**IO operations Assignments by gaurav diwan**

**Java with DSA and System design**

Q1] what is input and output stream in java ?

Ans. Input Stream: Reads data from a source (e.g., file, network).

Output Stream: Writes data to a destination (e.g., file, network).

Q2] what are the methods of outputstream ?

Ans. Common methods of `OutputStream` in Java include:

1. `write(int b)`: Writes the specified byte to the output stream.

2. `write(byte[] b)`: Writes an array of bytes to the output stream.

3. `close()`: Closes the output stream.

4. `flush()`: Flushes the output stream, forcing any buffered output bytes to be written.

Q3] what is serialization in java?

Ans. Serialization in Java is the process of converting an object's state into a byte stream. This allows the object to be easily saved to a file, sent over a network, or stored in a database.

Q4] what is the serializable interface in java?

Ans. The `Serializable` interface in Java is a marker interface that indicates a class can be serialized. Classes that implement this interface can be converted into a byte stream for various purposes like storage or network transmission.

Q5] what is deserialization in java?

Ans. Deserialization in Java is the process of reconstructing an object from a byte stream. It is the reverse of serialization and is commonly used to restore the state of an object that was previously serialized.

Q6] how is the serializable achieved in java?

Ans. To achieve serialization in Java:

1. Implement the **Serializable** interface in the class.
2. No methods need to be provided; it serves as a marker interface.

Q7] how is the deserializable achieved in java?

Ans. To achieve deserialization in Java:

1. Implement the **Serializable** interface in the class.
2. Use **ObjectInputStream** to read the serialized object from a file or another source.

Q8] how can you avoid certain member variables of class from getting serialized?

Ans. To avoid certain member variables from getting serialized in Java:

1. Declare the variables as **transient**.
2. The transient keyword instructs the serialization process to skip the specified variables.

import java.io.Serializable;

public class MyClass implements Serializable {

private int serializableField;

private transient int nonSerializableField;

// other class members

}

Q9] what classes are available in the java IO file classes API?

Ans. Some classes in the Java I/O file classes API include:

1. File: Represents a file or directory path.

2. FileInputStream: Reads bytes from a file.

3. FileOutputStream: Writes bytes to a file.

4. BufferedReader: Reads text from a character-input stream, buffering characters.

5. BufferedWriter: Writes text to a character-output stream, buffering characters.

Q10] what is the difference between externalizable and serializable interface ?

Ans. Serializable Interface:

- Provides default serialization.

- All fields are serialized by default.

Externalizable Interface:

- Allows custom serialization logic.

- Developers explicitly define how the object is serialized and deserialized.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***