**16. Java Input/Output (I/O)**

1. **Streams in Java (InputStream, OutputStream) Answer:**

Streams in Java are used to handle input and output (I/O) operations. Java provides two primary categories of streams for reading from and writing to different data sources (e.g., files, network connections, memory, etc.):

* 1. **InputStream**: Used for reading data.
  2. **OutputStream**: Used for writing data.

These are part of the java.io package, and they provide an abstraction for reading and writing bytes, allowing seamless interaction with various data sources and destinations.

1. **Reading and Writing Data Using Streams Answer:**

In Java, **streams** are used for reading from and writing to various data sources, including files, memory, or network resources. The InputStream and OutputStream classes (and their subclasses) handle byte-oriented data, while character-based streams (Reader and Writer classes) handle character-oriented data. In this section, we’ll focus on how to read and write data using bytebased streams (InputStream and OutputStream) and explain some of the commonly used classes for this purpose.

1. **Handling File I/O Operations Answer:**

In Java, handling file input and output (I/O) operations is primarily done using the java.io package, which provides classes for reading from and writing to files. Java's file I/O capabilities are divided into byte-oriented and character-oriented streams. Byte-oriented streams (such as FileInputStream and FileOutputStream) are used for binary data, while character-oriented streams (such as FileReader and FileWriter) are used for text data.

=====================================================================

=====================================================================

=====================================================================

==========================

=============

==============================