# **Java I/O:**

**What is a stream and what are the types of Streams and classes of the Streams?**

A stream is a sequence of data. In Java, a stream is composed of bytes. It's called a stream because it is like a stream of water that continues to flow. There are two types of Streams :

Byte Streams: Provide a convenient means for handling input and output of bytes.

Character Streams: Provide a convenient means for handling input & output of characters.

Byte Streams classes: Are defined by using two abstract classes, namely InputStream and OutputStream.

Character Streams classes: Are defined by using two abstract classes, namely Reader and Writer.

**What is a IO stream?**

It is a stream of data that flows from source to destination. Good example is file copying. Two streams are involved – input stream and output stream. An input stream reads from the file and stores the data in the process (generally in a temporary variable). The output stream reads from the process and writes to the destination file.

**What is the necessity of two types of streams – byte streams and character streams?**

Byte streams were introduced with JDK 1.0 and operate on the files containing ASCII characters. We know Java supports other language characters also known as Unicode characters. To read the files containing Unicode characters, the designers introduced character streams with JDK 1.1. As ASCII is a subset of Unicode, for the files of English characters, we can go with either byte streams or character streams.

**What are the super most classes of all streams?**

All the byte stream classes can be divided into two categories (input stream classes and output stream classes) and all character streams classes into two (reader classes and writer classes). There are four abstract classes from which all these streams are derived. The super most class of all byte stream classes is java.io.InputStream and for all output stream classes, java.io.OutputStream. Similarly for all reader classes is java.io.Reader and for all writer classes is java.io.Writer.

**What is the difference between the Reader/Writer class hierarchy and the InputStream/OutputStream class hierarchy?**

The Reader/Writer class hierarchy is character-oriented, and the InputStream/OutputStream class hierarchy is byte-oriented

**What value does read() return when it has reached the end of a file?**

The read() method returns -1 when it has reached the end of a file.

**What is the package name for ObjectInputStream class?**

java.io

**Which is the Parent Class of StringBufferInputStream class?**

InputStream

Note: Anything that ends with InputStream are subclasses of InputStream.

Anything that ends with OutputStream are subclasses of OutputStream.

Anything that ends with Reader are subclasses of Reader.

Anything that ends with Writer are subclasses of Writer.

**What are FileInputStream and FileOutputStream?**

These two are general purpose classes used by the programmer very often to copy file to file. These classes work well with files containing less data of a few thousand bytes as by performance these are very poor. For larger data, it is preferred to use BufferedInputStream (or BufferedReader) and BufferedOutputStream (or BufferedWriter).

**Which you feel better to use – byte streams or character streams?**

I feel personally to go with character streams as they are the latest. Many features exist in character streams that do not in byte streams like a) using BufferedReader in place of BufferedInputStreams and DataInputStream (one stream for two) and b) using newLine() method to go for next line and for this effect we must go for extra coding in byte streams etc.

**What System.out.println()?**

"println()" is a method of PrintStream class. "out" is a static object of PrintStream class defined in "System" class. System is a class from java.lang package used to interact with the underlying operating system by the programmer.

**What are filter streams?**

Filter streams are a category of IO streams whose responsibility is to add extra functionality (advantage) to the existing streams like giving line numbers in the destination file that do not exist int the source file or increasing performance of copying etc.

**Name the filter streams available?**

There are four filter streams in java.io package – two in byte streams side and two in character streams side. They are FilterInputStream, FilterOutputStream, FilterReader and FilterWriter. These classes are abstract classes and you cannot create of objects of these classes.

**Name the filter stream classes on reading side of byte stream?**

There are four classes – LineNumberInputStream (the extra functionality is it adds line numbers in the destination file), DataInputStream (contains special methods like readInt(), readDouble() and readLine() etc that can read an int, a double and a string at a time), BufferedInputStream (gives buffering effect that increases the performance to the peak) and PushbackInputStream (pushes the required character back to the system).

**What is the functionality of SequenceInputStream?**

It is very useful to copy multiple source files into one destination file with very less code.

**What is PrintStream and PrintWriter?**

Functionally both are same but belong to two different categories – byte streams and character streams. println() method exists in both classes.

**Which streams are advised to use to have maximum performance in file copying?**

BufferedInputStream and BufferedOutputStream on byte streams side and BufferedReader and BufferedWriter on character streams side.

**What are piped streams?**

There are four piped streams – PipedInputStream, PipedOutputStream, PipedReader and PipedWriter. These streams are very useful to pass data between two running threads (say, processes).

**What is File class?**

It is a non-stream (not used for file operations) class used to know the properties of a file like when it was created (or modified), has read and write permissions, size etc.