EXP-5

1. 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()).

Code:

Ayush_22BCS14610_Easy_5.java > Ayush_22BCS14610_Easy_5 > calculateSum(List<Intege

```java
import java.util.*;

public class Ayush_22BCS14610_Easy_5 {
    Run | Debug
    public static void main(String[] args) {
        List<Integer> numbers = new ArrayList<>();
        String[] inputs = {"10", "20", "30", "40", "50"};

        // Parsing strings and adding to the list (Autoboxing)
        for (String input : inputs) {
            numbers.add(parseToInteger(input));
        }

        // Calculating sum (Unboxing)
        int sum = calculateSum(numbers);

        // Display result
        System.out.println("Sum of numbers: " + sum);
    }

    public static Integer parseToInteger(String str) {
        return Integer.parseInt(str); // Autoboxing
    }

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

Output:

```
OUTPUT    TERMINAL    PORTS

PS D:\New folder>  & 'C:\Pro
c39c5b98507ba4538\redhat.jav
Sum of numbers: 150
PS D:\New folder>
```

2. 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. Handle FileNotFoundException, IOException, and ClassNotFoundException using exception handling.

Code:

D:\New folder\Ayush_22BCS14610_Medium_5.java (preview)

```java
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 Ayush_22BCS14610_Medium_5 {
    private static final String FILE_NAME = "student.ser";

    public static void serializeStudent(Student student) {
        try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(FILE_NAME))) {
            oos.writeObject(student);
            System.out.println(x:"Student object serialized successfully.");
        } catch (FileNotFoundException e) {
            System.err.println("File not found: " + e.getMessage());
        } catch (IOException e) {
            System.err.println("Error during serialization: " + e.getMessage());
        }
    }

    public static Student deserializeStudent() {
        try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(FILE_NAME))) {
            return (Student) ois.readObject();
        } catch (FileNotFoundException e) {
            System.err.println("File not found: " + e.getMessage());
        } catch (IOException e) {
            System.err.println("Error during deserialization: " + e.getMessage());
        } catch (ClassNotFoundException e) {
            System.err.println("Class not found: " + e.getMessage());
        }
        return null;
    }
```

```java
Run | Debug
public static void main(String[] args) {
    Student student = new Student(id:1, name:"Ayush", gpa:3.8);
    serializeStudent(student);

    Student deserializedStudent = deserializeStudent();
    if (deserializedStudent != null) {
        System.out.println("Deserialized Student: " + deserializedStudent);
    }
}
```

Output:

```
PS D:\New folder>  & 'C:\Program Files\Eclipse Adoptium\jdk-17.
c39c5b98507ba4538\redhat.java\jdt_ws\New folder_f73aa647\bin' '
Student object serialized successfully.
Deserialized Student: Student{id=1, name='Ayush', gpa=3.8}
PS D:\New folder>
```

3. 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.

Code:

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

class Employee implements Serializable {
    private static final long serialVersionUID = 1L;
    private int id;
    private String name;
    private String designation;
    private double salary;

    public Employee(int id, String name, String designation, double salary) {
        this.id = id;
        this.name = name;
        this.designation = designation;
        this.salary = salary;
    }

    @Override
    public String toString() {
        return "Employee ID: " + id + ", Name: " + name + ", Designation: " + designation + ", Salary: " + salary;
    }
}

public class Ayush_22BCS14610_Hard_5 {
    private static final String FILE_NAME = "employees.ser";

    Run | Debug
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        while (true) {
            System.out.println(x:"1. Add an Employee");
            System.out.println(x:"2. Display All Employees");
            System.out.println(x:"3. Exit");
            System.out.print(s:"Enter your choice: ");
            int choice = scanner.nextInt();
            scanner.nextLine();

            switch (choice) {
                case 1:
                    addEmployee(scanner);
                    break;
                case 2:
                    displayEmployees();
                    break;
                case 3:
                    System.out.println(x:"Exiting application.");
                    scanner.close();
                    return;
```

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

    private static void addEmployee(Scanner scanner) {
        System.out.print(s:"Enter Employee ID: ");
        int id = scanner.nextInt();
        scanner.nextLine();
        System.out.print(s:"Enter Employee Name: ");
        String name = scanner.nextLine();
        System.out.print(s:"Enter Designation: ");
        String designation = scanner.nextLine();
        System.out.print(s:"Enter Salary: ");
        double salary = scanner.nextDouble();

        Employee employee = new Employee(id, name, designation, salary);
        List<Employee> employees = readEmployees();
        employees.add(employee);
        writeEmployees(employees);
        System.out.println(x:"Employee added successfully!");
    }

    private static void displayEmployees() {
        List<Employee> employees = readEmployees();
        if (employees.isEmpty()) {
            System.out.println(x:"No employees found.");
        } else {
            for (Employee employee : employees) {
                System.out.println(employee);
            }
        }
    }

    private static List<Employee> readEmployees() {
        List<Employee> employees = new ArrayList<>();
        try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(FILE_NAME))) {
            employees = (List<Employee>) ois.readObject();
        } catch (FileNotFoundException e) {
            // File not found, return empty list
        } catch (IOException | ClassNotFoundException e) {
            System.err.println("Error reading employee data: " + e.getMessage());
        }
        return employees;
```

```java
    private static void writeEmployees(List<Employee> employees) {
        try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(FILE_NAME))) {
            oos.writeObject(employees);
        } catch (IOException e) {
            System.err.println("Error writing employee data: " + e.getMessage());
        }
    }
}
```

Output:

```
OUTPUT    TERMINAL    PORTS    DEBUG CONSOLE    PROBLEMS 1

PS D:\New folder>  & 'C:\Program Files\Eclipse Adoptium\jdk-17.0.8.7-ho
c39c5b98507ba4538\redhat.java\jdt_ws\New folder_f73aa647\bin' 'Ayush_22
1. Add an Employee
2. Display All Employees
3. Exit
Enter your choice: 1
Enter Employee ID: 101
Enter Employee Name: Ayush
Enter Designation: SDE1
Enter Salary: 10000
Employee added successfully!
1. Add an Employee
2. Display All Employees
3. Exit
Enter your choice: 2
Employee ID: 101, Name: Ayush, Designation: SDE1, Salary: 10000.0
1. Add an Employee
2. Display All Employees
3. Exit
Enter your choice: 3
Exiting application.
PS D:\New folder>
```