Experiment-5

Student Name: Kamalpreet Singh **UID: 22BCS11720**
Branch: B.E- CSE **Section/Group: IOT_643-A**
Semester: 6th **DateofPerformance:24-02-2025**
Subject Name: Project Based Learning Java with Lab **Subject Code: 22CSH-359**

Aim:

Develop Java programs using autoboxing, serialization, file handling, and efficient data processing and management.

Objective:

- Implement autoboxing and unboxing for efficient data handling.
- Serialize and deserialize objects using file handling.
- Develop a menu-based application for managing employee data with persistence.

Implementation/Code:

Easy Level: Sum of Integers Using Autoboxing & Unboxing

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

public class AutoboxingExample {
    public static void main(String[] args) {
        ArrayList<Integer> numbers = new ArrayList<>();
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of elements: ");
        int n = scanner.nextInt();

        System.out.println("Enter the numbers:");
        for (int i = 0; i < n; i++) {
            numbers.add(scanner.nextInt()); // Autoboxing: int → Integer
        }

        int sum = 0;
        for (Integer num : numbers) {
            sum += num; // Unboxing: Integer → int
        }

        System.out.println("Sum of numbers: " + sum);

        System.out.print("Enter a number as a string: ");
```

```
        scanner.nextLine(); // Consume newline
        String strNum = scanner.nextLine();
        int parsedNum = Integer.parseInt(strNum); // Parsing String to int
        System.out.println("Parsed number: " + parsedNum);

        scanner.close();
    }
}
```

Medium Level: Serialization & Deserialization of a Student Object

```
import java.io.*;

class Student implements Serializable {
    private static final long serialVersionUID = 1L;
    int id;
    String name;
    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("ID: " + id + ", Name: " + name + ", GPA: " + gpa);
    }
}

public class StudentSerialization {
    public static void main(String[] args) {
        String filename = "student_data.ser";
        Student student = new Student(101, "Kamalpreet", 3.9);

        try (ObjectOutputStream oos = new ObjectOutputStream(new
FileOutputStream(filename))) {
            oos.writeObject(student);
            System.out.println("Student object serialized successfully.");
        } catch (IOException e) {
            System.out.println("Serialization Error: " + e.getMessage());
        }

        try (ObjectInputStream ois = new ObjectInputStream(new
FileInputStream(filename))) {
            Student deserializedStudent = (Student) ois.readObject();
            System.out.println("Deserialized Student:");
            deserializedStudent.display();
        } catch (FileNotFoundException e) {
            System.out.println("File not found: " + e.getMessage());
        } catch (IOException | ClassNotFoundException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

Hard Level: Employee Management System Using File Handling

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

class Employee implements Serializable {
    private static final long serialVersionUID = 1L;
    int id;
    String name, designation;
    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 + ", Name: " + name + ", Designation: " +
designation + ", Salary: $" + salary);
    }
}

public class EmployeeManagement {
    private static final String FILE_NAME = "employees.dat";

    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();
            scanner.nextLine();

            switch (choice) {
                case 1:
                    addEmployee(scanner);
                    break;
                case 2:
                    displayEmployees();
                    break;
                case 3:
                    System.out.println("Exiting...");
                    break;
                default:
                    System.out.println("Invalid choice! Try again.");
            }
        } while (choice != 3);
```

```
        scanner.close();
    }

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

        Employee emp = new Employee(id, name, designation, salary);

        try (ObjectOutputStream oos = new ObjectOutputStream(new
FileOutputStream(FILE_NAME, true))) {
            oos.writeObject(emp);
            System.out.println("Employee added successfully.");
        } catch (IOException e) {
            System.out.println("Error saving employee: " + e.getMessage());
        }
    }

    private static void displayEmployees() {
        try (ObjectInputStream ois = new ObjectInputStream(new
FileInputStream(FILE_NAME))) {
            while (true) {
                Employee emp = (Employee) ois.readObject();
                emp.display();
            }
        } catch (EOFException e) {
            System.out.println("All employees displayed.");
        } catch (IOException | ClassNotFoundException e) {
            System.out.println("Error reading file: " + e.getMessage());
        }
    }
}
```

Output:

Easy-

```
Enter the number of elements: 3
Enter the numbers:
1
2
3
Sum of numbers: 6
Enter a number as a string: 10
Parsed number: 10
```

Medium-

```
Student object serialized successfully.
Deserialized Student:
ID: 11720, Name: Kamalpreet, GPA: 8.4
```

Hard-

```
Enter your choice: 1
Enter Employee ID: 11720
Enter Name: Kamalpreet Singh
Enter Designation: Data Scientist
Enter Salary: 450000
Employee added successfully.

Menu:
1. Add Employee
2. Display All Employees
3. Exit
Enter your choice: 1
Enter Employee ID: 10543
Enter Name: Deepak Rajput
Enter Designation: Cloud Expertt
Enter Salary: 350000
Employee added successfully.

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

All Employees:
ID: 11720, Name: Kamalpreet Singh, Designation: Data Scientist, Salary: $450000.0
ID: 10543, Name: Deepak Rajput, Designation: Cloud Expertt, Salary: $350000.0

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

Learning Outcomes:

- Understand autoboxing and unboxing for automatic type conversions.
- Learn object serialization and deserialization using file handling.
- Implement exception handling for robust applications.
- Manage employee records with file storage for data persistence.