**Java.io Package in Java**

**Java.io Package in Java**

This package provides for system input and output through data streams, serialization and the file system. Unless otherwise noted, passing a null argument to a constructor or method in any class or interface in this package will cause a NullPointerException to be thrown.  
**Following are the important classes in Java.io package:**

* [BufferedInputStream](https://www.geeksforgeeks.org/java-io-bufferedinputstream-class-java/)
* [BufferedOutputStream](https://www.geeksforgeeks.org/java-io-bufferedoutputstream-class-java/)
* [BufferedReader](https://www.geeksforgeeks.org/java-io-bufferedreader-class-java/)
* [BufferedWriter](https://www.geeksforgeeks.org/io-bufferedwriter-class-methods-java/)
* [ByteArrayInputStream](https://www.geeksforgeeks.org/io-bytearrayinputstream-class-java/)
* [ByteArrayOutputStream](https://www.geeksforgeeks.org/io-bytearrayoutputstream-class-java/)
* [CharArrayReader](https://www.geeksforgeeks.org/java-io-chararrayreader-class-java/)
* CharArrayWriter –[Set1](https://www.geeksforgeeks.org/java-io-chararraywriter-class-java-set-1/) [Set2](https://www.geeksforgeeks.org/java-io-chararraywriter-class-java-set-2/)
* [Console](https://www.geeksforgeeks.org/java-io-console-class-java/)
* DataInputStream – [Set1](https://www.geeksforgeeks.org/java-io-datainputstream-class-java-set-1/) [Set2](https://www.geeksforgeeks.org/java-io-datainputstream-class-java-set-2/)
* [DataOutputStream](https://www.geeksforgeeks.org/dataoutputstream-in-java/)
* [File](https://www.geeksforgeeks.org/file-class-in-java/)
* [FileDescriptor](https://www.geeksforgeeks.org/java-io-filedescriptor-java/)
* [FileInputStream](https://www.geeksforgeeks.org/java-io-fileinputstream-class-java/)
* [FileOutputStream](https://www.geeksforgeeks.org/creating-a-file-using-fileoutputstream/)
* [FilePermission](https://www.geeksforgeeks.org/java-io-filepermission-class-java/)
* [FileReader and FileWriter](https://www.geeksforgeeks.org/file-handling-java-using-filewriter-filereader/)
* [FilterInputStream](https://www.geeksforgeeks.org/java-io-fileinputstream-class-java/)
* [FilterOutputStream](https://www.geeksforgeeks.org/java-io-filteroutputstream-class-java/)
* [FilterReader](https://www.geeksforgeeks.org/java-io-filterreader-class-java/)
* [FilterWriter](https://www.geeksforgeeks.org/java-io-filterreader-class-java/)
* [InputStream](https://www.geeksforgeeks.org/java-io-inputstream-class-in-java/)
* [InputStreamReader](https://www.geeksforgeeks.org/java-io-inputstreamreader-class/)
* [LineNumberInputStream](https://www.geeksforgeeks.org/java-io-linenumberinputstream-class-java/)
* [LineNumberReader](https://www.geeksforgeeks.org/java-io-linenumberreader-class-java/)
* ObjectInputStream
* ObjectInputStream.GetField
* ObjectOutputStream – [Set 1](https://www.geeksforgeeks.org/java-io-objectoutputstream-class-java-set-1/) [Set2](https://www.geeksforgeeks.org/java-io-objectoutputstream-class-java-set-2/)
* ObjectOutputStream.PutField
* ObjectStreamClass
* ObjectStreamField
* [OutputStream](https://www.geeksforgeeks.org/java-io-outputstream-class-java/)
* OutputStreamWriter
* [PipedInputStream](https://www.geeksforgeeks.org/java-io-pipedinputstream-class-java/)
* [PipedOutputStream](https://www.geeksforgeeks.org/java-io-pipedoutputstream-class-java/)
* [PipedReader](https://www.geeksforgeeks.org/java-io-pipedreader-class-java/)
* [PipedWriter](https://www.geeksforgeeks.org/java-io-pipedwriter-class-java/)
* PrintStream – [Set1](https://www.geeksforgeeks.org/java-io-printstream-class-java-set-2/)   [Set2](https://www.geeksforgeeks.org/java-io-printstream-class-java-set-2/)
* PrintWriter – [Set1](https://www.geeksforgeeks.org/java-io-printwriter-class-java-set-1/) [Set2](https://www.geeksforgeeks.org/java-io-printwriter-class-java-set-2/)
* PushbackInputStream
* [PushbackReader](https://www.geeksforgeeks.org/java-io-pushbackreader-class-java/)
* RandomAccessFile
* [Reader](https://www.geeksforgeeks.org/java-io-reader-class-java/)
* [SequenceInputStream](https://www.geeksforgeeks.org/java-io-sequenceinputstream-java/)
* SerializablePermission
* StreamTokenizer –[Set1](https://www.geeksforgeeks.org/java-io-streamtokenizer-class-java/) [Set2](https://www.geeksforgeeks.org/java-io-streamtokenizer-class-java-set-2/)
* StringBufferInputStream
* [StringReader](https://www.geeksforgeeks.org/java-io-stringreader-class-java/)
* [StringWriter](https://www.geeksforgeeks.org/java-io-stringwriter-class-in-java/)
* [Writer](https://www.geeksforgeeks.org/java-io-writer-class-java-2/)
* [ZipInputStream class in Java](https://www.geeksforgeeks.org/java-util-zip-zipinputstream-class-java/)
* [ZipEntry class in Java](https://www.geeksforgeeks.org/java-util-zip-zipentry-class-java/)
* [JarEntry class in Java](https://www.geeksforgeeks.org/java-util-jar-jarentry-class-java/)
* [ZipOutputStream class in Java](https://www.geeksforgeeks.org/java-util-zip-zipoutputstream-class-java/)
* [Zip.InflaterInputStream class in Java](https://www.geeksforgeeks.org/java-util-zip-inflaterinputstream-class-java/)
* [Zip.DeflaterInputStream class in Java](https://www.geeksforgeeks.org/java-util-zip-deflaterinputstream-class-java/)
* [Zip.DeflaterOutputStream class in Java](https://www.geeksforgeeks.org/java-util-zip-deflateroutputstream-class-java/)

The java.io package contains nearly every class you might ever need to perform input and output (I/O) in Java. All these streams represent an input source and an output destination. The stream in the java.io package supports many data such as primitives, object, localized characters, etc.

## **Stream**

A stream can be defined as a sequence of data. There are two kinds of Streams −

* **InPutStream** − The InputStream is used to read data from a source.
* **OutPutStream** − The OutputStream is used for writing data to a destination.



Java provides strong but flexible support for I/O related to files and networks but this tutorial covers very basic functionality related to streams and I/O.

## **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.

Java IO

### OutputStream Hierarchy

Java output stream hierarchy

### InputStream Hierarchy

Java input stream hierarchy