VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Arnav Dinesh (1BM23CS052)

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 Arnav Dinesh **(1BM23CS052),** 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.

Lab faculty Incharge Name - Dr.Prasad G R Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE

Index

Github Link: https://github.com/ArnavRD/Javalab

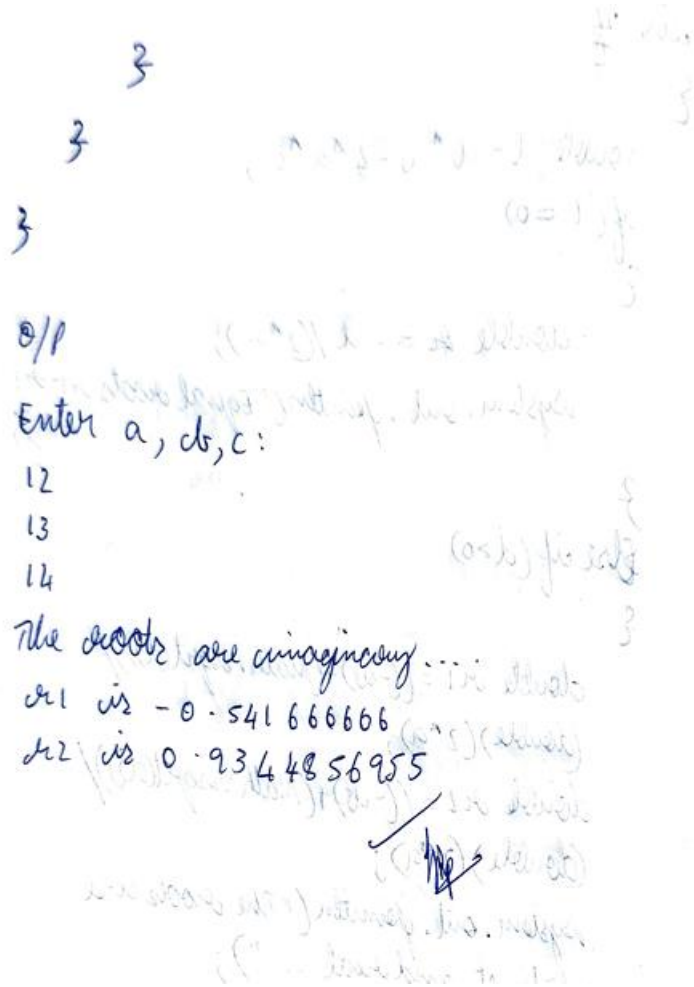
Week 1

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

```
import java.util.Scanner;
import java.lang.Math;
class quad {
    public static void main(String[] args)
    {
        double a,b,c;
        Scanner s= new Scanner(System.in);
        System.out.println("Enter a,b,c:");
        a= s.nextDouble();
        b= s.nextDouble();
        c= s.nextDouble();
        if(a=0)
        {
            System.out.println("Not quadratic");
        }
```

```
else if
{
    double d = b*b - 4*a*c;
    if(d==0)
    {
        double sr = -b/(2*a);
        System.out.println("Equal roots sr1=sr2"
                                        +sr);
    }
    else if(d>0)
    {
        double sr1 = ((-b)+(Math.sqrt(d)))/
        (double)(2*a);
        double sr2 = ((-b)+(Math.sqrt(d)))/
        (double)(2*a);
        System.out.println("The roots are
        distinct and real....");
        System.out.println("sr1 is" +sr1);
        System.out.println("sr2 is" +sr2);
    }
    else if(d<0)
    {
        double sr1 = (-b)/(2*a);
        double sr2 = Math.sqrt(-d)/(2*a);
        System.out.println("The roots are imaginary
        System.out.println("sr1 is" +sr1);
        System.out.println("sr2 is" +sr2);
    }
```

Output in observation book



Source code

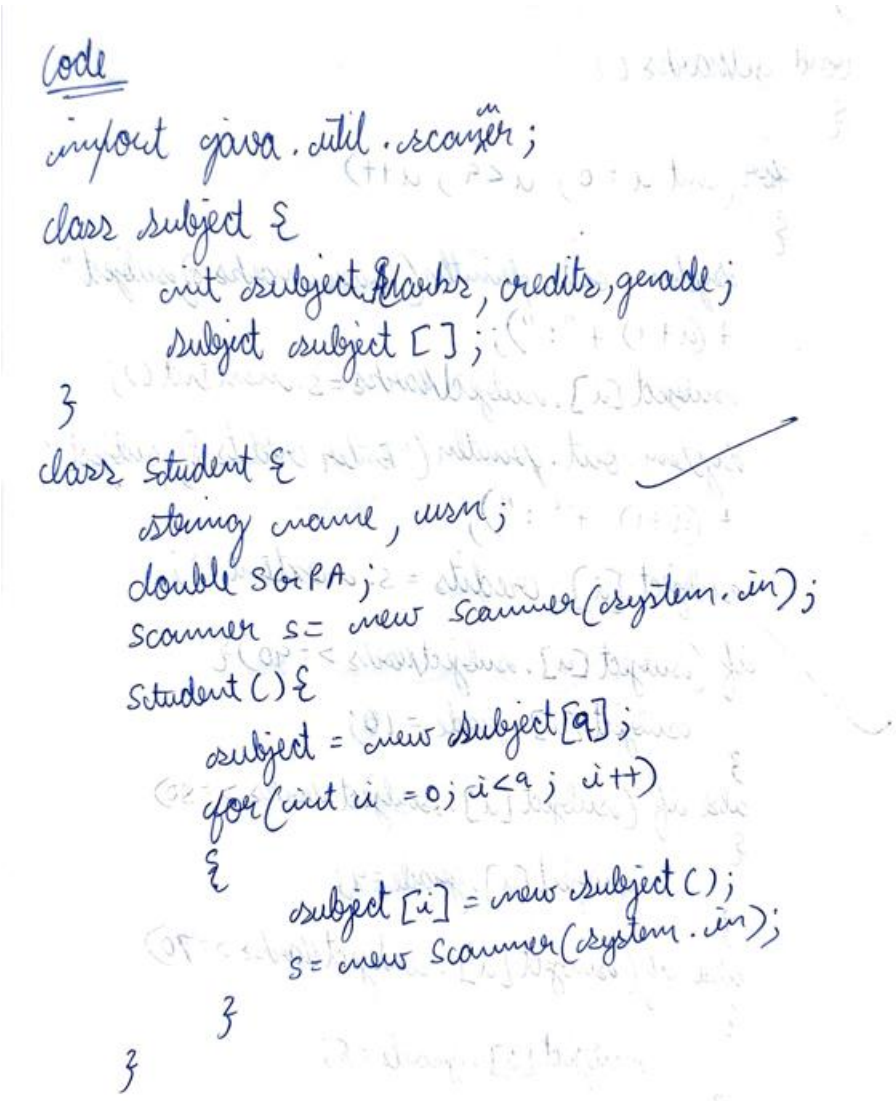
```
import java.util.Scanner;
import java.lang.Math;
class quad{
	public static void main(String[] args)
	{
		double a,b,c;
		Scanner s = new Scanner(System.in);
		System.out.println("Enter a,b,c:");
		a = s.nextDouble();
		b = s.nextDouble();
		c = s.nextDouble();
		if(a==0)
		{
			System.out.println("Not quadratic");
		}
		else
		{
			double d = b*b-4*a*c;
			if(d==0)
			{
				double r=-b/(2*a);
				System.out.println("Equal roots r1=r2=" + r);
			}
			else if(d>0)
			{
				double r1 = ((-b)+(Math.sqrt(d)))/(double)(2*a);
				double r2 = ((-b)-(Math.sqrt(d)))/(double)(2*a);
				System.out.println("The roots are distinct and real...");
				System.out.println("r1 is" + r1);
				System.out.println("r2 is" + r2);
			}
			else if(d<0)
			{
				double r1 = (-b)/(2*a);
				double r2 = Math.sqrt(-d)/(2*a);
				System.out.println("The roots are imaginary...");
				System.out.println("r1 is" + r1);
				System.out.println("r2 is" + r2);
			}


		}

	}

	System.out.println("Arnav Dinesh 1BM23CS052");
}
```

Output

```
Enter coefficients a, b, c:
12
12
12
The roots are imaginary.
r1 = -0.5 + 0.8660254037844387i
r2 = -0.5 - 0.8660254037844387i
Arnav Dinesh 1BM23CS052
```

Week 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.



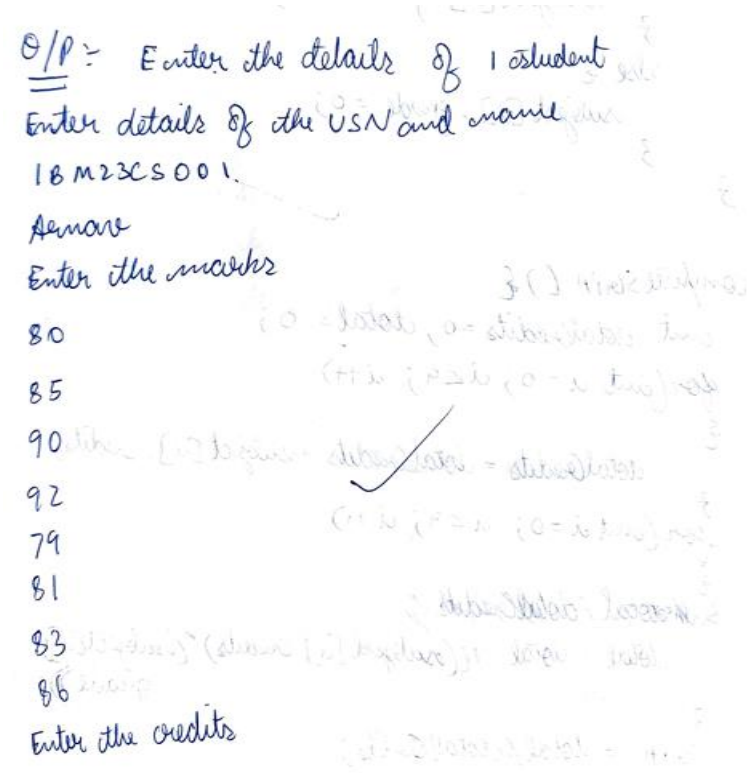
Code

```
import java.util.Scanner;
class subject {
    int subject_marks, credits, grade;
    subject subject[];
}
class Student {
    String name, usn;
    double SGPA;
    Scanner s= new Scanner(System.in);
    Student(){
        subject = new subject[a];
        for(int i =0; i<a; i++)
        {
            subject[i] = new subject();
            s= new Scanner(System.in);
        }
}
```

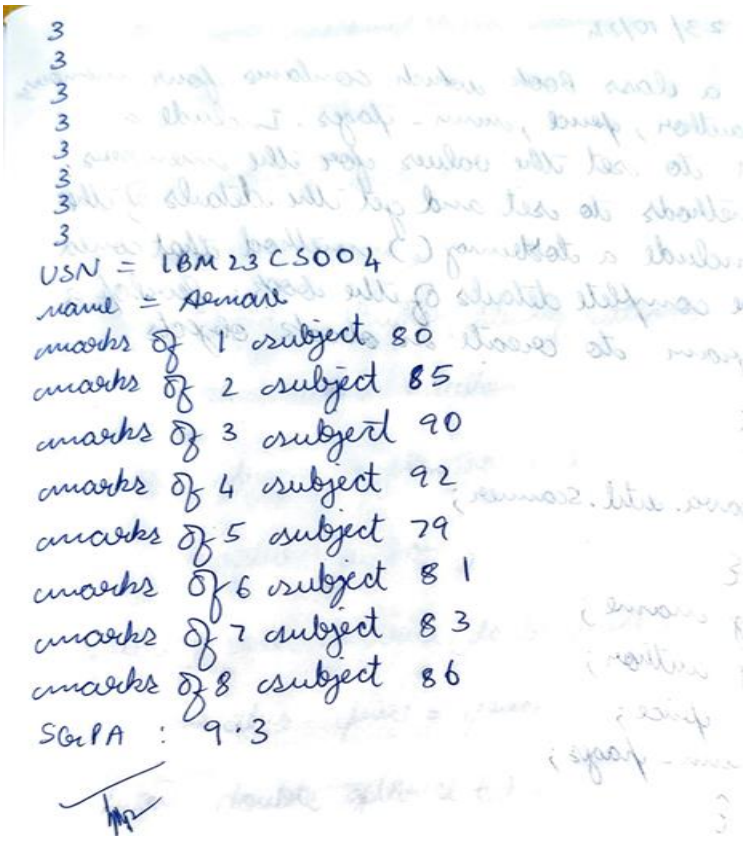
```
void getStudentDetails()
{
    System.out.println("Enter your name:");
    name = s.next();
    System.out.println("Enter your usn:");
    usn = s.next();
}
void getMarks()
{
    for(int i = 0; i<9; i++)
    {
        System.out.println("Enter marks of subject"
        +(i+1)+":");
        subject[i].subjectMarks = s.nextInt();
        System.out.println("Enter credits of subject"
        +(i+1)+":");
        subject[i].credits = s.nextInt();
        if(subject[i].subjectMarks >=90){
            subject[i].grade = 10;
        }
        else if(subject[i].subjectMarks >=80)
        {
            subject[i].grade=9;
        }
        else if(subject[i].subjectMarks >=90)
        {
            subject[i].grade=8;
        }
```

```
        else if(subject[i].subjectMarks >= 60)
        {
            subject[i].grade = 7;
        }
        else if(subject[i].subjectMarks >= 50)
        {
            subject[i].grade = 6;
        }
        else if(subject[i].subjectMarks >= 40)
        {
            subject[i].grade = 5;
        }
        else {
            subject[i].grade = 0;
        }
    }
}
void computeSGPA(){
    int totalCredits=0, total= 0;
    for(int i = 0; i<9; i++)
    {
        totalCredits = totalCredits + subject[i].credits;
    }
    for(int i=0; i<9; i++)
    {
        total = total +((subject[i].credits)*(subject[i].
                                              grade));
    }
    SGPA = total/totalCredits;
}

public static void main(String[] args)
    {
        Student S1 = new Student();
        S1.getStudentDetails();
        S1.getMarks();
        S1.computeSGPA();
    }
}
```

Output



O/P :- Enter the details of 1 student
Enter details of the USN and name
1BM23CS001.
Aenave
Enter the marks
80
85
90
92
79
81
83
86
Enter the credits



3
3
3
3
3
3
3
3
USN = 1BM23CS004
name = Aenave
marks of 1 subject 80
marks of 2 subject 85
marks of 3 subject 90
marks of 4 subject 92
marks of 5 subject 79
marks of 6 subject 81
marks of 7 subject 83
marks of 8 subject 86
SGPA : 9.3

Source code

```
import java.util.*;
class Stud_det
{ int m[]=new int[8];
 int c[]=new int[8];
 int p[]=new int[8];
 int g,sum;
 String name,usn;
 double sgpa;
  Scanner s=new Scanner(System.in);
 void getdetails()
 {
  System.out.println("Enter name:");
  name=s.next();
  System.out.println("Enter usn:");
  usn=s.next();
  for(int i=0;i<8;i++)
  {
   System.out.println("Enter marks of subject:"+(i+1));
   m[i]=s.nextInt();
   System.out.println("Enter credits for subject:" +(i+1));
   c[i]=s.nextInt();
  }
 }
 void gradepoint()
 {
  for(int i=0;i<8;i++)
  {
   if (m[i]>=90 && m[i]<=100)
    p[i]=10;
   else if (m[i]>=80 && m[i]<90)
    p[i]=9;
   else if (m[i]>=70 && m[i]<80)
    p[i]=8;
   else if (m[i]>=60 && m[i]<70)
    p[i]=7;
   else if (m[i]>=50 && m[i]<60)
    p[i]=6;
   else if (m[i]>=40 && m[i]<50)
    p[i]=5;
   else
    p[i]=0;
  }
 }
 void calculate()
 {
  for(int i=0;i<8;i++)
```

```
  {
   g+=c[i]*p[i];
  }
  for(int i=0;i<8;i++)
  {
   sum+=c[i];
  }
  sgpa=g/sum;
 }
 void display()
 {
  System.out.println("Name:"+name);
  System.out.println("USN:"+usn);
  System.out.println("SGPA=:"+sgpa);
 }
}
public class student
{
 public static void main(String a[]){
  Stud_det s1[]=new Stud_det[3];
for(int i=0;i<3;i++){
s1[i]=new Stud_det();


}

  for(int i=0;i<3;i++)
  {
   System.out.println("Enter details of student:"+(i+1));
   s1[i].getdetails();
  }
  for(int i=0;i<3;i++)
  {
   s1[i].gradepoint();
   s1[i].calculate();
  }
  for(int i=0;i<3;i++)
  {
   System.out.println("Student:"+(i+1));
   s1[i].display();
System.out.println ("Arnav Dinesh 1BM23CS052");
  }
 }
```

Output

```
Enter details of student:1
Enter name:
Arnav
Enter usn:
1BM23CS001
Enter marks of subject:1
87
Enter credits for subject:1
3
Enter marks of subject:2
93
Enter credits for subject:2
3
Enter marks of subject:3
89
Enter credits for subject:3
3
Enter marks of subject:4
78
Enter credits for subject:4
3
Enter marks of subject:5
98
Enter credits for subject:5
3
Enter marks of subject:6
87
Enter credits for subject:6
3
Enter marks of subject:7
76
Enter credits for subject:7
3
Enter marks of subject:8
89
Enter credits for subject:8
3
```

```
Enter details of student:2
Enter name:
Ariz
Enter usn:
1BM23CS002
Enter marks of subject:1
98
Enter credits for subject:1
3
Enter marks of subject:2
98
Enter credits for subject:2
3
Enter marks of subject:3
98
Enter credits for subject:3
3
Enter marks of subject:4
98
Enter credits for subject:4
3
Enter marks of subject:5
98
Enter credits for subject:5
3
Enter marks of subject:6
98
Enter credits for subject:6
3
Enter marks of subject:7
98
Enter credits for subject:7
3
Enter marks of subject:8
98
Enter credits for subject:8
3
```

```
Enter details of student:3
Enter name:
Vatsal
Enter usn:
1BM23CS003
Enter marks of subject:1
76
Enter credits for subject:1
3
Enter marks of subject:2
89
Enter credits for subject:2
3
Enter marks of subject:3
87
Enter credits for subject:3
3
Enter marks of subject:4
85
Enter credits for subject:4
3
Enter marks of subject:5
76
Enter credits for subject:5
3
Enter marks of subject:6
56
Enter credits for subject:6
3
Enter marks of subject:7
76
Enter credits for subject:7
3
Enter marks of subject:8
76
Enter credits for subject:8
3
```

```
Student:1
Name:Arnav
USN:1BM23CS001
SGPA=:9.0
Arnav Dinesh 1BM23CS052
Student:2
Name:Ariz
USN:1BM23CS002
SGPA=:10.0
Arnav Dinesh 1BM23CS052
Student:3
Name:Vatsal
USN:1BM23CS003
SGPA=:8.0
Arnav Dinesh 1BM23CS052
```

Week 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.

Main code

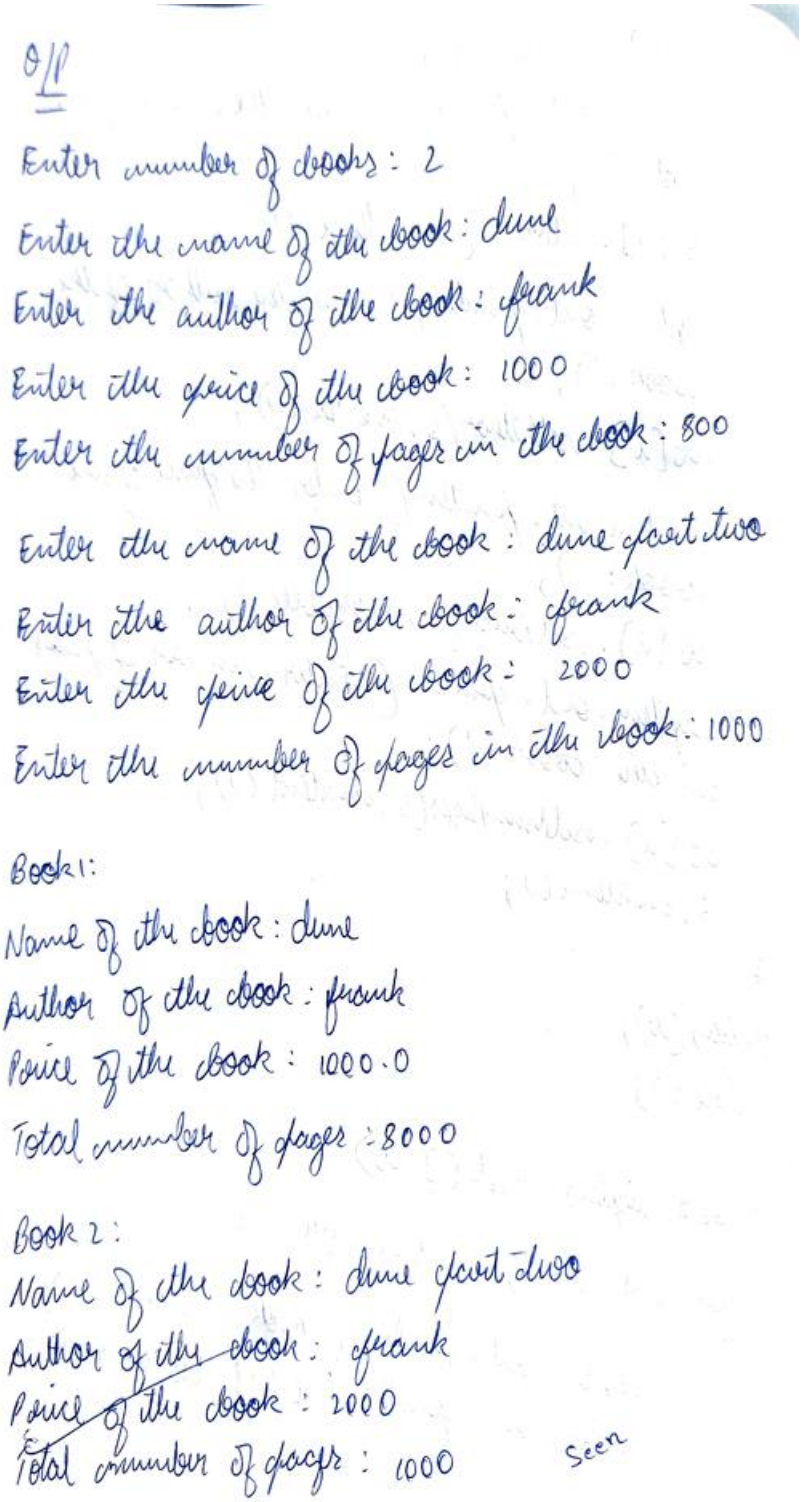
```
import java.util.Scanner;
class Book {
    String name;
    String author;
    double price;
    int num_pages;
    Book() {
        this.name = name;
        this.author = author;
        this.price = price;
        this.num_pages = num_pages;
}
```

```
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;
}
public String toString()
{
    return "Name of the book: " + name +
    " Author of the book: " + author + "Price of the
    book: " + price + " Total number of pages: "
    + num_pages;
}
}
class Run {
    public static void main (String [] args)
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter number of book:");
        int n = s.nextInt();
        s.nextLine();
        Book b[];
        b = new Book [n];
        for (int i = 0 ; i<n; i++)
        {
```

```
        db[i] = new Book();
        System.out.println("Enter the name of the
        book:");
        db[i].setName(s.nextLine());
        System.out.println("Enter the author of the
        book:");
        db[i].setAuthor(s.nextLine());
        System.out.println("Enter the price of the
        book:");
        db[i].setPrice(s.nextDouble());
        System.out.println("Enter number of pages
        in the book:");
        db[i].setNumPages(s.nextInt());
        s.nextLine();
    }
    display(db);
    s.close();
}
static void display(Book[] db)
{
    for(int i=0; i<db.length; i++)
    {
        System.out.println("Book"+(i+1)+":");
        System.out.println(db[i].toString());
    }
}
```

Output



O/P

Enter number of books : 2
Enter the name of the book : dune
Enter the author of the book : frank
Enter the price of the book : 1000
Enter the number of pages in the book : 800

Enter the name of the book : dune part two
Enter the author of the book : frank
Enter the price of the book : 2000
Enter the number of pages in the book : 1000

Book1:
Name of the book : dune
Author of the book : frank
Price of the book : 1000.0
Total number of pages : 8000

Book 2 :
Name of the book : dune part two
Author of the book : frank
Price of the book : 2000
Total number of pages : 1000

Seen

Source code

```
import java.util.Scanner;
System.out.println("Arnav Dinesh 1BM23CS052");
class Book {
    String name;
    String author;
    double price;
    int num_pages;

    Book() {
        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;
    }
```

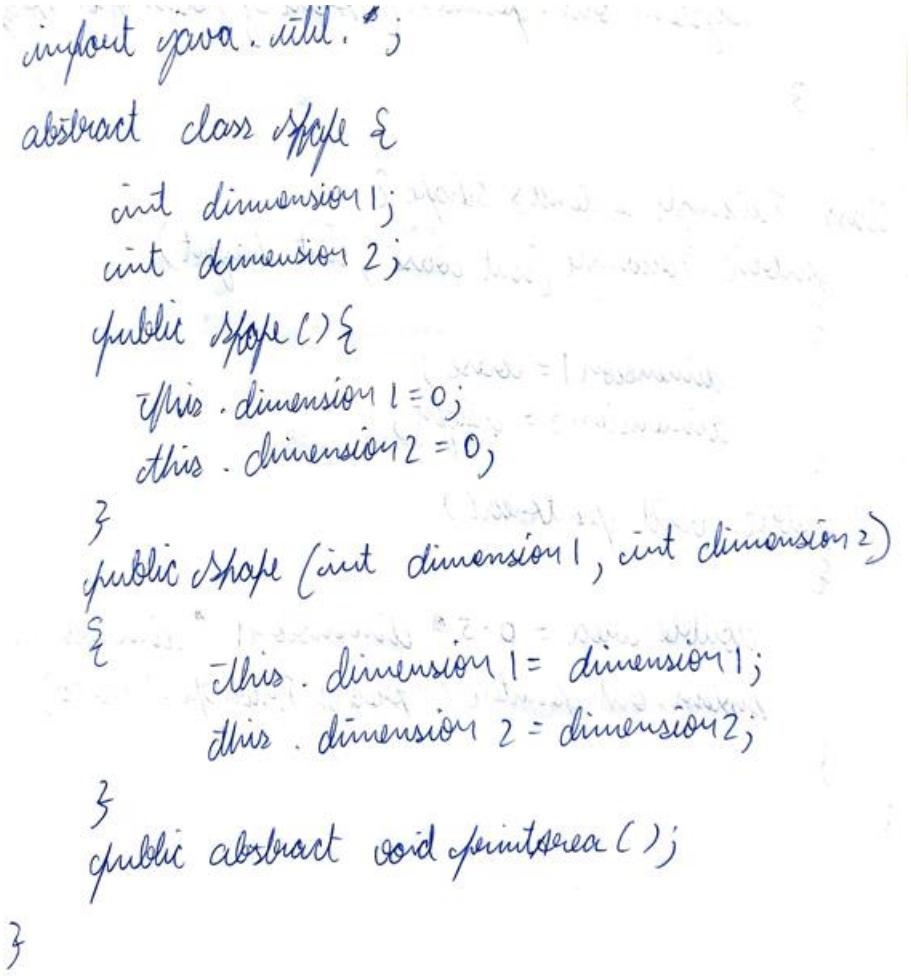
```
    public String toString() {
        return "Name of the book: " + name + ", Author of the book: " + author +
                ", Price of the book: " + price + ", Total number of pages: " + num_pages;
    }
}

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

        System.out.print("Enter number of books: ");
        int n = s.nextInt();
        s.nextLine();

        Book b[];
b = new Book[n];
        for (int i = 0; i < n; i++) {
            b[i] = new Book();

            System.out.print("Enter the name of the book: ");
            b[i].setName(s.nextLine());

            System.out.print("Enter the author of the book: ");
            b[i].setAuthor(s.nextLine());

            System.out.print("Enter the price of the book: ");
            b[i].setPrice(s.nextDouble());

            System.out.print("Enter the number of pages in the book: ");
            b[i].setNumPages(s.nextInt());
            s.nextLine();
        }

        display(b);

        s.close();
    }

    void display(Book[] b) {
        for (int i = 0; i < b.length; i++) {
            System.out.println("Book " + (i + 1) + ":");
            System.out.println(b[i].toString());
        }
    }
}
```

Output

```
Arnav Dinesh 1BM23CS052
Enter number of books: 2
Enter the name of the book: Dune
Enter the author of the book: Frank
Enter the price of the book: 2000
Enter the number of pages in the book: 500
Enter the name of the book: Dune part 2
Enter the author of the book: Frank
Enter the price of the book: 5000
Enter the number of pages in the book: 800
Book 1:
Name of the book: Dune, Author of the book: Frank, Price of the book: 2000.0, Total number of pages: 500
Book 2:
Name of the book: Dune part 2, Author of the book: Frank, Price of the book: 5000.0, Total number of pages: 800
```

Week 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.



```
import java.util.*;
abstract class Shape {
    int dimension1;
    int dimension2;
    public Shape() {
        this.dimension1 = 0;
        this.dimension2 = 0;
    }
    public Shape(int dimension1, int dimension2)
    {
        this.dimension1 = dimension1;
        this.dimension2 = dimension2;
    }
    public abstract void printArea();
}
```

```
class Rectangle extends Shape {
    public Rectangle(int length, int width)
    {
        dimension1 = length;
        dimension2 = width;
    }
    public void printArea() {
        int area = dimension1 * dimension2;
        System.out.println("Area of Rectangle:" +
    }
}
class Triangle extends Shape {
    public Triangle(int base, int height)
    {
        dimension1 = base;
        dimension2 = height;
    }
    public 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;
    }
    public void printArea() {
    double area = Math.PI * dimension1 * dimension2;
    System.out.println("Area of Circle: " + area);
    }
}
class shapearea {
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter length and width
                            for rectangle:");
        int length = scanner.nextInt();
        int width = scanner.nextInt();
        Shape rectangle = new Rectangle(length, width);
        rectangle.printArea();
        System.out.println("Enter base and height
          for Triangle:");
        int base = scanner.nextInt();
        int height = scanner.nextInt();
        Shape triangle = new Triangle(base, height);
```

```
        triangle.printArea();
        System.out.println("Enter radius for
        circle:");
        int radius = scanner.nextInt();
        Shape circle = new Circle(radius);
        circle.printArea();
        scanner.close();
    }
}
```

Output

```
O/P
Enter length & width for rectangle:
20
30
Area of rectangle: 600
Enter base & heigh for triangle:
10
20
Area of triangle: 100.0
Enter radius of Circle:
20
Area of Circle: 1256.6370
```

Executed
GL
23/10/24

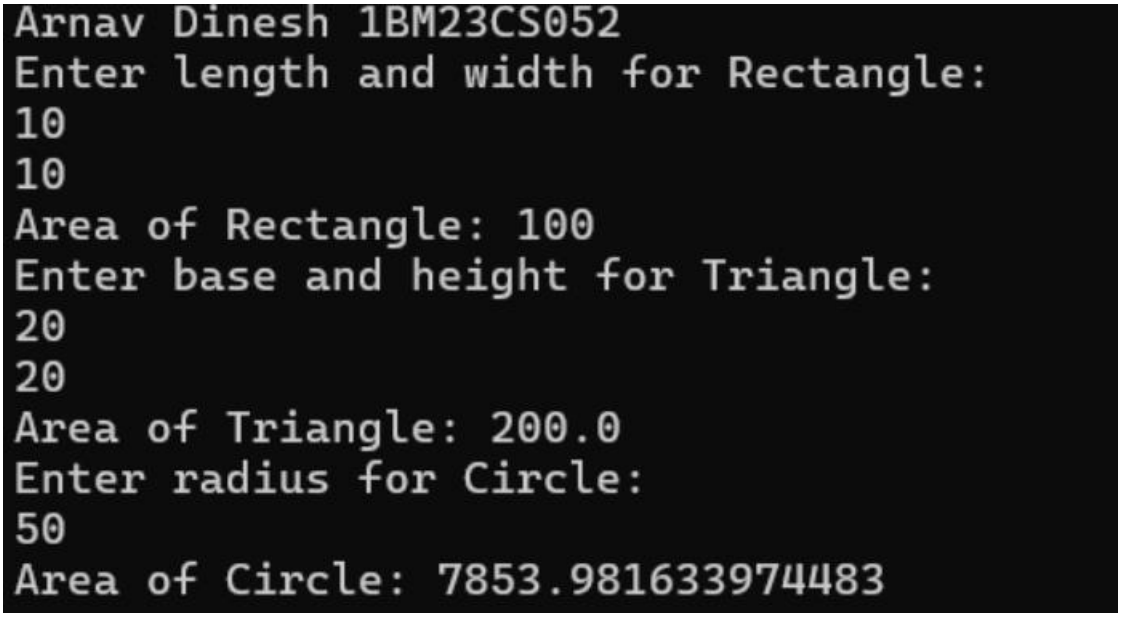
Source code

```
import java.util.*;
abstract class Shape {
   int dimension1;
   int dimension2;


   public Shape() {
      this.dimension1 = 0;
      this.dimension2 = 0;

   }

   public Shape(int dimension1, int dimension2) {
      this.dimension1 = dimension1;
      this.dimension2 = dimension2;
   }

   public abstract void printArea();

}

class Rectangle extends Shape {
   public Rectangle(int length, int width) {

      dimension1 = length;
      dimension2 = width;
   }

   public 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;
   }
```

```
 public 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;
    }

    public void printArea() {

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

class shapearea {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("ArnavDinesh 1BM23CS052);

        System.out.println("Enter length and width for Rectangle:");

        int length = scanner.nextInt();
        int width = scanner.nextInt();
        Shape rectangle = new Rectangle(length, width);
        rectangle.printArea();

        System.out.println("Enter base and height for Triangle:");

        int base = scanner.nextInt();
        int height = scanner.nextInt();
        Shape triangle = new Triangle(base, height);
        triangle.printArea();

        System.out.println("Enter radius for Circle:");

        int radius = scanner.nextInt();
        Shape circle = new Circle(radius);
        circle.printArea();
```

```
        scanner.close();
    }
}
```

Output

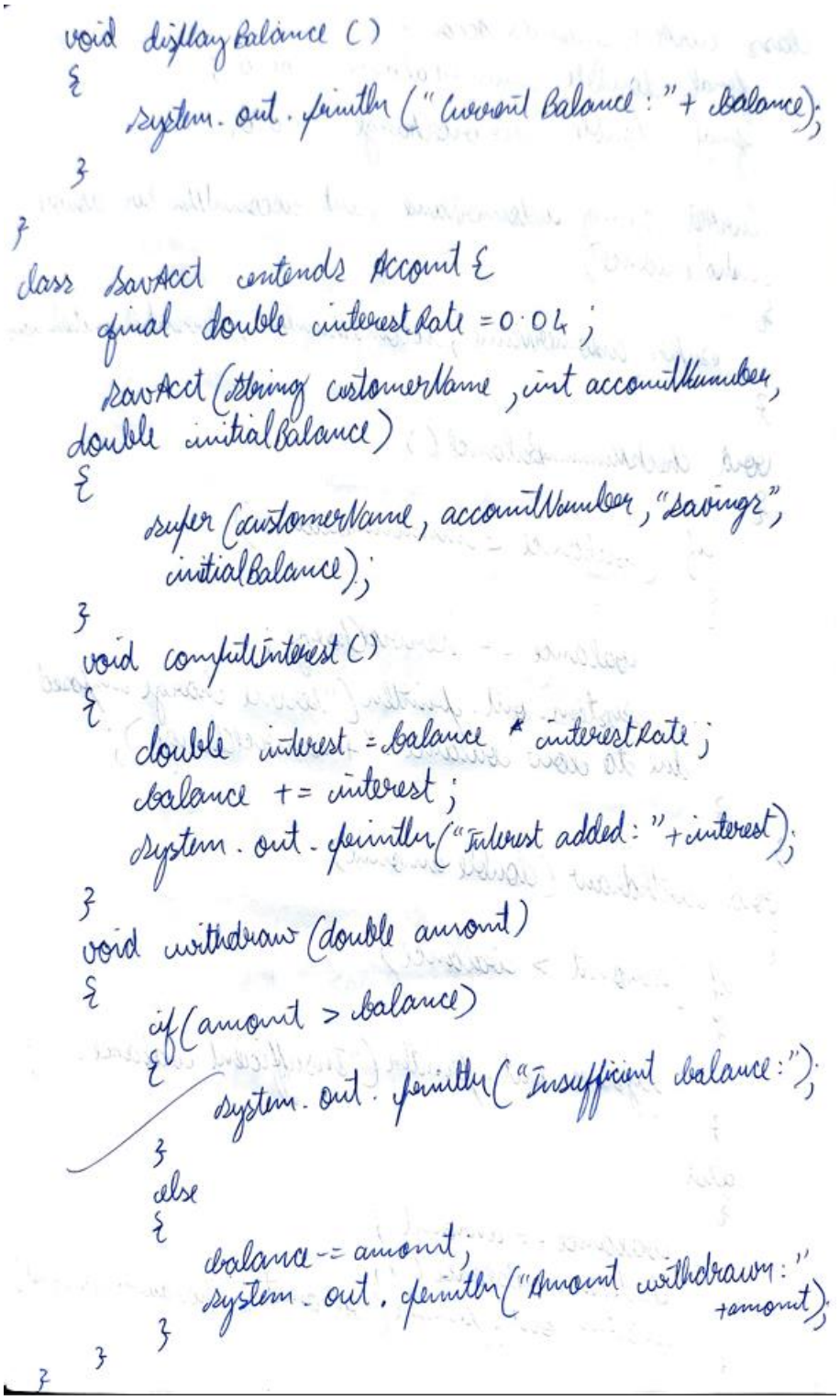


Week 5

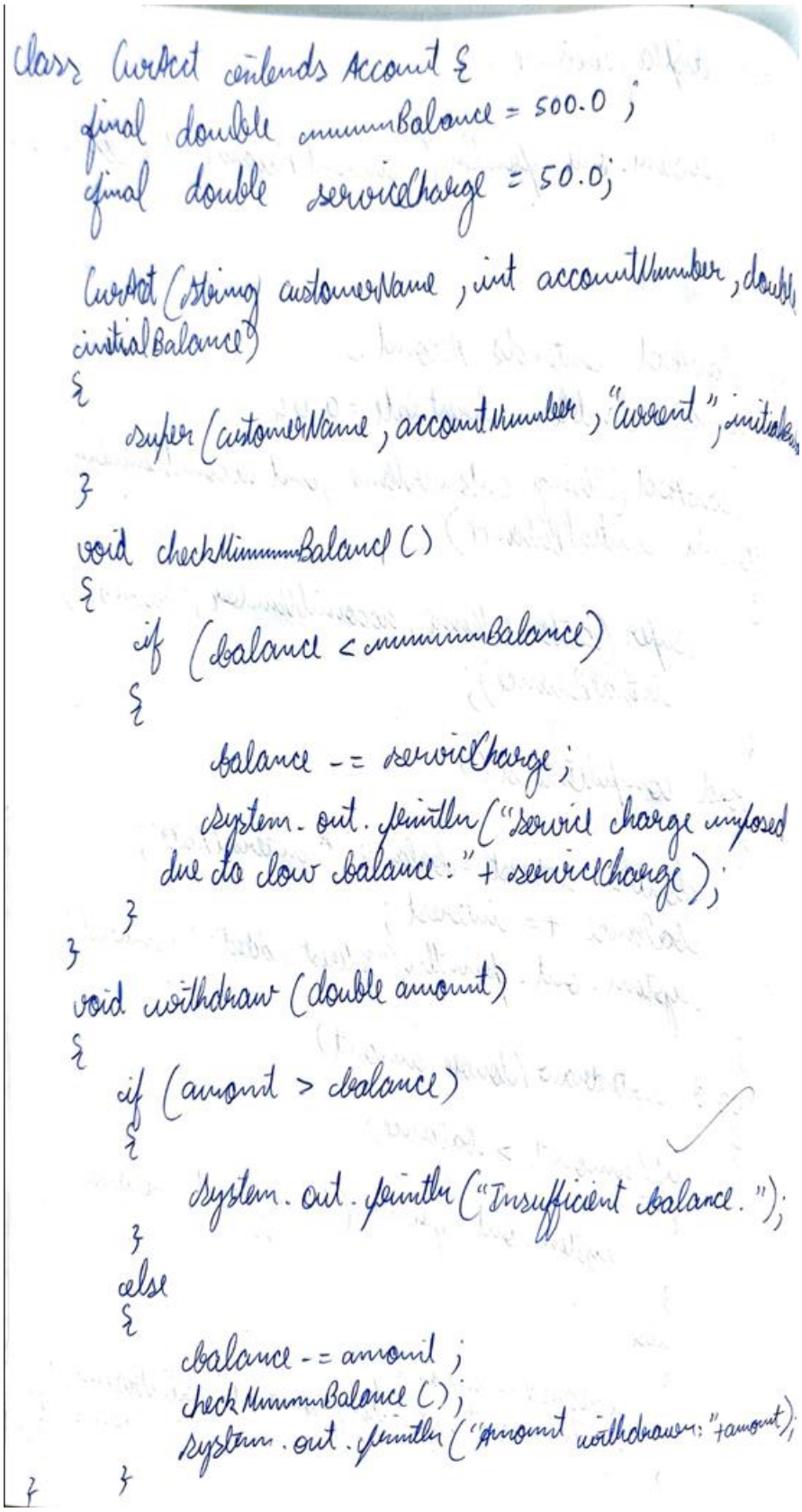
Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book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

Main Code

```
class Account {
    String customerName;
    int accountNumber;
    String accountType;
    double balance;
    Account (String customerName, int accountNumber,
    String accountType, double initialBalance)
    {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = initialBalance;
    }
    void deposit (double amount)
    {
        if (amount > 0)
        {
            balance += amount;
            System.out.println("Amount deposited:" + amount);
        }
        else
        {
            System.out.println("Invalid deposit amount.");
        }
    }
```



```
    void displayBalance ()
    {
        System.out.println ("Account Balance : " + balance);
    }
}
class SavAcct extends Account {
    final double interestRate = 0.04;
    SavAcct (String customerName, int accountNumber,
double initialBalance)
    {
        super (customerName, accountNumber, "savings",
         initialBalance);
    }
    void computeInterest ()
    {
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest added: " + interest);
    }
    void withdraw (double amount)
    {
        if(amount > balance)
        {
            System.out.println("Insufficient balance:");
        }
        else
        {
            balance -= amount;
            System.out.println("Amount withdrawn: "
                                        + amount);
        }
    }
}
```



```
class CurrAct extends Account {
    final double minimumBalance = 500.0;
    final double serviceCharge = 50.0;

    CurrAct (String customerName, int accountNumber, double
    initialBalance)
    {
        super (customerName, accountNumber, "current", initialBa
    }
    void checkMinimumBalance ()
    {
        if (balance < minimumBalance)
        {
            balance -= serviceCharge;
            System.out.println("Service charge imposed
          due to low balance: " + serviceCharge);
        }
    }
    void withdraw (double amount)
    {
        if (amount > balance)
        {
            System.out.println("Insufficient balance.");
        }
        else
        {
            balance -= amount;
            checkMinimumBalance ();
            System.out.println("Amount withdrawn: "+amount);
        }
}
```

```
class Bank {
    public static void main(String[] args)
    {
        SavAcct savingsAccount = new SavAcct("Alice", 101, 1000.
        CurAcct currentAccount = new CurAcct("Bob", 102, 600
        System.out.println(" Savings Account: ");
        savingsAccount.deposit(500);
        savingsAccount.computeInterest();
        savingsAccount.withdraw(200);
        savingsAccount.displayBalance();
        System.out.println(" In Current Account :");
        currentAccount.deposit(300);
        currentAccount.withdraw(700);
        currentAccount.displayBalance();
    }
}
```

Output

```
O/P
Savings Account:
Amount deposited : 500.0
Interest added : 60.0
Amount withdrawn : 200.0
Current balance : 1360.0

Current Account:
Amount deposited : 300.0
Service charge imposed due to low balance : 50.0
Amount withdrawn : 700.0
Current balance : 150.0
```

Source Code

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

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

    void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Amount deposited: " + amount);
        } else {
            System.out.println("Invalid deposit amount.");
        }
    }

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

class SavAcct extends Account {
    final double interestRate = 0.04;

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

    void computeInterest() {
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest added: " + interest);
    }

    void withdraw(double amount) {
        if (amount > balance) {
            System.out.println("Insufficient balance.");
        } else {
            balance -= amount;
            System.out.println("Amount withdrawn: " + amount);
```

```
        }
    }
}

class CurAcct extends Account {
    final double minimumBalance = 500.0;
    final double serviceCharge = 50.0;

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

    void checkMinimumBalance() {
        if (balance < minimumBalance) {
            balance -= serviceCharge;
            System.out.println("Service charge imposed due to low balance: " + serviceCharge);
        }
    }

    void withdraw(double amount) {
        if (amount > balance) {
            System.out.println("Insufficient balance.");
        } else {
            balance -= amount;
            checkMinimumBalance();
            System.out.println("Amount withdrawn: " + amount);
        }
    }
}

class Bank {
    public static void main(String[] args) {
        SavAcct savingsAccount = new SavAcct("Alice", 101, 1000.0);
        CurAcct currentAccount = new CurAcct("Bob", 102, 600.0);

        System.out.println("Savings Account:");
        savingsAccount.deposit(500);
        savingsAccount.computeInterest();
        savingsAccount.withdraw(200);
        savingsAccount.displayBalance();

        System.out.println("\nCurrent Account:");
        currentAccount.deposit(300);
        currentAccount.withdraw(700);
        currentAccount.displayBalance();
        System.out.println("Arnav Dinesh 1BM23CS052");
    }}
```

Output

```
Savings Account:
Amount deposited: 500.0
Interest added: 60.0
Amount withdrawn: 200.0
Current balance: 1360.0

Current Account:
Amount deposited: 300.0
Service charge imposed due to low balance: 50.0
Amount withdrawn: 700.0
Current balance: 150.0

Arnav Dinesh 1BM23CS052
```

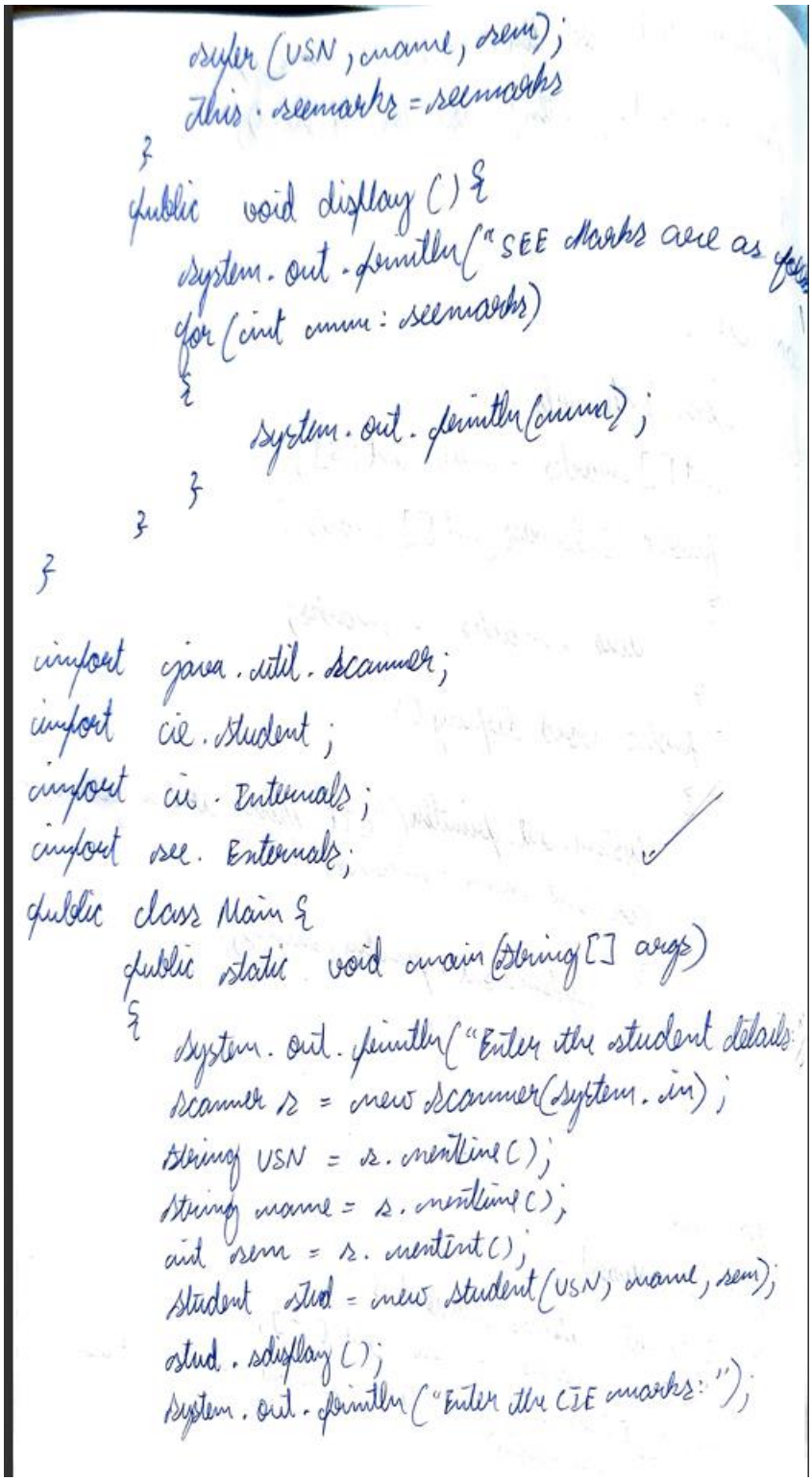
Week 6

Create a package CIE which has two classes- Student 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 Student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

Code

```
package cie;
public class student {
    String USN;
    String name;
    int sem;
    public student(String USN, String name, int sem)
    {
        this.USN = USN;
        this.name = name;
        this.sem = sem;
    }
    public void display()
    {
        System.out.println("student details are as follows:");
        System.out.println("USN: " + USN);
```

```
        System.out.println("Name:" + name);
        System.out.println("Semester:" + sem);
    }
}
package cie;
public class Internals {
    int[] marks = new int[5];
    public Internals(int[] marks)
    {
        this.marks = marks;
    }
    public void display()
    {
        System.out.println("CIE Marks are as follows:\n");
        for(int mm : marks)
        {
            System.out.println(mm);
        }
    }
}
package see;
import cie.Student;
class External extends Student {
    int[] semarks = new int[5];
    public External(String USN, String name, int sem,
      int[] seemarks)
    {
```



```
        super(USN, name, sem);
        this.seemarks = seemarks
    }
    public void display() {
        System.out.println("SEE Marks are as
        for(int mm : seemarks)
        {
            System.out.println(mm);
        }
    }
}

import java.util.Scanner;
import cie.Student;
import cie.Externals;
import see.Externals;
public class Main {
    public static void main(String[] args)
    {
        System.out.println("Enter the student details");
        Scanner s2 = new Scanner(System.in);
        String USN = s.nextLine();
        String name = s.nextLine();
        int sem = s.nextInt();
        Student stud = new Student(USN, name, sem);
        stud.sdisplay();
        System.out.println("Enter the CIE marks:");
```

```
int [] marks = new int [5];
for(int i=0; i<5 ; i++)
{
    marks[i] = s.nextInt();
}
Internals i = new Internals(marks);
i.display();
System.out.println("Enter the SEE marks:");
int [] smarks = new int [5];
for (int i = 0; i<5 ; i++)
{
    smarks[i] = s.nextInt();
}
Externals e = new Externals(USN, name, sem, smarks);
e.display();
s.close();
}
}
```

Output

```
O/P
Enter the number of students : 1
Entering details for Student : 1
Enter USN : 1BM23CS054
Enter Name : Akash
Enter Semester : 3
Enter Internal Marks for 5 subjects :
```

Subject 1 : 85
Subject 2 : 86
Subject 3 : 90
Subject 4 : 91
Subject 5 : 92

Enter SEE Marks for 5 subjects :
Subject 1 : 90
Subject 2 : 94
Subject 3 : 95
Subject 4 : 96
Subject 5 : 97

Final Marks of Students :
Student 1 :
USN : 1BM23CS051
Name : Akash
Semester : 3
Final Marks in 5 subjects :
Subject 1 : 175
Subject 2 : 180
Subject 3 : 185
Subject 4 : 187
Subject 5 : 189

Source code

```
package cie;

public class Internals {
    public int[] intMarks = new int[5];
    public Internals(int[] marks) {
        System.arraycopy(marks, 0, intMarks, 0, marks.length);
    }
}
---------------
package cie;

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

public Student(String usn,String name,int sem)
 {
  this.usn=usn;
  this.name=name;
  this.sem=sem;
 }
}
-------------
package see;

import cie.Student;

public class Externals extends Student {
    public int[] seeMarks = new int[5];
    public Externals(String usn, String name, int sem, int[] seeMarks) {
        super(usn, name, sem);
        System.arraycopy(seeMarks, 0, this.seeMarks, 0, seeMarks.length);
    }
}
----------------
import cie.*;
import see.*;
import java.util.*;

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

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

Student[] s = new Student[n];
Internals[] ie = new Internals[n];
Externals[] ee = new Externals[n];

for (int i = 0; i < n; i++) {
   System.out.println("Enter details for student " + (i + 1));

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

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

   System.out.print("Enter Semester: ");
   int sem= in.nextInt();

   System.out.print("Enter internal marks for 5 subjects: ");
   int[] intMarks = new int[5];
   for (int j = 0; j < 5; j++) {
      intMarks[j] = in.nextInt();
   }

   System.out.print("Enter SEE marks for 5 subjects: ");
   int[] seeMarks = new int[5];
   for (int j = 0; j < 5; j++) {
      seeMarks[j] = in.nextInt();
   }

   in.nextLine();

   s[i] = new Student(usn, name, sem);
   ie[i] = new Internals(intMarks);
   ee[i] = new Externals(usn, name, sem, seeMarks);
}

System.out.println("\nFinal marks of each student:");
for (int i = 0; i < n; i++) {
   System.out.println("Student " + (i + 1) + " (" + s[i].name + "):");
   for (int j = 0; j < 5; j++) {
      int finalMarks = ie[i].intMarks[j] + (ee[i].seeMarks[j] / 2);
      System.out.println("Course " + (j + 1) + ": " + finalMarks);
   }
```

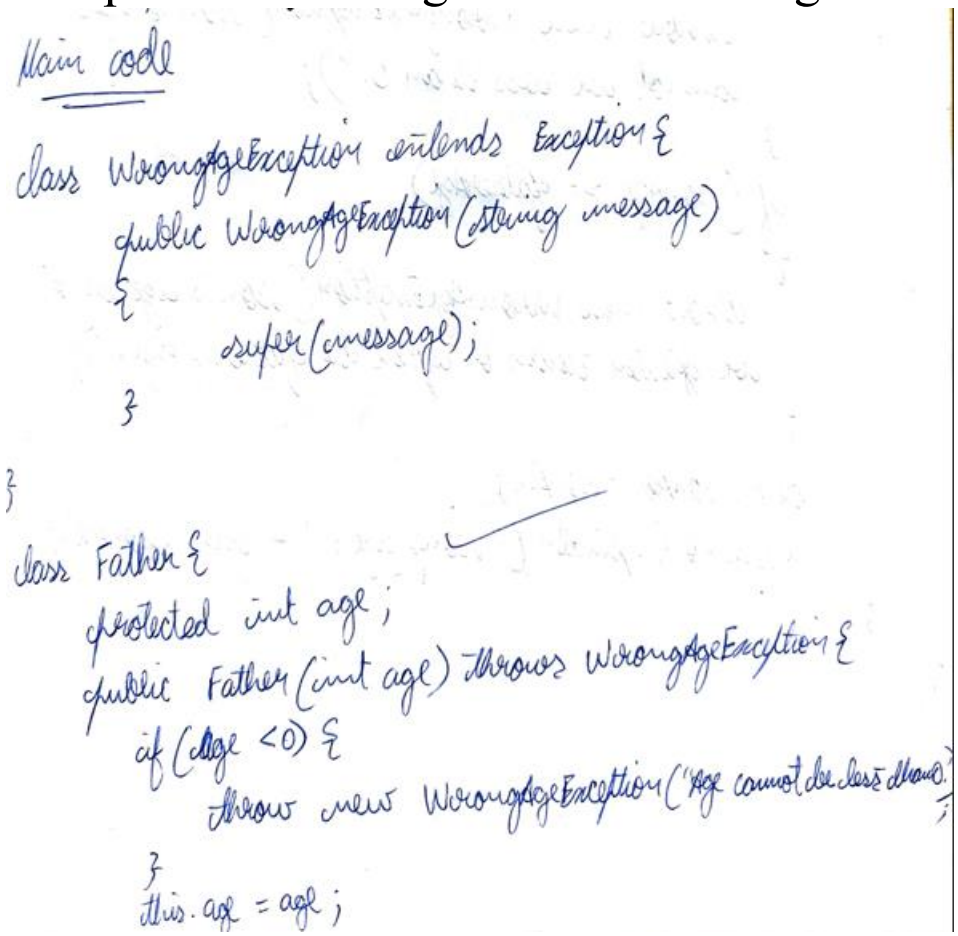
```
            System.out.println();
            System.out.println("Arnav Dinesh 1BM23CS052");
        }

        in.close();
    }
}
```

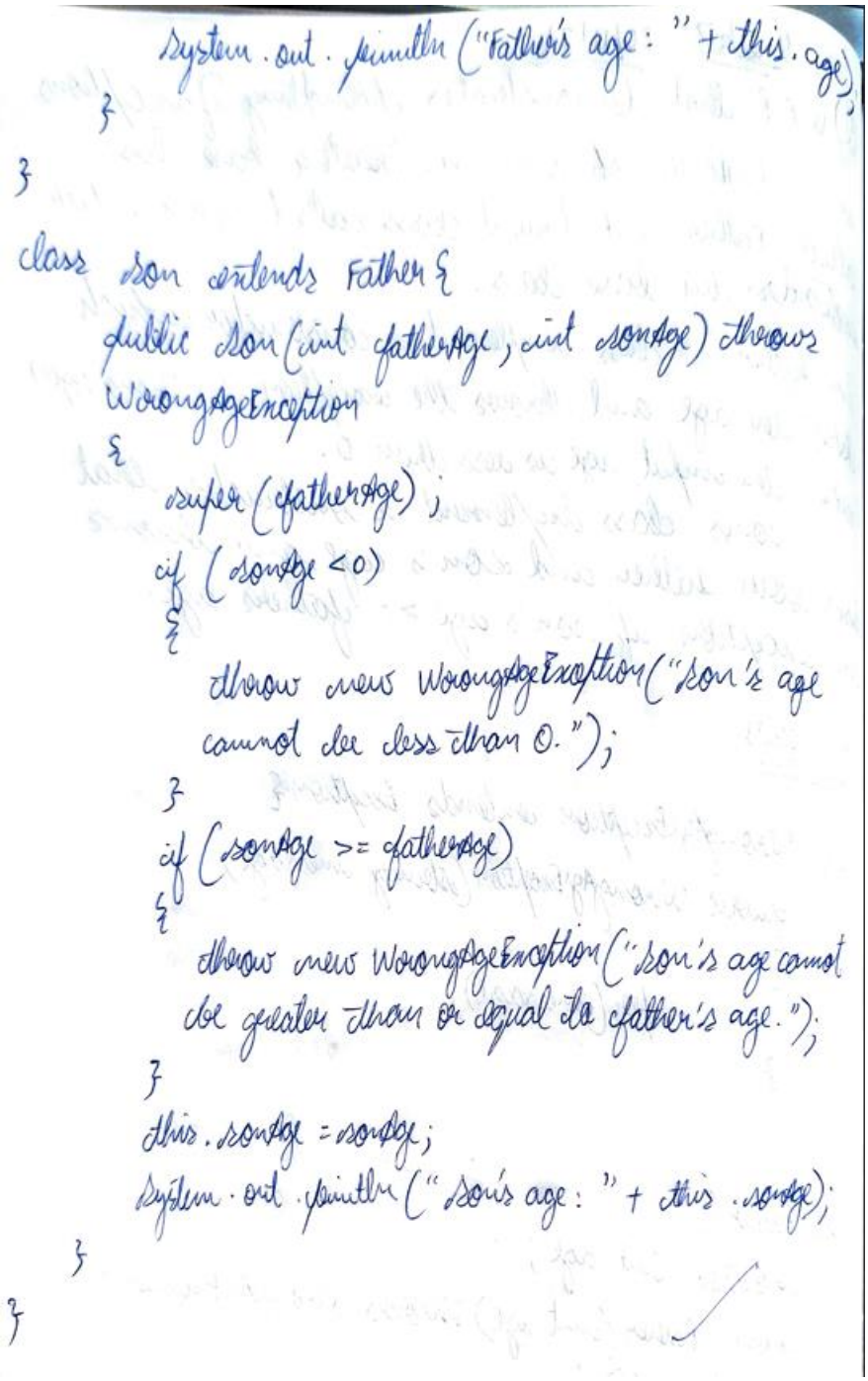
Output

```
Enter the number of students: 1
Enter details for student 1
Enter USN: 1BM23CS052
Enter Name: Arnav
Enter Semester: 3
Enter internal marks for 5 subjects: 45
45
45
45
45
Enter SEE marks for 5 subjects: 80
80
80
80
80

Final marks of each student:
Student 1 (Arnav):
Course 1: 85
Course 2: 85
Course 3: 85
Course 4: 85
Course 5: 85

Arnav Dinesh 1BM23CS052
```

Week 7

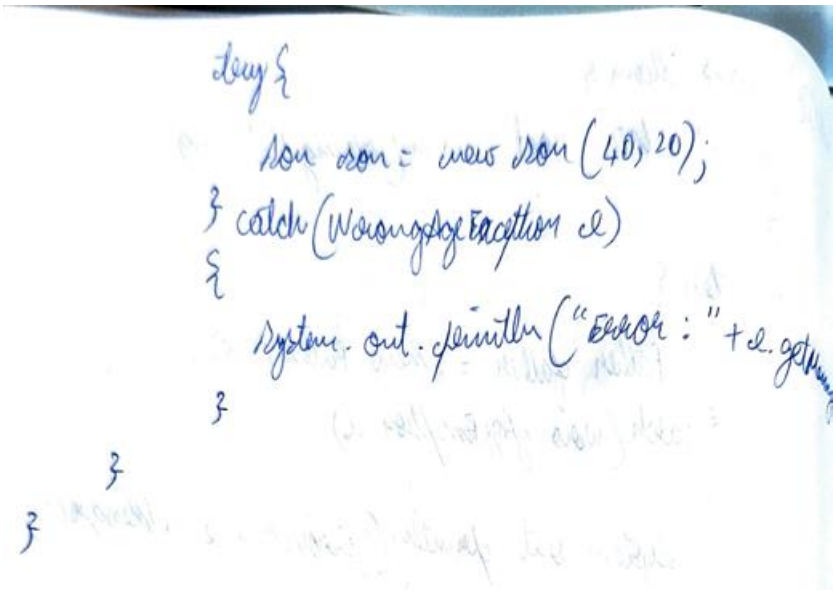
Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age<0. In Son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





```
        System.out.println("Father's age: " + this.age);
    }
}

class Son extends Father {
    public Son(int fatherAge, int sonAge) throws
    WrongAgeException
    {
        super(fatherAge);
        if (sonAge < 0)
        {
            throw new WrongAgeException("Son's age
            cannot be less than 0.");
        }
        if (sonAge >= fatherAge)
        {
            throw new WrongAgeException("Son's age cannot
            be greater than or equal to father's age.");
        }
        this.sonAge = sonAge;
        System.out.println("Son's age: " + this.sonAge);
    }
}
```

```
public class Main {
    public static void main(String[] args)
    {
        try {
            Father father = new Father(-5);
        } catch (WrongAgeException e)
        {
            System.out.println("Error:" + e.getMessage());
        }
        try {
            Son son = new Son(40,45);
        } catch (WrongAgeException e)
        {
            System.out.println("Error:" + e.getMessage());
        }
        try {
            Son son = new Son(40, -10);
        } catch (WrongAgeException e)
        {
            System.out.println("Error:" + e.getMessage());
        }
```



```
try {
    Son son = new Son(40, 20);
} catch (WrongAgeException e)
{
    System.out.println("Error : " + e.getMessage());
}
}
}
```

Output

O/P

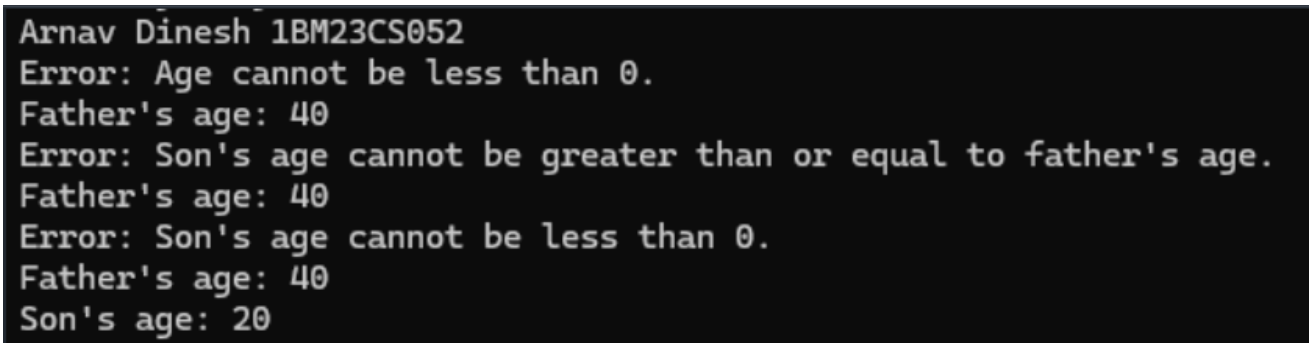
Error : Age cannot be less than 0.
Father's age : 40
Error : Son's age cannot be greater than or equal to father's age
Father's age : 40
Error : Son's age cannot be less than 0.
Father's age : 40
Son's age : 20

Source Code

```
class WrongAgeException extends Exception {
    public WrongAgeException(String message) {
        super(message);
    }
}
class Father {
    protected int age;
    public Father(int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException("Age cannot be less than 0.");
        }
        this.age = age;
        System.out.println("Father's age: " + this.age);
    }
}
class Son extends Father {
    private int sonAge;
    public Son(int fatherAge, int sonAge) throws WrongAgeException {
        super(fatherAge);
        if (sonAge < 0) {
            throw new WrongAgeException("Son's age cannot be less than 0.");
        }
        if (sonAge >= fatherAge) {
            throw new WrongAgeException("Son's age cannot be greater than or equal to father's age.");
        }
        this.sonAge = sonAge;
        System.out.println("Son's age: " + this.sonAge);
    }
}
public class Main {
    public static void main(String[] args) {
        System.out.println("Arnav Dinesh 1BM23CS052");
        try {
            Father father = new Father(-5);
        } catch (WrongAgeException e) {
            System.out.println("Error: " + e.getMessage());
        }
        try {
            Son son = new Son(40, 45);
        } catch (WrongAgeException e) {
            System.out.println("Error: " + e.getMessage());
        }
        try {
            Son son = new Son(40, -10);
```

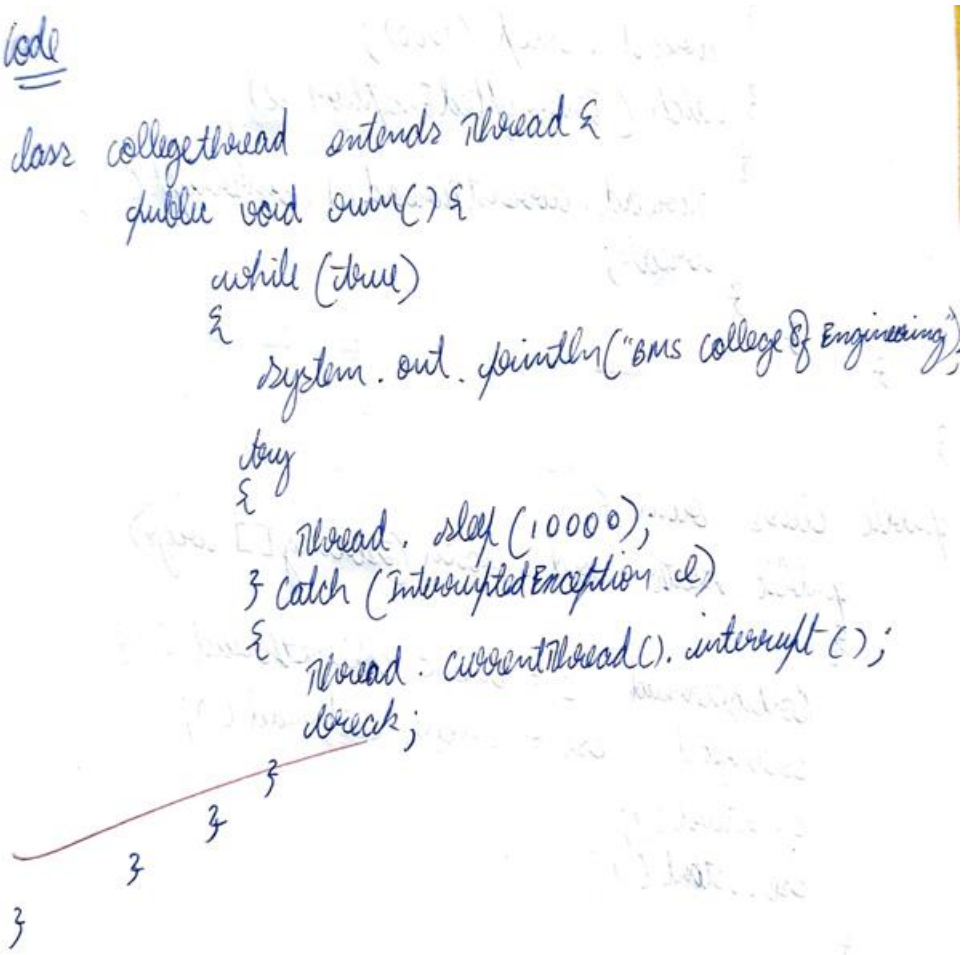
```
        } catch (WrongAgeException e) {
            System.out.println("Error: " + e.getMessage());
        }
        try {
            Son son = new Son(40, 20);
        } catch (WrongAgeException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

Output



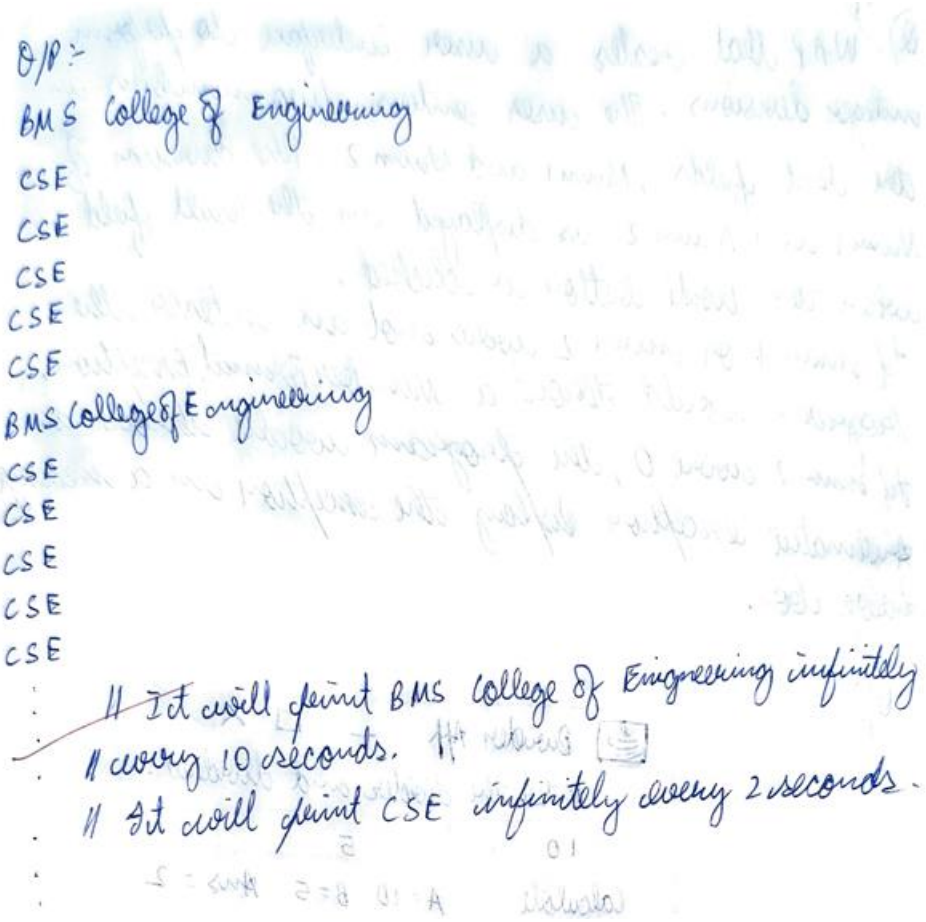
Week 8

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.



```
class csethread extends Thread {
    public void run()
    {
        while (true)
        {
            System.out.println("CSE");
            try
            {
                Thread.sleep(2000);
            } catch (InterruptedException e)
            {
                Thread.currentThread().interrupt();
                break;
            }
        }
    }
}
public class run {
    public static void main (String[] args)
    {
        collegethread c = new collegethread();
        csethread cse = new csethread();
        c.start();
        cse.start();
    }
}
```

Output


O/P :-
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS CollegeofEngineering
CSE
CSE
CSE
CSE
CSE
// It will print BMS College of Engineering infinitely
// every 10 seconds.
// It will print CSE infinitely every 2 seconds.

Source code

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

class CseThread extends Thread {
    public void run() {
        while (true) {
            System.out.println("Arnav Dinesh 1BM23CS052");
            System.out.println("CSE");
            try {
                Thread.sleep(2000);
            } catch (InterruptedException e) {
                Thread.currentThread().interrupt();
                break;
            }
        }
    }
}

public class run {
    public static void main(String[] args) {
        CollegeThread collegeThread = new CollegeThread();
        CseThread cseThread = new CseThread();
        collegeThread.start();
        cseThread.start();
    }
}
```

Output

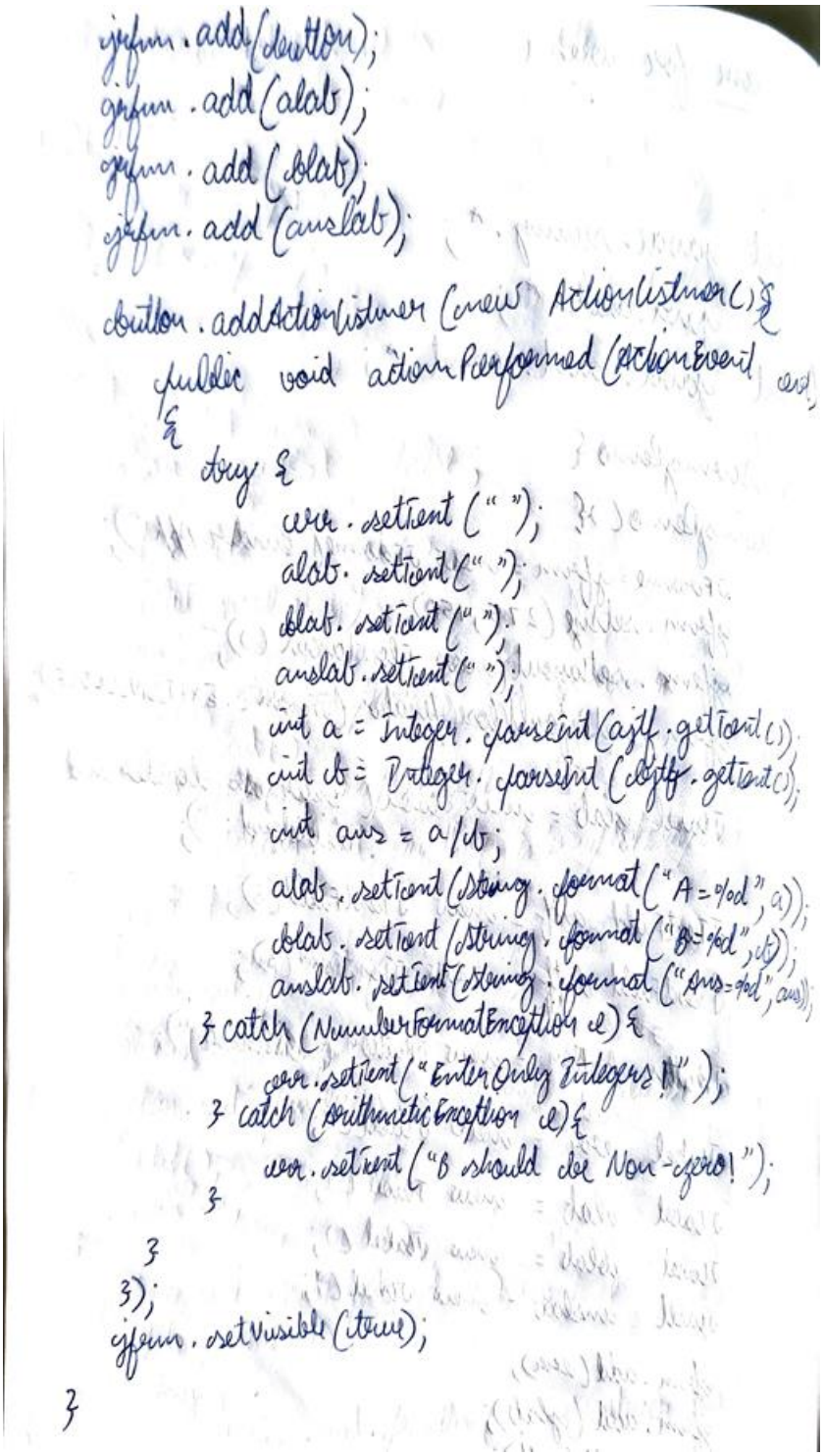
```
BMS College of Engineering
Arnav Dinesh 1BM23CS052
CSE
Arnav Dinesh 1BM23CS052
CSE
Arnav Dinesh 1BM23CS052
CSE
Arnav Dinesh 1BM23CS052
CSE
Arnav Dinesh 1BM23CS052
CSE
BMS College of Engineering
Arnav Dinesh 1BM23CS052
CSE
Arnav Dinesh 1BM23CS052
CSE
```

Week 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.

Code

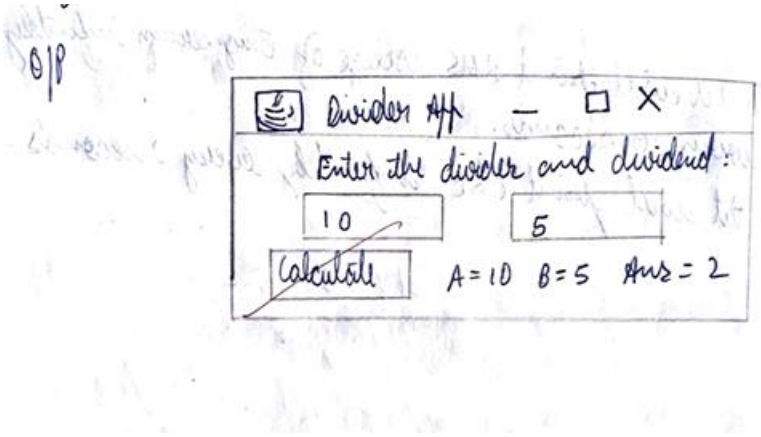
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class SwingDemo {
    SwingDemo() {
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JLabel jlab = new JLabel("Enter the divider and
                                  dividend: ");

        JTextField ajtf = new JTextField(8);

        JTextField bjtf = new JTextField(8);

        JButton button = new JButton("Calculate");

        JLabel err = new JLabel();

        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
        JLabel anslab = new JLabel();

        jfrm.add(err);
        jfrm.add(jlab);
        jfrm.add(ajtf);
        jfrm.add(bjtf);
```



```
ajfun.add(button);
gnfun.add(alab);
ajfun.add(blab);
ajfun.add(anslab);

button.addActionListner(new ActionListner(){
    public void actionPerformed(ActionEvent ae)
    {
        try {
            wor.setText(" ");
            alab.setText(" ");
            blab.setText(" ");
            anslab.setText(" ");
            int a = Integer.parseInt(ajtf.getText());
            int b = Integer.parseInt(bjtf.getText());
            int ans = a/b;
            alab.setText(String.format("A=%d", a));
            blab.setText(String.format("B=%d", b));
            anslab.setText(String.format("ans=%d", ans));
        } catch (NumberFormatException e) {
            wor.setText("Enter Only Integers !!");
        } catch (ArithmeticException e){
            wor.setText("B should be Non-zero!");
        }
    }
});
ajfun.setVisible(true);
}
```

```
public static void main(String args[])
{
    System.out.println("Im");
    SwingUtilities.invokeLater(new Runnable(){
        public void run(){
            new SwingDemo();
        }
    });
}
}
```

Output

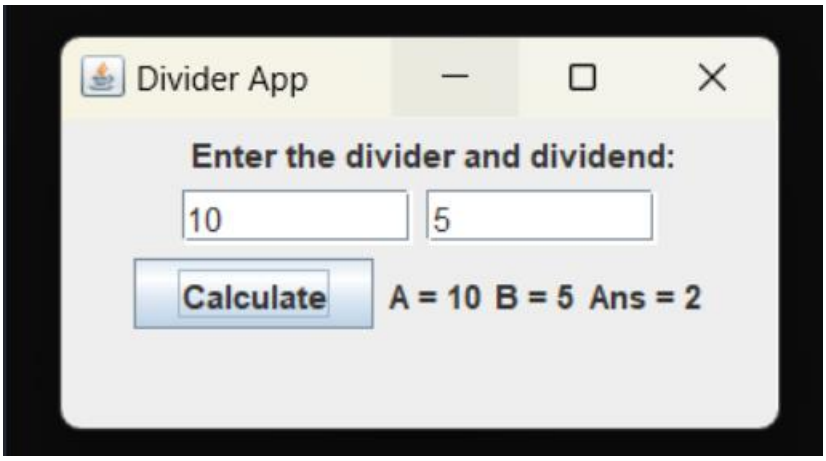


Source Code

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

class SwingDemo {
   SwingDemo() {
      JFrame jfrm = new JFrame("Divider App");
      jfrm.setSize(275, 150);
      jfrm.setLayout(new FlowLayout());
      jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

      JLabel jlab = new JLabel("Enter the divider and dividend:");

      JTextField ajtf = new JTextField(8);
      JTextField bjtf = new JTextField(8);

      JButton button = new JButton("Calculate");

      JLabel err = new JLabel();
      JLabel alab = new JLabel();
      JLabel blab = new JLabel();
      JLabel anslab = new JLabel();

      jfrm.add(err);
      jfrm.add(jlab);
      jfrm.add(ajtf);
      jfrm.add(bjtf);
      jfrm.add(button);
      jfrm.add(alab);
      jfrm.add(blab);
      jfrm.add(anslab);

      button.addActionListener(new ActionListener() {
         public void actionPerformed(ActionEvent evt) {
            try {
               err.setText("");
               alab.setText("");
               blab.setText("");
               anslab.setText("");

               int a = Integer.parseInt(ajtf.getText());
               int b = Integer.parseInt(bjtf.getText());
               int ans = a / b;
```

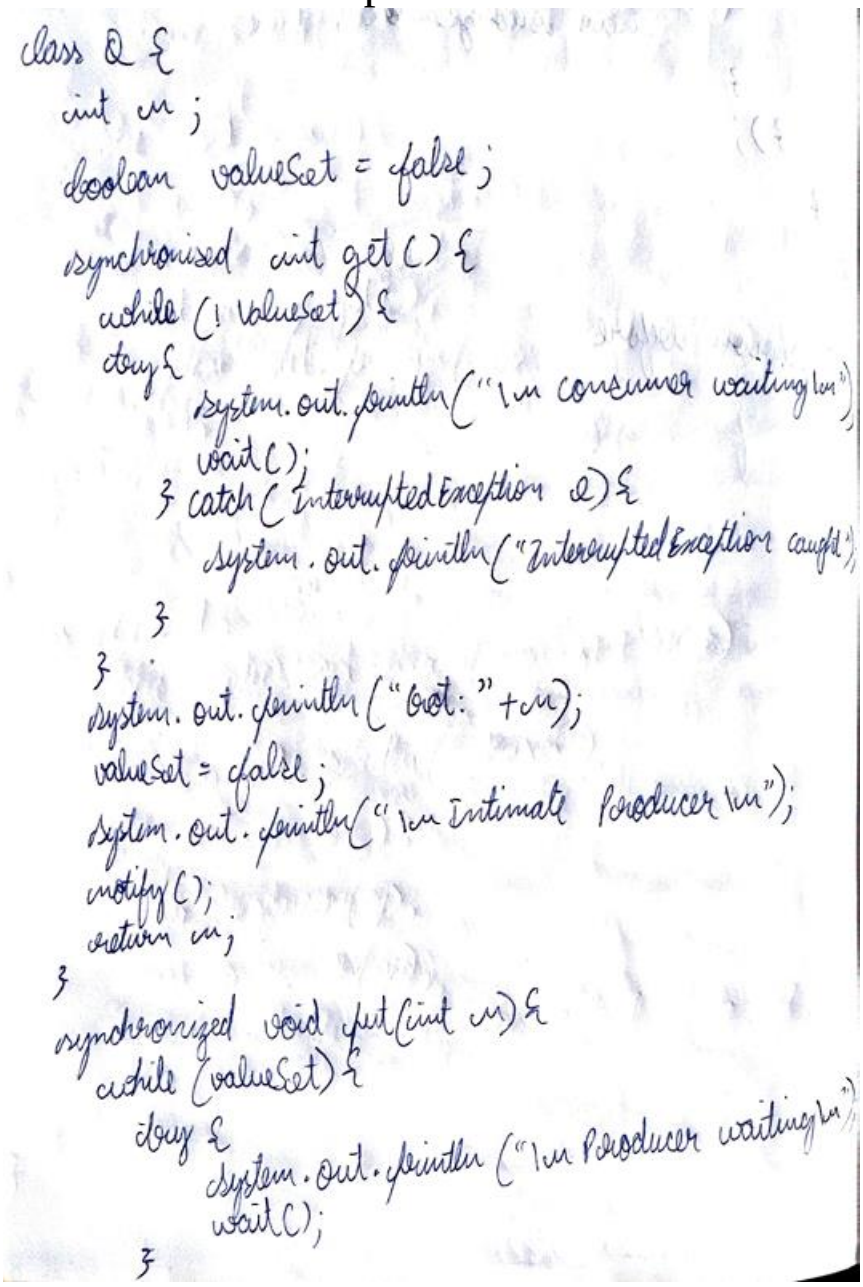
```
                alab.setText(String.format("A = %d", a));
                blab.setText(String.format("B = %d", b));
                anslab.setText(String.format("Ans = %d", ans));
            } catch (NumberFormatException e) {
                err.setText("Enter Only Integers!");
            } catch (ArithmeticException e) {
                err.setText("B should be NON-zero!");
            }
        }
    });

    jfrm.setVisible(true);
  }

  public static void main(String args[]) {
    System.out.println("Arnav Dinesh 1BM23CS052");
    SwingUtilities.invokeLater(new Runnable() {
      public void run() {
        new SwingDemo();
      }
    });
  }
}
```

Output



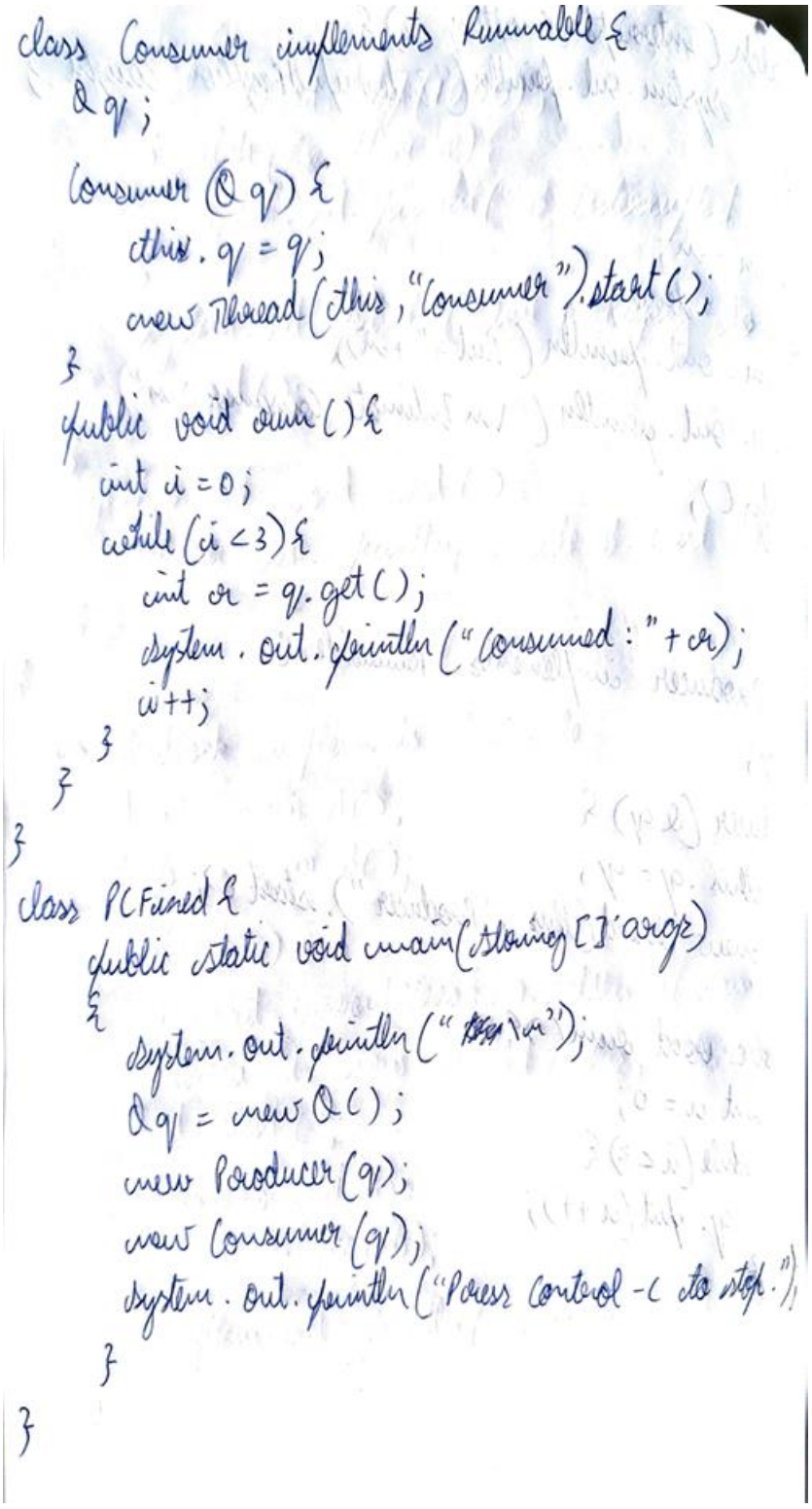
Week 10

Demonstrate Inter process Communication and deadlock



```
class Q {
  int n;
  boolean valueSet = false;
  synchronised int get() {
    while (!valueSet) {
    try {
        System.out.println("\n consumer waiting\n");
        wait();
      } catch (InterruptedException e) {
        System.out.println("InterruptedException caught");
      }
    }
    System.out.println("Got: " + n);
    valueSet = false;
    System.out.println("\n Intimate Producer\n");
    notify();
    return n;
  }
  synchronized void put(int n) {
    while (valueSet) {
      try {
        System.out.println("\n Producer waiting\n");
        wait();
      }
```

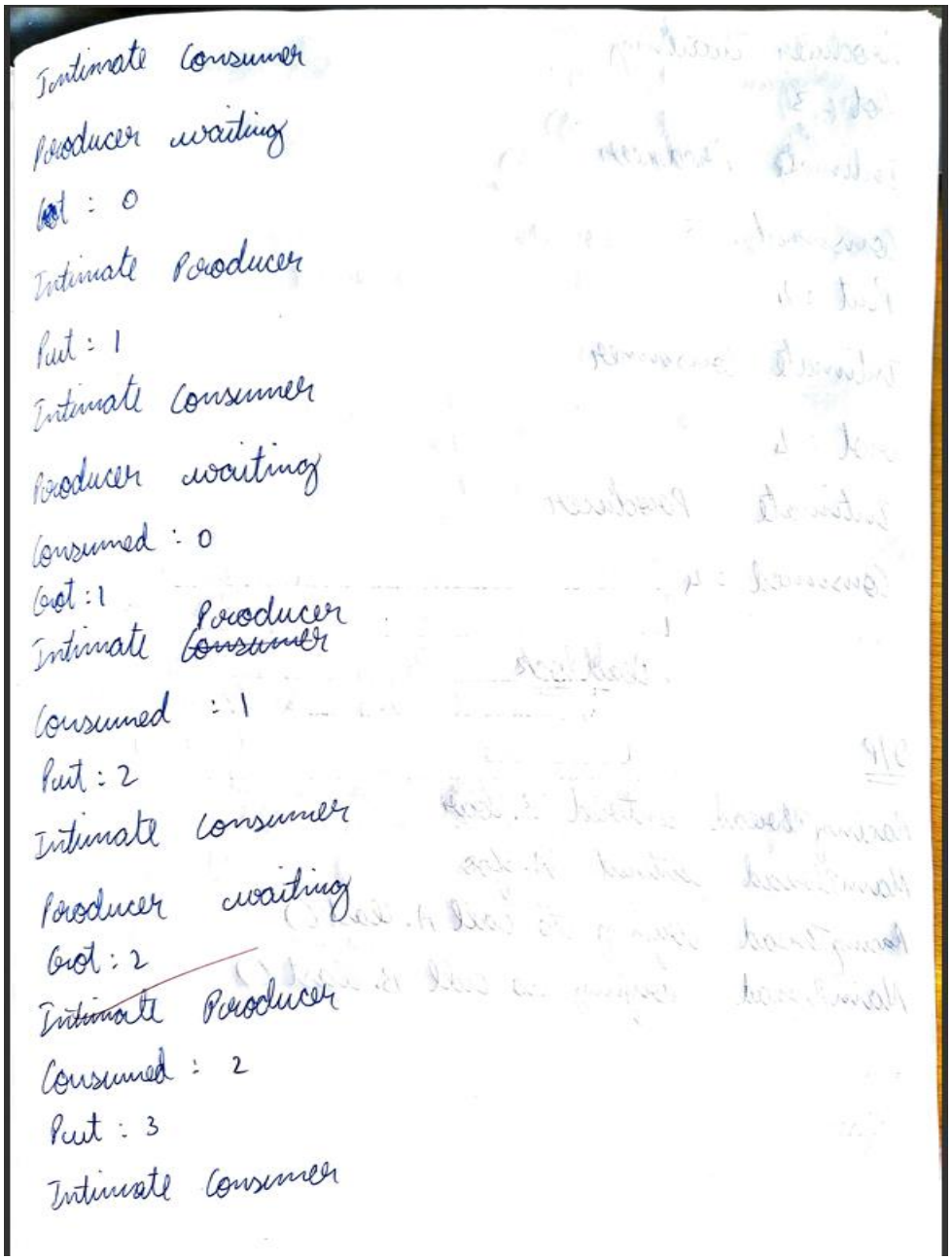
```
        catch (InterruptedException e) {
            System.out.println("InterruptedException caught");
        }
    }
    this.n = n;
    valueSet = true;
    System.out.println("Put: " + n);
    System.out.println("\n Intimate Consumer \n");
    notify();
}

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 < 3) {
            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 < 3) {
            int r = q.get();
            System.out.println("Consumed: " + r);
            i++;
        }
    }
}
class PCFixed {
    public static void main(String[] args)
    {
        System.out.println("Press Ctrl+C");
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C to stop.");
    }
}
```

Output

O/P
Press Control-C to stop.
Put : 0



Producer waiting
Got : 3
Intimate Producer
Consumed : 3
Put : 4
Intimate Consumer
Got : 4
Intimate Producer
Consumed : 4

Source 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();
   }
```

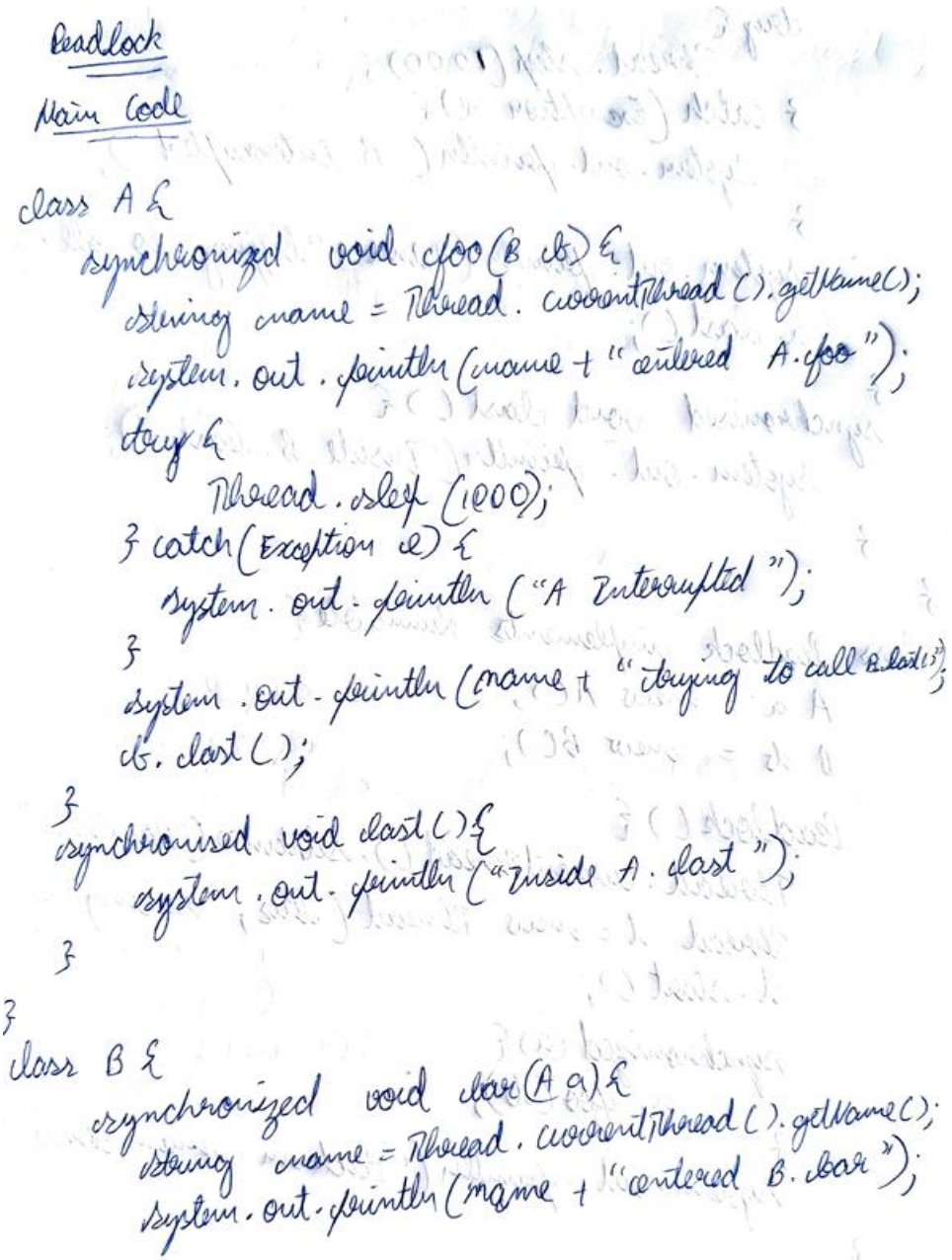
```
    public void run() {
        int i = 0;
        while (i < 3) {
            q.put(i++);
        }
    }
}

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 < 3) {
            int r = q.get();
            System.out.println("Consumed: " + r);
            i++;
        }
    }
}

class PCFixed {
    public static void main(String args[]) {
        System.out.println("Arnav Dinesh 1BM23CS052");
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C to stop.");
    }
}
```

Output

```
Arnav Dinesh 1BM23CS052
Put: 0

Intimate Consumer


Producer waiting

Press Control-C to stop.
Got: 0

Intimate Producer

Put: 1

Intimate Consumer


Producer waiting

Consumed: 0
Got: 1

Intimate Producer

Consumed: 1
Put: 2

Intimate Consumer

Got: 2

Intimate Producer

Consumed: 2
```

Week 10

Demonstrate Inter process Communication and deadlock



Deadlock

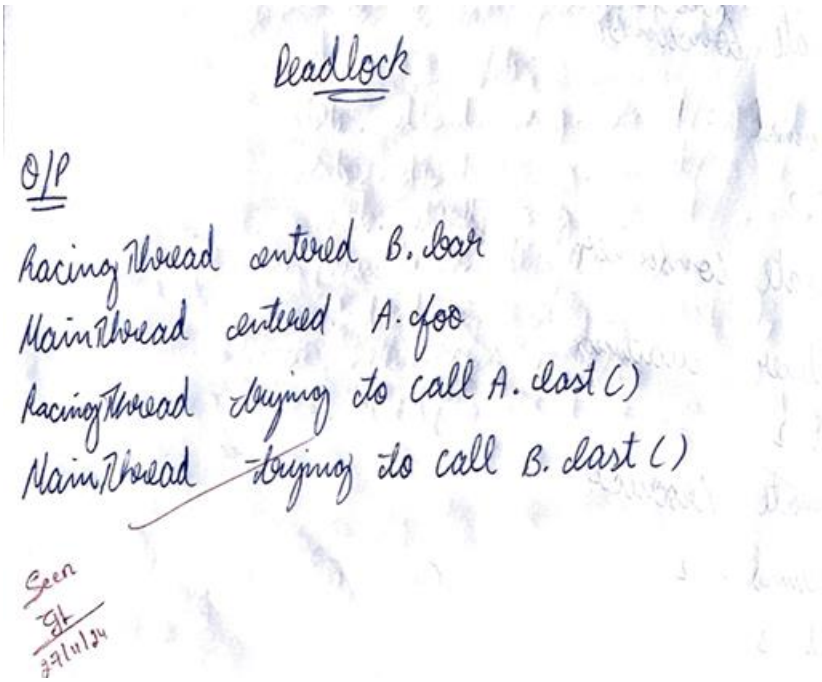
Main Code

```
class A {
    synchronized void cfoo(B b) {
        String cname = Thread.currentThread().getName();
        System.out.println(cname + "entered A.foo");
        try {
            Thread.sleep(1000);
        } catch(Exception e) {
            System.out.println("A Interrupted");
        }
        System.out.println(cname + "trying to call B.last()");
        b.last();
    }
    synchronised void last(){
        System.out.println("inside A.last");
    }
}
class B {
    synchronized void bar(A a) {
        String cname = 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
        a.last();
    }
    synchronised void last(){
        System.out.println("Inside B.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();
        synchronized(a){
            a.foo(b);
        }
        System.out.println("Back in main thread
    }
```

```
public void run() {
    synchronized (b) {
        b.bar(a);
    }
    System.out.println("Back in other thread");
}
public static void main(String[] args)
{
    System.out.println("\n");
    new Deadlock();
}
}
```

Output



Source 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();
```

```
    Deadlock() {
        Thread.currentThread().setName("MainThread");
        Thread t = new Thread(this, "RacingThread");
        t.start();

        synchronized (a) {
            a.foo(b);
        }

        System.out.println("Back in main thread");
    }

    public void run() {
        synchronized (b) {
            b.bar(a);
        }

        System.out.println("Back in other thread");
    }

    public static void main(String args[]) {
        System.out.println("Arnav Dinesh 1BM23CS052");
        new Deadlock();
    }
}
```

Output

```
Arnav Dinesh 1BM23CS052
MainThread entered A.foo
RacingThread entered B.bar
RacingThread trying to call A.last()
MainThread trying to call B.last()
```