**Points to remember:**

1.Serialization in Java

Serialization in java is a mechanism of writing the state of an object into a byte stream.

ObjectOutputStream class

public ObjectOutputStream(OutputStream out) throws IOException {}creates an ObjectOutputStream that writes to the specified OutputStream.

Refer example

2.Deserialization in java

Deserialization is the process of reconstructing the object from the serialized state.It is the reverse operation of serialization.

public ObjectInputStream(InputStream in) throws IOException {}

Refer exampe.

3.If a class implements serializable then all its sub classes will also be serializable

4.If a class has a reference of another class, all the references must be Serializable otherwise serialization process will not be performed. In such case, NotSerializableException is thrown at runtime.

5.If there is any static data member in a class, it will not be serialized because static is the part of class not object.

6.In case of array or collection, all the objects of array or collection must be serializable. If any object is not serialiizable, serialization will be failed.

3.Externalizable in java

The Externalizable interface provides the facility of writing the state of an object into a byte stream in compress format. It is not a marker interface.

The Externalizable interface provides two methods:

public void writeExternal(ObjectOutput out) throws IOException

public void readExternal(ObjectInput in) throws IOException

4.Java Transient Keyword

If you don't want to serialize any data member of a class, you can mark it as transient.

Questions:

1.What is the difference between Serializable and Externalizable interface in Java?

2.How many methods Serializable has? If no method then what is the purpose of Serializable interface?

3.What is serialVersionUID? What would happen if you don't define this?

4.While serializing you want some of the members not to serialize? How do you achieve it?

5.What will happen if one of the members in the class doesn't implement Serializable interface?

6.If a class is Serializable but its super class in not, what will be the state of the instance variables inherited from super class after deserialization?

7.Can you Customize Serialization process or can you override default Serialization process in Java?

8.Suppose super class of a new class implement Serializable interface, how can you avoid new class to being serialized?

9.Which methods are used during Serialization and DeSerialization process in Java?

10.Suppose you have a class which you serialized it and stored in persistence and later modified that class to add a new field. What will happen if you deserialize the object already serialized?

11.What are the compatible changes and incompatible changes in Java Serialization Mechanism?

12.Can we transfer a Serialized object vie network

13.Which kind of variables is not serialized during Java Serialization