Java I/O Tutorial

**Java I/O** (Input and Output) is used *to process the input* and *produce the output*.

Java uses the concept of stream to make I/O operation fast. The java.io package contains all the classes required for input and output operations.

We can perform **file handling in java** by Java I/O API.

## Stream

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.

In java, 3 streams are created for us automatically. All these streams are attached with console.

**1) System.out:**standard output stream

**2) System.in:**standard input stream

**3) System.err:**standard error stream

Let's see the code to print **output and error** message to the console.

1. System.out.println("simple message");
2. System.err.println("error message");

Let's see the code to get **input** from console.

1. **int** i=System.in.read();//returns ASCII code of 1st character
2. System.out.println((**char**)i);//will print the character

## OutputStream vs InputStream

The explanation of OutputStream and InputStream classes are given below:

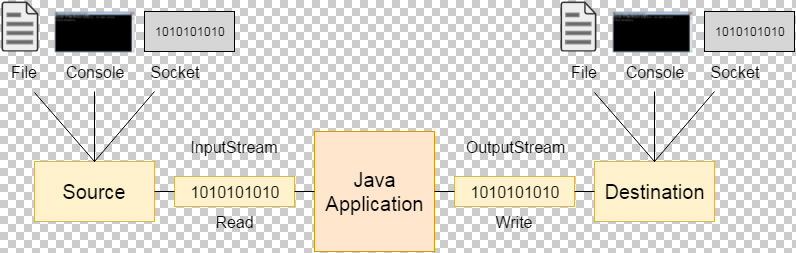
### OutputStream

Java application uses an output stream to write data to a destination, it may be a file, an array, peripheral device or socket.

### InputStream

Java application uses an input stream to read data from a source, it may be a file, an array, peripheral device or socket.

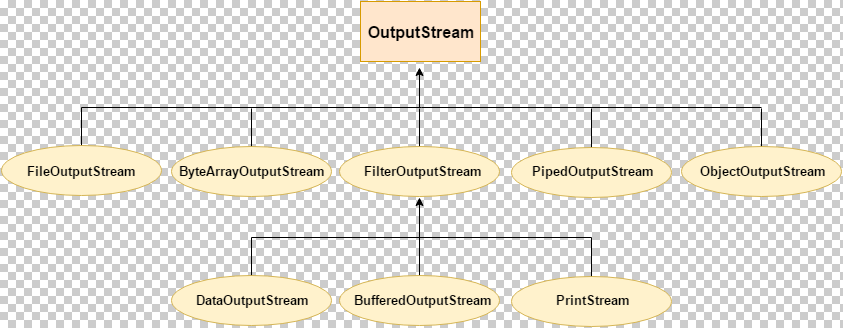
Let's understand working of Java OutputStream and InputStream by the figure given below.



## OutputStream class

OutputStream class is an abstract class. It is the super class of all classes representing an output stream of bytes. An output stream accepts output bytes and sends them to some sink.

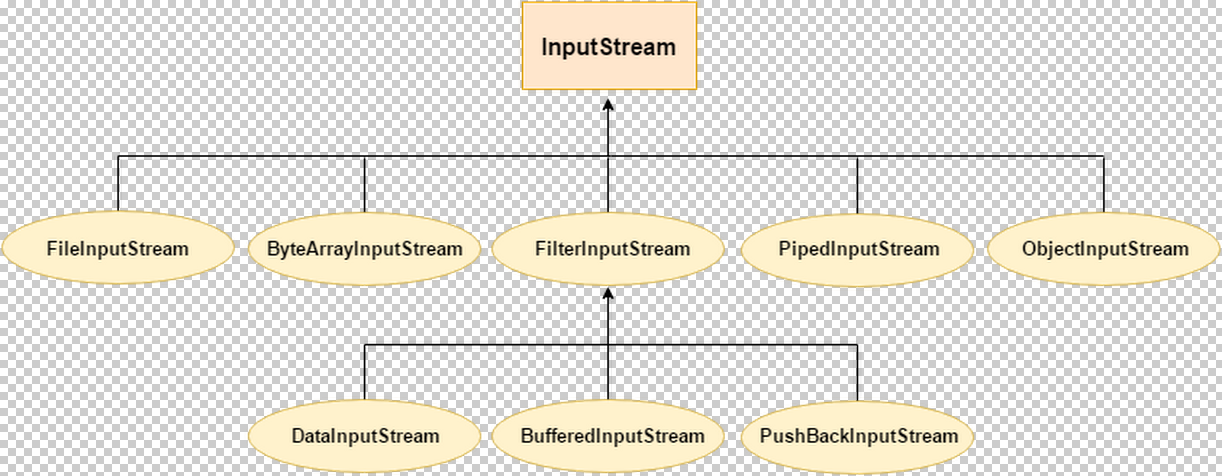
### OutputStream Hierarchy



## InputStream class

InputStream class is an abstract class. It is the super class of all classes representing an input stream of bytes.

### InputStream Hierarchy



# Java FileOutputStream Class

Java FileOutputStream is an output stream used for writing data to a file.

If you have to write primitive values into a file, use FileOutputStream class. You can write byte-oriented as well as character-oriented data through FileOutputStream class. But, for character-oriented data, it is preferred to use FileWriter than FileOutputStream.

## FileOutputStream class declaration

Let's see the declaration for Java.io.FileOutputStream class:

**public** **class** FileOutputStream **extends** OutputStream

# Java FileInputStream Class

Java FileInputStream class obtains input bytes from a file. It is used for reading byte-oriented data (streams of raw bytes) such as image data, audio, video etc. You can also read character-stream data. But, for reading streams of characters, it is recommended to use FileReader class.

## Java FileInputStream class declaration

Let's see the declaration for java.io.FileInputStream class:

**public** **class** FileInputStream **extends** InputStream