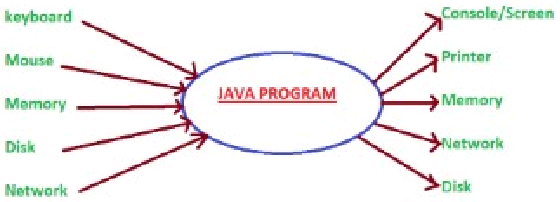
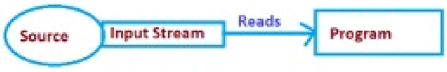
Managing Input/Output Files in JAVA

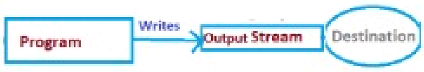


Streams:

- Collection of continues group of data which will travel from one point to another point is known as Stream.



(a) Reading data into a program.



(a) Writing data to destination

1) System.in (Input Stream)
2) System.out (Output Stream)
3) System.err (Error Stream)

java.io package

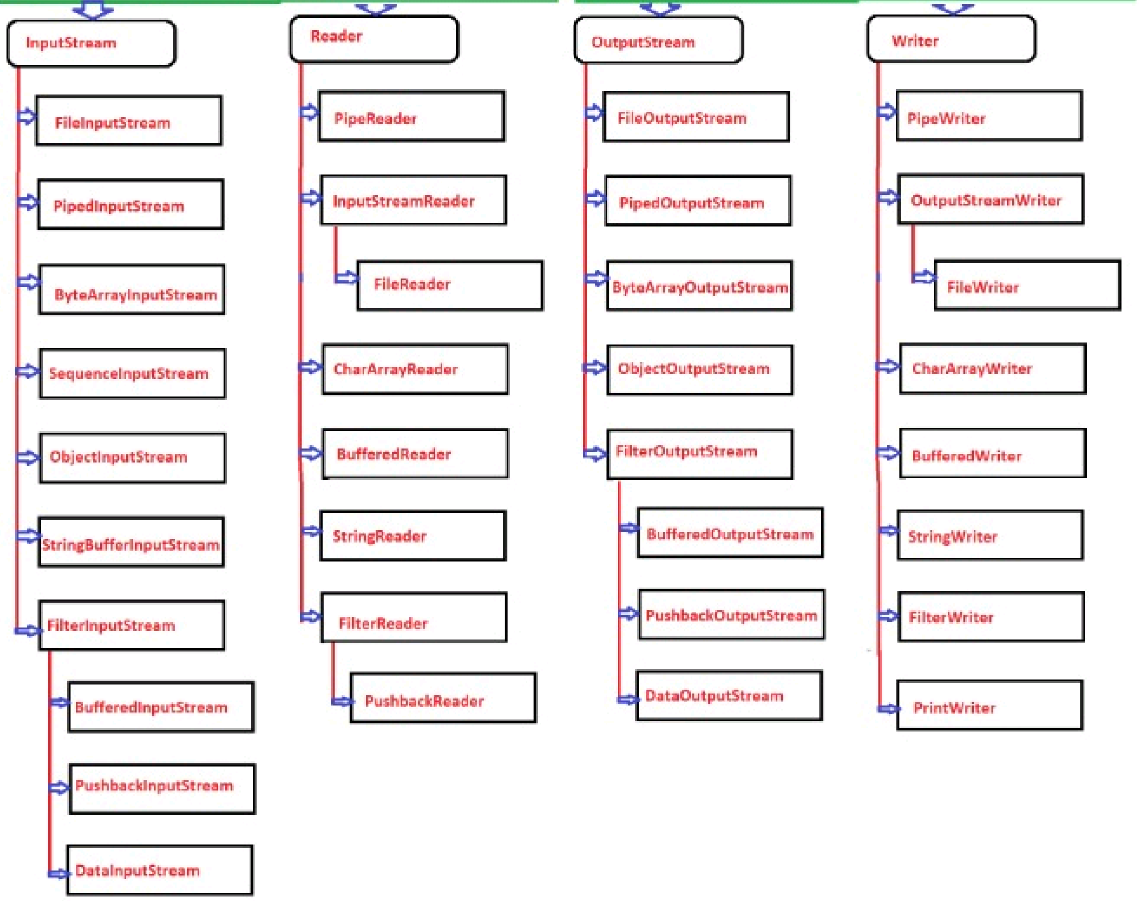
VJTech Academy

Byte Stream Class	Character Stream Class
1) Provides support for handling IO operations on byte.	1) Provides support for handling IO operations on characters.
2) Two types: i - Input Stream Classes ii - Output Stream Classes	2) Two types: i - Reader Classes ii - Writer Classes
3) Input stream classes are used to read 8 bit byte data.	3) Reader stream classes are used to read character from files.
4) Output stream classes are used to write 8 bit byte data.	4) Writer stream classes are used to write character into files.
5) We can translate character stream into byte stream with OutputStreamWriter.	5) We can translate byte stream into character stream with InputStreamReader.

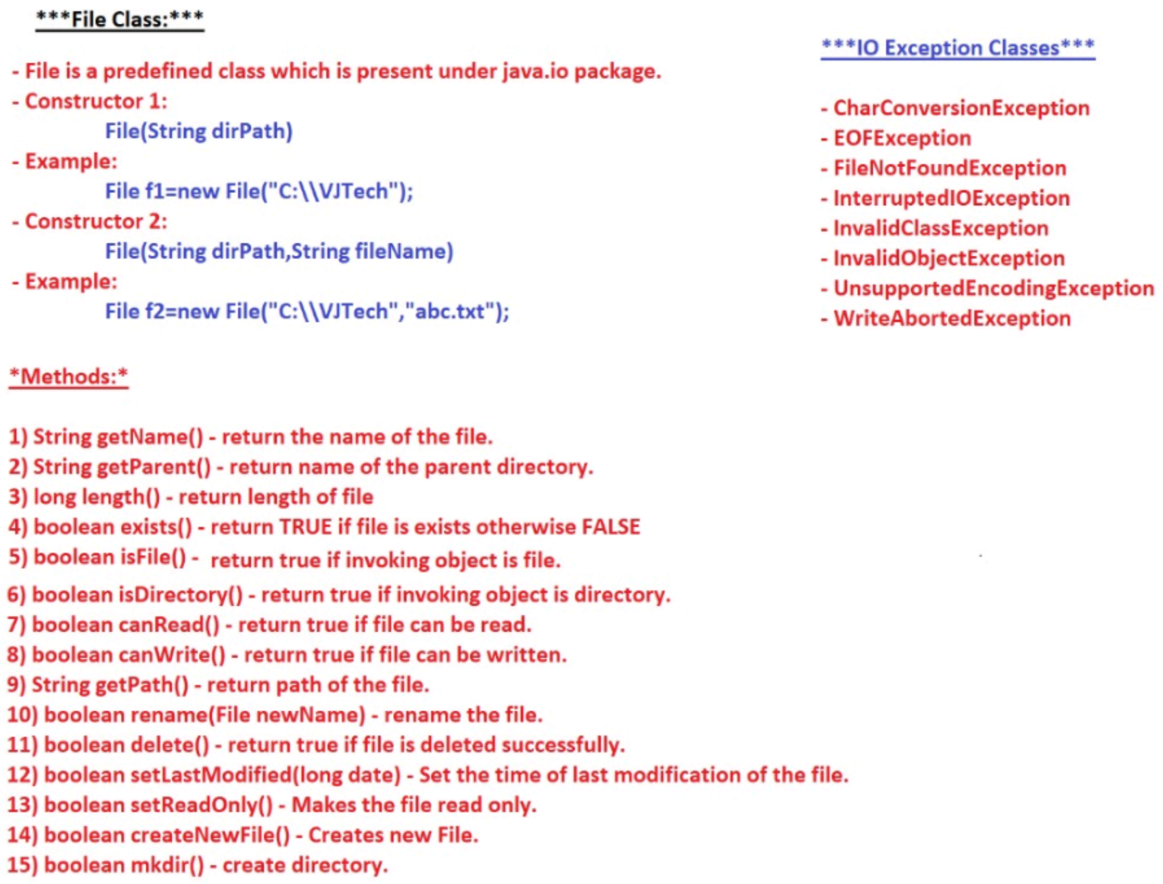
VJTech Academy

InputStream Class	Reader Class
1) InputStream classes are used to read 8 bit bytes data.	1) Reader class is used to read 16 bit unicode character stream.
2) InputStream classes are byte-oriented.	2) Reader class is character oriented.
3) Constructor: InputStream()	3) Constructor: Reader() Reader(Object lock)
4) Methods: i) read() - Reads the next byte of data ii) read(byte b[]) - Read some bytes from input stream & stored them into byte array b. iii) read(byte b[],int m,int n) - Read n bytes data from input stream starting from mth byte & stored them into byte array b. iv) close() - Closes the input stream. v) skip(n) - skips over n bytes from input stream vi) available() - No of bytes available	4) Methods: i) read() - Read single character ii) read(char c[]) - Read characters into an array. iii) read(char c[],int m,int n) - Read n characters data from input stream starting from mth character & stored them into char array c. iv) close() - Closes the input stream. v) skip(n) - skips over n characters from stream vi) reset() - Goes back to the beginning of the stream.

OutputStream Class	Writer Class
1)OutputStream classes are used to write 8 bit bytes data.	1) Writer class is used to 16 bit unicode character stream.
2)OutputStreamclasses are byte-oriented.	2) Writer class is character oriented.
3) Constructor: OutputStream()	3) Constructor: Writer() Writer(Object lock)
4) Methods: i) write() - Writes byte of data ii) write(byte b[]) - Writes byte array b to the output stream. iii)write (byte b[],int m,int n) - Write n bytes data from array b to output stream which is starting from mth byte iv) close() - Closes the output stream v) skip(n) - skips over n bytes from output stream	4) Methods: i) write(char c) - write character c ii) write(char c[]) - writes character array. iii) write(char c[],int m,int n) - write n characters from array c starting from m character. iv) write(String str)- writes string. v) close() - closes the writer stream. vi) flush() - flushes the output stream.



- InputStream
  - FileInputStream
  - PipedInputStream
  - ByteArrayInputStream
  - SequenceInputStream
  - ObjectInputStream
  - StringBufferInputStream
  - FilterInputStream
    - BufferedInputStream
    - PushbackInputStream
    - DataInputStream
- Reader
  - PipeReader
  - InputStreamReader
    - FileReader
  - CharArrayReader
  - BufferedReader
  - StringReader
  - FilterReader
    - PushbackReader
- OutputStream
  - FileOutputStream
  - PipedOutputStream
  - ByteArrayOutputStream
  - ObjectOutputStream
  - FilterOutputStream
    - BufferedOutputStream
    - PushbackOutputStream
    - DataOutputStream
- Writer
  - PipeWriter
  - OutputStreamWriter
    - FileWriter
  - CharArrayWriter
  - BufferedWriter
  - StringWriter
  - FilterWriter
  - PrintWriter



File Class:

- File is a predefined class which is present under java.io package.
- Constructor 1:

```
File(String dirPath)
```

- Example:

```
File f1=new File("C:\\VJTech");
```

- Constructor 2:

```
File(String dirPath,String fileName)
```

- Example:

```
File f2=new File("C:\\VJTech","abc.txt");
```

Methods:

1) String getName() - return the name of the file.
2) String getParent() - return name of the parent directory.
3) long length() - return length of file
4) boolean exists() - return TRUE if file is exists otherwise FALSE
5) boolean isFile() - return true if invoking object is file.
6) boolean isDirectory() - return true if invoking object is directory.
7) boolean canRead() - return true if file can be read.
8) boolean canWrite() - return true if file can be written.
9) String getPath() - return path of the file.
10) boolean rename(File newName) - rename the file.
11) boolean delete() - return true if file is deleted successfully.
12) boolean setLastModified(long date) - Set the time of last modification of the file.
13) boolean setReadOnly() - Makes the file read only.
14) boolean createNewFile() - Creates new File.
15) boolean mkdir() - create directory.

IO Exception Classes

- CharConversionException
- EOFException
- FileNotFoundException
- InterruptedIOException
- InvalidClassException
- InvalidObjectException
- UnsupportedEncodingException
- WriteAbortedException