



Name: Virak Rith

Student ID: P20230033

Course: OOP in java

Instructor: HOK Tin

Assignment: Lab05

Due Date: May 21, 2025 (11:59 AM)

## Exercise 1

---

### Source Codes

```
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.Scanner;

class MyDate {

    public void getCurrentDate() {
        SimpleDateFormat format = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");
        Date d = new Date();
        String formatted = format.format(d);
        System.out.println(formatted);
    }

    public long calculateDaysBtwDates(String date1, String date2) throws
    ParseException {
        SimpleDateFormat format = new SimpleDateFormat("dd/MM/yyyy");
        Date d1 = format.parse(date1);
        Date d2 = format.parse(date2);
        long diff = java.lang.Math.abs(d2.getTime() - d1.getTime());
        return diff / (1000 * 60 * 60 * 24);
    }

    public String findDay(String dateStr) throws ParseException {
        SimpleDateFormat format = new SimpleDateFormat("dd/MM/yyyy");
        Date date = format.parse(dateStr);
        Calendar calendar = Calendar.getInstance();
        calendar.setTime(date);
        String[] days = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday",
        "Friday", "Saturday"};
        int dayIndex = calendar.get(Calendar.DAY_OF_WEEK) - 1;
```

```
        return days[dayIndex];
    }

}

public class e1 {

    public static void main(String[] args) throws ParseException {

        Scanner scanner = new Scanner(System.in);
        MyDate myDate = new MyDate();
        int option;

        do {
            System.out.println("==== Menu ===");
            System.out.println("1. Current date and time");
            System.out.println("2. Calculate days btw two dates");
            System.out.println("3. Find the day of the week");
            System.out.println("4. Quit");
            System.out.print("Choose an option: ");
            option = scanner.nextInt();
            scanner.nextLine();

            switch (option) {
                case 1:
                    System.out.println("Current datetime is: ");
                    myDate.getCurrentDate();
                    break;

                case 2:
                    try {
                        System.out.print("First date (dd/MM/yyyy): ");
                        String d1 = scanner.nextLine();
                        System.out.print("Second date (dd/MM/yyyy): ");
                        String d2 = scanner.nextLine();
                        long days = myDate.calculateDaysBtwDates(d1, d2);
                        System.out.println("Difference between two dates is: " +
days + " days");
                    } catch (ParseException e) {
                        System.out.println("Invalid date format. Please use
dd/MM/yyyy.");
                    }
                    break;

                case 3:
                    try {
                        System.out.print("Input a date (dd/MM/yyyy): ");
                        String inputDate = scanner.nextLine();
                        String day = myDate.findDay(inputDate);
                        System.out.println("The day is: " + day);
                    } catch (ParseException e) {
                        System.out.println("Invalid date format. Please use
dd/mm/yyyy.");
                    }
            }
        }
    }
}
```

```
                break;

            case 4:
                System.out.println("Thank you for using our program><");
                break;

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

        System.out.println();
    } while (option != 4);

    scanner.close();
}
}
```

## Output

```
==== Menu ===
1. Current date and time
2. Calculate days btw two dates
3. Find the day of the week
4. Quit
Choose an option: 1
Current datetime is:
07/07/2025 16:13:52

==== Menu ===
1. Current date and time
2. Calculate days btw two dates
3. Find the day of the week
4. Quit
Choose an option: 2
First date (dd/MM/yyyy): 12/12/2003
Second date (dd/MM/yyyy): 12/12/2025
Difference between two dates is: 8036 days

==== Menu ===
1. Current date and time
2. Calculate days btw two dates
3. Find the day of the week
4. Quit
Choose an option: 3
Input a date (dd/MM/yyyy): 24/12/2003
The day is: Wednesday

==== Menu ===
1. Current date and time
```

```
2. Calculate days btw two dates
3. Find the day of the week
4. Quit
Choose an option: 4
Thank you for using our program><
```

## Exercise 2

---

### Source Code

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

class Color {

    public static final String RESET = "\u001B[0m";
    public static final String ORANGE = "\u001B[38;5;208m";
}

class Student {

    int id;
    String name;
    int age;

    Student(int id, String name, int age) {
        this.id = id;
        this.name = name;
        this.age = age;
    }

    @Override
    public String toString() {
        return String.format("%-8d | %-20s | %-5d |", id, name, age);
    }
}

public class ex2 {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);
        ArrayList<Student> stuArr = new ArrayList<>();
        ArrayList<String> resultMessage = new ArrayList<>();
        stuArr.add(new Student(101, "Alice", 20));
        stuArr.add(new Student(102, "Bob", 21));
        stuArr.add(new Student(103, "Charlie", 19));

        int choice;
```

```

do {

    System.out.println("==== Menu ====");
    System.out.println("1. Add new students");
    System.out.println("2. Delete multiple students");
    System.out.println("3. Quit");
    System.out.print("Choose an option: ");
    choice = scanner.nextInt();

    int count = 1;
    String answer;
    switch (choice) {
        case 1:

            do {

                System.out.println("Student #" + count);
                System.out.print("ID: ");
                int id = scanner.nextInt();
                scanner.nextLine();
                System.out.print("Name: ");
                String name = scanner.nextLine();
                System.out.print("Age: ");
                int age = scanner.nextInt();
                stuArr.add(new Student(id, name, age));
                count++;

                scanner.nextLine();
                System.out.print("Do you want to add more (y/n): ");
                answer = scanner.nextLine();

            } while (!answer.equalsIgnoreCase("n"));

            break;

        case 2:

            int deleteCount = 1;
            int no = 1;

            for (int i = 0; i < 54; i++) {
                System.out.print("=");
            }
            System.out.println();
            System.out.printf("| %-8s | %-8s | %-20s | %-5s |\n", "No",
"ID", "Name", "Age");
            for (int i = 0; i < 54; i++) {
                System.out.print("=");
            }
            System.out.println();
            for (Student stu : stuArr) {
                System.out.printf("| %-8d | %s \n", no++, stu);
            }

```

```
        for (int i = 0; i < 54; i++) {
            System.out.print("=");
        }
        System.out.println();

        System.out.println("==== DELETION ====");
        do {
            System.out.print("Input student #" + deleteCount + " ID:");

            int deleteId = scanner.nextInt();
            scanner.nextLine();

            boolean found = false;

            for (int i = 0; i < stuArr.size(); i++) {
                if (stuArr.get(i).id == deleteId) {
                    stuArr.remove(i);
                    resultMessage.add("Student with ID " + deleteId +
" deleted!");

                    found = true;
                    break;
                }
            }

            if (!found) {
                resultMessage.add("Student with ID " + deleteId + "
not found!");

                deleteCount++;
                System.out.print("Do you want to delete more (y/n)? ");
                answer = scanner.nextLine();

            } while (!answer.equalsIgnoreCase("n"));

            System.out.println();
            for (String rm : resultMessage) {
                System.out.println(Color.ORANGE + rm + Color.RESET);
            }
            resultMessage.clear();
            System.out.println();

            break;
        }

        case 3:

            System.out.println("\nThank you for using our program><");
            break;

        default:
            System.out.println("\nInvalid input, please try again...\n");
            break;
    }
}
```

```

    } while (choice != 3);

    scanner.close();
}
}

```

## Output:

```

==== Menu ====
1. Add new students
2. Delete multiple students
3. Quit
Choose an option: 1
Student #1
ID: 111
Name: rith
Age: 12
Do you want to add more (y/n): y
Student #2
ID: 222
Name: rith
Age: 22
Do you want to add more (y/n): y
Student #3
ID: 333
Name: rithy
Age: 13
Do you want to add more (y/n): n
==== Menu ====
1. Add new students
2. Delete multiple students
3. Quit
Choose an option: 2
=====
| No      | ID      | Name      | Age      |
=====
| 1       | 101     | Alice     | 20       |
| 2       | 102     | Bob       | 21       |
| 3       | 103     | Charlie   | 19       |
| 4       | 111     | rith      | 12       |
| 5       | 222     | rith      | 22       |
| 6       | 333     | rithy     | 13       |
=====
==== DELETION ====
Input student #1 ID: 111
Do you want to delete more (y/n)? : y
Input student #2 ID: 333
Do you want to delete more (y/n)? : n

Student with ID 111 deleted!

```

```
Student with ID 333 deleted!

==== Menu ====
1. Add new students
2. Delete multiple students
3. Quit
Choose an option: 3

Thank you for using our program><
```

## Exercise 3

---

### Source Code:

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

class Shape {

    public void display() {
        System.out.println("Generic Shape");
    }
}

class Line extends Shape {

    int length;

    public Line(int length) {
        this.length = length;
    }

    @Override
    public String toString() {
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < length; i++) {
            sb.append("_");
        }
        return sb.toString();
    }
}

class Rectangle extends Shape {

    int width;
    int height;

    public Rectangle(int width, int height) {
        this.width = width;
```



```

        this.height = height;
    }

    @Override
    public String toString() {
        StringBuilder sb = new StringBuilder("Width: " + width + ", Height: " +
height + "\n");
        for (int i = 0; i < height; i++) {
            for (int j = 0; j < width; j++) {
                if (i == 0 || i == height - 1 || j == 0 || j == width - 1) {
                    sb.append("*");
                } else {
                    sb.append(" ");
                }
            }
            sb.append("\n");
        }
        return sb.toString();
    }
}

class Triangle extends Shape {

    int height;

    public Triangle(int height) {
        this.height = height;
    }

    @Override
    public String toString() {
        StringBuilder sb = new StringBuilder("Height: " + height + "\n");
        for (int i = 1; i <= height; i++) {
            for (int j = 1; j <= i; j++) {
                sb.append("*");
            }
            sb.append("\n");
        }
        return sb.toString();
    }
}

public class e3 {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);
        ArrayList<Line> lines = new ArrayList<>();
        ArrayList<Rectangle> rectangles = new ArrayList<>();
        ArrayList<Triangle> triangles = new ArrayList<>();

        int option;

        do {

```

```
System.out.println("=== Menu ===");
System.out.println("1. View all lines");
System.out.println("2. View all rectangles");
System.out.println("3. View all triangles");
System.out.println("4. Add a new shape");
System.out.println("5. Quit");
System.out.print("Choose an Option: ");
option = scanner.nextInt();

int shapeOption;

switch (option) {

    case 1:

        int lineCount = 1;

        System.out.println("\n==== List all lines ====");
        for (Line line : lines) {
            System.out.println((lineCount++) + ". length: " +
line.length);
            System.out.println(line);
        }
        break;

    case 2:

        int rectangleCount = 1;

        System.out.println("\n==== List all Rectangles ====");
        for (Rectangle rectangle : rectangles) {
            System.out.print((rectangleCount++) + ". ");
            System.out.println(rectangle);
        }
        break;

    case 3:

        int triangleCount = 1;

        System.out.println("\n==== List all Triangles ====");
        for (Triangle triangle : triangles) {
            System.out.print((triangleCount++) + ". ");
            System.out.println(triangle);
        }
        break;

    case 4:

        System.out.println("==== Add new shape ====");
        System.out.println("Select a shape:");
        System.out.println("1. Line");
        System.out.println("2. Rectangle");
        System.out.println("3. Triangle");
        System.out.print("Choose an Option: ");
```

```

        shapeOption = scanner.nextInt();

        switch (shapeOption) {
            case 1:
                System.out.print("Input length: ");
                int length = scanner.nextInt();
                lines.add(new Line(length));
                break;

            case 2:
                System.out.print("Input width: ");
                int width = scanner.nextInt();
                System.out.print("Input height: ");
                int height = scanner.nextInt();
                rectangles.add(new Rectangle(width, height));
                break;

            case 3:
                System.out.print("Input hight: ");
                int heightForT = scanner.nextInt();
                triangles.add(new Triangle(heightForT));
                break;

            default:
                System.out.println("Invalid input, please try
again...");
                break;
        }
        break;

    case 5:
        System.out.println("Thank you for using our program><");
        break;

    default:
        System.out.println("Invalid input, please try again...");
        break;
}

} while (option != 5);

scanner.close();
}
}

```

## Output:

```

=== Menu ===
1. View all lines
2. View all rectangles

```

```
3. View all triangles
4. Add a new shape
5. Quit
Choose an Option: 4
==== Add new shape ====
Select a shape:
1. Line
2. Rectangle
3. Triangle
Choose an Option: 1
Input length: 5
=== Menu ===
1. View all lines
2. View all rectangles
3. View all triangles
4. Add a new shape
5. Quit
Choose an Option: 4
==== Add new shape ====
Select a shape:
1. Line
2. Rectangle
3. Triangle
Choose an Option: 2
Input width: 10
Input height: 2
=== Menu ===
1. View all lines
2. View all rectangles
3. View all triangles
4. Add a new shape
5. Quit
Choose an Option: 1

==== List all lines ====
1. length: 5

-----
=== Menu ===
1. View all lines
2. View all rectangles
3. View all triangles
4. Add a new shape
5. Quit
Choose an Option: 2

==== List all Rectangles ====
1. Width: 10, Height: 2
*****
*****

=== Menu ===
1. View all lines
2. View all rectangles
3. View all triangles
```

```
4. Add a new shape
5. Quit
Choose an Option: 5
Thank you for using our program><
```

## Exercise 4

---

### Source Code:

```
import java.io.*;
import java.util.*;

public class e4 {

    static final String FILE_NAME = "data.txt";

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        while (true) {
            System.out.println("==== Menu ====");
            System.out.println("1. View my note");
            System.out.println("2. Edit");
            System.out.println("3. Quit");
            System.out.print("Choose an option: ");
            int option = scanner.nextInt();
            scanner.nextLine();
            switch (option) {
                case 1:
                    viewNote();
                    break;
                case 2:
                    editMenu(scanner);
                    break;
                case 3:
                    System.out.println("Thank you for using our program><");
                    return;
                default:
                    System.out.println("Invalid choice.");
            }
        }
    }

    static void viewNote() {
        Scanner Scanner = new Scanner(System.in);
        List<String> lines = readLines();
        System.out.println("==== View note ====");
        System.out.println("*****");
        for (int i = 0; i < lines.size(); i++) {
            System.out.printf("%d | %s\n", (i + 1), lines.get(i));
        }
    }
}
```

```

    }
    System.out.println("*****");
    Scanner.nextLine();
}

static void editMenu(Scanner scanner) {
    List<String> lines = readLines();

    System.out.println("==== Edit following note ====");
    System.out.println("*****");
    for (int i = 0; i < lines.size(); i++) {
        System.out.printf("%d | %s\n", (i + 1), lines.get(i));
    }
    System.out.println("*****");
    System.out.println("1. Append new line");
    System.out.println("2. Update at line");
    System.out.println("3. Delete line");
    System.out.print("Choose an option: ");
    int choice = scanner.nextInt();
    scanner.nextLine();

    switch (choice) {
        case 1:
            System.out.printf("Input a string for line %d:\n", (lines.size()
+ 1));

            String newLine = scanner.nextLine();
            lines.add(newLine);
            System.out.printf("Line %d is appended to the note.\n",
lines.size());
            break;
        case 2:
            System.out.print("Enter line number to update: ");
            int updateIndex = scanner.nextInt() - 1;
            scanner.nextLine();
            if (updateIndex >= 0 && updateIndex < lines.size()) {
                System.out.print("Enter new text: ");
                String updatedText = scanner.nextLine();
                lines.set(updateIndex, updatedText);
                System.out.printf("Line %d updated.\n", updateIndex + 1);
            } else {
                System.out.println("Invalid line number.");
            }
            break;
        case 3:
            System.out.print("Delete line number: ");
            int deleteIndex = scanner.nextInt() - 1;
            scanner.nextLine();
            if (deleteIndex >= 0 && deleteIndex < lines.size()) {
                lines.remove(deleteIndex);
                System.out.printf("Line %d is deleted.\n", deleteIndex + 1);
            } else {
                System.out.println("Invalid line number.");
            }
            break;
    }
}

```

```

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

    writeLines(lines);
}

static List<String> readLines() {
    List<String> lines = new ArrayList<>();
    try (BufferedReader reader = new BufferedReader(new
FileReader(FILE_NAME))) {
        String line;
        while ((line = reader.readLine()) != null) {
            lines.add(line);
        }
    } catch (IOException e) {
        System.out.println("File not existed!");
    }
    return lines;
}

static void writelines(List<String> lines) {
    try (BufferedWriter writer = new BufferedWriter(new
FileWriter(FILE_NAME))) {
        for (String line : lines) {
            writer.write(line);
            writer.newLine();
        }
    } catch (IOException e) {
        System.out.println("Error writing file: " + e.getMessage());
    }
}
}

```

## Output:

```

==== Menu ====
1. View my note
2. Edit
3. Quit
Choose an option: 2
==== Edit following note ====
*****
1 | Hello world!
*****
1. Append new line
2. Update at line
3. Delete line
Choose an option: 1
Input a string for line #2:

```

```
this is second line
Line #2 is appended to the note.
==== Menu ====
1. View my note
2. Edit
3. Quit
Choose an option: 1
==== View note ====
*****
1 | Hello world!
2 | this is second line
*****
==== Menu ====
1. View my note
2. Edit
3. Quit
Choose an option: 3
Thank you for using our program><
```

## Exercise 5

---

### Source Code:

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

class Color {

    public static final String RESET = "\u001B[0m";
    public static final String RED = "\u001B[31m";
    public static final String GREEN = "\u001B[32m";
}

public class ex5 {

    static final String TEACHER_FILE = "Files\\teacher.txt";
    static final String STUDENT_FILE = "Files\\student.txt";
    static final String SECURITYGUARD_FILE = "Files\\securityguard.txt";

    public static void main(String[] args) throws Exception {

        Scanner scanner = new Scanner(System.in);

        int choice;
```



```
do {

    System.out.println("=== Menu ===");
    System.out.println("1. View all");
    System.out.println("2. Add new");
    System.out.println("3. Quit");
    System.out.print("Choose an option: ");
    choice = scanner.nextInt();

    switch (choice) {

        case 1:
            viewFile();
            break;

        case 2:

            while (true) {
                System.out.println("=== Add new resource ===");
                System.out.println("1. Teacher");
                System.out.println("2. Student");
                System.out.println("3. Security guard");
                System.out.print("Choose an option: ");
                int option = scanner.nextInt();
                scanner.nextLine();

                if (option == 1) {
                    addTeacher(scanner);
                    break;
                } else if (option == 2) {
                    addStudent(scanner);
                    break;
                } else if (option == 3) {
                    addSecurityguard(scanner);
                    break;
                } else {
                    System.out.println(Color.RED + "Invalid option, please
try again..." + Color.RESET);
                }
            }
            break;

        case 3:
            System.out.println("Thank you for using our program><");
            break;

        default:
            System.out.println(Color.RED + "Invalid choice, please try
again..." + Color.RESET);
            break;
    }
} while (choice != 3);
scanner.close();
```

```
}

static void addTeacher(Scanner scanner) {
    List<String> lines = readLines(TEACHER_FILE);
    System.out.print("First Name: ");

    String first_name = scanner.nextLine();
    System.out.print("Last Name: ");
    String last_name = scanner.nextLine();
    System.out.print("SEX: ");
    String sex = scanner.nextLine();
    System.out.print("Email: ");
    String email = scanner.nextLine();
    System.out.print("Subject: ");
    String subject = scanner.nextLine();
    System.out.print("Salary: ");
    String salary = Double.toString(scanner.nextDouble());

    String teacher = String.format("%d. [%s %s][%s][%s][%s][%s$]",
    (lines.size() + 1), first_name, last_name, sex.toUpperCase(),
        email, subject, salary);
    lines.add(teacher);
    writeLines(TEACHER_FILE, lines);
}

static void addStudent(Scanner scanner) {

    List<String> lines = readLines(STUDENT_FILE);

    System.out.print("First Name: ");
    String first_name = scanner.nextLine();
    System.out.print("Last Name: ");
    String last_name = scanner.nextLine();
    System.out.print("SEX: ");
    String sex = scanner.nextLine();
    System.out.print("Email: ");
    String email = scanner.nextLine();
    System.out.print("Year: ");
    String year = Integer.toString(scanner.nextInt());
    scanner.nextLine();
    System.out.print("Major: ");
    String major = scanner.nextLine();

    String student = String.format("%d. [%s %s][%s][%s][%s][%s]",
    (lines.size() + 1), first_name, last_name, sex.toUpperCase(),
        email, year, major);

    lines.add(student);
    writeLines(STUDENT_FILE, lines);

}

static void addSecurityguard(Scanner scanner) {
```

```

List<String> lines = readLines(SEcurityGUARD_FILE);

System.out.print("First Name: ");
String first_name = scanner.nextLine();
System.out.print("Last Name: ");
String last_name = scanner.nextLine();
System.out.print("SEX: ");
String sex = scanner.nextLine();
System.out.print("Email: ");
String email = scanner.nextLine();
System.out.print("Position: ");
String position = scanner.nextLine();

String securityguard = String.format("%d. [%s %s][%s][%s][%s]",
(lines.size() + 1), first_name, last_name, sex.toUpperCase(), email, position);

lines.add(securityguard);
writeLines(SEcurityGUARD_FILE, lines);

}

static void viewFile() {

    List<String> linesFromTeacher = readLines(TEACHER_FILE);
    System.out.println("\n==== Teacher ====");
    for (int i = 0; i < linesFromTeacher.size(); i++) {
        System.out.println(Color.GREEN + linesFromTeacher.get(i) +
Color.RESET);
    }

    List<String> linesFromStu = readLines(STUDENT_FILE);
    System.out.println("\n==== Student ====");
    for (int i = 0; i < linesFromStu.size(); i++) {
        System.out.println(Color.GREEN + linesFromStu.get(i) + Color.RESET);
    }

    List<String> linesFromSG = readLines(SEcurityGUARD_FILE);
    System.out.println("\n==== Security guard ====");
    for (int i = 0; i < linesFromSG.size(); i++) {
        System.out.println(Color.GREEN + linesFromSG.get(i) + Color.RESET);
    }
}

static List<String> readLines(String filename) {
    List<String> lines = new ArrayList<>();
    try (BufferedReader reader = new BufferedReader(new FileReader(filename)))
    {
        String line;
        while ((line = reader.readLine()) != null) {
            lines.add(line);
        }
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
}

```

```

        return lines;
    }

    static void writelines(String filename, List<String> lines) {
        try (BufferedWriter writer = new BufferedWriter(new FileWriter(filename)))
        {
            for (String line : lines) {
                writer.write(line);
                writer.newLine();
            }
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
}

```

## Output:

```

==== Add new resource ====
1. Teacher
2. Student
3. Security guard
Choose an option: 1
Files\teacher.txt (The system cannot find the file specified)
First Name: rith
Last Name: virak
SEX: M
Email: hello@gmail.com
Subject: math
Salary: 1200
=== Menu ===
1. View all
2. Add new
3. Quit
Choose an option: 1

==== Teacher ====
1. [rith virak][M][hello@gmail.com][math][1200.0$]
Files\student.txt (The system cannot find the file specified)

==== Student ====
Files\securityguard.txt (The system cannot find the file specified)

==== Security guard ====
=== Menu ===
1. View all
2. Add new
3. Quit
Choose an option: 3
Thank you for using our program><

```

## Exercise 6

---

### Source Code:

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class e6 {

    static final String FILE_NAME = "Files\\user.txt";

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        int option;

        do {

            System.out.println("=== Menu ===");
            System.out.println("1. Login");
            System.out.println("2. Register");
            System.out.println("3. Quit");
            System.out.print("Choose an option: ");
            option = scanner.nextInt();
            scanner.nextLine();

            switch (option) {
                case 1:

                    List<String> users = readLines();
                    System.out.println("\n=== Login ===");
                    System.out.print("Email or Username: ");
                    String input = scanner.nextLine().trim();
                    System.out.print("Password: ");
                    String inputPassword =
String.valueOf(System.console().readPassword()).trim();

                    boolean found = false;

                    for (String line : users) {
                        String[] parts = line.split(",");
                        if (parts.length >= 5) {
```

```

        String first_name = parts[0];
        String last_name = parts[1];
        String linesUsername = parts[2];
        String linesEmail = parts[3];
        String linesPassword = parts[4];

        if ((input.equals(linesUsername.trim()) ||
input.equals(linesEmail.trim())) && inputPassword.equals(linesPassword.trim())) {
            System.out.println("\n===== User Info
=====");
            System.out.println(Color.ORANGE + "Hi " +
first_name + " " + last_name + ",");
            System.out.println("Your username is: " +
linesUsername);
            System.out.println("Your email is: " + linesEmail
+ "\n" + Color.RESET);
            found = true;
            break;
        }
    }

    if (!found) {
        System.out.println("\nUser not found, please try again..."
+ Color.RESET);
    }
    break;

case 2:

    List<String> lines = readLines();
    System.out.println("\n=== Register a new user ===");
    System.out.print("First Name: ");
    String first_name = scanner.nextLine();
    System.out.print("Last Name: ");
    String last_name = scanner.nextLine();
    System.out.print("Username: ");
    String username = scanner.nextLine();
    System.out.print("Email: ");
    String email = scanner.nextLine();
    System.out.print("Password: ");
    String password =
String.valueOf(System.console().readPassword()).trim();

    String user = String.format("%s,%s,%s,%s,%s", first_name,
last_name, username, email, password);

    lines.add(user);
    writelines(lines);
    System.out.println("\nYou are successfully registered~" +
Color.RESET);

    break;

case 3:

```

```

        System.out.println("Exiting program...");
        break;

        default:
            System.out.println("Invalid input, please try again..." +
Color.RESET);
            break;
    }

    } while (option != 3);

    scanner.close();
}

static List<String> readLines() {
    List<String> lines = new ArrayList<>();
    try (BufferedReader reader = new BufferedReader(new
FileReader(FILE_NAME))) {
        String line;
        while ((line = reader.readLine()) != null) {
            lines.add(line);
        }
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
    return lines;
}

static void writelines(List<String> lines) {
    try (BufferedWriter writer = new BufferedWriter(new
FileWriter(FILE_NAME))) {
        for (String line : lines) {
            writer.write(line);
            writer.newLine();
        }
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
}
}

```

## Output:

```

=== Menu ===
1. Login
2. Register
3. Quit
Choose an option: 2

==== Register a new user ====

```

```
First Name: rith
Last Name: virak
Username: pop
Email: rith@gmail.com
Password:
```

You are successfully registered~

=== Menu ===

1. Login
2. Register
3. Quit

Choose an option: 1

==== Login ====

Email or Username: pop  
Password:

===== User Info =====

Hi rith virak,  
Your username is: pop  
Your email is: rith@gmail.com

=== Menu ===

1. Login
2. Register
3. Quit

Choose an option: 3

Exiting program...

Link to GitHub Account : [Click Here](#) 

Note: Viewing in VsCode IDE for better formatting!!!