Experiment 5

Student Name: Harshad Fozdar
UID: 22BCS10263
Branch: BE-CSE
Section/Group: 22BCS_DL-901
Semester: 6
Date of Performance: 06/02/2025
Subject Name: Project Based Learning in Java with Lab
Subject Code: 22CSH-359

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

2. **Objective:**
   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()).
   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.
   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.

3. **Implementation/Code:**
   1. ***SumCalWrapper***

```
import java.util.ArrayList;
import java.util.Scanner;
public class SumCalWrapper {
   public static void main(String[] args)
      { Scanner input = new Scanner(System.in);
```

```
        System.out.print("Enter the number of elements: ");
        int n = Integer.parseInt(input.nextLine());
        ArrayList<Integer> numbers = new ArrayList<>();
        int sum = 0;
        System.out.println("Enter elements:");
        for (int i = 0; i < n; i++) {
            int num = Integer.parseInt(input.next());
            numbers.add(num);
            sum += num;
        }
        System.out.println("The sum of the entered numbers is: " + sum);
        input.close();
    } }
```

2. *Serialisation*

```
import java.io.*;
import java.util.Scanner;
class Student implements Serializable
    { 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;
    }
    public void display()
        { System.out.println("\nDeserialization \nStudent Data:");
        System.out.println("ID: " + id);
        System.out.println("Name: " + name);
        System.out.println("GPA: " + gpa);
    }
}
public class Serialisation {
    public static void main(String[] args)
        { Scanner input = new
        Scanner(System.in);
        System.out.print("Enter Student ID: ");
        int id = input.nextInt();
```

```
        input.nextLine();
        System.out.print("Enter Student Name: ");
        String name = input.nextLine();
        System.out.print("Enter Student GPA: ");
        double gpa = input.nextDouble();
        Student student = new Student(id, name, gpa);
        String file = "D:\\22bcs13216\\6\\java\\code\\ser_file.txt";
        try (ObjectOutputStream out = new ObjectOutputStream(new
FileOutputStream(file))) {
            out.writeObject(student);
            System.out.println("\nSaved to file: " + file);
        } catch (IOException e)
            { System.out.println("Error : " +
            e.getMessage());
        }
        try (ObjectInputStream in = new ObjectInputStream(new FileInputStream(file)))
            { Student deserStudent = (Student) in.readObject();
            deserStudent.display();
        } catch (FileNotFoundException e) {
            System.out.println("Error: File not found - " + e.getMessage());
        } catch (IOException e) {
            System.out.println("Error: IO Exception - " + e.getMessage());
        } catch (ClassNotFoundException e) {
            System.out.println("Error: Class not found - " + e.getMessage());
        }}}
```

3. *EmployeeApp*

```
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;
    }
    public void display()
        { System.out.println("ID: " + id);
        System.out.println("Name: " + name);
        System.out.println("Designation: " + designation);
        System.out.println("Salary: " + salary);
        System.out.println("----------");
    }
}
public class EmployeeApp {
    private static final String FILE_NAME = "employees.dat";
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        List<Employee> employees = loadEmployees();
        while (true) {
            System.out.println("\n1. Add an Employee");
            System.out.println("2. Display All Employees");
            System.out.println("3. Exit");
            System.out.print("Enter your choice: ");
            int choice = input.nextInt();
            input.nextLine();
            if (choice == 1) {
                System.out.print("Enter Employee ID: ");
                int id = input.nextInt();
                input.nextLine();
                System.out.print("Enter Employee Name: ");
                String name = input.nextLine();
                System.out.print("Enter Designation: ");
                String designation = input.nextLine();
                System.out.print("Enter Salary: ");
                double salary = input.nextDouble();
                employees.add(new Employee(id, name, designation, salary));
                saveEmployees(employees);
                System.out.println("Employee added successfully!");
            } else if (choice == 2) {
                if (employees.isEmpty()) {
                    System.out.println("No employee records found.");
```

```
                } else {
                    for (Employee emp : employees)
                        { emp.display();
                    }
                }
            } else if (choice == 3)
                { System.out.println("Exiting...");
                input.close();
                return;
            } else {
                System.out.println("Invalid choice. Try again.");
            }}}
    private static List<Employee> loadEmployees()
        { List<Employee> employees = new ArrayList<>();
        try (ObjectInputStream in = new ObjectInputStream(new
FileInputStream(FILE_NAME))) {
            employees = (List<Employee>) in.readObject();
        } catch (FileNotFoundException e) {
        } catch (IOException | ClassNotFoundException e)
            { System.out.println("Error loading employees: " + e.getMessage());
        }    return employees;     }
    private static void saveEmployees(List<Employee> employees)
        { try (ObjectOutputStream out = new ObjectOutputStream(new
FileOutputStream(FILE_NAME)))
            { out.writeObject(employees);
        } catch (IOException e) {
            System.out.println("Error saving employees: " + e.getMessage());
        }}}
```

4. Output

1. SumCalWrapper

```
PS D:\22bcs13216\6\java\code> java SumCalWrapper
Enter the number of elements: 4
Enter elements:
56 89 5 23
The sum of the entered numbers is: 173
```

2. Serialisation

```
PS D:\22bcs13216\6\java\code> java Serialisation
Enter Student ID: 101
Enter Student Name: ABC
Enter Student GPA: 4.0

Saved to file: D:\22bcs13216\6\java\code\ser_file.txt

Deserialization
Student Data:
ID: 101
Name: ABC
GPA: 4.0
```

3. **EmployeeApp**

```
PS D:\22bcs13216\6\java\code> java EmployeeApp

1. Add an Employee
2. Display All Employees
3. Exit
Enter your choice: 1
Enter Employee ID: 101
Enter Employee Name: ABC
Enter Designation: SDE
Enter Salary: 10000
Employee added successfully!

1. Add an Employee
2. Display All Employees
3. Exit
Enter your choice: 2
ID: 100
Name: ABC
Designation: SDE
Salary: 10000.0
----------------------------
ID: 101
Name: ABC
Designation: SDE
Salary: 10000.0
----------------------------

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

5. Learning Outcome:

- Autoboxing, Unboxing
- Serialization, file handling
- Exception handling
- Collections in Java