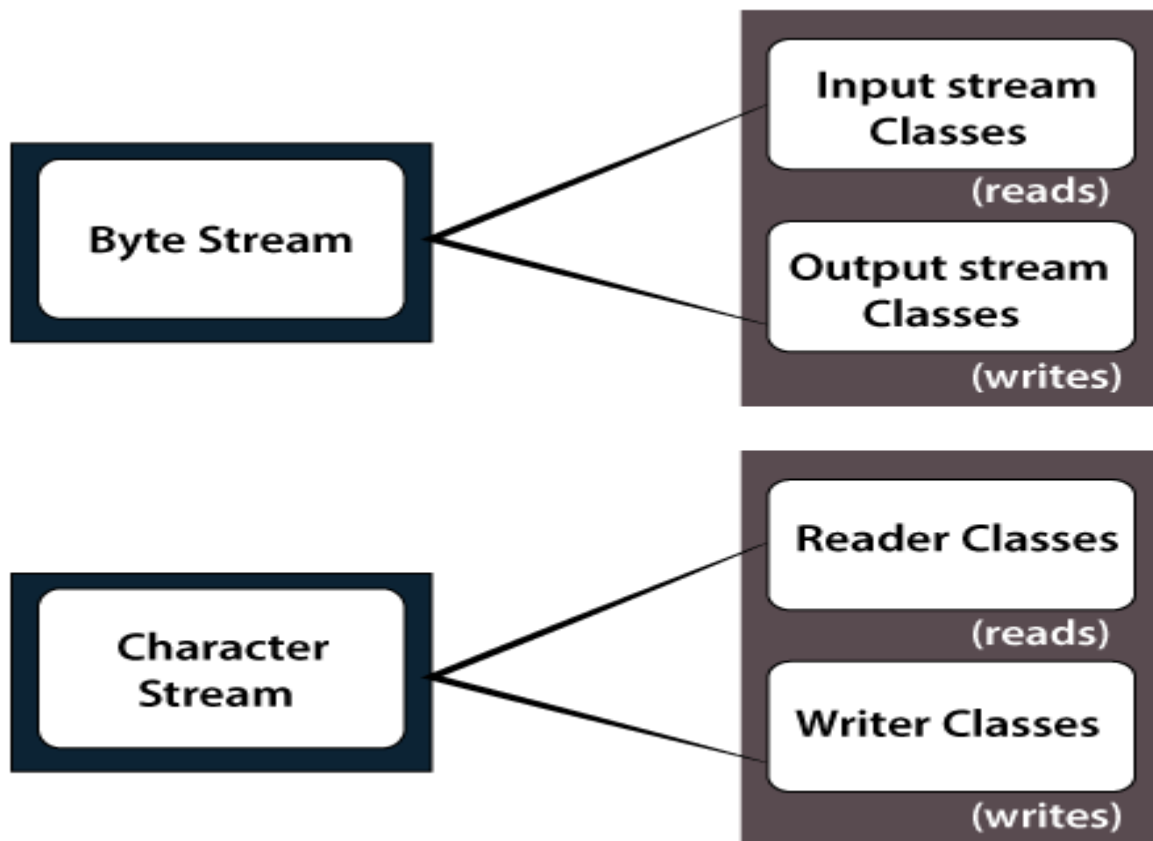


Stream

Dr. Monika Patel
Asst Professor
Computer Dept
Durga College Raipur(C.G.)

A series of data is referred to as a **stream**. In Java, **Stream** is classified into two types, i.e., **Byte Stream** and **Character Stream**.



Brief classification of I/O streams

Byte Stream

Byte Stream is mainly involved with byte data. A file handling process with a byte stream is a process in which an input is provided and executed with the byte data.

Character Stream

Character Stream is mainly involved with character data. A file handling process with a character stream is a process in which an input is provided and executed with the character data.

Java File Class Methods

S.No	Method	Return Type	Description
1.	canRead()	Boolean	The canRead() method is used to check whether we can read the data of the file or not.
2.	createNewFile()	Boolean	The createNewFile() method is used to create a new empty file.
3.	canWrite()	Boolean	The canWrite() method is used to check whether we can write the data into the file or not.
4.	exists()	Boolean	The exists() method is used to check whether the specified file is present or not.
5.	delete()	Boolean	The delete() method is used to delete a file.
6.	getName()	String	The getName() method is used to find the file name.
7.	getAbsolutePath()	String	The getAbsolutePath() method is used to get the absolute pathname of the file.
8.	length()	Long	The length() method is used to get the size of the file in bytes.
9.	list()	String[]	The list() method is used to get an array of the files available in the directory.
10.	mkdir()	Boolean	The mkdir() method is used for creating a new directory.

File Operations

We can perform the following operation on a file:

- Create a File
- Get File Information
- Write to a File
- Read from a File
- Delete a File

File Operations in Java



Create a File

Create a File operation is performed to create a new file. We use the `createNewFile()` method of file. The `createNewFile()` method returns true when it successfully creates a new file and returns false when the file already exists.

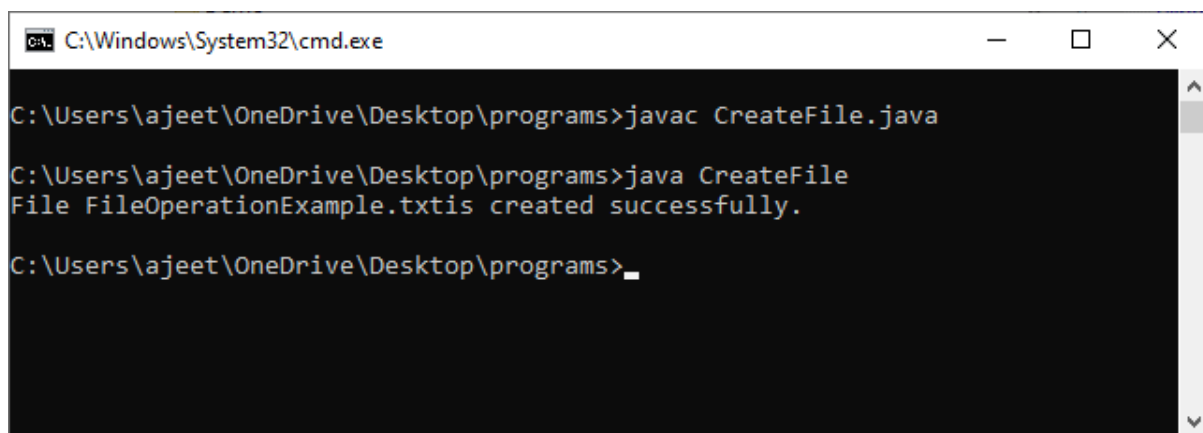
Let's take an example of creating a file to understand how we can use the `createNewFile()` method to perform this operation.

CreateFile.java

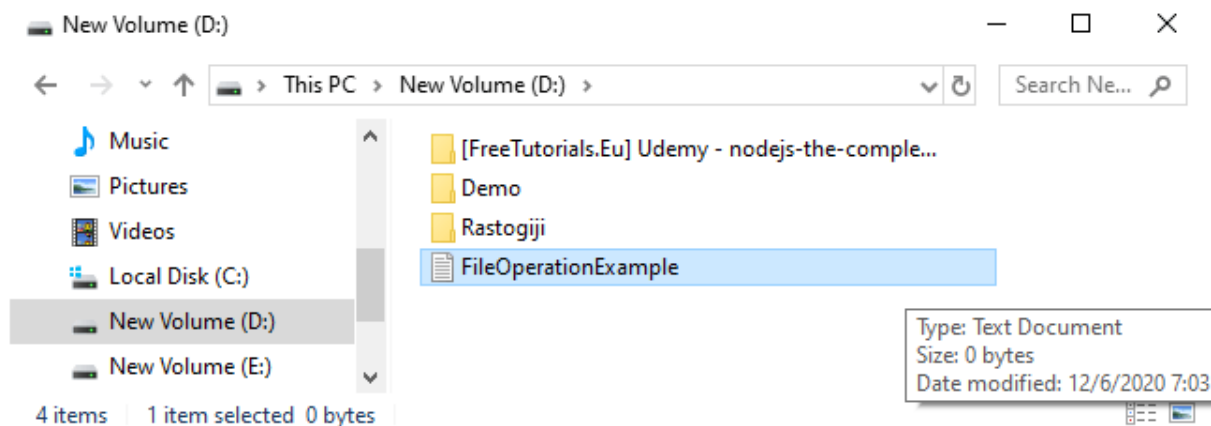
```
1. // Importing File class
2. import java.io.File;
3. // Importing the IOException class for handling errors
4. import java.io.IOException;
5. class CreateFile {
6.     public static void main(String args[]) {
7.         try {
8.             // Creating an object of a file
9.             File f0 = new File("D:FileOperationExample.txt");
10.            if (f0.createNewFile()) {
11.                System.out.println("File " + f0.getName() + " is created successfully.");
```

```
12.         } else {
13.             System.out.println("File is already exist in the directory.");
14.         }
15.     } catch (IOException exception) {
16.         System.out.println("An unexpected error is occurred.");
17.         exception.printStackTrace();
18.     }
19. }
20. }
```

Output:



```
C:\Windows\System32\cmd.exe
C:\Users\ajeet\OneDrive\Desktop\programs>javac CreateFile.java
C:\Users\ajeet\OneDrive\Desktop\programs>java CreateFile
File FileOperationExample.txt is created successfully.
C:\Users\ajeet\OneDrive\Desktop\programs>_
```



Explanation:

In the above code, we import the File and IOException class for performing file operation and handling errors, respectively. We create the **f0** object of the File class and specify the location of the directory where we want to create a file. In the try block, we call the **createNewFile()** method through the **f0** object to create a new file in the specified location. If the method returns false, it will jump to the else section. If there is any error, it gets handled in the catch block.

Get File Information

The operation is performed to get the file information. We use several methods to get the information about the file like name, absolute path, is readable, is writable and length.

Let's take an example to understand how to use file methods to get the information of the file.

FileInfo.java

```
1. // Import the File class
2. import java.io.File;
3. class FileInfo {
4.     public static void main(String[] args) {
5.         // Creating file object
6.         File f0 = new File("D:FileOperationExample.txt");
7.         if (f0.exists()) {
8.             // Getting file name
9.             System.out.println("The name of the file is: " + f0.getName());
10.
11.            // Getting path of the file
12.            System.out.println("The absolute path of the file is: " + f0.getAbsolutePath());
13.
14.            // Checking whether the file is writable or not
15.            System.out.println("Is file writeable?: " + f0.canWrite());
16.
17.            // Checking whether the file is readable or not
18.            System.out.println("Is file readable " + f0.canRead());
19.
20.            // Getting the length of the file in bytes
21.            System.out.println("The size of the file in bytes is: " + f0.length());
22.        } else {
23.            System.out.println("The file does not exist.");
24.        }
25.    }
26. }
```