Table of Contents

[1 Introduction 2](#_Toc433744854)

[2 File 2](#_Toc433744855)

[2.1 File class Constructors 3](#_Toc433744856)

[3 FileReader 3](#_Toc433744857)

[4 BufferedWriter 3](#_Toc433744858)

[5 BufferedReader 3](#_Toc433744859)

[6 PrintWriter 4](#_Toc433744860)

# Introduction

Java IO is one of the most important concept for day to day programming in java. To store small amount of data we use files, and it’s better to user some database to store large amount of data.

# File

File is class useful to perform most common operation on file, like create a file, rename, etc.

**File f = new File("abc.txt");**

This line **won't create any physical files**, first it will check whether any physical file already available with the name of "abc.txt" or not.

If it is not already avalilable this line won't create any physical file and just it creates a java file object to represent the name abc.txt

File f = new File("abc.txt"); -------🡪 This line creates only **java file Object**

System.out.println(f.exists());

f.createNewFile(); --------------------🡪 this line will create physical file

System.out.println(f.exists());

F

abc.txt

Output:

|  |  |
| --- | --- |
| 1st Run | 2nd Run |
| False | True |
| True | True |

We can use java file object to represent directory also.

File f = new File("abc.txt");

System.out.println(f.exists());

f.mkdir();-------------------🡪 this line will create a file

System.out.println(f.exists());

Note: Java file IO concept is implemented based on UNIX system and in UNIX everything is treat as file. Hence we can use java file object to represent both files and directories.

## File class Constructors

Below are the constructors of File class.

1. **File f = new File(String name);**

Creates a java file object to represent name of specified file or directory present in current working directory.

1. **File f = new File(String subdir, String name);**

Creates a java file object to represent name of the file or directory present in specific directory.

1. **File f = new File(String subdir, String name);**

* Below is the code to create a file named with abc.txt in current working directory

File f = new File(abc.txt);

f.createNewFile();

* Below is the code to create a directory nameed with "naren" in current working directory

and under that create a file named with "abc.txt".

File f = new File("naren");

f.mkdir();

File f1 = new File("naren", "abc.txt");

File f2 = new File(f, "abc.txt");

f.createNewFile();

# FileReader

Before you can do any inspection on a class you need to obtain its java.lang.Class object. All types in Java including the primitive types (int, long, float etc.) including arrays have an associated Class object.

# BufferedWriter

Before you can do any inspection on a class you need to obtain its java.lang.Class object. All types in Java including the primitive types (int, long, float etc.) including arrays have an associated Class object.

# BufferedReader

Before you can do any inspection on a class you need to obtain its java.lang.Class object. All types in Java including the primitive types (int, long, float etc.) including arrays have an associated Class object.

# PrintWriter

Before you can do any inspection on a class you need to obtain its java.lang.Class object. All types in Java including the primitive types (int, long, float etc.) including arrays have an associated Class object.