

Serialization over deserialization as its not mandatory to define readObject() and writeObject() methods.

You've great control over the serialization process as it is important to override writeExternal() and readExternal() method.

Constructor is called during deserialization.

Constructor is not called during serialization. Constructor is called during deserialization.



Ans:- File  
Random Access File  
File Input Stream  
File Reader  
File Output Stream  
File Writer

(10) What is difference between Externalizable and Serializable Interface?

Ans:-

	Serializable	Externalizable
Methods	It is a marker interface and it doesn't have any method.	It is not a marker interface. It has method called <code>writeExternal()</code> and <code>readExternal()</code> method.
Default Serialization process	Yes, serialization provides its own default serialization process, we just need to implement a <code>Serializable</code> interface.	No, we need to override <code>writeExternal()</code> and <code>readExternal()</code> for the serialization process to happen.

Custom Serialization process: We can customize default serialization process by defining following methods in our class: `readObject()` and `writeObject()`.

- Serialization process is completely customized. We need to override the `Externalizable` interface's `writeExternal()` and `readExternal()` method.

Control over

It provide less control

Externalizable provides.



## IO operation

Page No. 2/5  
Date      /      /     

① what is input and output stream in Java?

Ans:- A Stream can be defined as a sequence of data. The InputStream is used to read data from a source and the OutputStream is used for writing data to a destination.

② what are the methods of OutputStream?

Ans:-

- write() - writes the specified byte to the output stream.
- write(byte[] array) - writes the bytes from the specified array to the output stream.
- flush() - forces to write all data present in the output stream to the destination.
- close() - closes the output stream.

③ what is serialization in Java?

Ans:- Serialization is the process of converting an object into a stream of bytes to transfer it over a network or to store it in a file or database. In Java, serialization is done by implementing the Serializable interface.

④ what is serializable interface in Java?

Ans:- The Serializable interface in Java is a marker interface that has no methods. It is used to mark classes that can be serialized, meaning their ~~product~~ object instance can be converted into a stream of bytes.



⑤ What is deserialization in Java?  
Ans: Deserialization is the process of converting a stream of bytes back into an object instance. This is done after an object has been serialized.

⑥ How is serialization achieved in Java?  
Ans: Serialization achieved in Java by implementing the Serializable interface. When an object is serialized, its state is converted into a stream of byte, which can then be transferred over a network or stored in a file or database.

⑦ How is deserialization achieved in Java?  
Ans: Deserialization achieved in Java by reading a stream of bytes and using them to recreate the original object instance. This is done by calling the readObject() method of an ObjectInputStream instance.

⑧ How can you avoid certain member variables of class from getting serialized?  
Ans: Mark member variables as static and transient, and those member variable will no more be a part of serialization.

⑨ What classes are available in the Java IO file classes API?