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 serialisation 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, serialisation is done by implementing the Serializable interface.

4) What is the 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 serialised, meaning their object instances 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 into an object instance. This is done after an object has been serailized .

6) How is serialisation achieved in Java?

Ans: Serialization is achieved in Java by implementing the Serializable interface. When an object is Serialised,it's State is converting into a stream of bytes, which can be transferred over a network or stored in a file or database.

7) How is deserialization achieved in Java?

Ans: Deserialization is 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 or transient, and those member variables will no more be a part of Serialization.

9) What classes are available in the Java IO File Classes API?

Ans: The following classes are available in the Java IO API and are important to work with files in Java.

- File
- RandomAccessFile
- FileInputStream

- FileReader
- FileOutputStream
- FileWriter

10)What is the difference between Extenalizable and Serialization interface?
Ans:

	Serializable	**Externalizable**
Methods	It is a marker interface and it doesn't have any method.	It's not a marker interface. It has methods called writeExternal() and readExternal()
Default Serialization Process	YES, Serializable provides it's own default serialisation process, we just need to implement a Serializable interface.	NO, we need to override writeExternal() and readExternal() for the serialisation process to happen
Customise Serialisation process	We can customise default serialisation process by defining following methods in our class> readObject() and writeObject()	Serialization Process is completely customised We need to override the Externalizable interface's writeExternal() and readExternal() methods.
Control over	It provides less control	Externalizable provides.
Serialization	Over Serialization as it's mandatory to define readObject() and writeObject() methods.	You have great control over the serailization process as it is important to override withExternal() and readExternal() methods.
Constructor call during deserialization	Constructor is not called during deserialization.	Constructor is called during deserialization