LAB-MST

Student Name: Ishika Thakur
UID: 22BCS10765
Branch: BE/CSE
Section/Group: 22BCS_IOT-618/B
Semester: 6th
Date of Performance: 07/03/25
Subject Name: Project Based Learning in JAVA with Lab
Subject Code: 22CSH-359

1. **Aim:** "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".

2. **Objective:** The objective of Developing Java Programs Using Serialization, File Handling, and Efficient Data Processing and Management is to equip developers with the skills to write robust, scalable, and efficient Java applications.

3. **Implementation/Code:**

```
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;
    }

    public void display() {
        System.out.println("Student ID: " + id);
        System.out.println("Name: " + name);
        System.out.println("GPA: " + gpa);
```

```
    }
}

public class StudentSerialization {
    public static void main(String[] args) {
        String filePath = "C:/Users/Dell/OneDrive/Documents/student.dat";

        Student student = new Student(101, "Tommy ", 8.8);
        try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(filePath))) {
            oos.writeObject(student);
            System.out.println("Student object serialized successfully at: " + filePath);
        } catch (FileNotFoundException e) {
            System.err.println("Error: File not found - " + e.getMessage());
        } catch (IOException e) {
            System.err.println("Error during serialization - " + e.getMessage());
        }

        try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(filePath))) {
            Student deserializedStudent = (Student) ois.readObject();
            System.out.println("\nDeserialized Student:");
            deserializedStudent.display();
        } catch (FileNotFoundException e) {
            System.err.println("Error: File not found - " + e.getMessage());
        } catch (IOException e) {
            System.err.println("Error during deserialization - " + e.getMessage());
        } catch (ClassNotFoundException e) {
            System.err.println("Error: Class not found - " + e.getMessage());
        }
    }
}
```

Output:-

```
PS C:\Users\Dell\OneDrive\Desktop\coding\java> cd "c:\Users\Dell\OneDrive\Desktop\coding\
Student object serialized successfully at: C:/Users/Dell/OneDrive/Documents/student.dat

Deserialized Student:
Student ID: 101
Name: Tommy
GPA: 8.8
PS C:\Users\Dell\OneDrive\Desktop\coding\java>
```

4. Learning Outcomes:

- Understanding Serialization & Deserialization: Learners understand how Java automatically converts between objects and bytes.
- String to Integer Parsing: Demonstrates the use of Integer.parseInt() to convert string inputs into integer values.
- List Handling in Java: Shows how to store and manipulate integers using ArrayList<Integer>.
- Looping & Summation Logic: Reinforces iteration concepts using a for loop to sum up a list of numbers.
- User Input Handling: Teaches how to take space-separated user input and process it effectively using Scanner and split().