CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

UNIVERSITY INSTITUTE OF ENGINEERING

Project Based Learning in Java

Experiment 8

23CSP-304

Submitted To:
Faculty Name: Er. Deep Prakash

Submitted By: Name:
Aryan Srivastav
UID: 23BCS10135
Section: KRG - 2B
Semester: 5th

Experiment 8: Object Serialization and Deserialization in Java

Aim

To create a Java program that serializes and deserializes a **Student** object containing id, name, and GPA, using object streams and proper exception handling.

Objectives

- Implement **serialization** and **deserialization** using Java I/O streams.
- Understand how objects are converted into a byte stream and reconstructed.
- Handle exceptions like **FileNotFoundException**, **IOException**, and **ClassNotFoundException**.
- Demonstrate file operations using **ObjectOutputStream** and **ObjectInputStream**.
- Strengthen understanding of persistent object storage in Java.

Code Implementation

```
import java.io.*;

// Step 1: Define a Student class implementing Serializable
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 int getId() { return id; }
public String getName() { return name; }
public double getGpa() { return gpa; }

@Override
public String toString() {
return "ID: " + id + "\nName: " + name + "\nGPA: " + gpa;
}
}

public class SerializationDemo {
public static void main(String[] args) {

String fileName = "student.ser";
// Step 2: Create a Student object
Student student = new Student(101, "Alice", 9.1);
```

```
// Step 3: Serialize the object
try (ObjectOutputStream oos = new ObjectOutputStream(new
FileOutputStream(fileName))) {
oos.writeObject(student);
System.out.println("Student serialized successfully!");
} catch (FileNotFoundException e) {
System.out.println("Error: File not found!");
} catch (IOException e) {
System.out.println("Error: Unable to serialize object!");
e.printStackTrace();
}

// Step 4: Deserialize the object
try (ObjectInputStream ois = new ObjectInputStream(new
FileInputStream(fileName))) {
Student deserializedStudent = (Student) ois.readObject();
System.out.println("\nStudent deserialized:");
System.out.println(deserializedStudent);
} catch (FileNotFoundException e) {
System.out.println("Error: File not found!");
} catch (IOException e) {
System.out.println("Error: Unable to deserialize object!");
e.printStackTrace();
} catch (ClassNotFoundException e) {
System.out.println("Error: Class not found!");
}}}
```

Output

```
Student serialized successfully!

Student deserialized:
ID: 101
Name: Alice
GPA: 9.1
```

Learning Outcomes

- Learned how to **serialize and deserialize** Java objects.
- Understood how to use **Object streams** (ObjectOutputStream and ObjectInputStream).
- Implemented **exception handling** for file operations.
- Gained insight into **object persistence and data recovery** in Java.
- Practiced writing robust Java programs that interact with the filesystem.