How to read/ write objects to stream (at any end point like files)?

* The process is known as serialization.

**Serialization:**

It is the process to save the state of an object such that it can be written to a stream.

**De-serialization:**

The reverse process (i.e.; to retrieve the state of same object) is known as de-serialization.

A few authors also used the name of Marshalling and Un-Marshalling for the set purpose.

* **State of Object?**

- What is the state of an object?

- When we create objects, they are created on memory, heap. A memory-portion is made.

- Object? Instance of class. Block of memory.

**- Object is combination of data-members and functions.** Values of that data is present on heap, memory.

- Every object has its own values.

- The values that data members acquire are called state of object.

**- State of object means** values of data members of an object in memory.

* Serialization in Java can be **managed using following 2 methods:**

1. **Automatic (Using serializable interface).**
2. **Manual (Using externalizable interface).**

* Serialization -> Saving state of any object.
* Object State -> Value of data members in memory.
* Where do we need to save state? -> On stream (it can be a file or other terminal).
* Serialization -> means to write on stream.
* De-serialization -> Reverse process.
* Marshalling -> Serialization.
* Un-Marshalling -> De-serialization.
* Automatic serialization us done by Java itself.
* In manual serialization, we have to save the state.
* **How to save the state?**
* The class whose objects we want to save/ serialize, that class **must implement Serializable** interface.

**Class PersonInfo implements Serializable**

* PersonInfo signed a contract with Serializable interface.
* **Serializable interface:**
* It is just information.
* It is a tagging interface.
* **It has no function.** So there is no responsibility of PersonInfo class implementing it.
* Then, **what is the purpose of this interface?**
* It is a message, notification.
* The object of PersonInfo class is going to be serialized automatically.
* PersonInfo is a serializable. (Every object of PersonInfo class will be serialized by Java).
* We just need to write that object on stream

**new ObjectOutputStream(new FileOutputStream(‘test.txt’);**

* This OOS has a function with name of

**writeObject(\_\_);**

It will write your data object by object.

All the objects you want to write on file, will be passed to this function

Like PersonInfo -> p, s

**writeObject(p);**

**writeObject(s);**

**What type of parameter does it accept? -> Object type.**

* To read data:

**new ObjectInputStream(new FileInputStream(‘test.txt’);**

* This class provides a function with name of:

**readObject();**

**Return type of this function is Object** (because file can give different types of Objects).

To convert to appropriate Object form you can downcast.

To identify type-> use instance of

At first call, it will return first Object, then other object at 2nd call.

Until eof (end of file occurse), it should read Objects….. while loop

* **Can we partially serialize objects?**

Sometimes, there are objects whose data-members are critical (you don’t want to transfer them). For example, head-quarters want to send an info but they don’t want to send the pin-code. Head-Quarters want that customer name, address, debitcard no, pin (whole object) should go, values of all attributes get transferred but pincode value should not go where the object is being transferred to client/ branch. Pincode is secret.

* All data members that you don’t want to serialize must be declared as **transient.**
* **Transient data members:** Their values/ data will not be transferred/ saved in state.
* **Serialization process also takes care of recursive saving….**

Assume PersonInfo class is containing object of any other class like Object of City type or Object of Address type.

Those objects which PersonInfo is containing as data-members will be saved and if those objects are containing further classes as data members, they will also be saved.

If a class contains multiple objects, they will also be saved and retrievd.