

**VISVESVARAYA TECHNOLOGICAL
UNIVERSITY**

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT on
Object Oriented Java Programming
(23CS3PCOOJ)

Submitted by

Anshuman Gupta (**1BM23CS043**)

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 **Anshuman Gupta (1BM23CS043)**, who is bonafide student of **B.M.S. College of Engineering**. It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

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

Index

Sl. No.	Date	Experiment Title	Page No.
1	9/10/24	Quadratic equation	4
2	16/10/24	Sgpa calculatosgpa calculatorr	9

3	23/10/24	Book program	22
4	23/10/24	Abstract class shape program	29
5	13/11/24	Bank problem	34
6	13/11/24	Packages	43
7	20/11/24	Exception handling	51
8	27/11/24	Multithreading	56
9	27/11/24	Inter division with user interface	60
10	27/11/24	Inter process communication and deadlock	66

Github Link:

https://github.com/Anshuman262005/JAVA_record

Program 1

Implement Quadratic Equation

```

if(discriminant() > 0) {
    r1 = (-b + Math.sqrt(discriminant()) / (double)(2 * a));
    r2 = (-b - Math.sqrt(discriminant()) / (double)(2 * a));
    System.out.println("The roots are unique");
    System.out.println("First root: " + r1);
    System.out.println("Second root: " + r2);
}
else if(discriminant() == 0) {
    r1 = -b / (2 * a);
    System.out.println("The roots are equal");
    System.out.println("The roots: " + r1);
}
else if(discriminant() < 0) {
    r1 = -b / (2 * a);
    r2 = (-b + Math.sqrt(-discriminant()) / (double)(2 * a));
    System.out.println("The roots are imaginary");
    System.out.println("First root: " + r1 + "i" + r2);
    System.out.println("Second root: " + r1 + "i" + r2);
}

```

class Quad
 public static void main(String [] args) {
 Quadratic eq1 = new Quadratic();
 eq1.compute();
 Quadratic eq2 = new Quadratic();
 eq2.compute();
 Quadratic eq3 = new Quadratic();
 eq3.compute();
 }

DATE: PAGE:

DATE: PAGE:

Output

Equilateral

~~3x² - 9 = 0~~
~~2x~~

Q) Program to get quadratic equation roots.

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

```
class Quadratic {  
    int a, b, c;  
    double r1, r2;  
    Quadratic () {
```

```
        System.out.println ("Enter a, b and c from quadratic equation :");  
        Scanner sc = new Scanner (System.in);
```

```
        a = sc.nextInt();
```

```
        b = sc.nextInt();
```

```
        c = sc.nextInt();
```

```
}
```

```
    void discriminant () {
```

```
        return b * b - 4 * a * c;
```

```
}
```

```
    void compute () {
```

E: PAGE:

1) (24a);
a);

DATE: PAGE:

Output

Enter a, b and c from quadratic equation:

1

2

1

The roots are equal

The root is : -1.0

Enter a, b, c from quadratic equation:

3

4

9

The roots are Imaginary

First root : 0.0 + i0.6314438411042392

Second root : 0.0 - i0.6314438411042392

Enter a, b and c from quadratic equation:

1

5

6

The roots are unique

First root : -2.0

Second root : -3.0

~~AB~~
QW-21

```

import java.lang.Math;
import java.util.Scanner;

class Quadratic{
int a, b, c;
double r1, r2;
Quadratic(){}
System.out.println("Enter a, b and c from quadratic equation: ");
Scanner sc = new Scanner(System.in);
a = sc.nextInt();
b = sc.nextInt();
c = sc.nextInt();
}
double discriminant(){
return b*b-4*a*c;
}
void compute(){
if(discriminant() > 0){
r1 = (-b + Math.sqrt(discriminant()))/(double)(2*a);
r2 = (-b - Math.sqrt(discriminant()))/(double)(2*a);
System.out.println("The roots are unique");
System.out.println("First root: "+ r1);
System.out.println("Second root: "+ r2);
}
else if(discriminant() == 0){
r1 = -b/(2*a);
System.out.println("The roots are equal");
System.out.println("The root is: "+ r1);
}
else if(discriminant() < 0){
r1 = -b/(2*a);
r2 = (-b + Math.sqrt(-discriminant()))/(double)(2*a);
System.out.println("The roots are Imaginary");
System.out.println("First root: "+ r1 + "+i" + r2);
System.out.println("Second root: "+ r1 + "-i" + r2);
}
}
}
}

```

```

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

```

```
Quadratic eq1 = new Quadratic();
eq1.compute();
Quadratic eq2 = new Quadratic();
eq2.compute();
Quadratic eq3 = new Quadratic();
eq3.compute();
}
}
```

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

C:\Users\Admin>cd desktop

C:\Users\Admin\Desktop>javac Run.java

C:\Users\Admin\Desktop>java Run
Enter a, b and c from quadratic equation:
1
2
1
The roots are equal
The root is: -1.0
Enter a, b and c from quadratic equation:
3
4
9
The roots are Imaginary
First root: 0.0+i0.9319438411042397
Second root: 0.0-i0.9319438411042397
Enter a, b and c from quadratic equation:
1
5
6
The roots are unique
First root: -2.0
Second root: -3.0
```

2.Student information

Enter Marks for Subject 3:
45

Enter Credits for Subject 5:
4

Enter Marks for Subject 4:
88

Enter Credits for Subject 4:
4

Enter Marks for Subject 5:
77

Enter ~~Marks~~ for Subject 5:
4

Enter marks for Subject 2:
88

Enter credits for Subject 7:
1

Enter marks for Subject 1:
85

Enter ~~marks~~ for Subject 9:
1

Enter details for Student 3:

Enter name:
Yash

Enter USN
125

Enter Marks for Subject 2:
50

Enter Credits for Subject 4:
3

Enter Marks for Subject 5:

60

Enter Credits for Subject 5:

4

Enter Marks for Subject 6:

75

Enter Credits for Subject 6:

2

Enter marks for Subject 7:

90

Enter Credits for Subject 7:

1

Enter Marks for Subject 8:

90

Enter Credits for Subject 8:

1

Enter details for student 2:

Enter Student name:

Yash

Enter Student USN:

124

Enter marks for subject 1:

90

Enter Credits for Subject 1:

3

Enter Marks for Subject 2:

66

Enter Credits for Subject 2:

3

```
System.out.println ("\\n \\n Results for all student:");  
for (int i = 0; i < 3; i++) {  
    student[i].displayResult();  
}
```

3

Output

Enter details for Student 1:

Enter Student Name:

Anshuman

Enter Student USN

123

Enter marks for Subject 1:

90

Enter Credits for Subject 1:

3

Enter marks for Subject 2:

70

Enter Credits for Subject 2

3

Enter Marks for Subject 3:

80

Enter Credits for Subject 3

4

Enter Marks for Subject 4:

90

Enter Credits for Subject 4:

4

```
public void computeSGPA() {
    int totalCredits = 0;
    int effectiveScore = 0;
```

```
for (int i = 0; i < 3; i++) {
    effectiveScore += (subject[i].grade * subject[i].credits);
    totalCredits += subject[i].credits;
}
SGPA = (double) effectiveScore / totalCredits;
```

```
public void displayResult() {
    System.out.println("In Student Name: " + name);
    System.out.println("Student USN: " + usn);
    System.out.println("SGPA: " + SGPA);
```

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

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

PAGE: DATE: PAGE:
 Subject() subject = new Subject();
 Scanner s = new Scanner(System.in);
 public Student() {
 for (int i = 0; i < 8; i++) {
 subject[i] = new Subject();
 }
 }
 public void getStudentDetails() {
 System.out.println("Enter Student Name:");
 name = s.nextLine();
 System.out.println("Enter Student USN:");
 USN = s.nextLine();
 }
 public void getMarks() {
 for (int i = 0; i < 8; i++) {
 System.out.print("Enter Marks for Subject " + (i + 1) + ": ");
 subject[i].SubjectMarks = s.nextInt();
 }
 }
 if (subject[i].SubjectMarks > 100 || subject[i].SubjectMarks < 0) {
 System.out.println("Invalid marks! Please enter again");
 }
 else {
 count++;
 }
 System.out.println("Enter Credits for Subject " + (i + 1) + ": ");
 subject[i].Credits = s.nextInt();
 subject[i].calculateGPA();
}

a) Student information

```
import java.util.Scanner;
```

```
class SubjectMarks {
    int subjectMarks;
    int credits;
    int grade;
```

```
public void calculateGrade() {
    if (subjectMarks >= 90 && subjectMarks <= 100) {
        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;
    }
}
```

```
3 class Student {
```

```
String name;
String vsn;
double SGPA;
```

```

import java.util.Scanner;

class Student {
    int n;
    String usn, name;
    float marks[];
    float credits[];
    Scanner sc = new Scanner(System.in);

    Student() {
        System.out.println("ENTER THE NUMBER OF SUBJECTS");
        this.n = sc.nextInt();
    }

    void create() {
        System.out.println("Enter the name of the Student");
        this.name = sc.nextLine();
        sc.nextLine();
        System.out.println("Enter the USN of the Student");
        this.usn = sc.nextLine();

        this.marks = new float[n];
        this.credits = new float[n];

        for (int i = 0; i < n; i++) {
            System.out.println("Enter the marks of subject " + (i + 1));
            marks[i] = sc.nextFloat();

            while (marks[i] < 0 || marks[i] > 100) {
                System.out.println("Invalid marks! Please enter valid marks (0-100): ");
                marks[i] = sc.nextFloat();
            }
        }

        System.out.println("Enter the credits of subject " + (i + 1));
        credits[i] = sc.nextFloat();

        while (credits[i] <= 0) {
            System.out.println("Invalid credits! Please enter valid credits (>0): ");
            credits[i] = sc.nextFloat();
        }
    }
}

```

```

int grade(float a) {
    if (a <= 100 && a > 90) {
        return 10;
    } else if (a > 80) {
        return 9;
    } else if (a > 70) {
        return 8;
    } else if (a > 60) {
        return 7;
    } else if (a > 50) {
        return 6;
    } else if (a > 40) {
        return 5;
    } else if (a > 30) {
        return 4;
    } else if (a > 20) {
        return 3;
    } else if (a > 10) {
        return 2;
    } else if (a > 0) {
        return 1;
    } else {
        return 0;
    }
}

void sgpa() {
    float totalCredits = 0;
    float totalGradePoints = 0;

    for (int i = 0; i < n; i++) {
        totalCredits += credits[i];
        totalGradePoints += credits[i] * grade(marks[i]);
    }

    float sgpa = totalGradePoints / totalCredits;

    System.out.println("NAME: " + name);
    System.out.println("USN: " + usn);
    System.out.println("THE SGPA OF THE STUDENT IS: " + sgpa);
}
}

```

```
public class labpro {  
    public static void main(String[] args) {  
        Student S1 = new Student();  
        S1.create();  
        S1.sgpa();  
    }  
}
```

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

C:\Users\Admin>cd desktop

C:\Users\Admin\Desktop>javac Main.java

C:\Users\Admin\Desktop>java Main

Enter details for Student 1:
Enter Student Name:
akshat
Enter Student USN:
123
Enter Marks for Subject 1:
90
Enter Credits for Subject 1:
3
Enter Marks for Subject 2:
70
Enter Credits for Subject 2:
3
Enter Marks for Subject 3:
80
Enter Credits for Subject 3:
4
Enter Marks for Subject 4:
90
Enter Credits for Subject 4:
4
```

Results for all students:

Student Name: akshat

Student USN: 123

SGPA: 8.818181818181818

Student Name: ayush

Student USN: 124

SGPA: 7.954545454545454

Student Name: rajat

Student USN: 125

SGPA: 7.045454545454546

Enter Marks for Subject 3:	45
Enter Credits for Subject 3:	4
Enter Marks for Subject 4:	88
Enter Credits for Subject 4:	4
Enter Marks for Subject 5:	77
Enter Credits for Subject 5:	4
Enter Marks for Subject 6:	80
Enter Credits for Subject 6:	2
Enter Marks for Subject 7:	85
Enter Credits for Subject 7:	1
Enter Marks for Subject 8:	85
Enter Credits for Subject 8:	1
Enter details for Student 3:	
Enter Student Name:	rajat
Enter Student USN:	125
Enter Marks for Subject 1:	50
Enter Credits for Subject 1:	3
Enter Marks for Subject 2:	60
Enter Credits for Subject 2:	3
Enter Marks for Subject 3:	65
Enter Credits for Subject 3:	4
Enter Marks for Subject 4:	55
Enter Credits for Subject 4:	4
Enter Marks for Subject 5:	70
Enter Credits for Subject 5:	4
Enter Marks for Subject 6:	55
Enter Credits for Subject 6:	2
Enter Marks for Subject 7:	90
Enter Credits for Subject 7:	1
Enter Marks for Subject 8:	95
Enter Credits for Subject 8:	1
Enter Marks for Subject 4:	90
Enter Credits for Subject 4:	4
Enter Marks for Subject 5:	60
Enter Credits for Subject 5:	4
Enter Marks for Subject 6:	75
Enter Credits for Subject 6:	2
Enter Marks for Subject 7:	90
Enter Credits for Subject 7:	1
Enter Marks for Subject 8:	90
Enter Credits for Subject 8:	1
Enter details for Student 2:	
Enter Student Name:	ayush
Enter Student USN:	124
Enter Marks for Subject 1:	90
Enter Credits for Subject 1:	3
Enter Marks for Subject 2:	66
Enter Credits for Subject 2:	3
Enter Marks for Subject 3:	45
Enter Credits for Subject 3:	

3.book

PAGE:

304

370

DATE:

PAGE:

Author: Rich Riordan
Price: 500
Number of pages: 800

Details of Book 1:
Book Name: Game of Thrones
Author: George R. R. Martin
Price: 600.0
Number of Pages: 200

Details of Book 2:

Book Name: Percy Jackson
Author: Rich Riordan
Price: 500.0
Number of Pages: 800

DATE:

PAGE

```
String author = scanner.nextLine();
System.out.print("Price: ");
double price = scanner.nextDouble();
System.out.print("Number of Pages: ");
int numPages = scanner.nextInt();
scanner.nextLine();
```

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

```
for (int i = 0; i < n; i++) {
    System.out.println("In Detail of Book " + (i + 1));
    System.out.println(book[i].toString());
}
```

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

Output

Enter number of books: 2
Enter details for book 1.

Name: game of thrones

Author: george rr martin

Price: 600

Number of pages: 700

Enter details of book 2:

Name: Percy Jackson

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

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

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

toString
 public String toString () {
 return "Book Name: " + name + " by Author: " + author + "
 \n Price: " + price + "\n Number of pages: " + numPages
 } }

}
 public class Book Demo {
 public static void main (String [] args) {
 Scanner scanner = new Scanner (System.in);
 System.out.print ("Enter number of books: ");
 int n = scanner.nextInt ();
 scanner.nextLine ();

Book [] books = new Book [n];

for (int i = 0; i < n; i++) {
 System.out.println ("Enter details for book " + (i + 1)
 + ":");
 System.out.print ("Name: ");
 String name = scanner.nextLine ();
 System.out.print ("Author: ");

3)

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

```
    private String name;  
    private String author;  
    private double price;  
    private int numPages;
```

```
    public Book (String name, author, double price,  
                int numPages) {
```

```
        this.name = name;
```

```
        this.author = author;
```

```
        this.price = price;
```

```
        this.numPages = numPages;
```

```
    public String getName() {
```

```
        return name;
```

```
    public String getAuthor() {
```

```
        return author;
```

~~```
 public double getPrice() {
```~~~~```
        return price;
```~~~~```
 public int getNumPages() {
```~~~~```
        return numPages;
```~~~~```
 public void setName(String name) {
```~~~~```
        this.name = name;
```~~

```
import java.util.Scanner;

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

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

    public String getName() {
        return name;
    }

    public String getAuthor() {
        return author;
    }

    public double getPrice() {
        return price;
    }

    public int getNumPages() {
        return num_pages;
    }

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

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

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

```

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

@Override
public String toString() {
    return "Book Name: " + name + "\nAuthor: " + author + "\nPrice: " + price + "\nNumber of
Pages: " + num_pages;
}
}

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

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

        Book[] books = new Book[n];

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

            System.out.print("Name: ");
            String name = scanner.nextLine();
            System.out.print("Author: ");
            String author = scanner.nextLine();
            System.out.print("Price: ");
            double price = scanner.nextDouble();
            System.out.print("Number of Pages: ");
            int num_pages = scanner.nextInt();
            scanner.nextLine();

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

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

            System.out.println(books[i].toString());
        }
    }
}

```

```
        }

        scanner.close();
    }
}

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

C:\Users\Admin>cd desktop

C:\Users\Admin\Desktop>javac BookDemo.java

C:\Users\Admin\Desktop>java BookDemo
Enter number of books: 2
Enter details for book 1:
Name: game of thrones
Author: george rr martin
Price: 600
Number of Pages: 700
Enter details for book 2:
Name: percy jackson
Author: Rick Riordan
Price: 500
Number of Pages: 800

Details of Book 1:
Book Name: game of thrones
Author: george rr martin
Price: 600.0
Number of Pages: 700

Details of Book 2:
Book Name: percy jackson
Author: Rick Riordan
Price: 500.0
Number of Pages: 800

C:\Users\Admin\Desktop>Anshuman Gupta
```

4.abstract calss

class circle extends shape
circle (int radius)

{
 mr. r = radius;

void printarea()

{
 double area = 3.14 * r * r;
 System.out.println("Area of the circle = " + area);

{
 public class abstract class

{
 public static void main (String [] args)

{
 shape rectangle = new rectangle (5,10);
 shape triangle = new triangle (6,8);
 shape circle = new circle (7);

~~rectangle.printarea();~~
~~triangle.printarea();~~
~~circle.printarea();~~

outPut

Area of the rectangle = 50.0

Area of the triangle = 24.0

Area of the circle = 153.86

4) abstract class shape {

int l, b;

abstract void printarea();

5) class rectangle extends shape {

rectangle (int length, int breadth)

2) this.l = length;

this.b = breadth;

3) void printarea()

int area = l * b;

System.out.println("Area of the rectangle = " + area);

3)

class triangle extends shape {

triangle (int height, int width)

2) this.l = height;

this.b = width;

3) void printarea()

double area = 0.5 * l * b;

System.out.println("Area of the triangle = " + area);

```
import java.util.Scanner;

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

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

    public String getName() {
        return name;
    }

    public String getAuthor() {
        return author;
    }

    public double getPrice() {
        return price;
    }

    public int getNumPages() {
        return num_pages;
    }

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

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

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

```

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

@Override
public String toString() {
    return "Book Name: " + name + "\nAuthor: " + author + "\nPrice: " + price + "\nNumber of
Pages: " + num_pages;
}
}

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

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

        Book[] books = new Book[n];

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

            System.out.print("Name: ");
            String name = scanner.nextLine();
            System.out.print("Author: ");
            String author = scanner.nextLine();
            System.out.print("Price: ");
            double price = scanner.nextDouble();
            System.out.print("Number of Pages: ");
            int num_pages = scanner.nextInt();
            scanner.nextLine();

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

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

            System.out.println(books[i].toString());
        }
    }
}

```

```
        }

        scanner.close();
    }
}

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

C:\Users\Admin>cd desktop

C:\Users\Admin\Desktop>javac abstract_class.java
error: file not found: abstract_class.java
Usage: javac <options> <source files>
use --help for a list of possible options

C:\Users\Admin\Desktop>cd desktop
The system cannot find the path specified.

C:\Users\Admin\Desktop>javac abstract_class.java

C:\Users\Admin\Desktop>java abstract_class
Area of the rectangle =50
Area of the triangle =24.0
Area of the circle =153.86
C:\Users\Admin\Desktop>Anshuman gupta
```

5.accounts

default:

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

g

3

5

Output

Enter the name:

Ashwani

Enter the account number:

24229

Enter your choice

1) Savings account

2) Current Account

3) End.

1

Enter balance:

532353

Enter the interest rate:

9

The current balance is 5323530.0

switch (choice) 2

case 1:

```
System.out.println ("Enter initial balance");
double savingsBalance = sc.nextDouble();
System.out.println ("Enter the interest rate:");
double interestRate = sc.nextDouble();
SavingsAccount savingsAccount = new SavingsAccount(name, accountNumber, savingsBalance, interestRate);
savingsAccount.addInterest();
break;
```

case 2:

```
System.out.println ("Enter initial balance");
double currentBalance = sc.nextDouble();
System.out.println ("Enter the interest rate:");
System.out.println ("Enter minimum balance");
CurrentAccount currentAccount = new CurrentAccount(name, accountNumber, currentBalance, minBalance);
```

~~System.out.println ("Enter the amount to be withdrawn");~~

~~double q = sc.nextInt();~~
~~currentAccount.withdraw (q);~~
System.out.println ("Account created. Current balance: " + currentAccount.getAccount().getBalance());
break;

case 3:

```
System.out.println ("Enter i:");
sc.nextLine();
return i;
```

```

public CurrentAccount (String custName, String address,
double balance, double minBalance) {
    double balance, double minBalance;
    this. minBalance = minBalance;
    this. account = new Account (custName, minBalance);
}

```

```

public void withdraw (double amt) {
    if (amt > 0 && (account.getBalance () -
    > minBalance)) {
        account.withdraw (amt);
    } else {
        System.out.println ("Withdrawal is not
possible");
    }
}

```

```

public Account getAccount () {
    return account;
}

```

```

public class Bank {
    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter the name");
        String name = sc.nextLine ();
        System.out.println ("Enter the account number");
        String accnt = sc.nextLine ();
        while (true) {
            System.out.println ("Enter your Choice");
            System.out.println ("1. Saving account\n2.
Current Account\n3. Exit");
            int choice = sc.nextInt ();
            if (choice == 1) {
                SavingAccount sa =
                new SavingAccount (name, accnt);
                sa.deposit (1000);
                System.out.println ("Amount deposited
successfully");
            } else if (choice == 2) {
                CurrentAccount ca =
                new CurrentAccount (name, accnt);
                ca.deposit (1000);
                System.out.println ("Amount deposited
successfully");
            } else if (choice == 3) {
                System.out.println ("Thank you");
                break;
            }
        }
    }
}

```

PAGE: DATE: PAGE:

5
class SavingAccount {
 public double interestRate;
 private Account account;
}

6
public SavingsAccount (String custName, String accNo,
 double balana, double interestRate);
 this.interestRate = interestRate;
 this.account = new Account(custName, accNo,
 balana);

7
~~public void addInterest () {
 double interest = account.getBalance() *
 this.interestRate;
 account.deposit(interest);~~

8
public Account getAccount () {
 return account;

9
class CurrentAccount {
 private double minBalance;
 private Account account;

```

5) import java.util.Scanner;

class Account {
    private String custName;
    private String accNo;
    private double balance;

    public Account (String custName, String accNo, double balance) {
        this.custName = custName;
        this.accNo = accNo;
        this.balance = balance;
    }

    public double getBalance() {
        return this.balance;
    }

    public void deposit(double amount) {
        if (amount > 0) {
            this.balance += amount;
            System.out.println("The current balance is " + this.balance);
        } else {
            System.out.println("Amount should not be negative");
        }
    }

    public void withdraw(double amount) {
        if (amount > 0 && (balance - amount) >= 0) {
            this.balance -= amount;
            System.out.println("Withdraw successful. Current balance is " + this.balance);
        }
    }
}

```

```

import java.util.Scanner;

class Account {
    private String custName;
    private String accNo;
    private double balance;

    public Account(String custName, String accNo, double balance) {
        this.custName = custName;
        this.accNo = accNo;
        this.balance = balance;
    }

    public double getBalance() {
        return this.balance;
    }

    public void deposit(double amount) {
        if (amount > 0) {
            this.balance += amount;
            System.out.println("The current balance is " + this.balance);
        } else {
            System.out.println("Amount should not be negative");
        }
    }

    public void withdraw(double amount) {
        if (amount > 0 && (balance - amount) >= 0) {
            this.balance -= amount;
            System.out.println("Withdraw successful. Current balance: " + this.balance);
        } else {
            System.out.println("Withdraw is not possible");
        }
    }
}

class SavingsAccount {
    private double interestRate;
    private Account account;

    public SavingsAccount(String custName, String accNo, double balance, double interestRate) {
        this.interestRate = interestRate;
        this.account = new Account(custName, accNo, balance);
    }
}

```

```

}

public void addInterest() {
    double interest = account.getBalance() * this.interestRate;
    account.deposit(interest);
}

public Account getAccount() {
    return account;
}
}

class CurrentAccount {
    private double minBalance;
    private Account account;

    public CurrentAccount(String custName, String accNo, double balance, double minBalance) {
        this.minBalance = minBalance;
        this.account = new Account(custName, accNo, balance);
    }

    public void withdraw(double amt) {
        if (amt > 0 && (account.getBalance() - amt) >= minBalance) {
            account.withdraw(amt);
        } else {
            System.out.println("Withdraw is not possible");
        }
    }

    public Account getAccount() {
        return account;
    }
}

public class Bank {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the name:");
        String name = sc.nextLine();
        System.out.println("Enter the account number:");
        String accnt = sc.nextLine();

        while (true) {

```

```
System.out.println("Enter your choice:");
System.out.println("1. Savings Account");
System.out.println("2. Current Account");
System.out.println("3. Exit");
int choice = sc.nextInt();

switch (choice) {
    case 1:
        System.out.println("Enter initial balance:");
        double savingsBalance = sc.nextDouble();
        System.out.println("Enter the interest rate:");
        double interestRate = sc.nextDouble();
        SavingsAccount savingsAccount = new SavingsAccount(name, accnt,
savingsBalance, interestRate);
        savingsAccount.addInterest();
        break;

    case 2:
        System.out.println("Enter initial balance:");
        double currentBalance = sc.nextDouble();
        System.out.println("Enter minimum balance:");
        double minBalance = sc.nextDouble();
        CurrentAccount currentAccount = new CurrentAccount(name, accnt, currentBalance,
minBalance);

        System.out.println("Enter the amount to be withdraw");
        double q = sc.nextInt();
        currentAccount.withdraw(q);
        System.out.println("Account created. Current balance: " +
currentAccount.getAccount().getBalance());
        break;

    case 3:
        System.out.println("Exiting...");
        sc.close();
        return;

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

}
```

```
C:\ Command Prompt
35363
entr the amount to be withdraw
653543
withdraw is not possible
Account created. Current balance: 542538.0
Enter your choice:
1. Savings Account
2. Current Account
3. Exit
2
Enter initial balance:
563537
Enter minimum balance:
535
entr the amount to be withdraw
5243
Withdraw successful. Current balance: 558294.0
Account created. Current balance: 558294.0
Enter your choice:
1. Savings Account
2. Current Account
3. Exit
3
Exiting...
C:\Users\Admin\Desktop>_

C:\ Command Prompt
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>cd desktop

C:\Users\Admin\Desktop>javac Bank.java

C:\Users\Admin\Desktop>java Bank
Enter the name:
anshuman
Enter the account number:
24224
Enter your choice:
1. Savings Account
2. Current Account
3. Exit
1
Enter initial balance:
537353
Enter the interest rate:
9
The current balance is 5373530.0
Enter your choice:
1. Savings Account
2. Current Account
3. Exit
2
Enter initial balance:
542538
Enter minimum balance:
```

6.package

DATE:

Entire marks for 5 courses:

50

40

32

36

40

Entire SE marks for 5 courses:

21

72

73

79

95

Final marks of Students:

Student 1 Class : 10A23C8093

Course 2 Final marks : 66

Course 2 Final marks : 68

Course 3 Final marks : 69

Course 4 Final marks : 71

Course 5 Final marks : 72

```

int S) intialMarks = new int(S);
System.out.println("Enter initial marks for
S courses : ");
for (int j = 0; j < S; j++) {
    initialMarks[j] = scanner.nextInt();
}
int [] seeMarks = new int(S);
System.out.println("Enter see marks for scanner : ");
for (int j = 0; j < S; j++) {
    seeMarks[j] = scanner.nextInt();
}
internalStudents[i] = new InternalStudent (usn, name, sem,
intialMarks);
externalStudents[i] = new ExternalStudent (usn, name, sem,
seeMarks);
}
System.out.println("In Final Marks of Student : ");
for (int i = 0; i < n; i++) {
    System.out.println("In Student " + (i+1) + " USN."
    + internalStudents[i].usn);
}
for (int j = 0; j < S; j++) {
    int finalmark = (internalStudents[i].intialMark
    [j] + (externalStudents[i].seeMarks[j])/2);
}
System.out.println("Course " + (j+1) + "Final mark : "
finalmark);
}

```

Output

Ent detail for student 1:

USN : 13M2CS053

Name : Arshun

Semester : 3

DATE: PAGE:
 public external (String name, String name, int sem,
 int [] see marks);
 super (name, name, sem);
 other. see marks = see marks;
 }
 }
 import (I.E. Internal);
 import S.E. External;
 import java.util.Scanner;
 public class FinalMarks {
 public static void main (String [] args) {
 Scanner scanner = new Scanner (System.in);
 System.out.print ("Enter number of students");
 int n = scanner.nextInt();
 Internal [] internalStudents = new Internal [n];
 External [] externalStudents = new External [n];
 for (int i = 0; i < n; i++) {
 System.out.println ("Enter details for
 student " + (i + 1) + ":");
 System.out.print ("USN");
 String USN = scanner.next();
 System.out.print ("Name");
 String name = scanner.next();
 System.out.print ("Semester");
 int sem = scanner.nextInt();

6) package CIE;

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

} F

package CIE;

```
public class Internals extends Student {
    public int[] internalMarks;
```

```
public Internals (String usn, String name, int
    sem, int[] internalMarks) {
    super (usn, name, sem);
    this.internalMarks = internalMarks;
}
```

} .

package SIE;

import CIE.Student;

```
public class InternMarks Extends Student {
    public int[] seeMarks;
```

```

package CIE;

public class Internals extends Student {
    public int[] internalMarks;

    public Internals(String usn, String name, int sem, int[] internalMarks) {
        super(usn, name, sem);
        this.internalMarks = internalMarks;
    }
}

```

Studenpackage 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 External extends Student {
    public int[] seeMarks;

    public External(String usn, String name, int sem, int[] seeMarks) {
        super(usn, name, sem);
        this.seeMarks = seeMarks;
    }
}

```

```

import CIE.Internals;
import SEE.External;
import java.util.Scanner;

```

```

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

```

```

Scanner scanner = new Scanner(System.in);

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

Internals[] internalStudents = new Internals[n];
External[] externalStudents = new External[n];

// Input for each student
for (int i = 0; i < n; i++) {
    System.out.println("\nEnter details for student " + (i + 1) + ":");

    System.out.print("USN: ");
    String usn = scanner.next();

    System.out.print("Name: ");
    String name = scanner.next();

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

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

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

    // Creating objects for Internals and External
    internalStudents[i] = new Internals(usn, name, sem, internalMarks);
    externalStudents[i] = new External(usn, name, sem, seeMarks);
}

// Display final marks
System.out.println("\nFinal Marks of Students:");
for (int i = 0; i < n; i++) {
    System.out.println("\nStudent " + (i + 1) + " - USN: " + internalStudents[i].usn);
    for (int j = 0; j < 5; j++) {
}

```

```
        int finalMark = (internalStudents[i].internalMarks[j] +
(externalStudents[i].seeMarks[j]) / 2);
        System.out.println("Course " + (j + 1) + " Final Mark: " + finalMark);
    }
}

scanner.close();
}
}
```

```
Enter details for student 1:
```

```
USN: 1bm23cs041
```

```
Name: annas
```

```
Semester: 3
```

```
Enter internal marks for 5 courses:
```

```
31
```

```
32
```

```
33
```

```
34
```

```
35
```

```
Enter SEE marks for 5 courses:
```

```
71
```

```
72
```

```
73
```

```
74
```

```
75
```

```
Final Marks of Students:
```

```
Student 1 - USN: 1bm23cs041
```

```
Course 1 Final Mark: 66
```

```
Course 2 Final Mark: 68
```

```
Course 3 Final Mark: 69
```

```
Course 4 Final Mark: 71
```

```
Course 5 Final Mark: 72
```

DATE:

PAGE:

DATE:

PAGE:

5
6
3
F

Output

Enter Father's age : 34

Enter Son's age : 33

Son's age cannot be greater than or equal to father's age

Would you like to re-enter details (Y/N)

g

Enter Father's age : 44

Enter Son's age : 33

Accepted Successfully

Would you like to re-enter details (Y/N)

n

sys:
Fully %

```

    & age");
    & this.sonAge = sonAge;
}

public class Father {
    public static void main (String [] args) {
        while (true) {
            Scanner sc = new Scanner (System.in);
            System.out.print ("Enter Father's Age: ");
            int fatherAge = sc.nextInt ();
            System.out.print ("Enter Son's Age: ");
            int sonAge = sc.nextInt ();
            try {
                Son son = new Son (fatherAge, sonAge);
                System.out.println ("Created Successfully!");
            } catch (AgeException e) {
                System.out.println (e.getMessage ());
            } catch (SonAgeException e) {
                System.out.println (e.getMessage ());
            }
            System.out.println ("Would you like to re-run");
            detail (1/n);
        }
    }
}

```

Story input = sc.nextInt();
if (input.equals ("n")) {
break;

DATE: _____ PAGE: _____

```

(1) import java.util.Scanner;
class WrongAgeException (String message) {
    super(message);
}

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

class Father {
    private int age;
    public Father (int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException ("Wrong age");
        }
        this.age = age;
    }
    public int getAge () {
        return age;
    }
}

class Son extends Father {
    private int sonAge;
    public Son (int fatherAge, int sonAge) throws
        WrongAgeException, SonAgeException {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new SonAgeException ("Son's age
                cannot be greater than or equal to father's");
        }
    }
}

```

```

import java.util.Scanner;
class WrongAgeException extends Exception {
    public WrongAgeException(String message) {
        super(message);
    }
}

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

class Father {
    private int age;
    public Father(int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException("Wrong age");
        }
        this.age = age;
    }
    public int getAge() {
        return age;
    }
}

class Son extends Father {
    private int sonAge;
    public Son(int fatherAge, int sonAge) throws WrongAgeException, SonAgeException {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new SonAgeException("Son's age cannot be greater than or equal to father's age");
        }
        this.sonAge = sonAge;
    }
    public int getSonAge() {
        return sonAge;
    }
}
public class FatherSon{
    public static void main(String[] args) {
        while(true){
            Scanner sc = new Scanner(System.in);

```

```

System.out.print("Enter Father's Age: ");
int fatherAge = sc.nextInt();
System.out.print("Enter Son's Age: ");
int sonAge = sc.nextInt();
try {
    Son son = new Son(fatherAge, sonAge);
    System.out.println("Accepted Succesfully");
}
catch (WrongAgeException e) {
    System.out.println(e.getMessage());
}
catch (SonAgeException e) {
    System.out.println(e.getMessage());
}
System.out.println("Would you like to re-enter details (Y/n)");
String input = sc.next();
if (input.equalsIgnoreCase("n")) {
    break;
}
}
}

```

C:\ Command Prompt

```

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

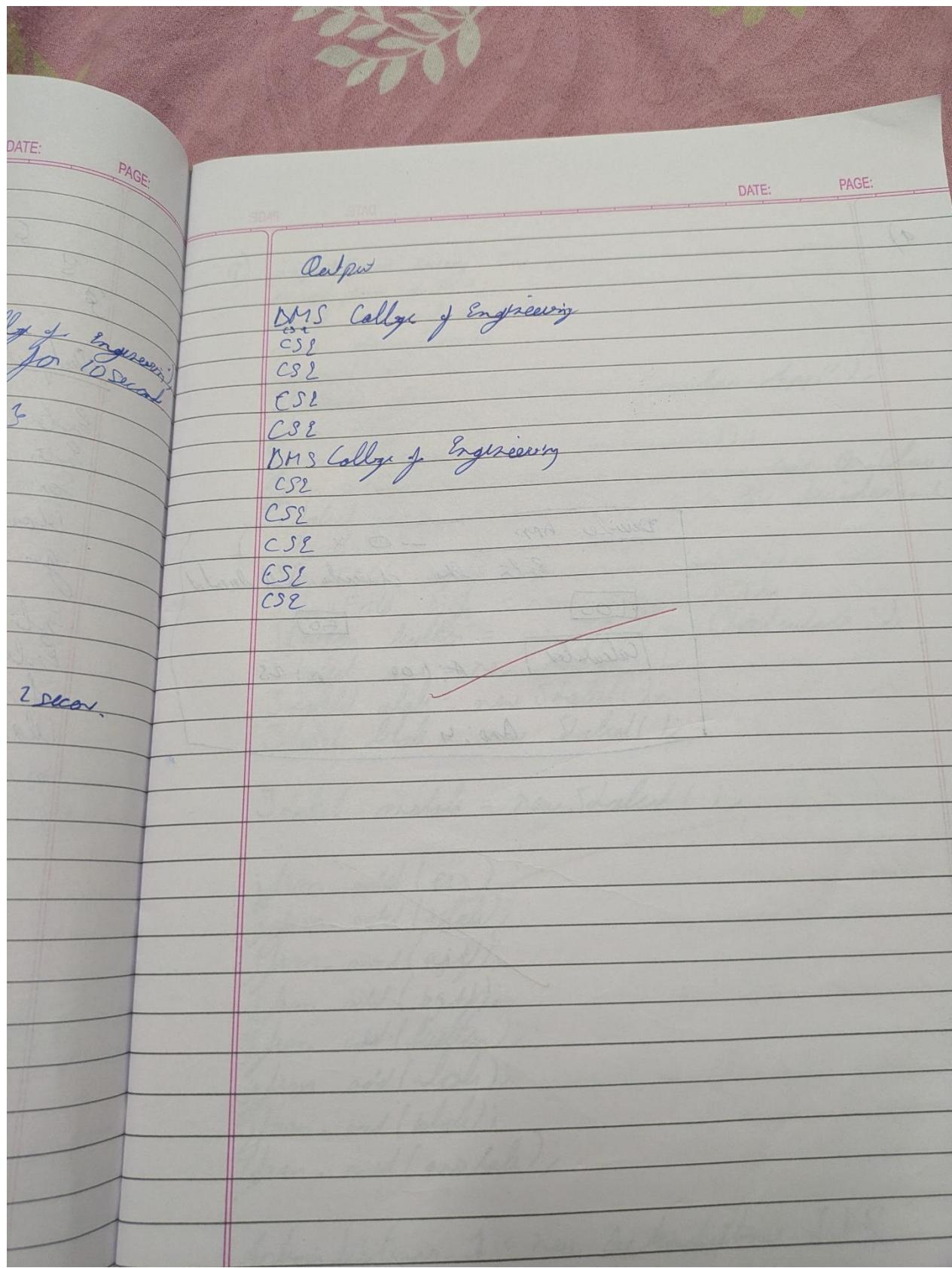
C:\Users\Admin>cd desktop

C:\Users\Admin\Desktop>javac FatherSon.java

C:\Users\Admin\Desktop>java FatherSon
Enter Father's Age: 53
Enter Son's Age: 50
Accepted Succesfully
Would you like to re-enter details (Y/n)
y
Enter Father's Age: 45
Enter Son's Age: 19
Accepted Succesfully
Would you like to re-enter details (Y/n)
n

C:\Users\Admin\Desktop>Anshuman gupta 1bm23cs043_

```



Q) 8) class DMS extends Thread {
public void run() {
try {
while (true) {
System.out.println("DMS Call of engineering");
Thread.sleep(1000); // sleep for 10 second
}
} catch (InterruptedException e) {}
}

3
class CS extends Thread {
public void run() {
try {
while (true) {
System.out.println("CS");
Thread.sleep(2000); // sleep for 2 sec.
}
} catch (InterruptedException e) {}
}

public class MultiThreading {
public static void main (String [] args) {
DMS dms = new DMS();
CS cs = new CS();
dms.start();
cs.start();
}

```

class BMS extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000); // Sleep for 10 seconds
            }
        }catch (InterruptedException e) {}
    }
}

class CSE extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("CSE");
                Thread.sleep(2000); // Sleep for 2 seconds
            }
        }catch (InterruptedException e) {}
    }
}

public class Multithreading{
    public static void main(String[] args) {
        BMS bms = new BMS();
        CSE cse = new CSE();
        bms.start();
        cse.start();
    }
}

```

```
C:\Users\Admin\Desktop>java Multithreading
C:\Users\Admin\Desktop>java Multithreading
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
BMS College of Engineering
CSE
```

9.Swing

DATE: PAGE:
Sout)
Date: PAGE:
3
3
3;
jfrom .setVisible(true);
{
public static void main (String []){
System.out.println ("
SwingUtilities.invokeLater (new Runnable () {
public void run () {
new SavingFrame ();
}
});
}
};
};

Output

| | | |
|-----------------------------|---------|--------|
| DeivideApp | | |
| Est. the divide and conquer | | |
| (10) | (10) | |
| calculable | 1 : 100 | 0 : 15 |
| Ans: 9 | | |

public void actionPerformed(ActionEvent evt) {
 System.out.println("Action event from a text
 field");

{

}

aifl.addActionListener(x);

bifl.addActionListener(l);

button.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent evt) {

try {

int a = Integer.parseInt(tf1.getText());

int b = Integer.parseInt(tf2.getText());

int ans = a / b;

alab.setText("n A = " + a);

blab.setText("n B = " + b);

anslab.setText("n Ans = " + ans);

}

catch (NumberFormatException e) {

alab.setText("n ");

blab.setText("n ");

anslab.setText("n ");

err.setText("n Only Integers!");

}

catch (ArithmException e) {

alab.setText("n ");

blab.setText("n ");

anslab.setText("n ");

err.setText("n D should be Non zero!");

DATE:

PAGE:

DATE: PAGE:

DATE:

PAGE:

9) import java.awt.*;
import java.awt.event.*;
import java.awt.BorderLayout;
class SwingingDemo extends Frame
Swinging Demo(12)
JFrame jfrm = new JFrame("Dividers App");
jfrm.setSize(225, 180);
jfrm.setLayout(new FlowLayout());
jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
JLabel jlable = new JLabel ("Enter the Dividers and
Dividers :");
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 onlab = new JLabel();

jfrm.add(err);
jfrm.add(jlable);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(onlab);

ActionListener 1 = new ActionListener (13)

```

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

JLabel anslab = new JLabel();
// add in order :)
jfrm.add(err); // to display error bois
jfrm.add(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);
ActionListener l = new ActionListener() {
public void actionPerformed(ActionEvent evt) {
System.out.println("Action event from a text field");
}
};
ajtf.addActionListener(l);
bjtf.addActionListener(l);
button.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent evt) {
try{

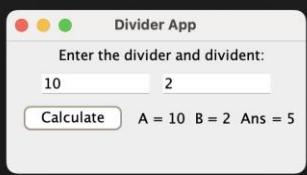
```

```

int a = Integer.parseInt(ajtf.getText());
int b = Integer.parseInt(bjtf.getText());
int ans = a/b;
alab.setText("\nA = " + a);
blab.setText("\nB = " + b);
anslab.setText("\nAns = " + ans);
}
catch(NumberFormatException e){
alab.setText("");
blab.setText("");
anslab.setText("");
err.setText("Enter Only Integers!");
}
catch(ArithmeticException e){
alab.setText("");
blab.setText("");
anslab.setText("");
err.setText("B should be NON zero!");
}
}
});
// display frame
jfrm.setVisible(true);
}

public static void main(String args[]){
System.out.println("Ritesh Mohan Nayak 1BM23EC212");
// create frame on event dispatching thread
SwingUtilities.invokeLater(new Runnable(){
public void run(){
new SwingDemo();
}
});
}
}
}

```



10.1. Deadlock

Wanel();
");

B. last();

{}

Cars Deadlock implement Runnable

A a = new A();

B b = new B();

Deadlock(1)

Thread -> create Thread(). Set Name ("Main Thread");

Thread t = new Thread(this, "Racing Thread");

t.start();

a.foo(b);

System.out.println("Bar in main thread");

public void run() {

b.bar(a);

System.out.println("Bar in other thread");

public static void main (String args[]) {

new Deadlock();

{}

{}

Output

Main Thread enters A.foo

Racing Thread enters B.bar

Main Thread trying to call B.last()

Inside A.last

Bar in main thread

Racing Thread trying to call A.last()

Inside A.last()

Bar to other thread,

Deadlock

DATE:

PAGE:

(i) 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");
}
}

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("inside B.last");
a.last();
}

System.out.println(name + " trying to call A.last");
}

void last() {
 System.out.println("inside A.last");
}

```

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

```

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

```
MainThread entered A.foo  
RacingThread entered B.bar  
MainThread trying to call B.last()  
Inside A.last  
Back in main thread  
RacingThread trying to call A.last()  
Inside A.last  
Back in other thread
```

10.2. interprocessing

Consumed = 0
Yat = 1

External Producer

Consumed = 1

Yat = 2

Internal Consumer

Product marketing

Yat = 62

Internal Producer

Consumed = 2

Yat = 3

Internal Consumer

Product marketing

Yat = 5

External Producer

Consumed = 3

Yat = 9

Shantanu
4-11-21

Internal Consumer

Yat = 9

External Producer

Consumed = 4

class C1C15 {

```
    int n = q->get();
    system.out.println("Consumed : " + n);
    i++;
}
```

3

5

5

class P {

```
    public static void main (String args []) {
        Q q = new Q ();
        new producer (q);
        new consumer (q);
    }
}
```

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

3

Output

Press Control+C to stop

Pat = 0

Internal Consumer

producer working.

Pat = 0

Internal Producer

Pat = 1

Internal consumer

producer working

```

this. n = n;
valueset = true;
system.out.println ("Pad = " + n);
System.out.println ("Intervale consumer(" + n + ")");

```

analyse();

33

class Producer implements runnable {

```

    Ag;
    Producer (Ag) {
        this. q = q;
    }

```

```

    new Thread (this, "Producer").start();
}

```

public void run () {

int i = 0;

while (i < 15) {

q. put (i++);

33

class Consumer implements runnable {

Ag;

```

    Consumer (Ag) {
        this. q = q;
    }

```

```

    new Thread (this, "Consumer").start();
}

```

public void run () {

int i = 0;

10) (i) class Q2
int n;
boolean valueSet = false;
synchronized int get() {
 if (!valueSet)
 System.out.println("Value set");
 try {
 System.out.println("In consumer waiting");
 wait();
 } catch (InterruptedException e) {
 System.out.println("InterruptedException caught");
 }
 System.out.println("Cnf = " + n);
 valueSet = true;
 System.out.println("End Inside Producer");
}
notify();
return n;
}
synchronized void put(int n) {
 valueSet (valueSet);
 try {
 System.out.println("In Producer waiting");
 wait();
 } catch (InterruptedException e) {
 System.out.println("InterruptedException caught");
 }
 System.out.println("Intput");
}

```

class Q {
int n;
boolean valueSet = false;
synchronized int get() {
while(!valueSet)
try {
System.out.println("\nConsumer waiting\n");
wait();
} catch(InterruptedException e) {
System.out.println("InterruptedException caught");
}
System.out.println("Got: " + n);
valueSet = false;
System.out.println("\nIntimate Producer\n");
notify();
return n;
}
synchronized void put(int n) {
while(valueSet)
try {
System.out.println("\nProducer waiting\n");
wait();
} catch(InterruptedException e) {
System.out.println("InterruptedException caught");
}
this.n = n;
valueSet = true;
System.out.println("Put: " + n);
System.out.println("\nIntimate Consumer\n");
notify();
}
}
class Producer implements Runnable {
Q q;
Producer(Q q) {
this.q = q;
new Thread(this, "Producer").start();
}
public void run() {
int i = 0;
while(i<15) {
q.put(i++);
}
}

```

```
}

}

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

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

```

    Oct: 0
    Intimate Consumer
    Producer waiting
    Oct: 0
    Intimate Producer
    Oct: 1
    Intimate Consumer
    Producer waiting
    consumer:0
    Oct: 1
    Intimate Producer
    consumer:1
    Oct: 2
    Intimate Consumer
    Producer waiting
    Oct: 2
    Intimate Producer
    consumer:2
    Oct: 3
    Intimate Consumer
    Producer waiting
    Oct: 3
    Intimate Producer
    consumer:3
    Oct: 4
    Intimate Consumer
    Producer waiting
    Oct: 4
    Intimate Producer
    consumer:4
    Oct: 5
    Intimate Consumer
    Producer waiting
    Oct: 5
    Intimate Producer
    consumer:5
    Oct: 6
    Intimate Consumer
    Producer waiting
    Oct: 6
    Intimate Producer
    consumer:6
    Oct: 7
    Intimate Consumer
    Producer waiting
    Oct: 7
    Intimate Producer
    consumer:7
    Oct: 8
    Intimate Consumer
    Producer waiting
    consumer:7
    Oct: 8
    Intimate Producer
    consumer:8
    Oct: 9
    Intimate Consumer
    Producer waiting
    Oct: 9
    Intimate Producer
    consumer:9
    Oct: 10
    Intimate Consumer
    Producer waiting
    Oct: 10
    Intimate Producer
    consumer:10
    Oct: 11
    Intimate Consumer
    Producer waiting
    Oct: 11
    Intimate Producer
    consumer:11
    Oct: 12
    Intimate Consumer
    Producer waiting
    Oct: 12
    Intimate Producer
    consumer:12
    Oct: 13
    Intimate Consumer
    Producer waiting
    Oct: 13
    Intimate Producer
    consumer:13
    Oct: 14
    Intimate Consumer
    Oct: 14
    Intimate Producer
    consumer:14

```