VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

on

Object Oriented Java Programming (23CS3PCOOJ)

Submitted by

Aaryan Prakash (1BM23CS006)

in partial fulfilment 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 **Aaryan Prakash (1BM23CS006),** who is bona fide student of **B. M. S. College of Engineering.** It is in partial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2024. 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.

Geetha N Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor HOD Department of CSE, BMSCE

Index

Github Link:
https://github.com/TheAaryanPrakash/JavaLab/

Name: Aaryan Prakash
USN: 1BM23SC006
Class: 3A

Java Week 1: Quadratic Equations

Question: 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 b^2-4ac is negative, display a message stating that there are no real solutions.

Algorithm:

NO

DATE

```
import java.util.Scanner;

public class QuadraticEquations {

    public static void main(String a[]){

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a: ");
        double a = scanner.nextDouble();
        System.out.print("Enter b: ");
        double b = scanner.nextDouble();
        System.out.print("Enter c: ");
        double c = scanner.nextDouble();

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

        if (d>0){
            double root1 = (-b+ Math.sqrt(d))/2*a);
            double root2 = (-b - Math.sqrt(d) / 2*a);

            System.out.println("Real Roots");
            System.out.println("Root1: " + root1);
            System.out.println("Root2:" + root2);
        }
```

NO

DATE

```
else if (d == 0) {
    double root root1 = -b / (2 * d);
    System.out.println("Roots are Real and Equal");
    System.out.println("Root 1 and 2:" + root1);
}

else {
    System.out.println("Roots are Complex");
}

scanner.close();
}
}
```

Output

→ Enter a: 12
Enter b: 10
Enter c: 9
Roots are Complex

→ Enter a: 2
Enter b: 4
Enter c: 2

NO

DATE

Roots are real and equal

Root : -1.0

→ Enter a: 2

Enter b: 8

Enter c: 2

Real Roots

Root 1: -0.27

Root 2: -3.73

Seen

9/10/24

Code:

```
import java.util.Scanner;

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

      System.out.print("Enter a: ");
      double a = scanner.nextDouble();
      System.out.print("Enter b: ");
      double b = scanner.nextDouble();
```

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

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

            System.out.println("aaryan");

        if (d > 0) {
            double root1 = (-b + Math.sqrt(d)) / (2 * a);
            double root2 = (-b - Math.sqrt(d)) / (2 * a);

            System.out.println("Real Roots");
            System.out.println("Root 1: " + root1);
            System.out.println("Root 2: " + root2);
        }
            else if (d == 0) {
            double root = -b / (2 * a);
            System.out.println("Roots are real and equal");
            System.out.println("Root: " + root);
        }
            else {
            System.out.println("Roots are complex");
        }

        scanner.close();
    }
}
```

Output:

```
D:\1bm23cs006>javac QuadraticEquations.java

D:\1bm23cs006>java QuadraticEquations
Enter a: 12
Enter b: 10
Enter c: 9
aaryan
Roots are complex

D:\1bm23cs006>java QuadraticEquations
Enter a: 2
Enter b: 4
Enter c: 2
aaryan
Roots are real and equal
Root: -1.0

D:\1bm23cs006>java QuadraticEquations
Enter a: 2
Enter b: 8
Enter c: 2
aaryan
Real Roots
Root 1: -0.2679491924311228
Root 2: -3.732050807568877

D:\1bm23cs006>
```

Java Week 2: SGPA Calculator

Question: Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

Algorithm:

NO

DATE 16/10/2024

Lab 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.

```
import java.util.Scanner;

class Subject{
    int subjectMarks;
    int credits;
    int grade;

    void calculateGrade(){
        if(subjectMarks >= 90){
            grade = 10;
        else if (subjectMarks >= 80)
            grade = 9;
        else if (subjectMarks >= 70)
            grade = 8;
        else if (subjectMarks >=60)
            grade = 7;
        else if (subjectMarks ≥ 50)
            grade = 6;
        else if (subjectMarks ≥ 40)
            grade = 5;
```

NO

DATE

```
        else
            grade = 0;
        }
    }

class Student {
    String usn;
    String name;
    double SGPA;
    Subject [] subjects = new Subject [8];
    Scanner scanner = new Scanner ( system.in);

    Student (){
        for( int i=0; i<8; i++) {
            subjects [i] = new Subject ();
        }
    }

    void getStudentDetails {
        System. out. println ("Enter USN:");
        usn = scanner. next();
        System. out. println ("Enter Name:");
        name = scanner. next();
    }
```

NO

DATE

```
void getMarks(){
    for (int i=0; i<8; i++){
        System.out.print("Enter Marks for Subject "+
        (i+1)+":");
        subjects[i].subjectMarks = scanner.nextInt(
        System.out.print("Enter Credits for Subject "+
        (i+1)+":");
        subjects[i].credits = scanner.nextInt();
        subject[i].calculateGrade();
    }
}

Void calculateSGPA(){
    double effectiveScore = 0;
    int totalCredits = 0;

    for (int i=0; i<8; i++){
        effectiveScore += (subjects[i].grade * subjects[i].
                                                    credits)
        totalCredits += subjects[i].credits;
    }
    SGPA = effectiveScore/totalCredits;
}
```

NO

DATE

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

class StudDetails{
    public static void main (String[] args){
        Student[] students = new Student[3];
        for(int i=0; i<3; i++){
            System.out.println("Enter Student Details:");
            students[i] = new Student();
            students[i].getStudentDetails();
            students[i].getMarks();
            students[i].calculateSGPA();
    }   }
    for(int i=0; i<3; i++){
            students[i].display();
        }
    }
}
```

Seen
execute

NO

DATE

Output

Enter the details for Student 1:

Enter the USN: 1BM23CS006

Enter the Name: Aaryan

Enter Marks for Subject 1: 98

Enter Credits for Subject 1: 4

Enter Marks for Subject 2: 91

Enter Credits for Subject 2: 4

Enter Marks for Subject 3: 78

Enter Credits for Subject 3: 3

Enter Marks for Subject 4: 89

Enter Credits for Subjects 4: 3

Enter Marks for Subject 5: 98

Enter Credits for Subject 5: 3

Enter Marks for Subject 6: 71

Enter Credits for Subject 6: 1

Enter Marks for Subject 7: 81

Enter Credits for Subject 7: 1

Enter Marks for Subject 8: 99

Enter Credits for Subject 8: 1

→ USN: 1BM23CS006

Name: Aaryan

SGPA: 9.4

Seen

16/10/24

Code:

```
import java.util.Scanner;

class Subject {
   int subjectMarks;
   int credits;
   int grade;

   void calculateGrade() {
      if (subjectMarks >= 90) {
         grade = 10;
      } else if (subjectMarks >= 80) {
         grade = 9;
      } else if (subjectMarks >= 70) {
         grade = 8;
      } else if (subjectMarks >= 60) {
         grade = 7;
      } else if (subjectMarks >= 50) {
         grade = 6;
      } else if (subjectMarks >= 40) {
         grade = 5;
      } else {
         grade = 0;
      }
   }
}

class Student {
   String usn;
   String name;
   double SGPA;
   Subject[] subjects = new Subject[8];
   Scanner scanner = new Scanner(System.in);

   Student() {
      for (int i = 0; i < 8; i++) {
         subjects[i] = new Subject();
      }
   }

   void getStudentDetails() {
      System.out.print("Enter the USN: ");
      usn = scanner.next();
      System.out.print("Enter the Name: ");
      name = scanner.next();
   }
```

```
    void getMarks() {
        for (int i = 0; i < 8; i++) {
            System.out.print("Enter marks for subject " + (i + 1) + ": ");
            subjects[i].subjectMarks = scanner.nextInt();
            System.out.print("Enter credits for subject " + (i + 1) + ": ");
            subjects[i].credits = scanner.nextInt();
            subjects[i].calculateGrade();
        }
    }

    void computeSGPA() {
        double effectiveScore = 0;
        int totalCredits = 0;

        for (int i = 0; i < 8; i++) {
            effectiveScore += (subjects[i].grade * subjects[i].credits);
            totalCredits += subjects[i].credits;
        }
        SGPA = (totalCredits > 0) ? (effectiveScore / totalCredits) : 0;
    }

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

public class StudDetails {
    public static void main(String[] args) {
            System.out.println("aaryan prakash");
        Student[] students = new Student[3];

        for (int j = 0; j < 3; j++) {
            System.out.println("Enter the details for student " + (j + 1) + ":");
            students[j] = new Student();
            students[j].getStudentDetails();
            students[j].getMarks();
            students[j].computeSGPA();
        }

        for (int i=0;i<3;i++) {
            students[i].display();
        }
    }
}
```

Output:

```
C:\Users\Admin>D:

D:\>cd 1BM23CS006

D:\1BM23CS006>javac StudDetails.java

D:\1BM23CS006>java StudDetails
aaryan prakash
Enter the details for student 1:
Enter the USN: 1BM23CS006
Enter the Name: Aaryan
Enter marks for subject 1: 98
Enter credits for subject 1: 4
Enter marks for subject 2: 91
Enter credits for subject 2: 4
Enter marks for subject 3: 78
Enter credits for subject 3: 3
Enter marks for subject 4: 89
Enter credits for subject 4: 3
Enter marks for subject 5: 98
Enter credits for subject 5: 3
Enter marks for subject 6: 71
Enter credits for subject 6: 1
Enter marks for subject 7: 81
Enter credits for subject 7: 1
Enter marks for subject 8: 99
Enter credits for subject 8: 1
Enter the details for student 2:
Enter the USN: 1BM23CS003
Enter the Name: Aaron
Enter marks for subject 1: 94
Enter credits for subject 1: 4
Enter marks for subject 2: 99
Enter credits for subject 2: 4
Enter marks for subject 3: 93
Enter credits for subject 3: 3
Enter marks for subject 4: 97
Enter credits for subject 4: 3
Enter marks for subject 5: 81
Enter credits for subject 5: 3
Enter marks for subject 6: 99
Enter credits for subject 6: 1
Enter marks for subject 7: 99
Enter credits for subject 7: 1
Enter marks for subject 8: 99
Enter credits for subject 8:
1
```

```
Enter the details for student 3:
Enter the USN: 1BM23CS016
Enter the Name: Afreen
Enter marks for subject 1: 97
Enter credits for subject 1: 4
Enter marks for subject 2: 82
Enter credits for subject 2: 4
Enter marks for subject 3: 91
Enter credits for subject 3: 3
Enter marks for subject 4: 93
Enter credits for subject 4: 3
Enter marks for subject 5: 85
Enter credits for subject 5: 3
Enter marks for subject 6: 89
Enter credits for subject 6: 1
Enter marks for subject 7: 91
Enter credits for subject 7: 1
Enter marks for subject 8: 99
Enter credits for subject 8: 1
```

```
USN: 1BM23CS006
Name: Aaryan
SGPA: 9.4
USN: 1BM23CS003
Name: Aaron
SGPA: 9.85
USN: 1BM23CS016
Name: Afreen
SGPA: 9.6
```

Java Week 3: Book Class

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

Algorithm:

NO

DATE

```
import java.util.Scanner;

class Book{
    String name;
    String author;
    int price;
    int numPages;

    Book(String name, String author, int price, int numPages){
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }

    @Override
    public String toString(){
        String bookDetails = "Book Name: " + this.name + "\n" +
            "Author Name: " + this.author + "\n" +
            "Price: " + this.price + "\n" + "Number of
            Pages: " + this.numPages + "\n";
        return bookDetails;
    }
}
```

NO

DATE

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

        System.out.print ("Enter the No. of Books: ");
        int n = s.nextInt();

        Book[] books = new Book[n];

        for (int i=0; i<n ; i++){
            System.out.print ("Enter Name of Book "+(i+1)+ ": ");
            String name = s.next();
            System.out.print ("Enter Author of Book "+(i+1)+ ":");
            String author = s.next();
            System.out.print = ("Enter Price of Book "+(i+1)+":");
            int price = s.nextInt();
            System.out.print = ("Enter No. of Pages in Book"+(i+1)+":");
            int numPages= s.nextInt();

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

NO

DATE

```
        System.out.println("\n Book Details: ");
        for (Book
                System.out.println(book);
        }

        s.close();
    }
}
```

Output

```
Enter the Number of Books: 3
Enter Name of Book 1: Divergent
Enter Author of Book 1: R.Veronica
Enter Price of Book 1: 499
Enter No. of Pages in Book 1: 390
Enter Name of Book 2: Emma
Enter Author of Book 2: J. Austen
Enter Price of Book 2: 399
Enter No. of Pages in Book 2: 300
Enter Name of Book 3: Rebecca
Enter Author of Book 3: D.Maurier
Enter Price of Book 3: 699
Enter No of Pages in Book 3: 469
```

NO

DATE

Book Details:

Book Name: Divergent

Author Name: R. Veronica

Price: 499

Number of Pages: 390

Book Name: Emma

Author Name: J. Austen

Price: 399

Number of Pages: 300

Book Name: Rebecca

Author Name: D. Maurier

Price: 699

Number of Pages: 469

O/P seen

Code:

```
import java.util.Scanner;

class Book {
   String name;
   String author;
   int price;
   int numPages;

   Book(String name, String author, int price, int numPages) {
      this.name = name;
      this.author = author;
      this.price = price;
      this.numPages = numPages;
   }

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

public class BooksData {
   public static void main(String[] args) {
         System.out.println("Aaryan Prakash");
      Scanner s = new Scanner(System.in);

      System.out.print("Enter the Number of Books: ");
      int n = s.nextInt();

      Book[] books = new Book[n];

      for (int i = 0; i < n; i++) {
         System.out.print("Enter name of book " + (i + 1) + ": ");
         String name = s.next();
         System.out.print("Enter author of book " + (i + 1) + ": ");
         String author = s.next();
         System.out.print("Enter price of book " + (i + 1) + ": ");
         int price = s.nextInt();
         System.out.print("Enter number of pages in book " + (i + 1) + ": ");
         int numPages = s.nextInt();

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

```
        System.out.println("\nBook Details:");
        for (Book book : books) {
            System.out.println(book);
        }

        s.close();
    }
}
```

Output:

```
C:\Users\Admin>D:

D:\1BM23CS006>javac BooksData.java

D:\1BM23CS006>java BooksData
Aaryan Prakash
Enter the Number of Books: 3
Enter name of book 1: Divergent
Enter author of book 1: R.Veronica
Enter price of book 1: 500
Enter number of pages in book 1: 390
Enter name of book 2: Emma
Enter author of book 2: J.Austen
Enter price of book 2: 399
Enter number of pages in book 2: 300
Enter name of book 3: Rebecca
Enter author of book 3: D.Maurier
Enter price of book 3: 699
Enter number of pages in book 3: 469

Book Details:
Book Name: Divergent
Author Name: R.Veronica
Price: 500
Number of Pages: 390

Book Name: Emma
Author Name: J.Austen
Price: 399
Number of Pages: 300

Book Name: Rebecca
Author Name: D.Maurier
Price: 699
Number of Pages: 469


D:\1BM23CS006>
```

Java Week 4: Shape Area

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

Algorithm:

NO

DATE 23/10/2024

Lab Program 4:

Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named rectangle, triangle, 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.Scanner;

abstract class Shape{
        int dim1;
        int dim2;
}

        public Shape (){
                this.dim1 = 0;
                this.dim2 = 0;
        }

        public abstract void printArea();
}
```

NO

DATE

```
class Rectangle extends Shape{
    public Rectangle (int length, int width){
        dim1 = length;
        dim 2 = width;
    }

    public void printArea(){
        int area = dim1 * dim2;
        System.out.println("Area of Rectangle : "+area);
    }
}

class Triangle extends Shape{
    public Triangle (int base , int height){
        dim1= base;
        dim2= height;
    }
```

NO

DATE

```
    public void printArea (){
        double area = 0.5 * dim1 * dim2;
        System.out.println ("Area of Triangle: " + area);
    }
}

class Circle extends Shape{
    public Circle (int radius){
        dim1 = radius;
        dim2 = 0;
    }

    public void printArea (){
        double area = Math.PI * dim1 * dim1;
        System.out.println ("Area of Circle: " + area);
    }
}

public class Shapes{
    public static void main (String[] args){
        Scanner in = new Scanner (System.in);
        System.out.println ("Enter length and width of
        Rectangle: ");
        int length = in.nextInt();
```

NO

DATE

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

System.out.println("Enter Base and Height of
Triangle: ");
int base = in.nextInt();
int height = in.nextInt();
Shape triangle = new Triangle (base, height);
triangle.printArea();

System.out.println("Enter Radius of Circle:");
int radius = in.nextInt();
Shape circle = new Circle (radius);
circle.printArea();

in.close();
    }
}
```

NO

DATE

Output

Enter length and width for Rectangle:

12

6

Area of Rectangle: 72

Enter base and height for Triangle:

8

44

Area of Triangle: 176

Enter radius for Circle:

16

Area of Circle: 804.2477

O/P Seen

GL

23/10/24

Code:

```
import java.util.Scanner;

abstract class Shape {
   int dim1;
   int dim2;


   public Shape() {
      this.dim1 = 0;
      this.dim2 = 0;

   }

   public Shape(int dim1, int dim2) {
      this.dim1 = dim1;
      this.dim2 = dim2;
   }

   public abstract void printArea();

}

class Rectangle extends Shape {
   public Rectangle(int length, int width) {
      dim1 = length;
      dim2 = width;
   }

   public void printArea() {

      int area = dim1 * dim2;
      System.out.println("Area of Rectangle: " + area);

   }
}

class Triangle extends Shape {
   public Triangle(int base, int height) {
      dim1 = base;
      dim2 = height;
   }

   public void printArea() {

      double area = 0.5 * dim1 * dim2;
      System.out.println("Area of Triangle: " + area);
```

```
    }
}

class Circle extends Shape {
    public Circle(int radius) {

        dim1 = radius;
        dim2 = 0;
    }

    public void printArea() {

        double area = Math.PI * dim1 * dim1;
        System.out.println("Area of Circle: " + area);
    }
}

public class shapes {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
            System.out.println("Aaryan Prakash");
        System.out.println("Enter length and width for Rectangle:");

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

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

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

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

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

        in.close();
    }
}
```

Output:

```
D:\1BM23CS006>javac shapes.java

D:\1BM23CS006>java shapes
Aaryan Prakash
Enter length and width for Rectangle:
12
6
Area of Rectangle: 72
Enter base and height for Triangle:
8
44
Area of Triangle: 176.0
Enter radius for Circle:
16
Area of Circle: 804.247719318987

D:\1BM23CS006>
```

Java Week 5: Bank Account

Question: Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

i. Accept deposit from customer and update the balance.
ii. Display the balance.
iii. Compute and deposit interest
iv. Permit withdrawal and update the balance
v. Check for the minimum balance, impose penalty if necessary and update the balance.

Algorithm:

NO

DATE

```
import java.util.Scanner;

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

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

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

NO

DATE

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

class SavAcct extends Account{
    private double interstRate;

    public SavAcct (String customerName, int accountNumber,
    double Balance, double interestRate){
        Super( customerName, accountNumber, balance);
        this.interestRate = interestRate;
    }

    public void compDepInterest (){
        double interest = balance * (interestRate /100);
        balance += interset;
        System.out.println ("Interest Added: " + interest);
    }

    public void withdraw (double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println ("Withdrawn: " + amount);
```

NO

DATE

```
        } else{
            System.out.println("Insufficient Balance
                for Withdrawal");
        }
    }
}

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

    public CurAcct(String customerName, int accountNumber,
        double balance, double minimumBalance, double
        serviceCharge){
            super(customerName, accountNumber, balance);
            this.minimumBalance = minimumBalance;
            this.serviceCharge = serviceCharge;
    }

    public void withdraw(double amount){
        if(amount <= balance){
            balance -= amount;
            System.out.println("Withdrawn: " + amount);
```

DATE

```
        if (balance < minimumBalance){
            balance -= serviceCharge;
            System.out.println("Service
            Charge imposed: " + serviceCharge);
        }
    } else {
        System.out.println("Insufficient Balance for
            Withdrawal");
    }
}

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

        SavAcct savAcc = new SavAcct ("Aaryan", 12345,
            1000, 5);

        CurAcct curAcc = new CurAcct ("Jay", 67890, 2000,
            500, 50);

        System.out.println("Choose Account Type: \n1:
            Savings Account \n2: Current Account");
```

NO

DATE

```
int choice = sc.nextInt();

switch (choice) {
    case 1:
        System.out.println("Savings Account
            Selected");
        savAcc.deposit(500);
        savAcc.compDepInterest();
        savAcc.withdraw(300);
        savAcc.displayBalance();
        break;

    case 2:
        System.out.println("Current Account
            Selected");
        curAcc.deposit(500);
        curAcc.withdraw(1800);
        curAcc.displayBalance();
        break;

    default:
        System.out.println("Invalid Choice");
}
sc.close();
}
}
```

DATE

Output

1. Savings Account
2. Current Account

1

Savings Account Selected

Deposited : 500.0

Interest added: 75.0

Withdrawn : 300.0

Balance : 1275.0

Choose Account Type :

1. Savings Account
2. Current Account

2

Current Account Selected

Deposited : 500.0

Withdrawn : 1800.0

Balance : 7000

13/11/24

Code:

```
import java.util.Scanner;

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

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

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

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

class SavAcct extends Account {
   private double interestRate;

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

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

   public void withdraw(double amount) {
      if (amount <= balance) {
         balance -= amount;
         System.out.println("Withdrawn: " + amount);
      } else {
```

```
            System.out.println("Insufficient balance for withdrawal");
        }
    }
}

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

    public CurAcct(String customerName, int accountNumber, double balance, double
minimumBalance, double serviceCharge) {
        super(customerName, accountNumber, balance);
        this.minimumBalance = minimumBalance;
        this.serviceCharge = serviceCharge;
    }

    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn: " + amount);

            if (balance < minimumBalance) {
                balance -= serviceCharge;
                System.out.println("Service charge imposed: " + serviceCharge);
            }
        } else {
            System.out.println("Insufficient balance for withdrawal");
        }
    }
}

public class Bank {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        SavAcct savAcc = new SavAcct("Alice", 12345, 1000, 5);
        CurAcct curAcc = new CurAcct("Bob", 67890, 2000, 500, 50);

        System.out.println("Choose Account Type:\n1. Savings Account\n2. Current Account");
        int choice = sc.nextInt();

        switch (choice) {
            case 1:
                System.out.println("Savings Account Selected");
                savAcc.deposit(500);
                savAcc.computeAndDepositInterest();
                savAcc.withdraw(300);
                savAcc.displayBalance();
                break;
```

```
            case 2:
                System.out.println("Current Account Selected");
                curAcc.deposit(500);
                curAcc.withdraw(1800);
                curAcc.displayBalance();
                break;

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

        sc.close();
    }
}
```

Output:

```
Choose Account Type:
1. Savings Account
2. Current Account
1
Savings Account Selected
Deposited: 500.0
Interest added: 75.0
Withdrawn: 300.0
Balance: 1275.0
```

```
Choose Account Type:
1. Savings Account
2. Current Account
2
Current Account Selected
Deposited: 500.0
Withdrawn: 1800.0
Balance: 700.0
```

Java Week 6: SEE Marks

Question: Create a package CIE which has two 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 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.

Algorithm:

NO

DATE

CIE/Student.java

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

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

    public void inputStudentDetails(){
        Scanner s = new Scanner(System.in)
        System.out.print("Enter USN: ");
        usn = s.nextline();
        System.out.print("Enter Name: ");
        name = s.nextline();
        System.out.print("Enter Semester: ");
        sem = s.nextInt();
    }

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

NO

DATE

CIE/Internals.java

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

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

    public void inputCIEmarks() {
        Scanner S = new Scanner(System.in);
        System.out.println("Enter Internal Marks for 5
        Subjects: ");
        for(int i=0; i<5; i++) {
            System.out.print("Subject " + (i+1) + ":");
            internalMarks[i] = S.nextInt();
        }
    }
}
```

SEE/Externals.java

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

NO

DATE

```
public class Externals extends Internals {
    protected int[] seeMarks = new int[5];
    protected int[] finalMarks = new int[5];

    public void inputSEEmarks() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter SEE Marks for 5
        Subjects");
        for(int i=0; i<5; i++) {
            System.out.println("Subject " + (i+1) + ":");
            seeMarks[i]=s.nextInt();
        }
    }

    public void calculateFinalMarks() {
        for(int i=0; i<5; i++) {
            finalMarks[i] = internalMarks[i] + seeMarks[i];
        }
    }

    public void displayFinalMarks() {
        displayStudentDetails();
        System.out.println("Final Marks for 5 Subjects:");
        for(int i=0; i<5; i++)
```

NO

DATE

```
        {
            System.out.println("Subject" + (i+1) + ":" +
            finalMarks[i]);
        }
    }
}
```

Main.java

~~import SEE.*~~

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

class Main{
    public static void main(String[] args){
        Scanner s = new Scanner(System.in);
        System.out.print("Enter Number of Students: ");
        int n = s.nextInt();

        Externals[] students = new Externals[n];

        for(int i=0; i<n; i++){
            System.out.println("Enter Details for Student " +
            (i+1) + ":");
```

NO

DATE

```
        students[i] = new External();
        students[i].inputStudentDetails();
        students[i].inputCIEmarks();
        students[i].inputSEEmarks();
        students[i].calculateFinalMarks();
    }

    System.out.println("\n Final Marks of Students: ");
    for(int i=0; i<n; i++){
        System.out.println("\n Student "+(i+1)+":");
        students[i].displayFinalMarks();
    }
}
}
```

Scan

Output

```
Enter number of students: 1

Enter details for student 1:
Enter USN: Aaryan
Enter Name: 1BM23CS006
Enter Semester: 3
Enter Internal Marks for 5 Subjects:
Subject 1: 45
```

DATE

Subject 2: 49

Subject 3: 42

Subject 4: 48

Subject 5: 46

Enter SEE Marks for 5 Subjects:

Subject 1: 44

Subject 2: 48

Subject 3: 50

Subject 4: 48

Subject 5: 49

Final Marks of Students:

Student 1:

USN: 1BM23CS006

Name: Aaryan

Semester: 3

Subject 1: 89

Subject 2: 94

Subject 3: 92

Subject 4: 96

Subject 5: 95

Executed

13/11/24

Code:

CIE/Student.java

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

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

   public void inputStudentDetails() {
      Scanner s = new Scanner(System.in);
      System.out.print("Enter USN: ");
      usn = s.nextLine();
      System.out.print("Enter Name: ");
      name = s.nextLine();
      System.out.print("Enter Semester: ");
      sem = s.nextInt();
   }

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

CIE/Internals.java

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

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

   public void inputCIEmarks() {
      Scanner s = new Scanner(System.in);
      System.out.println("Enter Internal Marks for 5 subjects:");
      for (int i = 0; i < 5; i++) {
         System.out.print("Subject " + (i + 1) + ": ");
         internalMarks[i] = s.nextInt();
      }
   }
}
```

SEE/Externals.java

```
package SEE;
import CIE.Internals;
```

```
import java.util.Scanner;

public class Externals extends Internals {
   private int[] seeMarks = new int[5];
   private int[] finalMarks = new int[5];

   public void inputSEEmarks() {
      Scanner s = new Scanner(System.in);
      System.out.println("Enter SEE Marks for 5 subjects:");
      for (int i = 0; i < 5; i++) {
         System.out.print("Subject " + (i + 1) + ": ");
         seeMarks[i] = s.nextInt();
      }
   }

   public void calculateFinalMarks() {
      for (int i = 0; i < 5; i++) {
         finalMarks[i] = internalMarks[i] + seeMarks[i];
      }
   }

   public void displayFinalMarks() {
      displayStudentDetails();
      System.out.println("Final Marks for 5 subjects:");
      for (int i = 0; i < 5; i++) {
         System.out.println("Subject " + (i + 1) + ": " + finalMarks[i]);
      }
   }
}
```

Main.java

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

class Main {
   public static void main(String[] args) {
      Scanner s = new Scanner(System.in);
      System.out.print("Enter number of students: ");
      int n = s.nextInt();

      Externals[] students = new Externals[n];

      for (int i = 0; i < n; i++) {
         System.out.println("\nEnter details for student " + (i + 1) + ":");
         students[i] = new Externals();
         students[i].inputStudentDetails();
         students[i].inputCIEmarks();
         students[i].inputSEEmarks();
```

```
            students[i].calculateFinalMarks();
        }

        System.out.println("\nFinal Marks of Students:");
        for (int i = 0; i < n; i++) {
            System.out.println("\nStudent " + (i + 1) + ":");
            students[i].displayFinalMarks();
        }
    }
}
```

Output:

```
D:\>cd javaoops

D:\javaoops>java Main
Enter number of students: 3

Enter details for student 1:
Enter USN: 1BM23CS006
Enter Name: Aaryan
Enter Semester: 3
Enter Internal Marks for 5 subjects:
Subject 1: 45
Subject 2: 49
Subject 3: 42
Subject 4: 48
Subject 5: 46
Enter SEE Marks for 5 subjects:
Subject 1: 44
Subject 2: 48
Subject 3: 50
Subject 4: 48
Subject 5: 49

Enter details for student 2:
Enter USN: 1BM23CS006
Enter Name: Avinash
Enter Semester: 4
Enter Internal Marks for 5 subjects:
Subject 1: 37
Subject 2: 39
Subject 3: 41
Subject 4: 43
Subject 5: 45
Enter SEE Marks for 5 subjects:
Subject 1: 45
Subject 2: 45
Subject 3: 45
Subject 4: 40
Subject 5: 40

Enter details for student 3:
Enter USN: 1BM23CS003
Enter Name: Chandan
Enter Semester: 5
Enter Internal Marks for 5 subjects:
Subject 1: 38
Subject 2: 39
Subject 3: 41
Subject 4: 41
Subject 5: 47
Enter SEE Marks for 5 subjects:
Subject 1: 44
Subject 2: 45
Subject 3: 43
Subject 4: 27
Subject 5: 50
```

```
Final Marks of Students:

Student 1:
USN: 1BM23CS006
Name: Aaryan
Semester: 3
Final Marks for 5 subjects:
Subject 1: 89
Subject 2: 97
Subject 3: 92
Subject 4: 96
Subject 5: 95

Student 2:
USN: 1BM23CS006
Name: Avinash
Semester: 4
Final Marks for 5 subjects:
Subject 1: 82
Subject 2: 84
Subject 3: 86
Subject 4: 83
Subject 5: 85

Student 3:
USN: 1BM23CS003
Name: Chandan
Semester: 5
Final Marks for 5 subjects:
Subject 1: 82
Subject 2: 84
Subject 3: 84
Subject 4: 68
Subject 5: 97

D:\javaoops>
```

Java Week 7: Father-Son Age

Question: Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age=father's age.

Algorithm:

NO

DATE

```
import java.util.Scanner;

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

class Father {
    int age;
    public Father (int age) throws WrongAge {
        if (age < 0) {
            throw new WrongAge ("Age cannot be
                Negative !");
        }
        this.age = age;
        System.out.println("Father's Age: " + this.age);
    }
}

class Son extends Father {
    int sonAge ;
    public Son ( int fatherAge , int sonAge) throws WrongAge {
        super (fatherAge);
```

NO

DATE

```
        if (sonAge < 0){
            throw new WrongAge("Son's Age Cannot
                be Negative!");
        }
        if (sonAge >= fatherAge){
            throw new WrongAge("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 FatherSon{
    public static void main(String[] args){
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter Father's Age: ");
        int fatherAge = scanner.nextInt();
        System.out.print("Enter Son's Age: ");
        int sonAge = scanner.nextInt();
        try{
            Son son = new Son(fatherAge, sonAge);
        } catch(WrongAge e){
            System.out.println("Exception: "+e.getMessage());
        }
```

```
        Scanner.close();
    }
}
```

Output

i)

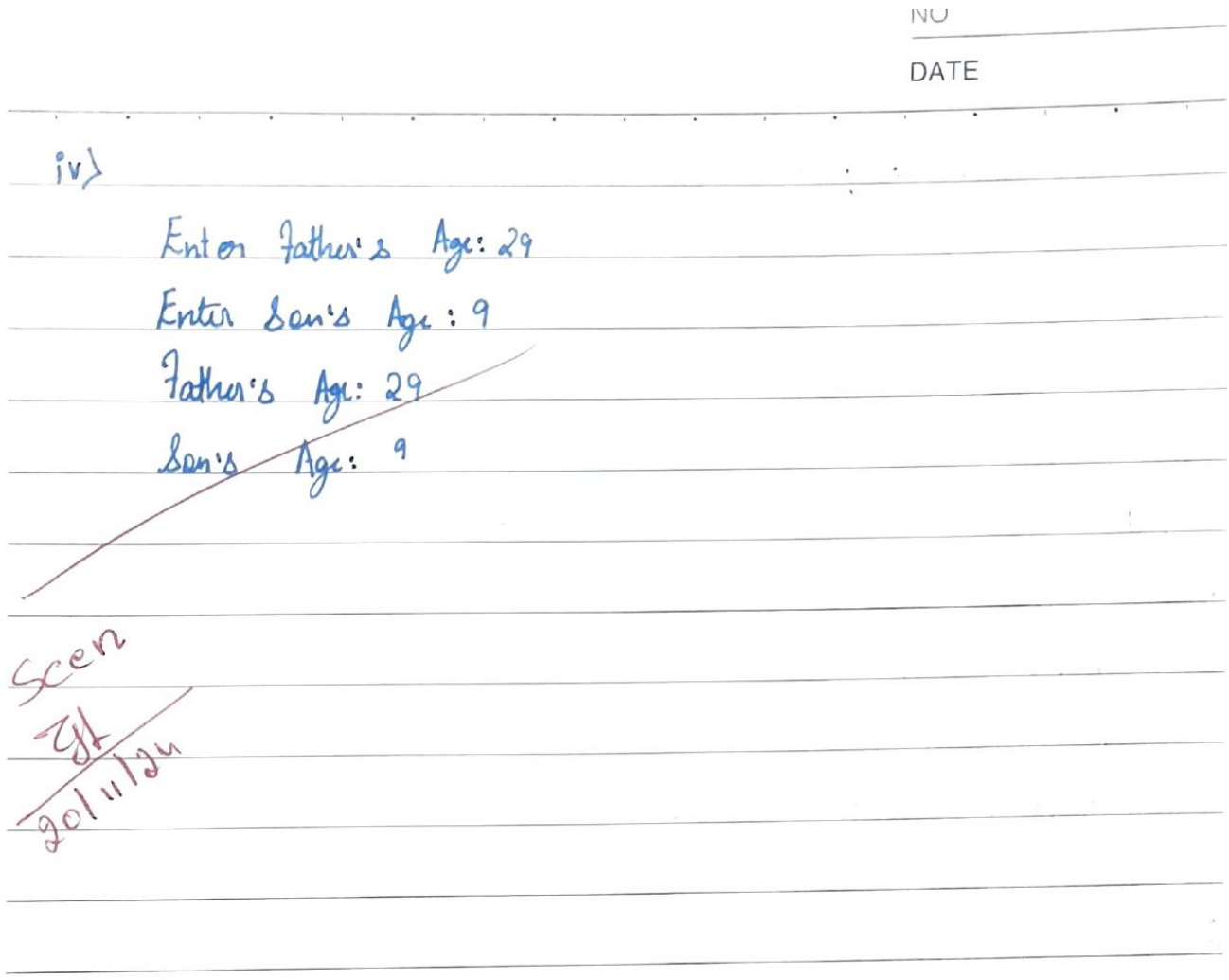
```
Enter Father's Age: -1
Enter Son's Age: 10
Exception: Age Cannot be Negative
```

ii)

```
Enter Father's Age: 25
Enter Son's Age: -3
Father's Age : 25
Exception: Son's Age Cannot be Negative
```

iii)

```
Enter Father's Age: 25
Enter Son's Age: 30
Father's Age: 25
Exception: Son's Age Cannot be greater than Father's Age
```



Code:

```
import java.util.Scanner;

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

class Father {
   int age;

   public Father(int age) throws WrongAge {
      if (age < 0) {
         throw new WrongAge("Age Cannot be Negative");
      }
```

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

class Son extends Father {
    int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAge {
        super(fatherAge);
        if (sonAge < 0) {
            throw new WrongAge("Son's Age Cannot be Negative");
        }
        if (sonAge >= fatherAge) {
            throw new WrongAge("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 FatherSon {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter Father's Age: ");
        int fatherAge = scanner.nextInt();

        System.out.print("Enter Son's Age: ");
        int sonAge = scanner.nextInt();

        try {
            Son son = new Son(fatherAge, sonAge);
        } catch (WrongAge e){
            System.out.println("Exception: " + e.getMessage());
        }

        scanner.close();
    }
}
```

Output:

```
D:\1BM23CS006>javac FatherSon.java

D:\1BM23CS006>java FatherSon
Enter Father's Age: -1
Enter Son's Age: 10
Exception: Age Cannot be Negative

D:\1BM23CS006>java FatherSon
Enter Father's Age: 25
Enter Son's Age: -3
Father's Age: 25
Exception: Son's Age Cannot be Negative

D:\1BM23CS006>java FatherSon
Enter Father's Age: 25
Enter Son's Age: 30
Father's Age: 25
Exception: Son's Age Cannot be Greater than or Equal to Father's Age

D:\1BM23CS006>java FatherSon
Enter Father's Age: 29
Enter Son's Age: 9
Father's Age: 29
Son's Age: 9

D:\1BM23CS006>
```

Java Week 8: Threads

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

Algorithm:

NO

DATE 27/11/2024

Lab 8

Q) 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 CollegeThread extends Thread {
    public void run() {
        try {
            for (int i=0; i<5; i++) {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000);
            }
        } catch (InterruptedException e) {
            System.out.println("CollegeThread Interrupted");
        }
    }
}

class CSEThread extends Thread {
    public void run() {
        try {
            for (int i=0; i<25; i++) {
                System.out.println("CSE");
```

NO

DATE

```
            Thread.sleep(2000);
        }
    } catch (InterruptedException e){
        System.out.println("CSE Thread Interrupted");
    }
  }
}
```

Output

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

NO

DATE

CSE

CSE

CSE

BNS College of Engineering

CSE

CSE

CSE

CSE

CSE

BNS College of Engineering

CSE

CSE

CSE

CSE

CSE

Code:

```
class CollegeThread extends Thread {
    public void run() {
        try {
            for (int i = 0; i < 5; i++) {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000);
            }
        } catch (InterruptedException e) {
            System.out.println("CollegeThread interrupted.");
        }
    }
}

class CSEThread extends Thread {
    public void run() {
        try {
            for (int i = 0; i < 25; i++) {
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        } catch (InterruptedException e) {
            System.out.println("CSEThread interrupted.");
        }
    }
}

public class ThreadExample {
    public static void main(String[] args) {
        CollegeThread collegeThread = new CollegeThread();
        CSEThread cseThread = new CSEThread();

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

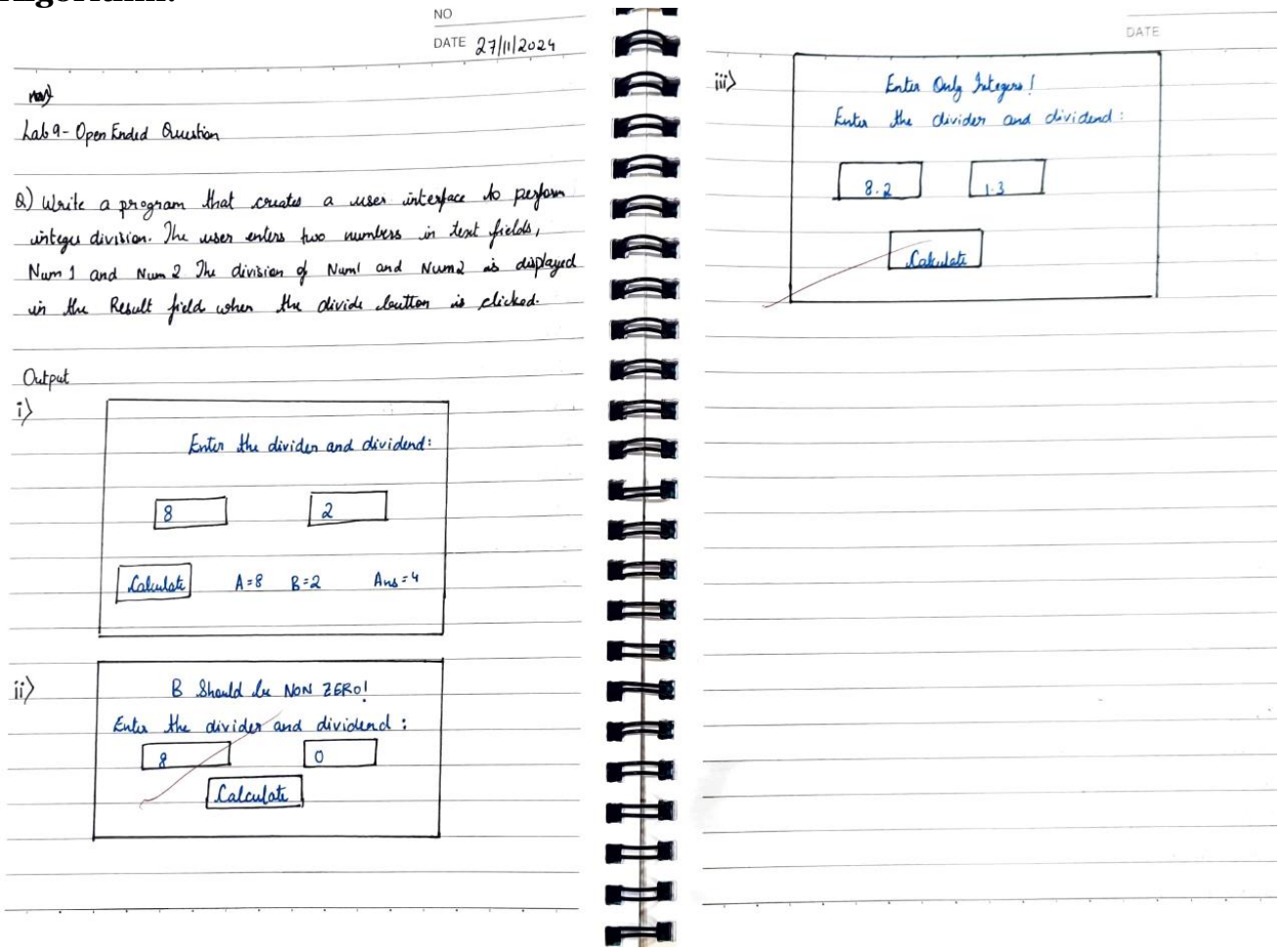
Output:

```
F:\Year 2\Semester 3\Object Oriented Programming in Java\Lab Class\Week 8>javac ThreadExample.java

F:\Year 2\Semester 3\Object Oriented Programming in Java\Lab Class\Week 8>java ThreadExample
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE

F:\Year 2\Semester 3\Object Oriented Programming in Java\Lab Class\Week 8>
```

Java Week 9: User Interface

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

Algorithm:



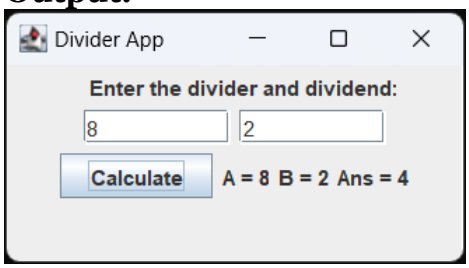
Code:

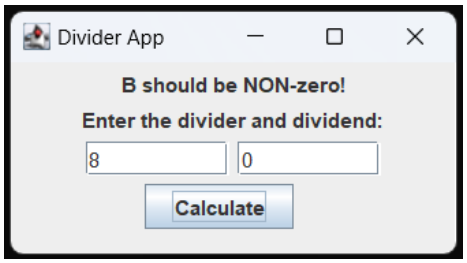
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class SwingDemo {
   SwingDemo() {
      // Create JFrame container
      JFrame jfrm = new JFrame("Divider App");
      jfrm.setSize(275, 150);
      jfrm.setLayout(new FlowLayout());
```

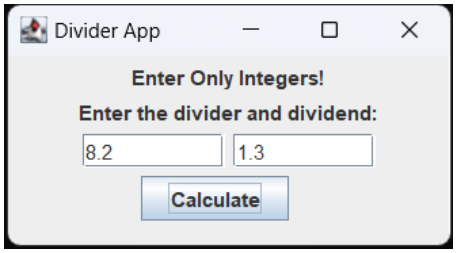
```
// Terminate on close
jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

// Create components
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();

// Add components in order
jfrm.add(err); // To display errors
jfrm.add(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);

// Add ActionListeners
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("A = " + a);
         blab.setText("B = " + b);
         anslab.setText("Ans = " + ans);
         err.setText(""); // Clear error message
      } 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 the 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:



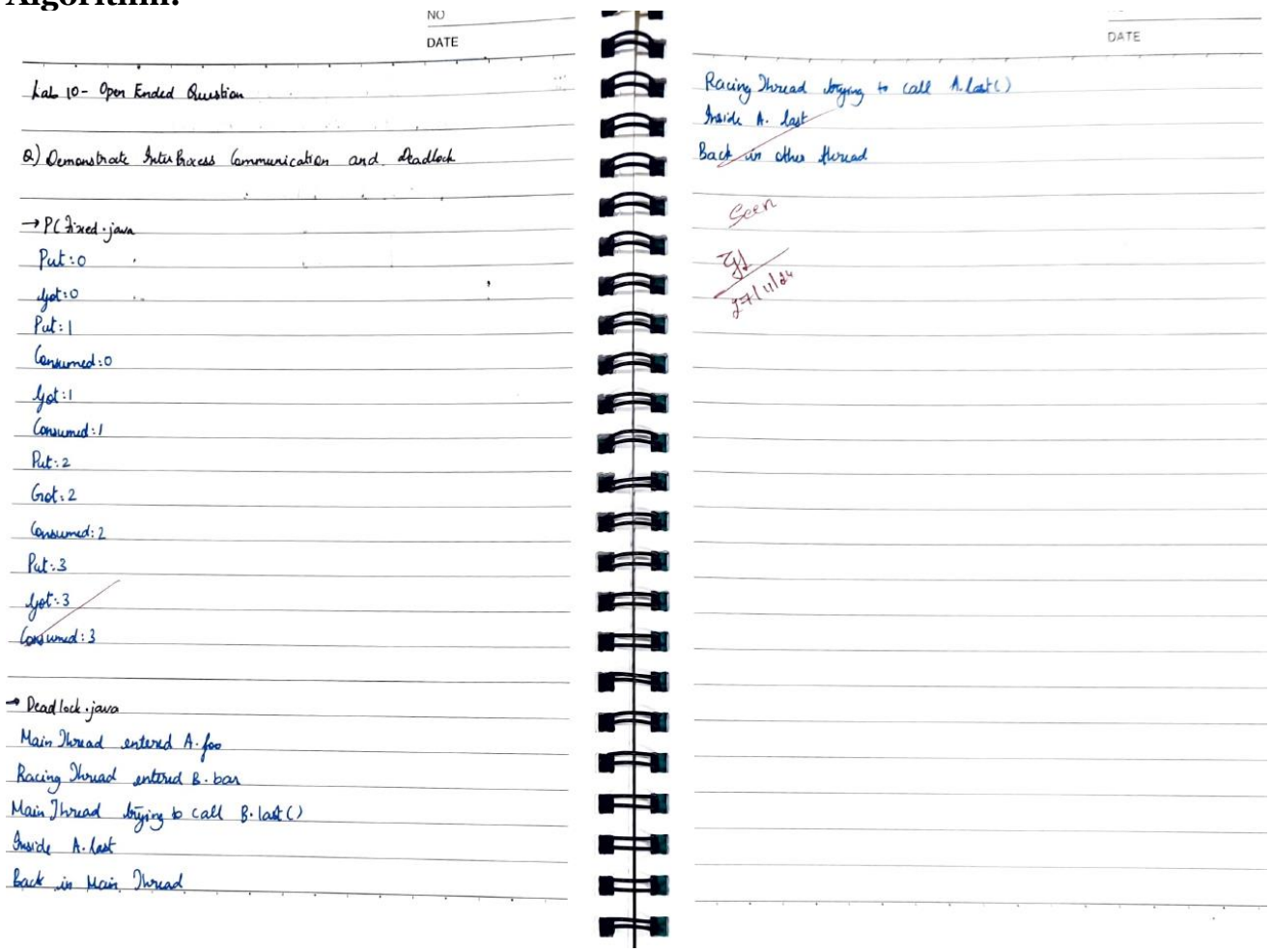




Java Week 10: Inter-Process Communication and Deadlocks

Question: Demonstrate Inter-Process Communication and Deadlock.

Algorithm:



PCFixed.java
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++;
      }
    }
}

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

Output:

```
F:\Year 2\Semester 3\Object Oriented Programming in Java\Lab Class\Week 9 and 10>java PCFixed
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

Put: 5

Intimate Consumer


Producer waiting

Consumed: 4
Got: 5
```

```
Intimate Producer

Consumed: 5
Put: 6

Intimate Consumer


Producer waiting

Got: 6

Intimate Producer

Consumed: 6
Put: 7

Intimate Consumer


Producer waiting

Got: 7

Intimate Producer

Consumed: 7
Put: 8

Intimate Consumer


Producer waiting

Got: 8
```

```
Consumed: 8
Got: 9

Intimate Producer

Put: 10

Intimate Consumer


Producer waiting

Consumed: 9
Got: 10

Intimate Producer

Consumed: 10
Put: 11

Intimate Consumer


Producer waiting

Got: 11

Intimate Producer

Consumed: 11
Put: 12

Intimate Consumer


Producer waiting

Got: 12
```

```
Intimate Producer

Consumed: 12
Put: 13

Intimate Consumer


Producer waiting

Got: 13

Intimate Producer

Consumed: 13
Put: 14

Intimate Consumer

Got: 14

Intimate Producer

Consumed: 14

F:\Year 2\Semester 3\Object Orien
```

Deadlock.java
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();

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

    public void run() {
        b.bar(a);
        System.out.println("Back in other thread");
    }

    public static void main(String args[]) {
        new Deadlock();
    }
}
```

Output:

```
F:\Year 2\Semester 3\Object Oriented
MainThread entered A.foo
RacingThread entered B.bar
MainThread trying to call B.last()
RacingThread trying to call A.last()
```