IO Streams Overview Java

Data can be stored on a computer system, as per the code requirements, in two ways, either **permanently**or **temporarily**

Temporary storage can be accomplished by storing the data in datastructures or instance variables.

The data is temporary because it is stored in RAM. For a permanent storage, the data should be stored on the hard disk either in the form of database tables or files.

There comes two entities – a **source** and a **destination**.

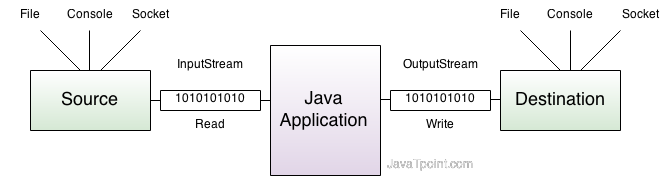
Source is that from where data is read and the destination is that one to where data is written.

The source and destination need not be a file only; it can be a socket or keyboard input etc.

To do the job of reading and writing, there comes two types of streams – **input streams** and **output streams**. An input stream job is to read from the source and the output stream job is to write to the destination.

**I/O streams are carriers of data from one place to another**. The input stream carries data from the source and places it temporarily in a variable (like int k or String str etc.) in the process (program). The output stream takes the data from the variable and writes to the destination. The variable works like a temporary buffer between input stream and output stream.

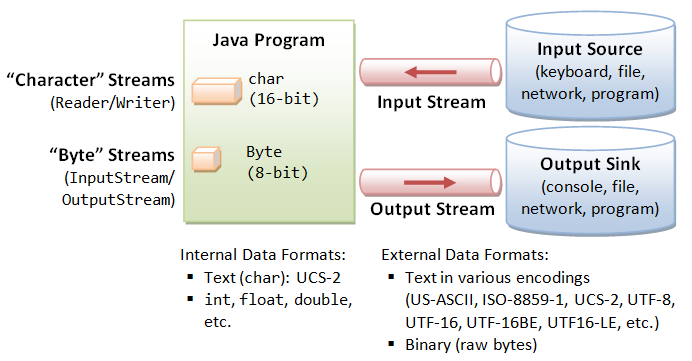
Let's understand working of Java OutputStream and InputStream by the figure given below.



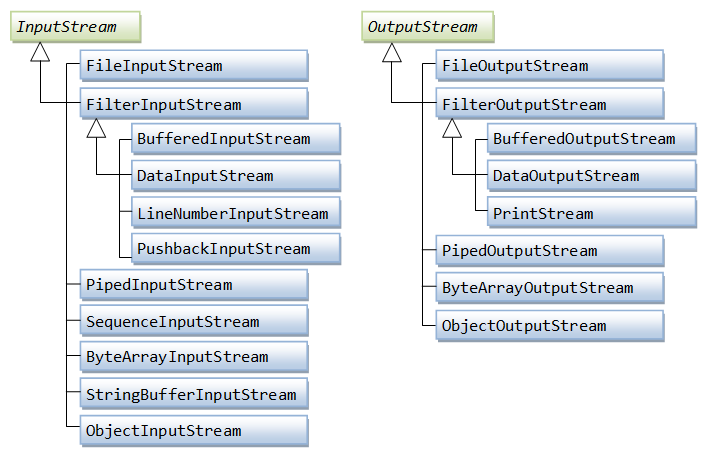
###### Byte streams Character streams

File copying is done, in JDK 1.0, with [**byte streams**](http://way2java.com/io/semantics-of-file-copying/). **Byte streams can read or write the files containing ASCII characters**that range from 0 to 255. That is, byte streams can copy the files containing English letters only but not of other languages.

If the file contains other than English characters, as Java supports **Unicode** characters, byte streams fail to do the job. To overcome this, in JDK 1.1 version, designers introduced **character streams**. **Character streams operate on Unicode characters**. That is, character streams can read, write and copy the files containing other than English characters. Characters streams can do with English characters also as [ASCII code is a subset of Unicode](http://way2java.com/java-introduction/primitive-data-types/).



### .  Byte-Based I/O & Byte Streams



### **Character-Based I/O & Character Streams**

