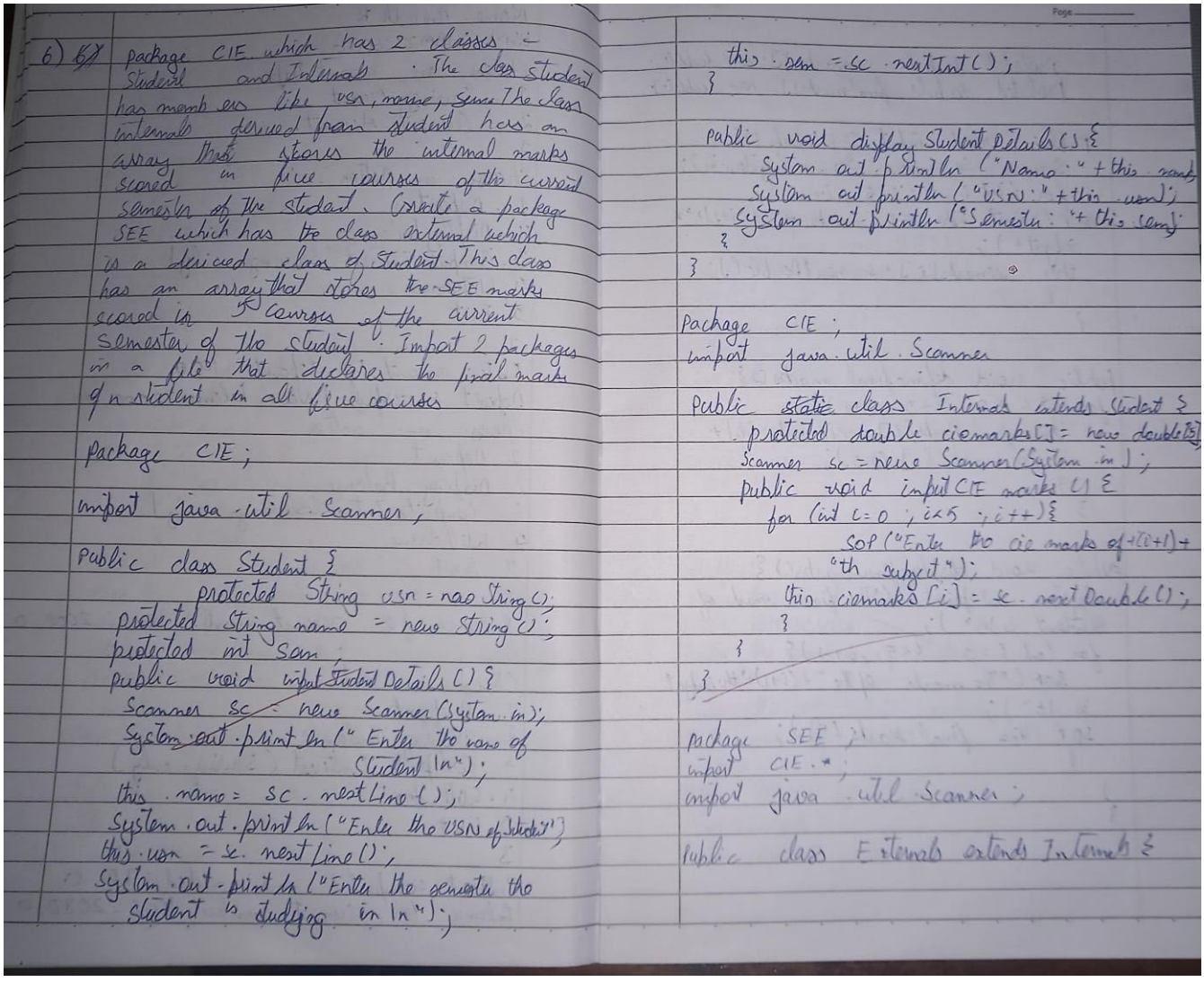
Output:

```
C:\JavaProg>java Bank
Name:Anirudh R
USN:1BM23CS036
Choose an account type:
1. Savings Account
2. Current Account
1

Choose an action:
1. Deposit
2. Display Balance
3. Compute Interest (Savings only)
4. Withdraw
5. Exit
1
Enter amount to deposit: 1000
Deposit successful! New balance: 2000.0

Choose an action:
1. Deposit
2. Display Balance
3. Compute Interest (Savings only)
4. Withdraw
5. Exit
2
Balance for account number 1001: 2000.0

Choose an action:
1. Deposit
2. Display Balance
3. Compute Interest (Savings only)
4. Withdraw
5. Exit
3
Interest computed and added: 80.0
Balance for account number 1001: 2080.0
```

```
Choose an action:
1. Deposit
2. Display Balance
3. Compute Interest (Savings only)
4. Withdraw
5. Exit
4
Enter amount to withdraw: 900
Withdrawal successful! New balance: 1180.0

Choose an action:
1. Deposit
2. Display Balance
3. Compute Interest (Savings only)
4. Withdraw
5. Exit
5
Thank you for banking with us!
```

Program 6
Packages

Algorithm



6) package CIE which has 2 classes - Student and Internals. The class Student has members like usn, name, sem. The class internals derived from Student has an array that stores the internal marks scored in five courses of the current semester of the student. Create a package SEE which has the class external which is a derived class of Student. This class has an array that stores the SEE marks scored in 5 courses of the current semester of the student. Import 2 packages in a file that declares the final marks of n student in all five courses

```
package CIE;

import java.util.Scanner;

Public class Student {
        protected String usn = nao String();
    protected String name = new String();
    protected int sem;
    public void inputStudentDetails() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the name of
                            Student\n");
        this.name = sc.nextLine();
        System.out.println("Enter the USN of Student");
        this.usn = sc.nextLine();
        System.out.println("Enter the semester the
          student is studying in\n");
        this.sem = sc.nextInt();
    }

    public void displayStudentDetails() {
        System.out.println("Name:" + this.name);
        System.out.println("USN:" + this.usn);
        System.out.println("Semester:" + this.sem);
    }
}

Package CIE;
import java.util.Scanner

Public static class Internals extends Student {
    protected double ciemarks[] = new double[5];
    Scanner sc = new Scanner(System.in);
    public void inputCIEmarks() {
        for (int i=0; i<5; i++) {
            SOP("Enter the cie marks of"+(i+1)+
                "th subject");
            this.ciemarks[i] = sc.nextDouble();
        }
    }
}

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

Public class Externals extends Internals {
```

```
protected double seemarks[] = new double[5];
protected double finalmarks[] = new double[5];

public void input SEE marks () {
  Scanner sc = new Scanner (System.in);
  for (int i = 0; i<5 ; i++) {
SOP ("Enter the SEE marks of " + (i+1) + "th subject");
this.seemarks[i] = sc.nextDouble();
  }
}

public void definefinal marks () {
for (int i = 0; i<5 ; i++) {
  this.finalMarks[i] = ciemarks[i] +
  (seemarks[i] /2.0);
  }
}

public void displayfinalmarks() {
System.out.println("The final marks of
student is \n");
for (int i = 0; i<5 ; i++) {
  SOP ("The marks of the " + (i+1) + "th subject
  is ");
  SOP (this.finalMarks[i]);
  }
}
}
```

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

class Main {
    PSVm (String args[]) {
      int n;
      Scanner sc = new Scanner(System.in);
SOP("Name : AnirudhR");
SOP(" USN : 1BM23CS036");
SOP("Enter the number of Students");
n = sc.nextInt();
Externals e[] = new Externals[n];
for(int i=0; i<n;i++){
e[i] = new Externals();
SOP("Enter the " + (i+1) + "th student details \n");
e[i].inputStudentDetails();
  }
}
}
```

Output

```
Name : Anirudh R
USM : 1BM23CS036
Enter the number of students
1
Enter the 1st Student details
Enter name of student
Anoop
Enter the USN of student
1bm23cs042
Enter semester 3
Enter the CIE marks
20
30
40
45
50
Enter SEE marks
50
60
70
20
90
The details are
Name : Anoop
USM : 1bm23cs042
Semester : 3
The marks of student is
45.0
60.0
75.0
85.0
90.0
```

Code:

```
package CIE;
import java.util.Scanner;

public class Student {
		protected String usn = new String();
		protected String name = new String();
		protected int sem;

		public void inputStudentDetails(){
			Scanner sc = new Scanner(System.in);
			System.out.println("Enter the name of the student \n");
			this.name=sc.nextLine();
			System.out.println("Enter the USN of the student \n");
			this.usn=sc.nextLine();
			System.out.println("Enter the semester the student is studying in \n");
			this.sem=sc.nextInt();
		}

		public void displayStudentDetails(){
			System.out.println("Name: " + this.name);
			System.out.println("USN: " +this.usn);
			System.out.println("Semester: " +this.sem);
		}
}

package CIE;
import java.util.Scanner;

public class Internals extends Student {
		protected double ciemarks[] = new double[5];
		Scanner sc= new Scanner(System.in);
		public void inputCIEmarks(){
			for(int i=0; i<5; i++){
				System.out.println("Enter the CIE marks of"+ (i+1)+"th subject");
				this.ciemarks[i]= sc.nextDouble();
			}
		}
}

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

public class Externals extends Internals{
				protected double seemarks[] = new double[5];
				protected double finalMarks[] = new double[5] ;
```

```
                public void inputSEEmarks() {
                        Scanner sc = new Scanner(System.in);
                        for(int i=0; i<5; i++){
                                System.out.println("Enter the SEE marks of"+ (i+1)+"th
subject");
                                this.seemarks[i]= sc.nextDouble();
                        }
                }

                public void definefinalmarks(){
                        for(int i=0; i<5; i++){
                                this.finalMarks[i] = ciemarks[i] + (seemarks[i]/2.0);
                        }

                }

                public void displayfinalmarks(){
                        System.out.println("The final marks of the student is \n");
                        for(int i=0; i<5; i++){
                                System.out.println("The marks of the" + (i+1)+"th
subject is \t");
                                System.out.println(this.finalMarks[i]);
                        }

                }
}

import SEE.Externals;
import java.util.Scanner;

class Main{
        public static void main(String args[]){
                int n;
                Scanner sc = new Scanner(System.in);
                System.out.println("Name: Anirudh R");
                System.out.println("USN: 1BM23CS036");
                System.out.println("Enter the number of students \t");
                n= sc.nextInt();
                Externals e[] = new Externals[n];
                for(int i=0; i<n; i++){
                        e[i] = new Externals();
                        System.out.println("Enter the" +(i+1)+"th student details \n");
                        e[i].inputStudentDetails();
                        System.out.println("Enter the" +(i+1)+"th student's CIE marks\n");
                        e[i].inputCIEmarks();
                        System.out.println("Enter the" +(i+1)+"th student's SEE marks\n");
```

```
                e[i].inputSEEmarks();
                System.out.println("The details of the" +(i+1)+"th student is");
                e[i].displayStudentDetails();
                e[i].definefinalmarks();
                System.out.println("The final marks of the" +(i+1)+"th student is");
                e[i].displayfinalmarks();
            }

        }
}
```

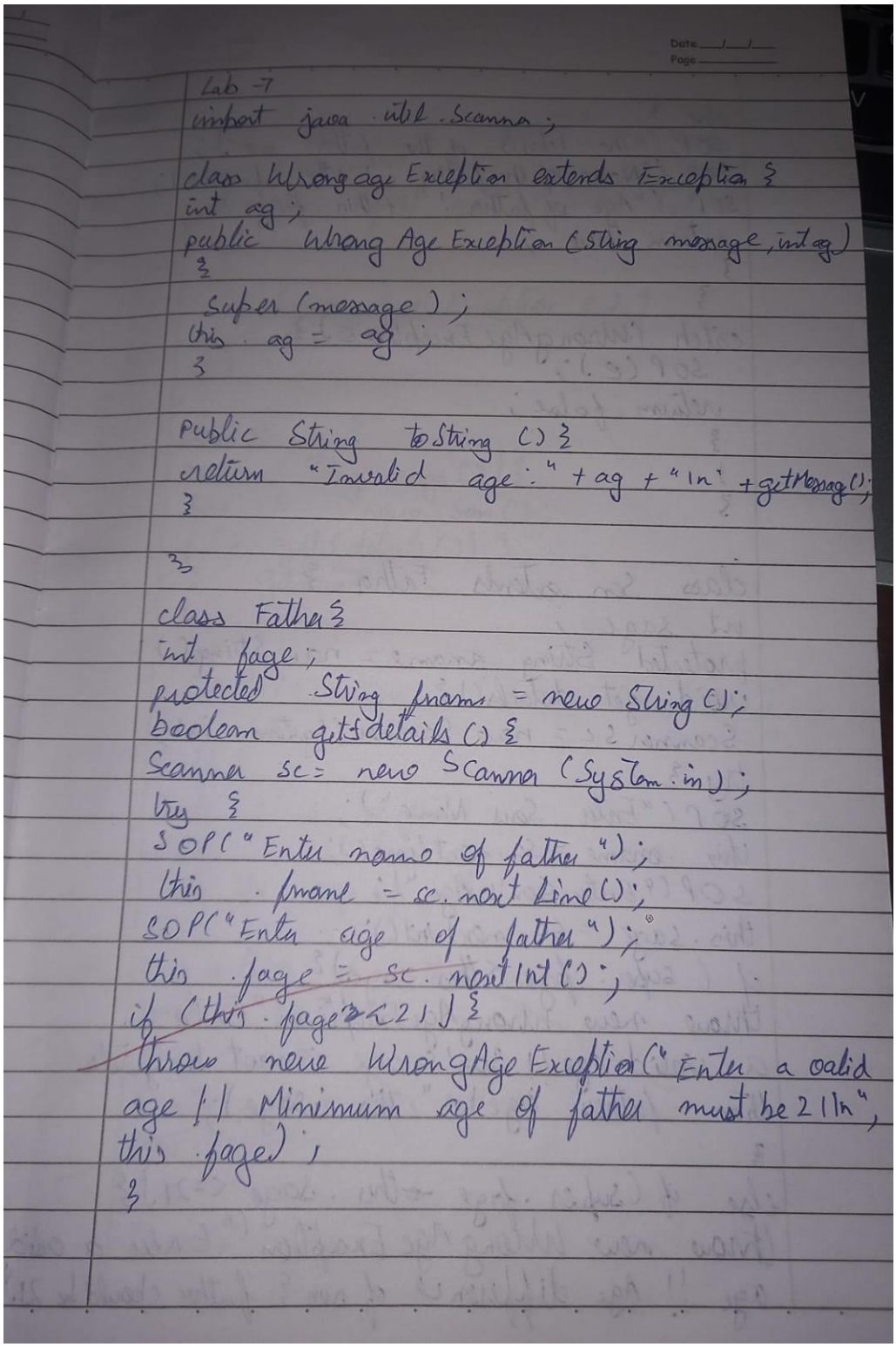
```
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>D:

D:\>cd 1bm23cs036

D:\1bm23cs036>javac Main.java

D:\1bm23cs036>java Main
Name: Anirudh R
USN: 1BM23CS036
Enter the number of students
2
Enter the1th student details

Enter the name of the student

anoop
Enter the USN of the student

1bm23cs042
Enter the semester the student is studying in

3
Enter the1th student's CIE marks

Enter the CIE marks of1th subject
20
Enter the CIE marks of2th subject
30
Enter the CIE marks of3th subject
40
Enter the CIE marks of4th subject
45
Enter the CIE marks of5th subject
50
Enter the1th student's SEE marks

Enter the SEE marks of1th subject
50
Enter the SEE marks of2th subject
60
Enter the SEE marks of3th subject
70
Enter the SEE marks of4th subject
80
Enter the SEE marks of5th subject
90
```

```
The details of the1th student is
Name: anoop
USN: 1bm23cs042
Semester: 3
The final marks of the1th student is
The final marks of the student is

The marks of the1th subject is
45.0
The marks of the2th subject is
60.0
The marks of the3th subject is
75.0
The marks of the4th subject is
85.0
The marks of the5th subject is
95.0
Enter the2th student details

Enter the name of the student

akul
Enter the USN of the student

1bm23cs023
Enter the semester the student is studying in

3
Enter the2th student's CIE marks

Enter the CIE marks of1th subject
20
Enter the CIE marks of2th subject
25
Enter the CIE marks of3th subject
30
Enter the CIE marks of4th subject
35
Enter the CIE marks of5th subject
40
Enter the2th student's SEE marks

Enter the SEE marks of1th subject
50
Enter the SEE marks of2th subject
60
Enter the SEE marks of3th subject
70
Enter the SEE marks of4th subject
80
```

```
Enter the SEE marks of5th subject
90
The details of the2th student is
Name: akul
USN: 1bm23cs023
Semester: 3
The final marks of the2th student is
The final marks of the student is

The marks of the1th subject is
45.0
The marks of the2th subject is
55.0
The marks of the3th subject is
65.0
The marks of the4th subject is
75.0
The marks of the5th subject is
85.0
```

Program 7
Exception handling

Algorithm



```
Lab -7
import java.util.Scanner;

class WrongAge Exception extends Exception {
int ag;
public WrongAge Exception (String message, int ag)
{
 super (message);
this.ag = ag;
}

Public String toString () {
return "Invalid age:" + ag + "\n" + getMessage();
}

}

class Father {
int fage;
protected String fname = new String();
boolean getDetails () {
Scanner sc = new Scanner (System.in);
try {
SOP("Enter name of father");
this.fname = sc.nextLine();
SOP("Enter age of father");
this.fage = sc.nextInt();
if (this.fage < 21) {
throw new WrongAge Exception("Enter a valid
age || Minimum age of father must be 21\n",
this.fage);
}
```

```
else {
SOP ("The details of the father are");
SOP ("Name of the father : " + this.fname);
SOP ("Age of father : " + this.fage);
return true;
}
}
catch (WrongAgeException e) {
SOP (e);
return false;
}
}
}

class Son extends Fatha {
int sage;
protected String sname = new String();
void getdetails() {
Scanner sc = new Scanner (System.in);
try {
SOP ("Enter Sons Name");
this.sname = sc.nextLine();
SOP ("Enter Son's Age");
this.sage = sc.nextInt();
if (super.fage < this.sage) {
throw new WrongAgeException ("Enter a valid age !! Son's age cannot be greater than fathers age in", this.sage);
}
else if (super.fage - this.sage <= 21) {
throw new WrongAgeException ("Enter a valid age !! Age difference of son & father should be 21");
```

```
else {
SOP ("The details of the Son are");
SOP ("The Name is " + this.sname);
SOP ("Age of the Son: " + this.sage);
}
}
catch (WrongAgeException e) {
SOP (e)
}}}

class Driver {
psvm (String args[]) {
Son s = new Son();
if (s.getdetails()) {
s.getdetails();
}
else {
SOP ("Fathers details are Invalid");

Output
Enter Name of father
ABC
Enter the age of father
32
The details of father are
Name : ABC
Age : 32
Enter name of SON
DEF
Age 29
Invalid age.
```

Seen
Sol

Code:

```
import java.util.Scanner;

class WrongAgeExcpetion extends Exception{
int ag;
public WrongAgeExcpetion(String message, int ag){
super(message);
this.ag = ag;
}

@Override
public String toString(){
return "Invalid age: "+ ag +"\n"+getMessage();
}

}

class Father{
int fage;
protected String fname = new String();
boolean getfdetails(){
Scanner sc = new Scanner(System.in);
try{
System.out.println("Enter the name of the father");
this.fname = sc.nextLine();
System.out.println("Enter the age of the father");
this.fage = sc.nextInt();
if(this.fage<21){
throw new WrongAgeExcpetion("Enter a valid age!! Minimum age of father must be 21 \n",
this.fage);
}
else{
System.out.println("The details of the father are");
System.out.println("Name of the father: "+this.fname);
System.out.println("Age of the father: "+this.fage);
return true;
}
  }
catch (WrongAgeExcpetion e){
System.out.println(e);
return false;
}
}
}

class Son extends Father{
int sage;
```

```
protected String sname = new String();
void getsdetails(){
Scanner sc = new Scanner(System.in);
try{
System.out.println("Enter the name of the son");
this.sname = sc.nextLine();
System.out.println("Enter the age of the son");
this.sage = sc.nextInt();
if(super.fage<this.sage){
throw new WrongAgeExcpetion("Enter a valid age!! Son's age cannot be greater than father's age\n",
this.sage);
}
else if(super.fage-this.sage<=21){
throw new WrongAgeExcpetion("Enter a valid age!! Age difference of son and father must be atleast
21 years!! \n", this.sage);

}
else{
System.out.println("The details of the son are");
System.out.println("Name of the son: "+this.sname);
System.out.println("Age of the son: "+this.sage);
}

  }
catch (WrongAgeExcpetion e){
System.out.println(e);
}
}

}


class Driver{
public static void main(String args[]){
Son s = new Son();
if(s.getfdetails()){
s.getsdetails();
}
else{
System.out.println("Father's details are invalid; so you cannot enter son's details \n");
}
}
}
```

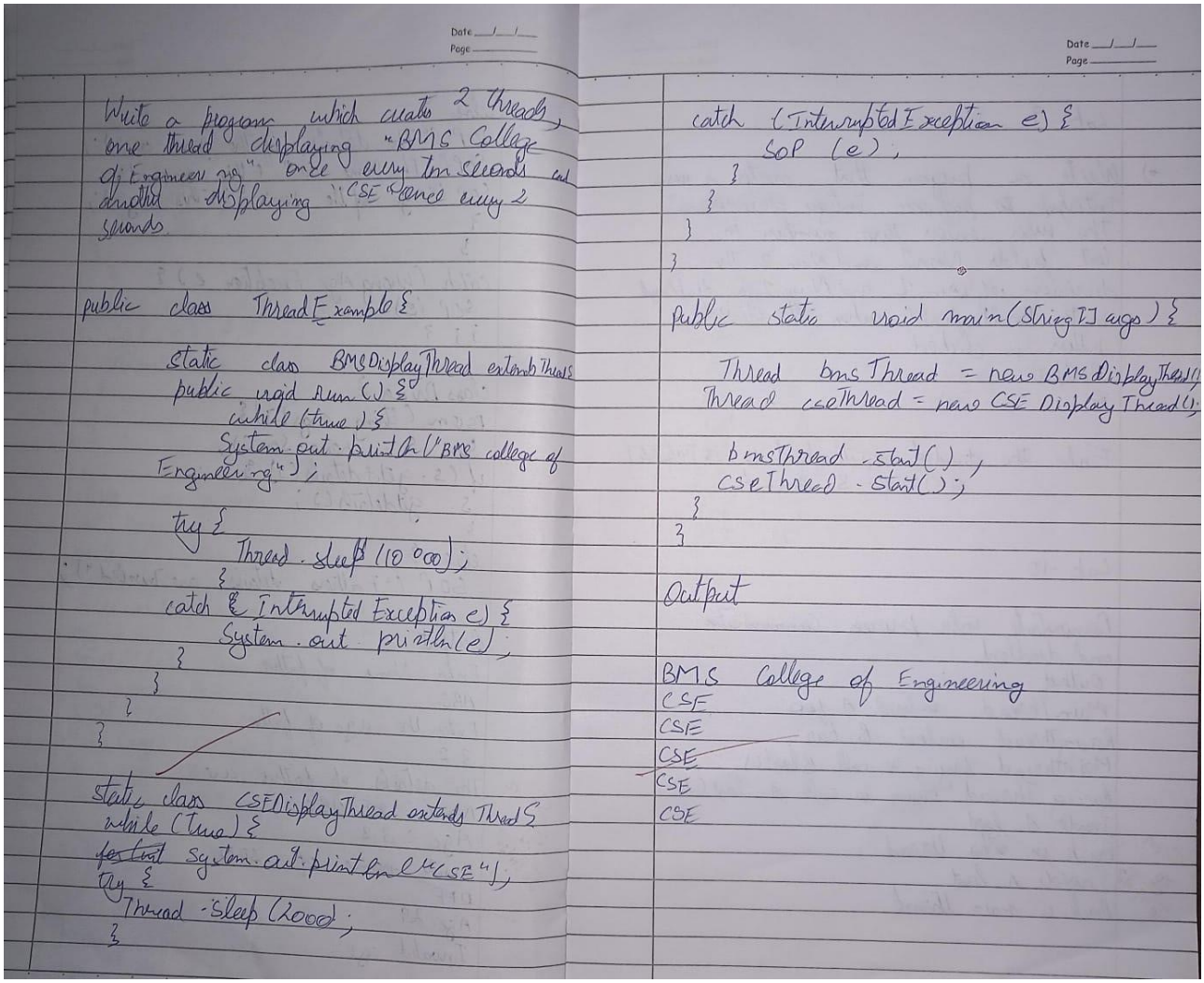
Output

```
C:\Anirudh\programs\JavaProg>javac Driver.java

C:\Anirudh\programs\JavaProg>java Driver
Name:Anirudh R
USN:1BM23CS036
Enter the name of the father
abc
Enter the age of the father
25
The details of the father are
Name of the father: abc
Age of the father: 25
Enter the name of the son
def
Enter the age of the son
30
Invalid age: 30
Enter a valid age!! Son's age cannot be greater than father's age


C:\Anirudh\programs\JavaProg>java Driver
Name:Anirudh R
USN:1BM23CS036
Enter the name of the father
abc
Enter the age of the father
29
The details of the father are
Name of the father: abc
Age of the father: 29
Enter the name of the son
def
Enter the age of the son
22
Invalid age: 22
Enter a valid age!! Age difference of son and father must be atleast 21 years!!
```

Program 8

Multithreading

Algorithm



Write a program which creates 2 threads, one thread displaying "BMS College of Engineering" once every ten seconds and another displaying "CSE" once every 2 seconds.

```
public class ThreadExample {

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

    static class CSEDisplayThread extends Thread {
        while (true) {
            System.out.println("CSE");
            try {
                Thread.sleep(2000);
            } catch (InterruptedException e) {
                SOP(e);
            }
        }
    }

    public static void main(String[] args) {
        Thread bmsThread = new BMSDisplayThread();
        Thread cseThread = new CSEDisplayThread();
        bmsThread.start();
        cseThread.start();
    }
}
```

Output

```
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
```

Code:

```
public class ThreadExample {

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

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

    public static void main(String[] args) {
        // Create two threads
        Thread bmsThread = new BMSDisplayThread();
        Thread cseThread = new CSEDisplayThread();

        bmsThread.start();
        cseThread.start();
    }
}
```

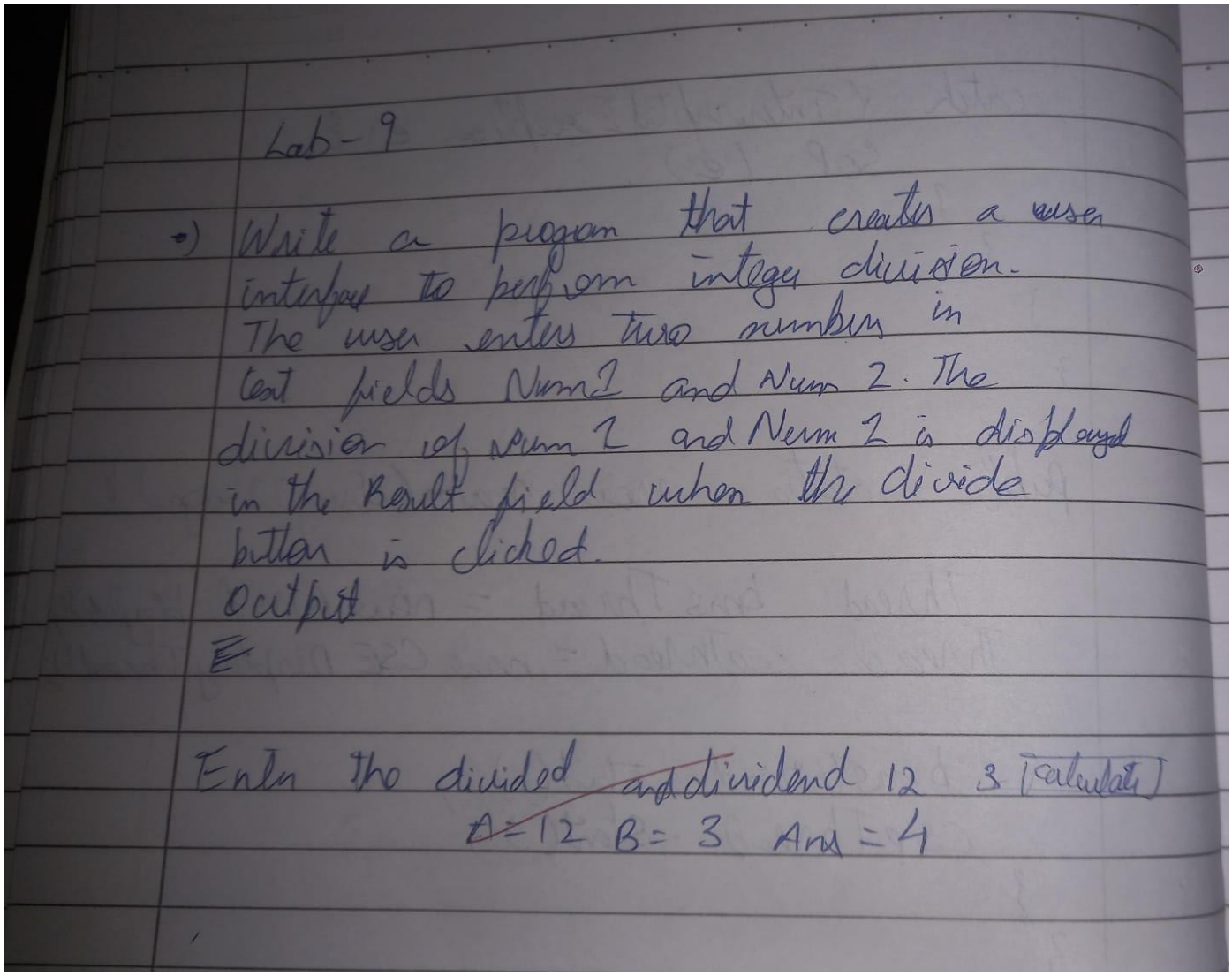
Output

```
Name:Anirudh
USN:1BM23CS036
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
```

Program 9

Integer division with user interface

Algorithm

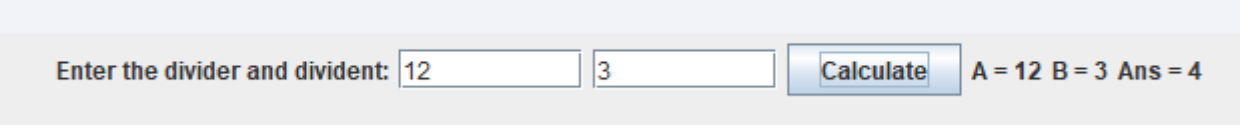


Code:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class SwingDemo{
 SwingDemo(){
 // create jframe container
 JFrame jfrm = new JFrame("Divider App");
 jfrm.setSize(275, 150);
 jfrm.setLayout(new FlowLayout());
 // to terminate on close
 jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
 // text label
 JLabel jlab = new JLabel("Enter the divider and divident:");
 // add text field for both numbers
 JTextField ajtf = new JTextField(8);
 JTextField bjtf = new JTextField(8);
 // calc button
 JButton button = new JButton("Calculate");
 // labels
 JLabel err = new JLabel();
 JLabel alab = new JLabel();
 JLabel blab = new JLabel();
 JLabel anslab = new JLabel();
 // add in order :)
 jfrm.add(err); // to display error bois
 jfrm.add(jlab);
 jfrm.add(ajtf);
 jfrm.add(bjtf);
 jfrm.add(button);
 jfrm.add(alab);
 jfrm.add(blab);
 jfrm.add(anslab);

 ActionListener l = new ActionListener() {
 public void actionPerformed(ActionEvent evt) {
 System.out.println("Action event from a text field");
 }
 };
 ajtf.addActionListener(l);
 bjtf.addActionListener(l);
 button.addActionListener(new ActionListener() {
 public void actionPerformed(ActionEvent evt) {
 try{
 int a = Integer.parseInt(ajtf.getText());
 int b = Integer.parseInt(bjtf.getText());
 int ans = a/b;
```

```
alab.setText("\nA = " + a);
blab.setText("\nB = " + b);
anslab.setText("\nAns = "+ ans);
}
catch(NumberFormatException e){
alab.setText("");
blab.setText("");
anslab.setText("");
err.setText("Enter Only Integers!");
}
catch(ArithmeticException e){
alab.setText("");
blab.setText("");
anslab.setText("");
err.setText("B should be NON zero!");
}
}
});
// display frame
jfrm.setVisible(true);
}
public static void main(String args[]){
// create frame on event dispatching thread
SwingUtilities.invokeLater(new Runnable(){
public void run(){
new SwingDemo();
}
});
}
}
```

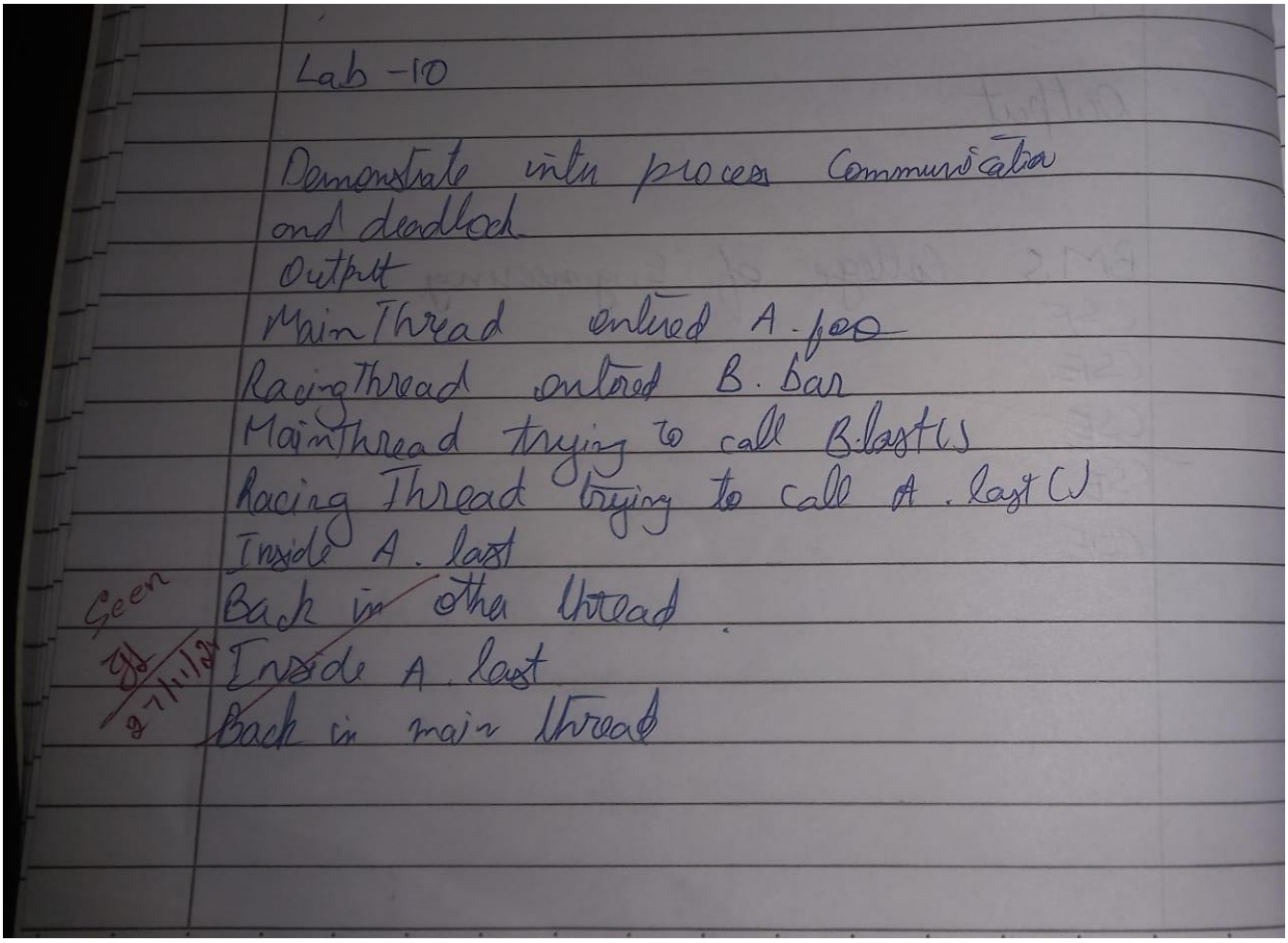
Output

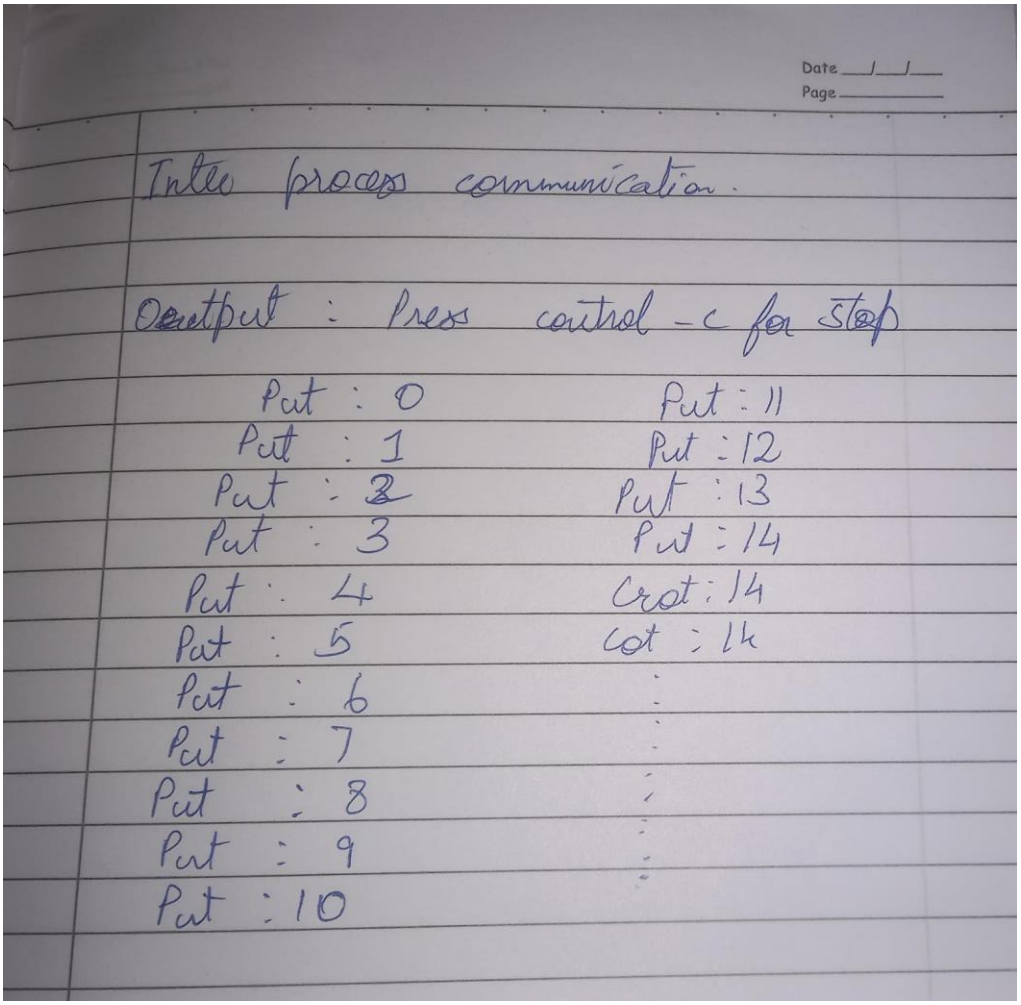


Program 10

Inter process communication and deadlock

Algorithm




Date __/__/__
Page _____

Inter process communication.

Output : Press control - c for stop

Put : 0
Put : 1
Put : 2
Put : 3
Put : 4
Put : 5
Put : 6
Put : 7
Put : 8
Put : 9
Put : 10
Put : 11
Put : 12
Put : 13
Put : 14
Cut : 14
Cut : 14
.
.
.
.
.
.
.

Code:

```
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[]) {
System.out.println("Name:Anirudh");
System.out.println("USN:1BM23CS036");
Q q = new Q();

new Producer(q);

new Consumer(q);

System.out.println("Press Control-C to stop.");

}

}

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();

}

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();
```

```
}

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();

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[]) {

System.out.println("Name:Anirudh");
System.out.println("USN:1BM23CS036");

new Deadlock();

}}
```

Output

```
C:\Users\Admin\Desktop\1bm23cs042>java Deadlock
Name:Anirudh
USN:1BM23CS036
MainThread entered A.foo
RacingThread entered B.bar
MainThread trying to call B.last()
RacingThread trying to call A.last()
Inside A.last
Back in other thread
Inside A.last
Back in main thread
```

```
Name:Anirudh
USN:1BM23CS036
Press Control-C to stop.
Put: 0

Intimate Consumer


Producer waiting

Got: 0

Intimate Producer

Put: 1

Intimate Consumer


Producer waiting

consumed:0
Got: 1

Intimate Producer

consumed:1
Put: 2

Intimate Consumer


Producer waiting

Got: 2
```

```
Intimate Producer

consumed:2
Put: 3

Intimate Consumer


Producer waiting

Got: 3

Intimate Producer

consumed:3
Put: 4

Intimate Consumer


Producer waiting

Got: 4

Intimate Producer

consumed:4
Put: 5

Intimate Consumer


Producer waiting

Got: 5

Intimate Producer

consumed:5
Put: 6
```