**JAC444 - Lecture 6**

Java Input / Output

Segment 1- Basics

**Objectives**

**Upon completion of this lecture, you should be able to:**

* Examine Input / Output classes in Java
* Create and Use I/O Streams in Java
* Distinguish Byte, Character, and Buffered Stream
* Design and Develop File I/O programs

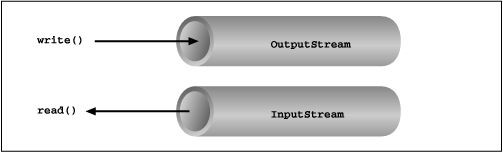
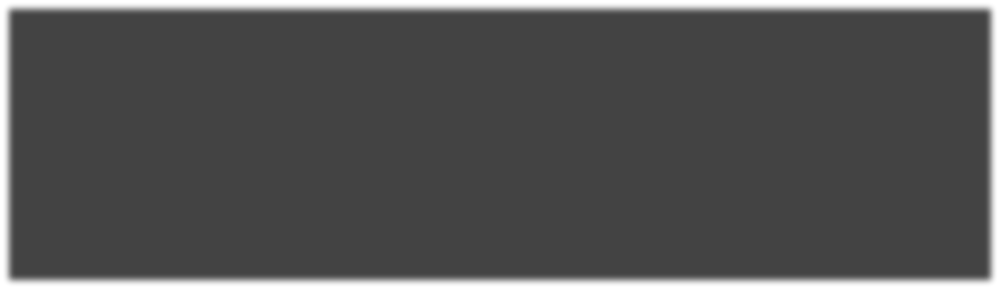
Input / Output

**In this lesson you will be learning about:**

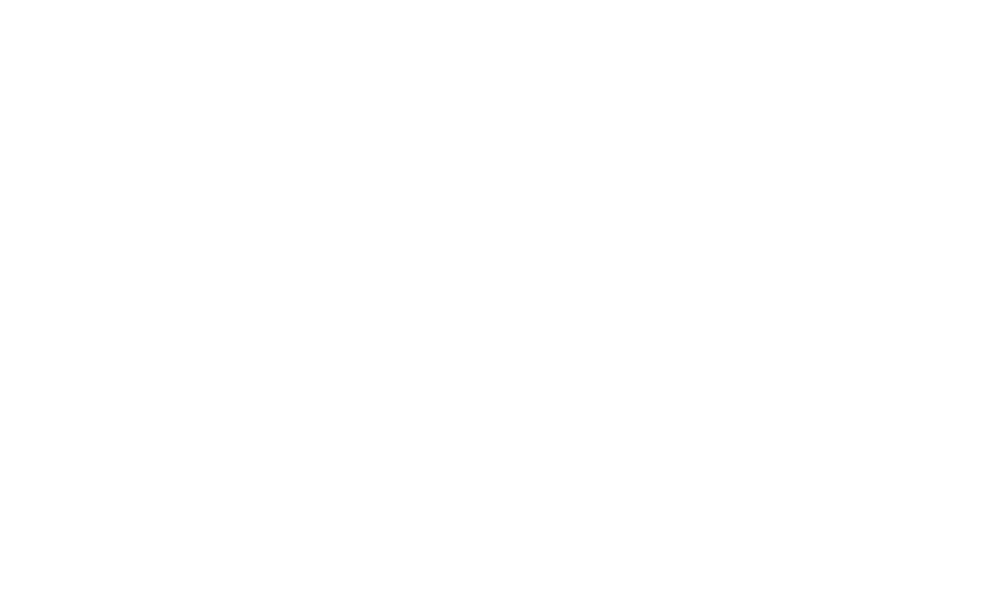
* Types of Input / Output Stream
* Typical use of IO Streams
* Character and Byte Streams.

**Reading / Writing Data**

|  |  |
| --- | --- |
| **Reading**  *Open a stream while more information read information*  *close the stream* | **Writing**  *open a stream while more information write information*  *close the stream* |



**I/O Package**



1.

Contains a collection of classes that support I/O algorithms.

2.

Classes are divided into two class hierarchies based on the data type o

●

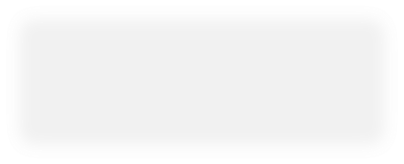
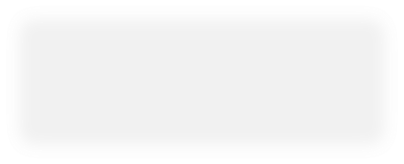
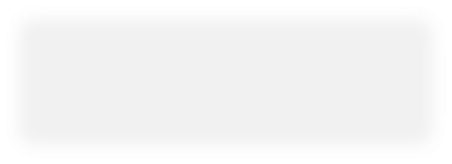
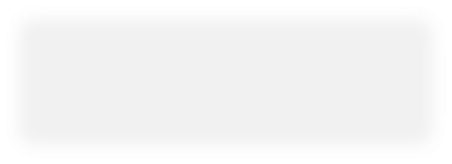
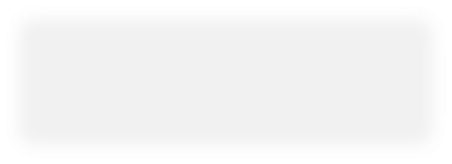
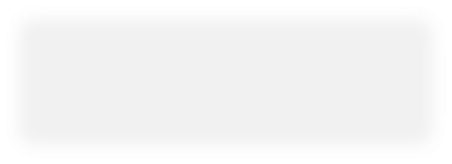
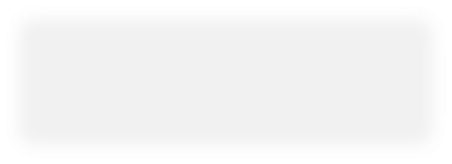
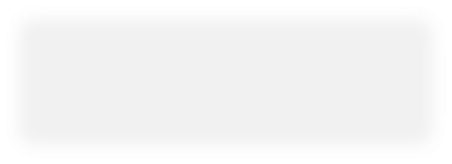
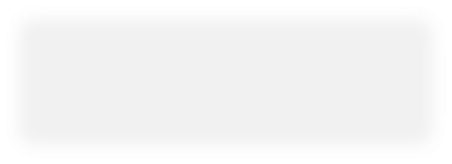
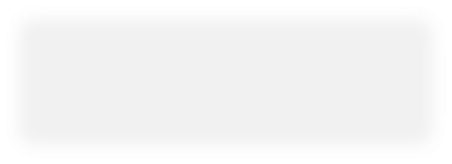
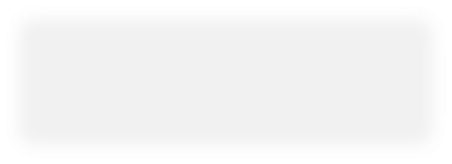
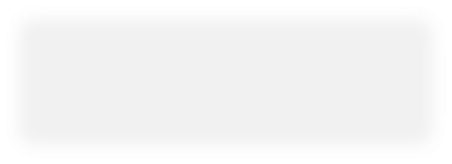
char

●

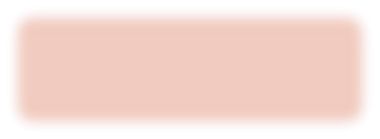
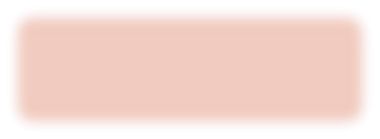
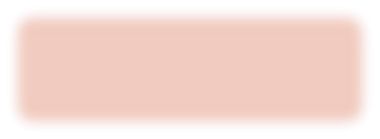
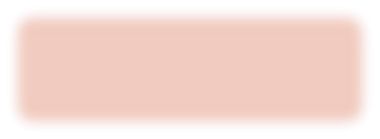
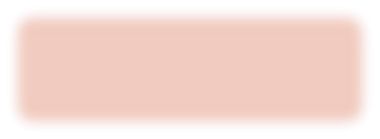
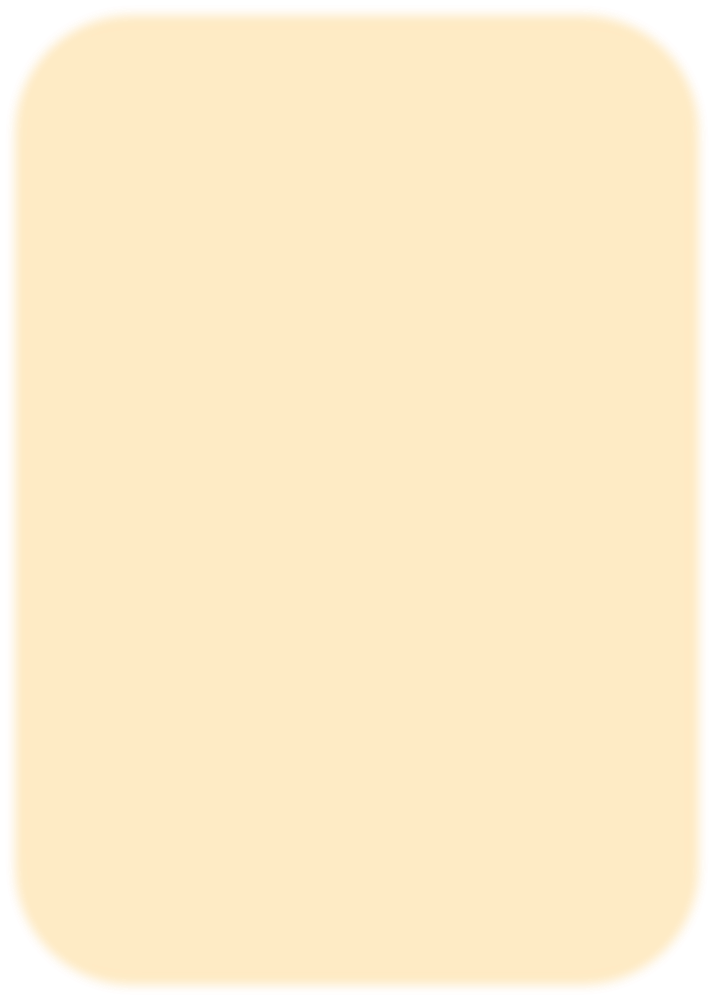
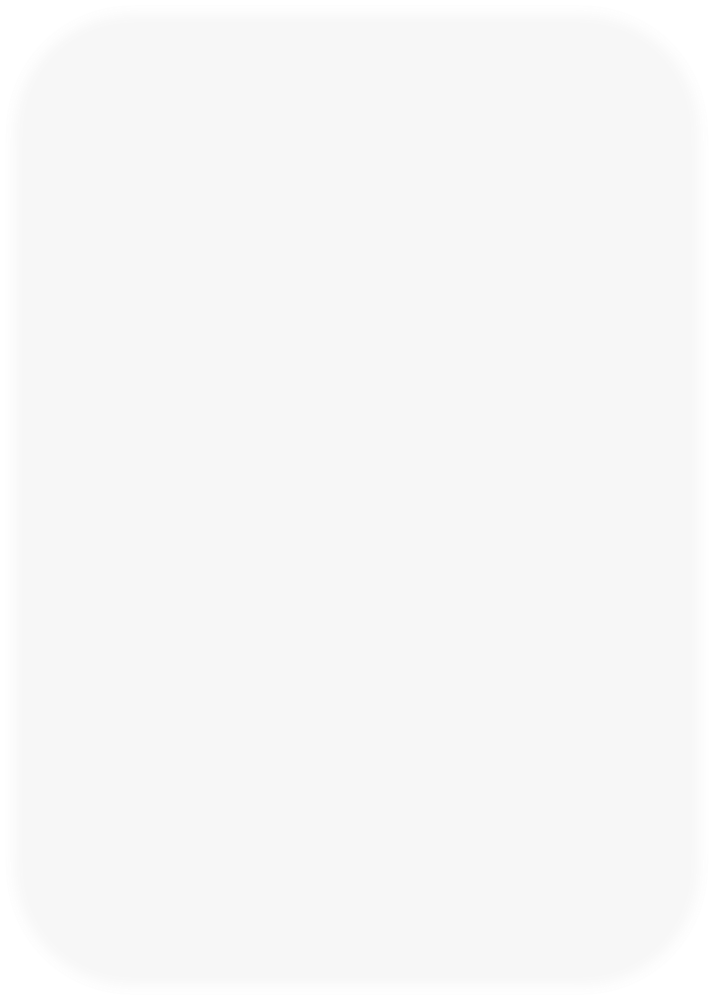
byte

**CHARS Reader/Writer**

**BYTES Input/Output Streams**



**Logical Group of IO Classes**



**Character Stream**

**Byte Stream**

**Data Sink**

**Stream**

**Processing**

**Stream**

**Character Stream**

* **Reader/Writer** are abstract superclasses for character streams in **java.io**package.
* **Reader** Abstract class for reading character streams and provides methods for reading 16-bit characters.
* The only methods that a subclass must implement are **read(char[],int, int)** and **close()***.*
* **Writer**Abstract class for writing to character streams and provides methods for writing 16-bit characters.
* The only methods that a subclass must implement are **write(char[], int, int), flush(), and close().**

**Reader**



**Reader**



**BufferedReader**



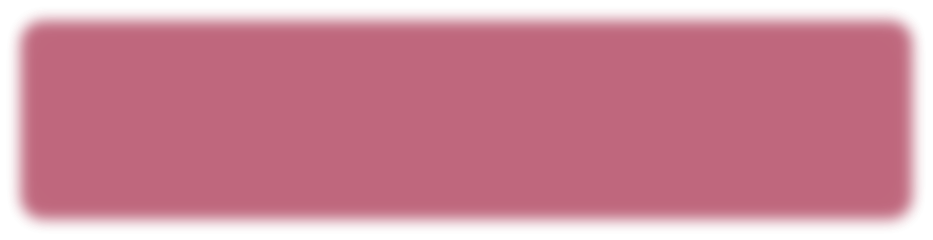
**LineNumberReader**



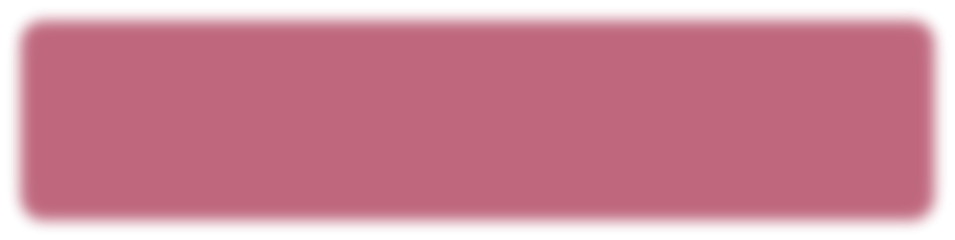
**CharArrayReader**



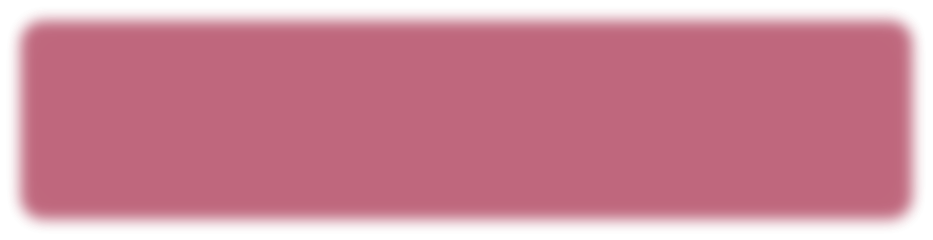
**InputStreamReader**



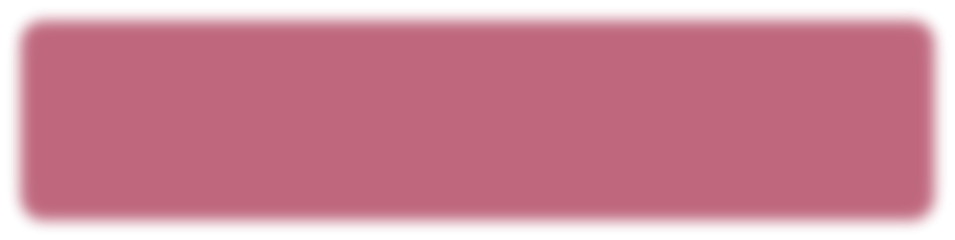
**FilterReader**



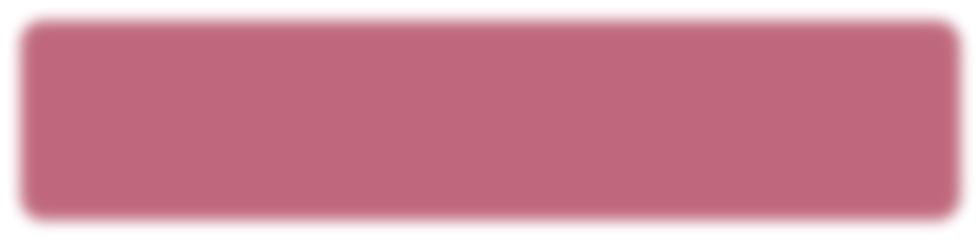
**PipedReader**



**StringReader**



**FileReader**



**PushbackReader**

Data sinks

**Writer**



**Writer**



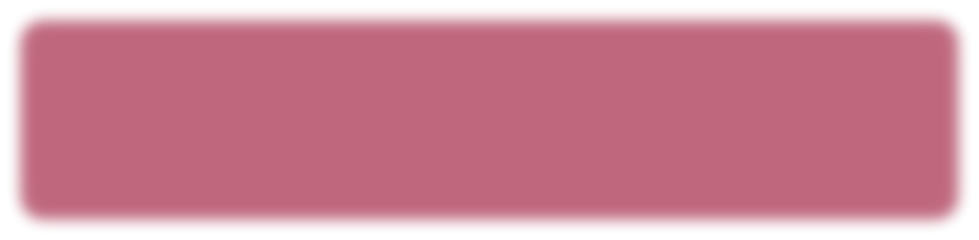
**BufferedWriter**



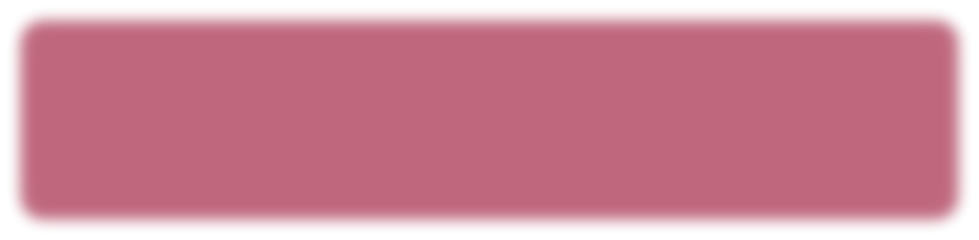
**CharArrayWriter**



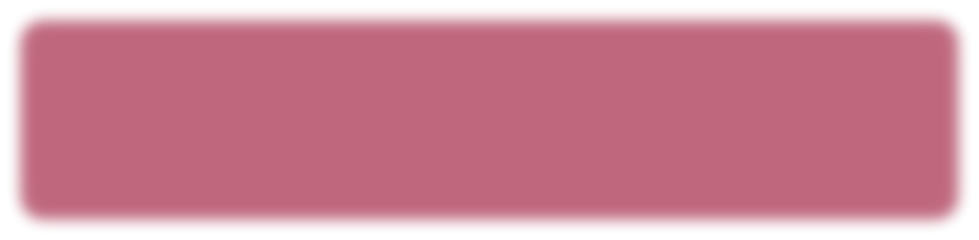
**OutputStreamWriter**



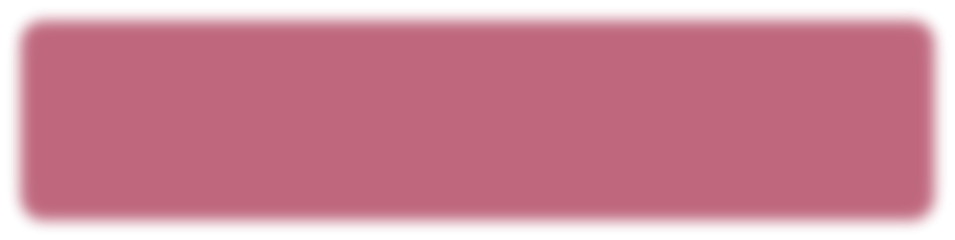
**FilterWriter**



**PipedWriter**



**StringWriter**



**FileWriter**

Data sinks

**Byte Streams**

* All byte stream classes are descended from **InputStream** and **OutputStream**
* **InputStream** and **OutputStream** provide the methods and some implementation for read and write 8 bit bytes.
* Streams are typically used to read and write binary data such images and sounds
* Input/Output subclasses provide specialized I/O for streams

**Input Stream**



**InputStream**



**FileInputStream**



**LineNumber**

**…**



**FilterInputStream**



**ByteArrayInputStre**

**~~a~~**

**m**



**SequenceInputStream**



**StringReaderObjectOutputStream**



**Data**

**…**



**Buffered**

**…**



**StringBufferedInputStream**

**Output Stream**



**OutputStream**



**FilterOutputStream**



**LineNumber**

**…**



**PipedOutputStream**



**ObjectOutputStream**



**ByteArrayOutputStream**



**Data**

**…**



**Buffered**

**…**



**FileOutputStream**

**Byte Stream Question**

* **InputStream** and **OutputStream** provide the methods and some implementation for read and write 8 bit bytes.
* **int read() throws IOException**

It returns: the next byte of data, or -1 if the end of the stream is reached

*Question*: The value of a byte b is in the range -128 <= b < 127

*When you invoke* ***read****() method and the returned value is -1, how do you know that this is the end of the stream or is the byte value from your stream*