Serialization in Java

What is Serialization?

Serialization is the process of converting an object into a byte stream so that it can be:

1. Saved to a file or database (persistence)

2. Sent over a network (communication)

3. Passed between JVMs or application layers

Why Use Serialization?

---

1. Persistence - Save and load objects.

2. Communication - Transmit objects between systems.

3. Deep Cloning - Copy complete object graphs.

4. Caching - Store objects for reuse.

5. Framework Integration - Used in Hibernate, Spring, RMI, etc.

Serializable Class Example:

```
import java.io.Serializable;

public class Student implements Serializable {
    private static final long serialVersionUID = 1L;
    String name;
    int age;
    transient String password; // will not be serialized
```

}

Serialization Code:

```
import java.io.FileOutputStream;

import java.io.ObjectOutputStream;

public class SerializeExample {

    public static void main(String[] args) {

        Student student = new Student("Alice", 20, "secret123");

        try (FileOutputStream fileOut = new FileOutputStream("student.ser");

             ObjectOutputStream out = new ObjectOutputStream(fileOut)) {

            out.writeObject(student);

            System.out.println("Serialized data is saved in student.ser");

        } catch (Exception e) {

            e.printStackTrace();

        }

    }

}
```

Deserialization Code:

```
import java.io.FileInputStream;

import java.io.ObjectInputStream;


public class DeserializeExample {

    public static void main(String[] args) {
```

```
Student student = null;
try (FileInputStream fileIn = new FileInputStream("student.ser");
     ObjectInputStream in = new ObjectInputStream(fileIn)) {
    student = (Student) in.readObject();
    System.out.println("Deserialized Student:");
    System.out.println("Name: " + student.name);
    System.out.println("Age: " + student.age);
        System.out.println("Password: " + student.password); // null due to transient
    } catch (Exception e) {
        e.printStackTrace();
    }
  }
}
```