File Handling

File:

It is an entity where we can store data organise data according to our requirements.

There are two types of files:

1. **Target file:**
   If the data is getting written inside a file, then it is called as target file.

2. **Source file:**
   If the data is getting fetched from the file, then it is called as source file.

   File Handling:
   Performing crud operations on file is called as file handling.
   1. Creating new file
   2. Deleting file
   3. Writing data inside file
   4. Fetching data from the file

1. **Creating New file:**

   Note:

   In order to perform any operation on file we need to create object for File class present inside java.io package.

   It will accept one argument of string type, i.e. name and location of file along with extension.

   createNewFile():

   It is a non-static method present inside File class. It will return true if file is not present at specified location otherwise false.

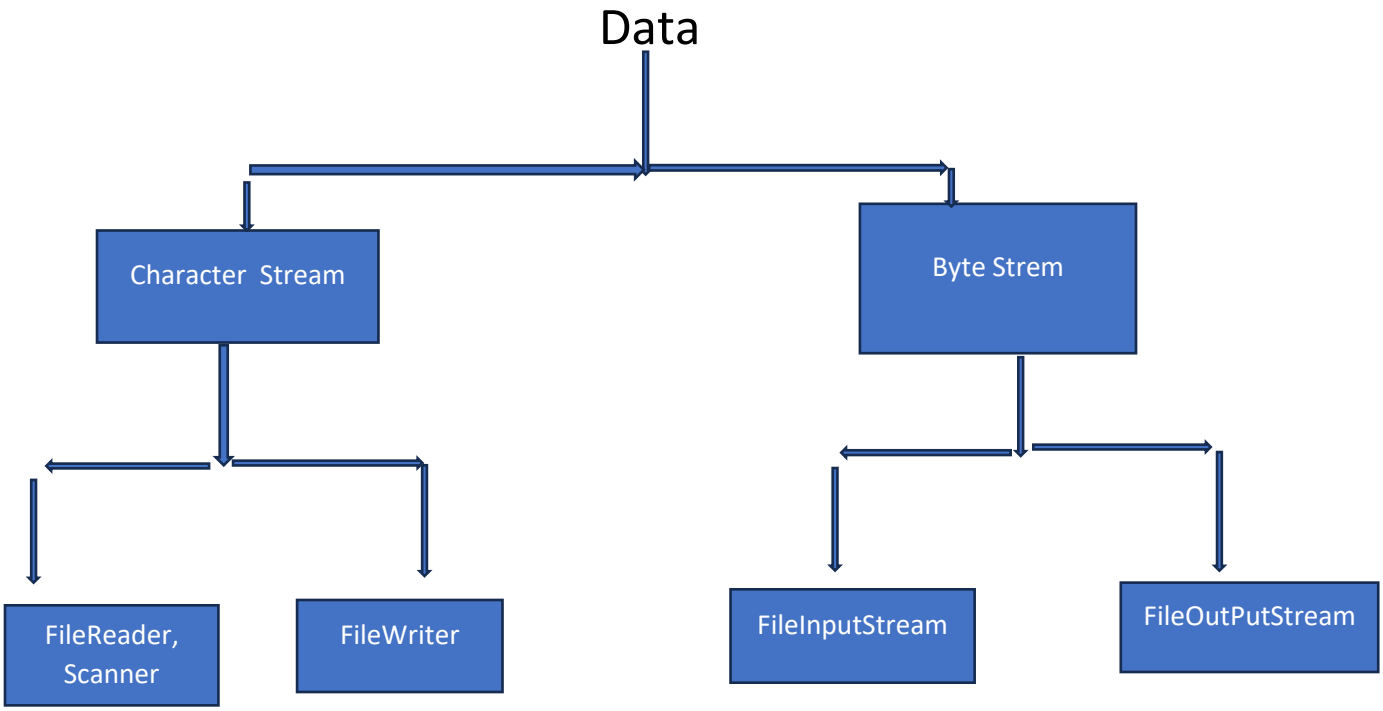
2. **Deleting a file**:

   delete():

   It is a non-static method present inside File class. It is used to delete a file. If file gets deleted then it returns true otherwise it returns false.

   exists():

   It is a non static method present inside File class. It returns true if file is present otherwise it returns false.



Some Important Methods of File Class

1.**getName():**

It is a non static method present inside a File class which is used to fetch the name of the file.

2.**getAbsolutePath():**

It is a non static method present inside a File class which is used to fetch the exact location of file.

3. **length():**

It is a non static method present inside a File class which is used to fetch the no. of characters present inside a file.

4. **canWrite():**

It is a non static method present inside a File class. It returns true if file is writable otherwise false.

5. canRead():

It is a non static method present inside a File class. It returns true if file is readable otherwise false.

6. canExecute():

It is a non static method present inside a File class. It returns true if file is executable otherwise false.

Note:

1. To write Character stream data inside file we need to create object of FileWriter Class.

2. To read Character stream data from file we need to create object of FileReader or Scanner Class.

3. To write Byte stream data inside file we need to create object of FileOutPutStream Class.

4. To read Byte stream data from file we need to create object of FileInputStream Class.