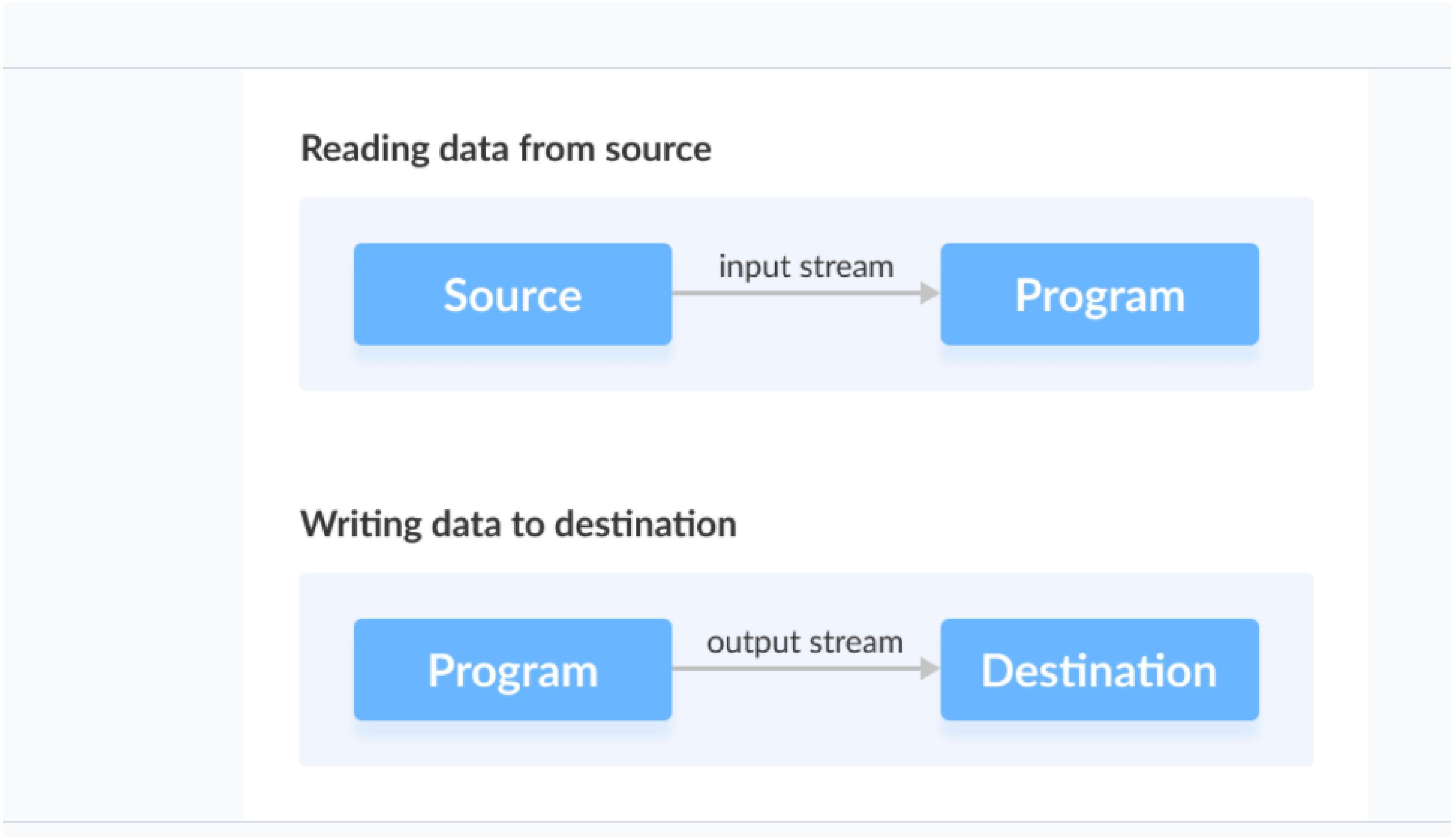
Java File Input/output

File Handling in Java

File handling refers to working with the file in java. Reading files & writing into java files is known as file handling in java.

File Handing in java comes under Java I/O package. java.io classes are specially provided for file handling in java.

File Handling in Java



File Handling in Java

Some of the common file handling operations are;

- Create file
- Delete file
- Read file
- Write file
- Change file permissions
- Opening file
- Closing file

File Handling Methods

Some of the methods are given below for performing different operations in java:

createNewFile(): createNewFile method used to create an empty file. It returns the response as boolean.
getName(): This method is used to get the file name. It returns the string i.e. name of the file in response.
getAbsolutePath(): It returns the absolute path of the file. The return type of this method is a string.
canRead(): This method used to check whether the file is readable or not. It returns a boolean value.
canWrite(): This method used to check whether the file is writable or not. It returns a boolean value.
delete(): This method used in deleting a file. It returns a boolean value.
exists(): This method used to check whether a file exists or not. It returns a boolean value.
length(): This method returns the size of the file in bytes. The return type of this method is long.
list(): This method returns an array of the files available in the directory. It returns an array of string values.
mkdir(): This method is used to create a directory. It returns a boolean value.

Java Reader Class
Java Writer Class

Java Reader Class

Using Java BufferedReader Class

The BufferedReader class of the java.io package can be used with other readers to read data (in characters) more efficiently.

Create a BufferedReader

- ❑ In order to create a BufferedReader, we must import the java.io.BuferedReader package first.
- ❑ Here is how we can create the reader.

```
// Creates a FileReader
FileReader file = new FileReader(String file);
// Creates a BufferedReader
BufferedReader buffer = new BufferedReader(file);
```

Using scanner to read text file in java

- Scanner breaks its input into tokens using a delimiter pattern, which by default matches whitespace.
- The resulting tokens may then be converted into values of different types using the various next methods

```
Path path = Paths.get(filename);
Scanner scanner = new Scanner(path);
```

Then use while loop to read file content

```
while(scanner.hasNextLine()){
    //process each line
    String line = scanner.nextLine();
    System.out.println(line);
}
```

Java write to files

Using FileWriter

In order to create a file writer, we must import the Java.io.FileWriter package first. Once we import the package, here is how we can create the file writer.

```
// Creates a Writer using FileWriter
FileWriter output = new FileWriter(filename);
```

write() Method

•write() - writes a single character to the writer

•write(char[] array) - writes the characters from the specified array to the writer

•write(String data) - writes the specified string to the writer

Java write to files

Using Java PrintWriter Class

PrintWriter converts the primitive data (int, float, char, etc.) into the text format. It then writes that formatted data to the writer.

```
// Creates a PrintWriter
PrintWriter output = new PrintWriter("output.txt");
```

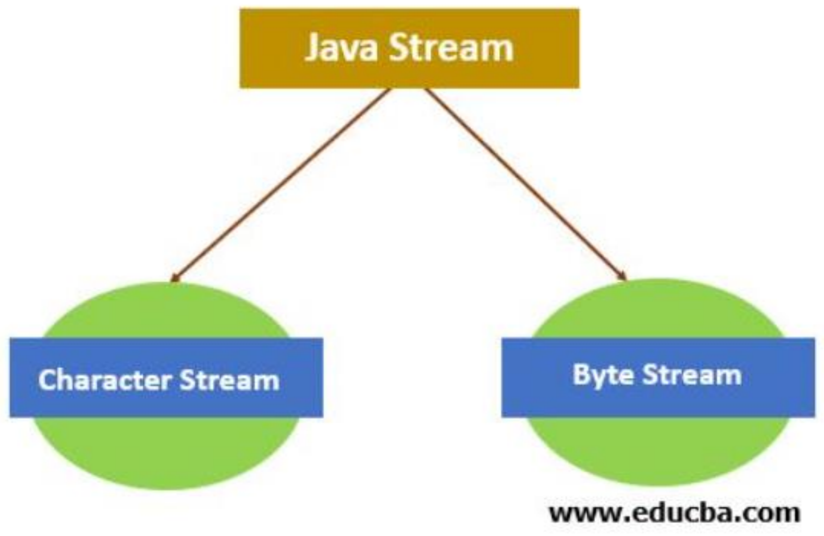
write() Method

- write() - writes a single character to the writer
- write(char[] array) - writes the characters from the specified array to the writer
- write(String data) - writes the specified string to the writer

How does File Handling work?

- In Java, File handling takes place by streaming concepts.
- Input/output operations on a file perform through the streaming.
- Stream refers to a sequence of data.

In java, Stream is of two types:


Java Stream
Character Stream
Byte Stream
www.educba.com