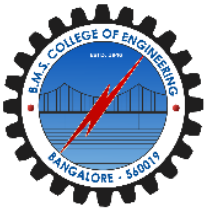
B.M.S. COLLEGE OF ENGINEERING
Basavanagudi, Bengaluru- 560019
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



LAB REPORT

On

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted By:
Bhanu Prakash M
1BM22CS067

In partial fulfilment of
BACHELOR OF ENGINEERING
In
COMPUTER SCIENCE AND ENGINEERING
2023-24

Faculty-In-Charge
Shravya A R
Assistant Professor
Department of Computer Science and Engineering

Table of Contents:

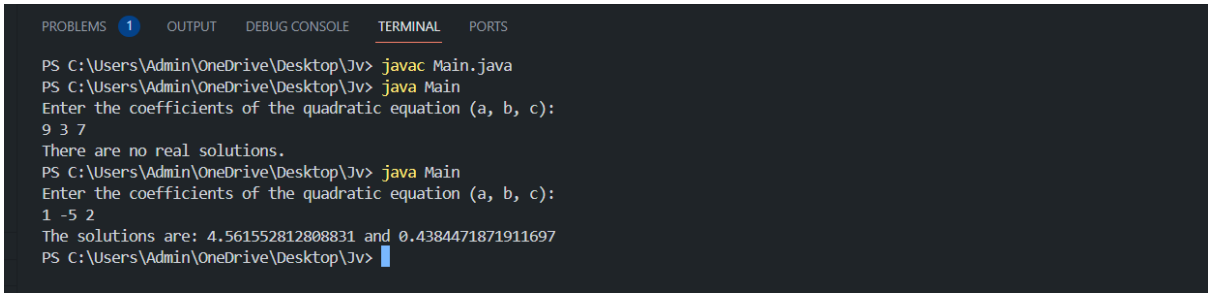
LAB1:

Develop a Java program that prints all real solutions to the quadratic equation a^2+bx+c=0 Read in a, b, c and use the quadratic formula. If the discriminate b2 - 4ac is negative, display a message stating that there are no real solutions.
import java.util.Scanner;

CODE:

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

        System.out.println("Enter the coefficients of the quadratic equation (a, b, c):");
        double a = scanner.nextDouble();
        double b = scanner.nextDouble();
        double c = scanner.nextDouble();

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

        if (discriminant < 0) {
            System.out.println("There are no real solutions.");
        } else if (discriminant == 0) {
            double solution = -b / (2 * a);
            System.out.println("The solution is: " + solution);
        } else {
            double solution1 = (-b + Math.sqrt(discriminant)) / (2 * a);
            double solution2 = (-b - Math.sqrt(discriminant)) / (2 * a);
            System.out.println("The solutions are: " + solution1 + " and " + solution2);
        }
    }
}
```

OUTPUT:



LAB2:

Write a Java program to create a class Student with members USN, name, marks(6 subjects). Include methods to accept student details and marks, Also include a method to calculate the percentage and display appropriate details. (Array of student object to be created)

CODE:

```
import java.util.Scanner;
class Student {
    private String USN;
    private String name;
    private int[] marks = new int[6];

    public void acceptDetails() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter USN:");
        USN = scanner.nextLine();
        System.out.println("Enter name:");
        name = scanner.nextLine();
        System.out.println("Enter marks for 6 subjects:");
        for (int i = 0; i < marks.length; i++) {
            System.out.print("Subject " + (i + 1) + ": ");
            marks[i] = scanner.nextInt();
        }
    }

    public double calculatePercentage() {
        int totalMarks = 0;
        for (int mark : marks) {
            totalMarks += mark;
        }
        return (double) totalMarks / marks.length;
    }

    public void displayDetails() {
        System.out.println("USN: " + USN);
        System.out.println("Name: " + name);
        System.out.println("Marks:");
        for (int i = 0; i < marks.length; i++) {
            System.out.println("Subject " + (i + 1) + ": " + marks[i]);
        }
        System.out.println("Percentage: " + calculatePercentage() + "%");
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the number of students:");
        int numStudents = scanner.nextInt();
```

```
        Student[] students = new Student[numStudents];

        for (int i = 0; i < numStudents; i++) {
            students[i] = new Student();
            System.out.println("Enter details for student " + (i + 1) + ":");
            students[i].acceptDetails();
        }

        System.out.println("\nStudent details:");
        for (int i = 0; i < numStudents; i++) {
            System.out.println("\nDetails for student " + (i + 1) + ":");
            students[i].displayDetails();
        }
    }
}
```

OUTPUT:

```
PROBLEMS 2  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Admin\OneDrive\Desktop\Jv> javac Main.java
PS C:\Users\Admin\OneDrive\Desktop\Jv> java Main
Enter the number of students:
2
Enter details for student 1:
Enter USN:
1BM22CS088
Enter name:
Rahul
Enter marks for 6 subjects:
Subject 1: 88
Subject 2: 98
Subject 3: 78
Subject 4: 86
Subject 5: 56
Subject 6: 99
Enter details for student 2:
Enter USN:
1BM22IS054
Enter name:
Rohan
Enter marks for 6 subjects:
Subject 1: 78
Subject 2: 89
Subject 3: 97
Subject 4: 95
Subject 5: 94
Subject 6: 93

Student details:

Details for student 1:
USN: 1BM22CS088
Name: Rahul
Marks:
Subject 1: 88
Subject 2: 98
Subject 3: 78
Subject 4: 86
Subject 5: 56
```

```
Subject 6: 99
Percentage: 84.16666666666667%

Details for student 2:
USN: 1BM22IS054
Name: Rohan
Marks:
Subject 1: 78
Subject 2: 89
Subject 3: 97
Subject 4: 95
Subject 5: 94
Subject 6: 93
Percentage: 91.0%
PS C:\Users\Admin\OneDrive\Desktop\Jv>
```

LAB3:

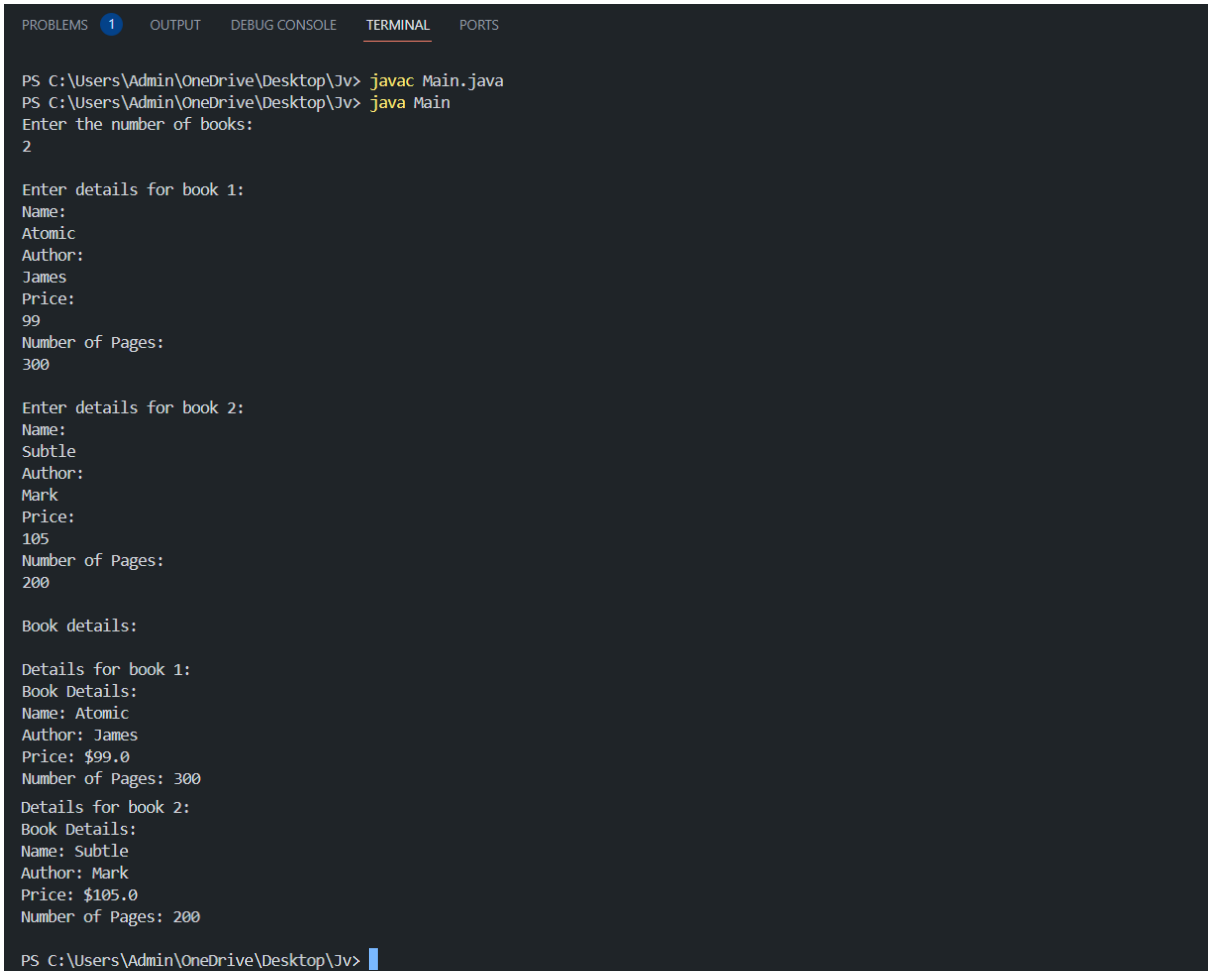
Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString() method that could display the complete details of the book. Develop a Java program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

CODE:

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

    public String getName() {
        return name;
    }

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

    public String getAuthor() {
        return author;
    }

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

    public double getPrice() {
        return price;
    }

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

```
    public int getNumPages() {
        return num_pages;
    }

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

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

        System.out.println("Enter the number of books:");
        int numBooks = scanner.nextInt();

        Book[] books = new Book[numBooks];

        for (int i = 0; i < numBooks; i++) {
            scanner.nextLine(); // Consume newline left-over from nextInt()
            System.out.println("\nEnter details for book " + (i + 1) + ":");
            System.out.println("Name:");
            String name = scanner.nextLine();
            System.out.println("Author:");
            String author = scanner.nextLine();
            System.out.println("Price:");
            double price = scanner.nextDouble();
            System.out.println("Number of Pages:");
            int numPages = scanner.nextInt();
            books[i] = new Book(name, author, price, numPages);
        }

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

OUTPUT:


PROBLEMS 1
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
PS C:\Users\Admin\OneDrive\Desktop\Jv> javac Main.java
PS C:\Users\Admin\OneDrive\Desktop\Jv> java Main
Enter the number of books:
2
Enter details for book 1:
Name:
Atomic
Author:
James
Price:
99
Number of Pages:
300
Enter details for book 2:
Name:
Subtle
Author:
Mark
Price:
105
Number of Pages:
200
Book details:
Details for book 1:
Book Details:
Name: Atomic
Author: James
Price: $99.0
Number of Pages: 300
Details for book 2:
Book Details:
Name: Subtle
Author: Mark
Price: $105.0
Number of Pages: 200
PS C:\Users\Admin\OneDrive\Desktop\Jv>

LAB4:

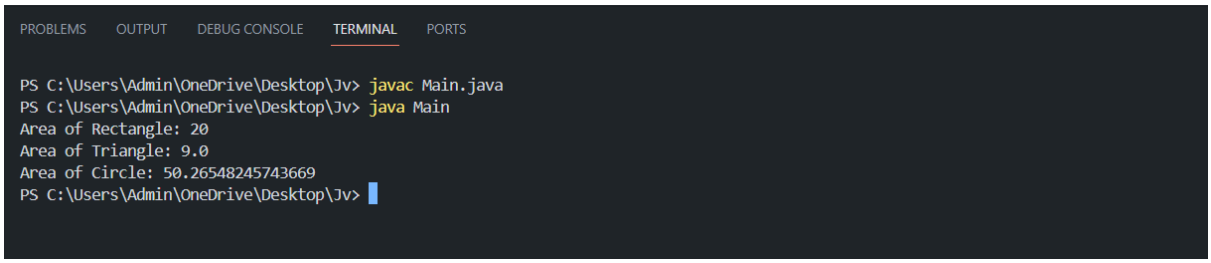
Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea()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

CODE:

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

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

    abstract void printArea();
}

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


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

class Triangle extends Shape {
    public Triangle(int base, int height) {
        super(base, height);
    }


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

class Circle extends Shape {
    public Circle(int radius) {
        super(radius, 0); // For circle, only one dimension is needed (radius)
    }
```

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

public class Main {
    public static void main(String[] args) {
        Rectangle rectangle = new Rectangle(5, 4);
        rectangle.printArea();

        Triangle triangle = new Triangle(3, 6);
        triangle.printArea();

        Circle circle = new Circle(4);
        circle.printArea();
    }
}
```

OUTPUT:



LAB5:

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

a) Accept deposit from customer and update the balance.

b) Display the balance.

c) Compute and deposit interest

d) Permit withdrawal and update the balance

Check for the minimum balance, impose penalty if necessary and update the balance.

CODE:

```
import java.util.Scanner;

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

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

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

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

    public void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
```

```
            System.out.println("Withdrawal of $" + amount + " successful.");
        } else {
            System.out.println("Insufficient balance.");
        }
    }
}

class CurrentAccount extends Account {
    private final double minimumBalance = 1000;
    private final double serviceCharge = 50;

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


    public void withdraw(double amount) {
        if (balance - amount >= minimumBalance) {
            balance -= amount;
            System.out.println("Withdrawal of $" + amount + " successful.");
        } else {
            System.out.println("Insufficient balance. Service charge of $" + serviceCharge + "
imposed.");
            balance -= serviceCharge;
        }
    }
}

class SavingsAccount extends Account {
    private final double interestRate = 0.05;

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

    public void computeInterest() {
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest computed and deposited: $" + interest);
    }
}

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


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

```
        System.out.println("Enter account number:");
        int curNumber = scanner.nextInt();
        System.out.println("Enter initial balance:");
        double curBalance = scanner.nextDouble();
        CurrentAccount currentAccount = new CurrentAccount(curName, curNumber,
curBalance);


        currentAccount.deposit(500);
        currentAccount.displayBalance();
        currentAccount.withdraw(2000); // Should impose service charge
        currentAccount.displayBalance();


        System.out.println("\nCreating Savings Account:");
        scanner.nextLine(); // Consume newline
        System.out.println("Enter customer name:");
        String savName = scanner.nextLine();
        System.out.println("Enter account number:");
        int savNumber = scanner.nextInt();
        System.out.println("Enter initial balance:");
        double savBalance = scanner.nextDouble();
        SavingsAccount savingsAccount = new SavingsAccount(savName, savNumber,
savBalance);

        savingsAccount.deposit(1000);
        savingsAccount.displayBalance();
        savingsAccount.withdraw(500);
        savingsAccount.displayBalance();
        savingsAccount.computeInterest();
        savingsAccount.displayBalance();
    }
}
```

OUTPUT:

```
PS C:\Users\Admin\OneDrive\Desktop\Jv> javac Main.java
PS C:\Users\Admin\OneDrive\Desktop\Jv> java Main
Creating Current Account:
Enter customer name:
rahul
Enter account number:
223344
Enter initial balance:
10000
Deposit of $500.0 successful.
Account Balance: $10500.0
Withdrawal of $2000.0 successful.
Account Balance: $8500.0

Creating Savings Account:
Enter customer name:
Rohan
Enter account number:
334455
Enter initial balance:
20000
Deposit of $1000.0 successful.
Account Balance: $21000.0
Withdrawal of $500.0 successful.
Account Balance: $20500.0
Interest computed and deposited: $1025.0
Account Balance: $21525.0
PS C:\Users\Admin\OneDrive\Desktop\Jv>
```

LAB6:

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

CODE:

```
package CIE;

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

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

```
package CIE;

public class Internals {
    protected int[] marks = new int[5]; // Internal marks scored in five courses

    public Internals(int[] marks) {
        this.marks = marks;
    }
}
```

```
package SEE;
import CIE.Student;

public class External extends Student {
    protected int[] marks = new int[5]; // SEE marks scored in five courses

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

```
import CIE.Student;
import CIE.Internals;
import SEE.External;
```

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

        Student student1 = new Student("1", "Alice", 3);
        Internals internals1 = new Internals(new int[]{80, 75, 85, 90, 70});
        External external1 = new External("1", "Alice", 3, new int[]{70, 65, 75, 80, 60});


        System.out.println("Final Marks for " + student1.name + ":");
        for (int i = 0; i < 5; i++) {
            int finalMark = internals1.marks[i] + external1.marks[i];
            System.out.println("Course " + (i+1) + ": " + finalMark);
        }
    }
}
```

OUTPUT:

```
Enter the number of students: 2

Enter details for Student 1:
USN: 1bm22cs061
Name: Ayman Amjad
Semester: 3
Enter Internal marks for Student 1:
Internal marks for Course 1: 45
Internal marks for Course 2: 46
Internal marks for Course 3: 42
Internal marks for Course 4: 49
Internal marks for Course 5: 40
Enter External marks for Student 1:
External marks for Course 1: 90
External marks for Course 2: 94
External marks for Course 3: 96
External marks for Course 4: 93
External marks for Course 5: 99

Enter details for Student 2:
USN: 1bm22cs333
Name: Cassian Black
Semester: 3
Enter Internal marks for Student 2:
Internal marks for Course 1: 49
Internal marks for Course 2: 48
Internal marks for Course 3: 50
Internal marks for Course 4: 48
Internal marks for Course 5: 49
Enter External marks for Student 2:
External marks for Course 1: 99
External marks for Course 2: 100
External marks for Course 3: 100
External marks for Course 4: 99
External marks for Course 5: 97
```

LAB7:

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

CODE:

```
class WrongAge extends Exception {
    public WrongAge() {
        super("Age cannot be negative.");
    }
}

class Father {
    protected int age;

    public Father(int age) throws WrongAge {
        if (age < 0) {
            throw new WrongAge();
        }
        this.age = age;
    }
}

class Son extends Father {
    private int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAge, IllegalArgumentException {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new IllegalArgumentException("Son's age cannot be greater than or equal to
father's age.");
        }
        this.sonAge = sonAge;
    }
}

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

            Father father = new Father(40);


            Son son = new Son(40, 20);
```

```
            System.out.println("Father's age: " + father.age);
            System.out.println("Son's age: " + son.sonAge);
        } catch (WrongAge e) {
            System.out.println("Caught WrongAge exception: " + e.getMessage());
        } catch (IllegalArgumentException e) {
            System.out.println("Caught IllegalArgumentException: " + e.getMessage());
        }
    }
}
```

OUTPUT:

LAB8:

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

CODE:

```
public class Main {
    public static void main(String[] args) {
        Thread bmsThread = new Thread(new Runnable() {

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

        Thread cseThread = new Thread(new Runnable() {

            public void run() {
                while (true) {
                    System.out.println("CSE");
                    try {
                        Thread.sleep(2000); // Display every 2 seconds
                    } catch (InterruptedException e) {
                        e.printStackTrace();
                    }
                }
            }
        });

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

OUTPUT:

```
PS C:\Users\Admin\OneDrive\Desktop\Jv> javac Main.java
PS C:\Users\Admin\OneDrive\Desktop\Jv> java Main
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
```