**\*\* Java Input/Output \*\***

**Java I/O :**

1. **Java I/O** (Input and Output) is used to process the inputandproduce the output.
2. Java uses the concept of a stream to make I/O operation fast.
3. The java.io package contains all the classes required for input and output operations.
4. We can perform **file handling in Java** by Java I/O API.

## Stream :

1. A stream is a sequence of data. In Java, a stream is composed of bytes.
2. 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 the 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 an error** message to the console.

### InputStream

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

### OutputStream

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

Java IO

### Useful methods of InputStream

|  |  |  |
| --- | --- | --- |
| **No.** | **Method** | **Description** |
|  | public abstract int read()throws IOException | reads the next byte of data from the input stream. It returns -1 at the end of the file. |
|  | public int available()throws IOException | returns an estimate of the number of bytes that can be read from the current input stream. |
|  | public void close()throws IOException | is used to close the current input stream. |

### InputStream Hierarchy

Java input stream hierarchy

## OutputStream class

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

### Useful methods of OutputStream

|  |  |  |
| --- | --- | --- |
| **No.** | **Method** | **Description** |
|  | public void write (int) throws IOException | is used to write a byte to the current output stream. |
|  | public void write(byte[]) throws IOException | is used to write an array of byte to the current output stream. |
|  | public void flush() throws IOException | flushes the current output stream. |
|  | public void close() throws IOException | is used to close the current output stream. |

### OutputStream Hierarchy

Java output stream hierarchy

# FileOutputStream Class

Java FileOutputStream is an output stream used for writing data to a [file](https://www.javatpoint.com/java-file-class).

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](https://www.javatpoint.com/java-filterwriter-class) than FileOutputStream.

# File Class

1. The File class is an abstract representation of file and directory pathname.
2. A pathname can be either absolute or relative.

### [Constructors](https://www.javatpoint.com/java-constructor)

|  |  |  |
| --- | --- | --- |
| **No .** | **Constructor** | **Description** |
|  | File(File parent, String child) | It creates a new File instance from a parent abstract pathname and a child pathname string. |
|  | File(String pathname) | It creates a new File instance by converting the given pathname string into an abstract pathname. |
|  | File(String parent, String child) | It creates a new File instance from a parent pathname string and a child pathname string. |
|  | File(URL) | It creates a new File instance by converting the given file: URI into an abstract pathname. |

### Useful Methods

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Modifier and Type** | **Method** | **Description** |
|  | static File | createTempFile(String  prefix, String suffix) | It creates an empty file in the default temporary-file directory, using the given prefix and suffix to generate its name. |
|  | boolean | createNewFile() | It atomically creates a new, empty file named by this abstract pathname if and only if a file with this name does not yet exist. |
|  | boolean | canWrite() | It tests whether the application can modify the file denoted by this abstract pathname.String[] |
|  | boolean | canExecute() | It tests whether the application can execute the file denoted by this abstract pathname. |
|  | boolean | canRead() | It tests whether the application can read the file denoted by this abstract pathname. |
|  | boolean | isAbsolute() | It tests whether this abstract pathname is absolute. |
|  | | boolean | isDirectory() | It tests whether the file denoted by this abstract pathname is a directory. |
|  | | boolean | isFile() | It tests whether the file denoted by this abstract pathname is a normal file. |
|  | | String | getName() | It returns the name of the file or directory denoted by this abstract pathname. |
|  | | String | getParent() | It returns the pathname string of this abstract pathname's parent, or null if this pathname does not name a parent directory. |
|  | | Path | toPath() | It returns a java.nio.file.Path object constructed from the this abstract path. |
|  | | URI | toURI() | It constructs a file: URI that represents this abstract pathname. |
|  | | File[] | listFiles() | It returns an [array](https://www.javatpoint.com/array-in-java) of abstract pathnames denoting the files in the directory denoted by this abstract pathname |
|  | | long | getFreeSpace() | It returns the number of unallocated bytes in the partition named by this abstract path name. |
|  | | String[] | list(FilenameFilter filter) | It returns an array of strings naming the files and directories in the directory denoted by this abstract pathname that satisfy the specified filter. |
|  | | boolean | mkdir() | It creates the directory named by this abstract pathname. |