Experiment 5.2

Student Name: Akshat Srivastava
UID: 22BCS11740
Branch: BE CSE
Section/Group: 22BCS_IOT_618_A
Semester: 6th
DoP: 21/02/2025
Subject Name: PBLJ Lab
Subject Code: 22CSH-359

1. **Aim:** To implement a Java program that serializes and deserializes a `Student` object using `ObjectOutputStream` and `ObjectInputStream` while handling exceptions like `FileNotFoundException`, `IOException`, and `ClassNotFoundException`.

2. **Objective:**
   - Create a serializable `Student` class with id, name, and GPA.
   - Serialize the object to a file named `student.ser`.
   - Deserialize the object from the file and display its details.
   - Handle exceptions during serialization and deserialization.

3. **Implementation/Code:**

```
import java.io.*;

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 displayStudent() {
        System.out.println("Student ID: " + id + ", Name: " + name + ", GPA: " + gpa);
    }
}
public class StudentSerialization {
    public static void serializeStudent(Student student, String filename) {
        try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(filename)))
{
            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: Unable to serialize object.");
        }
    }
    public static Student deserializeStudent(String filename) {
        try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(filename))) {
            System.out.println("Student object has been deserialized.");
            return (Student) ois.readObject();
        } catch (FileNotFoundException e) {
            System.out.println("Error: File not found.");
        } catch (IOException e) {
            System.out.println("Error: Unable to deserialize object.");
        } catch (ClassNotFoundException e) {
```

```
            System.out.println("Error: Class not found.");
        }
        return null;
    }
    public static void main(String[] args) {
        String filename = "student.ser";
        Student student1 = new Student(1, "John Doe", 3.75);
        serializeStudent(student1, filename);
        Student deserializedStudent = deserializeStudent(filename);
        if (deserializedStudent != null) {
            System.out.println("Deserialized Student Details:");
            deserializedStudent.displayStudent();
        }
        deserializeStudent("nonexistent.ser");
    }
}
```

4. Output

```
Student object has been serialized and saved to file.
Student object has been deserialized.
Deserialized Student Details:
Student ID: 1, Name: John Doe, GPA: 3.75
Error: File not found.
PS D:\java lab>
```

5. Learning Outcome:

- Understand Java serialization using `Serializable` interface.
- Use `ObjectOutputStream` and `ObjectInputStream` for object I/O.
- Implement exception handling for file and class-related errors.
- Gain experience with file input/output operations in Java.