Experiment 2.2

Student Name: Nikhil Arya **UID: 22BCS15728**
Branch: CSE **Section: 640-A**
Semester: 6th **DOP:24-02-2025**
Subject: PBLJ **Subject Code: 22CSH-359**

1. **Aim: Develop Java programs using autoboxing, serialization, file handling, and efficient data processing and management.**

2. **Problem Statements:**

- **Easy Level:** Write a Java program to calculate the sum of a list of integers using autoboxing and unboxing. Include methods to parse strings into their respective wrapper classes (e.g., Integer.parseInt()).

- **Medium Level:** Create a Java program to serialize and deserialize a Student object. The program should: Serialize a Student object (containing id, name, and GPA) and save it to a file. Deserialize the object from the file and display the student details.

- **Hard Level:** Create a menu-based Java application with the following options. 1.Add an Employee 2. Display All 3. Exit If option 1 is selected, the application should gather details of the employee like employee name, employee id, designation and salary and store it in a file. If option 2 is selected, the application should display all the employee details. If option 3 is selected the application should exit..

3. **Implementation/Code:**

Problem: Easy

```
import java.util.ArrayList;

import java.util.List;


public class exp5 {
    public static Integer parseStringToInteger(String str) {
        return Integer.parseInt(str);
    }

    public static int calculateSum(List<Integer> numbers) {
        int sum = 0;
        for (Integer number : numbers) {
            sum += number;
        }
        return sum;
    }

    public static void main(String[] args) {
        List<String> stringNumbers = new ArrayList<>();
        stringNumbers.add("10");
        stringNumbers.add("20");
        stringNumbers.add("30");
        stringNumbers.add("40");

        List<Integer> numbers = new ArrayList<>();
        for (String str : stringNumbers) {
            numbers.add(parseStringToInteger(str));
        }

        int sum = calculateSum(numbers);

        System.out.println(" sum= " + sum);
    }
}
```

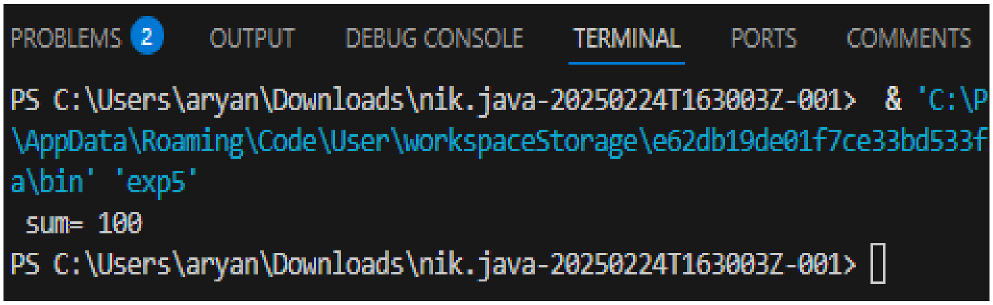
Problem:medium

```
import java.io.*;
class Student implements Serializable {
    private static final long serialVersionUID = 1L;
    private int id;
    private String name;
    private double gpa;
    public Student(int id, String name, double gpa) {
        this.id = id;
        this.name = name;
        this.gpa = gpa;
    }
    @Override
    public String toString() {
        return "Student ID: " + id + ", Name: " + name + ", GPA: " + gpa;
    }
}
public class exp5med {
    public static void serializeStudent(Student student) {
        try (ObjectOutputStream oos = new ObjectOutputStream(new
FileOutputStream("student.ser"))) {
            oos.writeObject(student);
            System.out.println("Student object has been serialized and saved to
file.");
        } catch (FileNotFoundException e) {
            System.out.println("Error: File not found.");
        } catch (IOException e) {
            System.out.println("Error: IOException occurred during
serialization.");
        }
    }
    public static Student deserializeStudent() {
        try (ObjectInputStream ois = new ObjectInputStream(new
FileInputStream("student.ser"))) {
            Student student = (Student) ois.readObject();
            System.out.println("Student object has been deserialized.");
            return student;
        } catch (FileNotFoundException e) {
            System.out.println("Error: File not found.");
        } catch (IOException e) {
            System.out.println("Error: IOException occurred during
deserialization.");
        } catch (ClassNotFoundException e) {
            System.out.println("Error: Class not found.");
        }
        return null;
    }
    public static void main(String[] args) {
        Student student1 = new Student(13251, "Manas", 8.3);
        serializeStudent(student1);
        Student deserializedStudent = deserializeStudent();
        if (deserializedStudent != null) {
            System.out.println("Deserialized Student Details:");
            System.out.println(deserializedStudent);
        }
        System.out.println("\nTest Case 2: Attempting to deserialize from a
non-existent file.");
        new File("student.ser").delete();
        deserializeStudent();
        System.out.println("\nTest Case 3: Simulating
ClassNotFoundException.");
        deserializeStudent();
    }
}
```

Problem:Hard

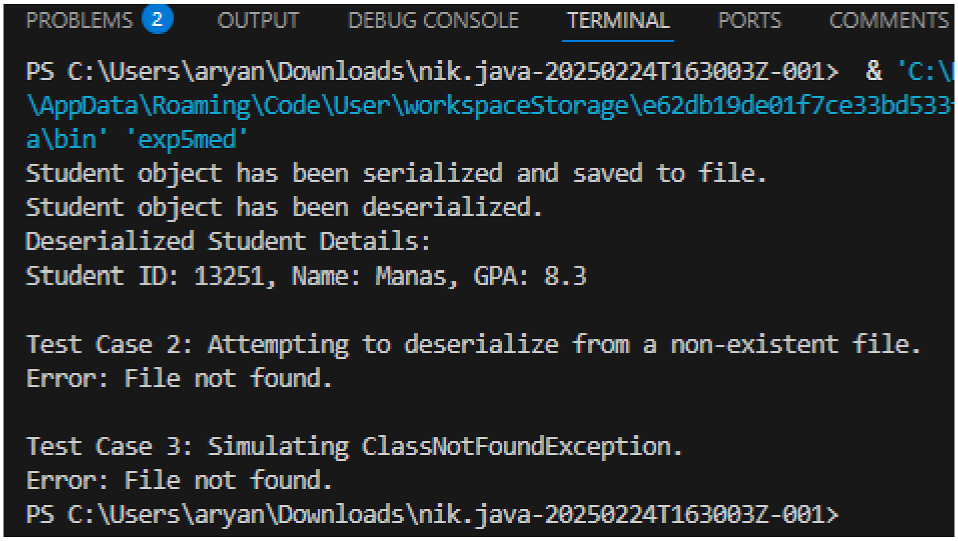
```
import java.io.*;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

class Employee implements Serializable {
    private static final long serialVersionUID = 1L;
    private String name;
    private int id;
    private String designation;
    private double salary;
    public Employee(String name, int id, String designation, double salary) {
        this.name = name;
        this.id = id;
        this.designation = designation;
        this.salary = salary;
    }
    @Override
    public String toString() {
        return "Employee ID: " + id + ", Name: " + name + ", Designation: " +
designation + ", Salary: " + salary;
    }
}
public class exp53 {
    private static final String FILE_NAME = "employees.ser";
    public static void addEmployee() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Employee Name: ");
        String name = scanner.nextLine();
        System.out.print("Employee ID: ");
        int id = scanner.nextInt();
        scanner.nextLine();
        System.out.print("Designation: ");
        String designation = scanner.nextLine();
        System.out.print("Salary: ");
        double salary = scanner.nextDouble();
        Employee employee = new Employee(name, id, designation, salary);
        saveEmployeeToFile(employee);
        System.out.println("Employee added successfully!");
    }
    private static void saveEmployeeToFile(Employee employee) {
        List<Employee> employees = readEmployeesFromFile();
        employees.add(employee);
        try (ObjectOutputStream oos = new ObjectOutputStream(new
FileOutputStream(FILE_NAME))) {
            for (Employee emp : employees) {
                oos.writeObject(emp);
            }
        } catch (IOException e) {
            System.out.println("Error saving employee to file: " + e.getMessage());
        }
    }
    public static void displayAllEmployees() {
        List<Employee> employees = readEmployeesFromFile();
        if (employees.isEmpty()) {
            System.out.println("No employees found.");
        } else {
            for (Employee employee : employees) {
                System.out.println(employee);
            }
        }
    }
    private static List<Employee> readEmployeesFromFile() {
        List<Employee> employees = new ArrayList<>();
        try {
            File file = new File(FILE_NAME);
            if (file.exists()) {
                try (ObjectInputStream ois = new ObjectInputStream(new
FileInputStream(file))) {
                    while (true) {
                        Employee employee = (Employee) ois.readObject();
                        employees.add(employee);
                    }
                } catch (EOFException e) {
                } catch (IOException | ClassNotFoundException e) {
                    System.out.println("Error reading employees from file: " +
e.getMessage());
                }
            }
        } catch (Exception e) {
            System.out.println("Error: " + e.getMessage());
        }
        return employees;
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int choice;
        do {
            System.out.println("\nMenu:");
            System.out.println("1. Add Employee");
            System.out.println("2. Display All Employees");
            System.out.println("3 . Exit");
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt();
            switch (choice) {
                case 1:
                    addEmployee();
                    break;
                case 2:
                    displayAllEmployees();
                    break;
                case 3:
                    System.out.println("Exiting...");
                    break;
                default:
                    System.out.println("Invalid choice. Please try again.");
            }
        } while (choice != 3);
    }
}
```

4. Output:



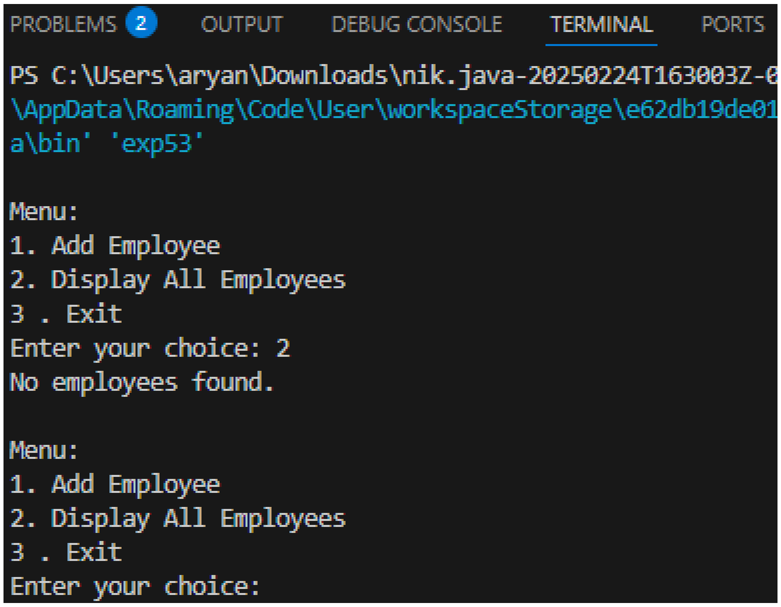
```
PS C:\Users\aryan\Downloads\nik.java-20250224T163003Z-001>  & 'C:\P
\AppData\Roaming\Code\User\workspaceStorage\e62db19de01f7ce33bd533f
a\bin' 'exp5'
 sum= 100
PS C:\Users\aryan\Downloads\nik.java-20250224T163003Z-001> 
```

(Fig. 1)



```
PS C:\Users\aryan\Downloads\nik.java-20250224T163003Z-001>  & 'C:\
\AppData\Roaming\Code\User\workspaceStorage\e62db19de01f7ce33bd533
a\bin' 'exp5med'
Student object has been serialized and saved to file.
Student object has been deserialized.
Deserialized Student Details:
Student ID: 13251, Name: Manas, GPA: 8.3

Test Case 2: Attempting to deserialize from a non-existent file.
Error: File not found.

Test Case 3: Simulating ClassNotFoundException.
Error: File not found.
PS C:\Users\aryan\Downloads\nik.java-20250224T163003Z-001>
```

(Fig. 2)



```
PS C:\Users\aryan\Downloads\nik.java-20250224T163003Z-0
\AppData\Roaming\Code\User\workspaceStorage\e62db19de01
a\bin' 'exp53'

Menu:
1. Add Employee
2. Display All Employees
3 . Exit
Enter your choice: 2
No employees found.

Menu:
1. Add Employee
2. Display All Employees
3 . Exit
Enter your choice:
```

(Fig. 3)

5. Learning Outcome:

1. Learn to manage collections dynamically using ArrayList and HashMap.
2. Understand thread synchronization and priority handling in Java.
3. Develop real-world applications with efficient data management techniques.