**JAC444 - Lecture 6**

Java Input / Output

Segment 3 - Serialization

Input Stream / Output Stream

**In this segment you will be learning about:**

* Serializable Objects
* FileInputStream / FileOutputStream
* ObjectInputStream / ObjectOutputStream

# Object Serialization

**Reading and writing objects** is a process called ***object serialization***.

1. One you need to know how to serialize objects by writing them to an **ObjectOutputStream** and reading them in again using an **ObjectInputStream**.
2. An object can be serialized only if its class implements the **Serializable** interface.

**FileOutputStream out = new FileOutputStream("fileName"); ObjectOutputStream stream = new ObjectOutputStream(out); stream.writeObject("Today"); stream.writeObject(new Date()); stream.flush();**

# ObjectInputStream

**FileInputStream in = new FileInputStream("fileName");**

**ObjectInputStream stream = new ObjectInputStream(in);**

**String today = (String)stream.readObject();**

**Date date = (Date)stream.readObject();**

**ObjectInputStream** stream implements the **DataInput** interface that defines methods for reading primitive data types.

**readObject** method deserializes the next object in the stream

# Customizing Serialization

* An object is serializable only if its class implements the **Serializable** interface.
* **Serializable** is an empty interface, it doesn't contain any method declarations. It is what is called a marker interface.
* The serialization of instances of this class are handled by the **defaultWriteObject** method of **ObjectOutputStream**.
* This method automatically writes out everything required to reconstruct an instance of the class, including the following:
* Class of the object.
* Class signature. • Values of all non-transient and and non-static members, including members that refer to other objects.
* One can customize serialization for his/her classes by providing two methods for it: **writeObject**and **readObject***.*

# Externalizable Interface

For complete, explicit control of the serialization process, a class must implement the **Externalizable** interface

**package java.io;**

**public interface Externalizable extends Serializable { public void writeExternal(ObjectOutput out) throws IOException; public void readExternal(ObjectInput in) throws IOException, java.lang.ClassNotFoundException;**

**}**

Particularly sensitive classes should not be serialized. To accomplish this, the class should not implement either the **Serializable** or **Externalizable** interface

# Conclusion

**After completion of this segment you should know:**

* How to serialize Java objects.
* How to deserialize and construct Java objects
* How to read/write objects using serialization.

