

1.What is Input and Output stream in java?

Ans: A stream can be defined as a sequence of data. There are two kinds of Streams
– InPutStream – The InputStream is used to read data from a source. OutPutStream – The OutputStream is used for writing data to a destination.

2. What are the methods of OutputStream?

Ans: Methods are:

- Write() : writes the specified byte to output stream.
- write(byte[] array) : writes the specified array to output stream.
- Flush() : forces to write all data present in the output stream to the destination.
- Close() : close the output stream.

3. What is serialization in java?

Ans: Serialization is a mechanism of converting the state of an object into a byte stream.

4. What is the serializable interface in java?

Ans: serializable interface in java is a marker interface that has no methods. It is used to mark classes but can be serialized, it means the object instances can be converted into a stream of bytes.

5. What is deserialization in java?

Ans: Deserialization is the reverse process where the byte stream is used to recreate the actual Java object in memory.

6. How to serialization achieve in java?

Ans: To make a Java object serializable we implement the java.io.Serializable interface. The ObjectOutputStream class contains writeObject() method for serializing an Object. The ObjectInputStream class contains readObject() method for deserializing an object.

7. How to deserialization achieve in java?

Ans: or deserialization we call the readObject() method of ObjectInputStream class.

8. How can you avoid member variables of class from getting serialized?

Ans: Mark member variables as a static and those member variables will no more be part of serialization.

9. What classes are available in Java IO File classed API?

Ans: Classes are:

- RandomAccessFile
- FileInputStream
- FileReader
- FileOutputStream
- FileWriter

10. Difference between Externalizable and Serialization interface.

Ans:

<i>Serializable</i>	<i>Externalizable</i>
Serializable is a marker interface i.e., does not contain any method.	Externalizable interface methods writeExternal() and readExternal() v override.
Serializable interface passes the responsibility of serialization to JVM and its default algorithm.	Externalizable provides control of serialization custom logic.
Mostly, default serialization is easy to implement, but has a higher performance cost.	Serialization done using Externalizable, add more control but often results in better performance.
It's hard to analyse and modify class structure because any change may break the serialization.	It's easier to analyse and modify class structure because serialization logic is under control.
Default serialization does not call any class constructor.	A public no-arg constructor is required which is called by the JVM.

