**IO Operations**

1. In Java, input and output streams are used for reading from and writing to various sources and destinations, such as files, network connections, or in-memory buffers. Input streams are used for reading data, while output streams are used for writing data.

2. Some methods of the OutputStream class in Java include write(byte[] b), write(int b), flush(), and close().

3. Serialization in Java refers to the process of converting an object into a stream of bytes, which can then be saved to a file or transmitted over a network. This allows objects to be saved, sent, and reconstructed later.

4. The Serializable interface in Java is a marker interface that indicates that the objects of a class can be serialized. Classes that implement this interface can be written to streams and reconstructed later.

5. Deserialization in Java is the process of reconstructing an object from its serialized form, i.e., converting a stream of bytes back into an object.

6. Serialization in Java is achieved by implementing the Serializable interface in the class whose objects need to be serialized. This interface doesn't have any methods; it simply serves as a marker to indicate that the class is serializable.

7. Deserialization in Java is achieved by reading the serialized bytes from a stream and converting them back into an object using the ObjectInputStream class.

8. To avoid certain member variables of a class from getting serialized in Java, you can declare them as transient. Transient variables are not serialized.

9. Some classes available in the Java IO File Classes API include File, FileInputStream, FileOutputStream, FileReader, FileWriter, BufferedReader, BufferedWriter, and RandomAccessFile.

10. The main difference between Externalizable and Serializable interface in Java is that Serializable interface provides default serialization mechanism whereas Externalizable interface provides a custom serialization mechanism where the class must implement its own serialization and deserialization methods: writeExternal() and readExternal(). Additionally, Serializable interface serializes all non-transient fields by default, while Externalizable allows more control over the serialization process.