1. **What is Input and Output Stream in Java?**
   - **Input Stream**: Used to read data from a source (e.g., file, network).
   - **Output Stream**: Used to write data to a destination (e.g., file, network).
2. **What are the methods of OutputStream?**
   - `write(int b)`: Writes the specified byte to this output stream.
   - `write(byte[] b)`: Writes `b.length` bytes from the specified byte array to this output stream.
   - `write(byte[] b, int off, int len)`: Writes `len` bytes from the specified byte array starting at offset `off` to this output stream.
   - `flush()`: Flushes this output stream and forces any buffered output bytes to be written out.
   - `close()`: Closes this output stream and releases any system resources associated with this stream.
3. **What is serialization in Java?**
   - Serialization is the process of converting an object into a byte stream, so it can be easily saved to a file or transmitted over a network.
4. **What is the Serializable interface in Java?**
   - The `Serializable` interface is a marker interface (contains no methods) that enables the serialization of an object by the Java serialization mechanism.
5. **What is deserialization in Java?**
   - Deserialization is the process of converting a byte stream back into a copy of the original object.
6. **How is serialization achieved in Java?**
   - By implementing the `Serializable` interface and using `ObjectOutputStream` to write the object to a stream.

7. **How is deserialization achieved in Java?**

   - By using `ObjectInputStream` to read the object from a stream.

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

   - By marking them as `transient`.

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

   - Some of the classes include `File`, `FileReader`, `FileWriter`, `BufferedReader`, `BufferedWriter`, `FileInputStream`, `FileOutputStream`, etc.

10. **What is the difference between Externalizable and Serializable interface?**

    - `Serializable` is a marker interface, while `Externalizable` extends `Serializable` and requires the implementation of `writeExternal` and `readExternal` methods for custom serialization.