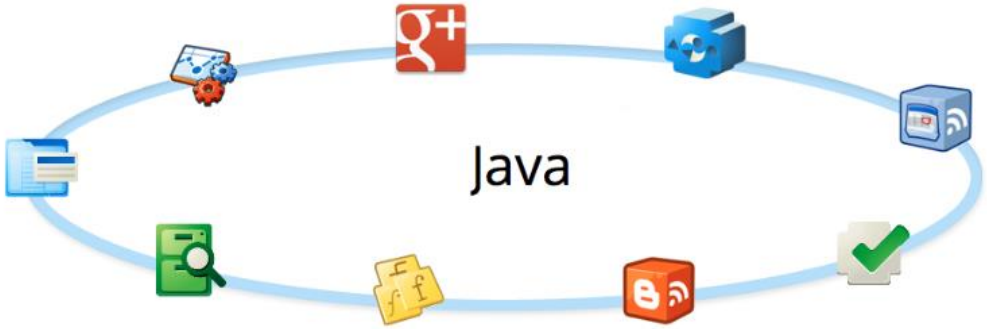
Iostream

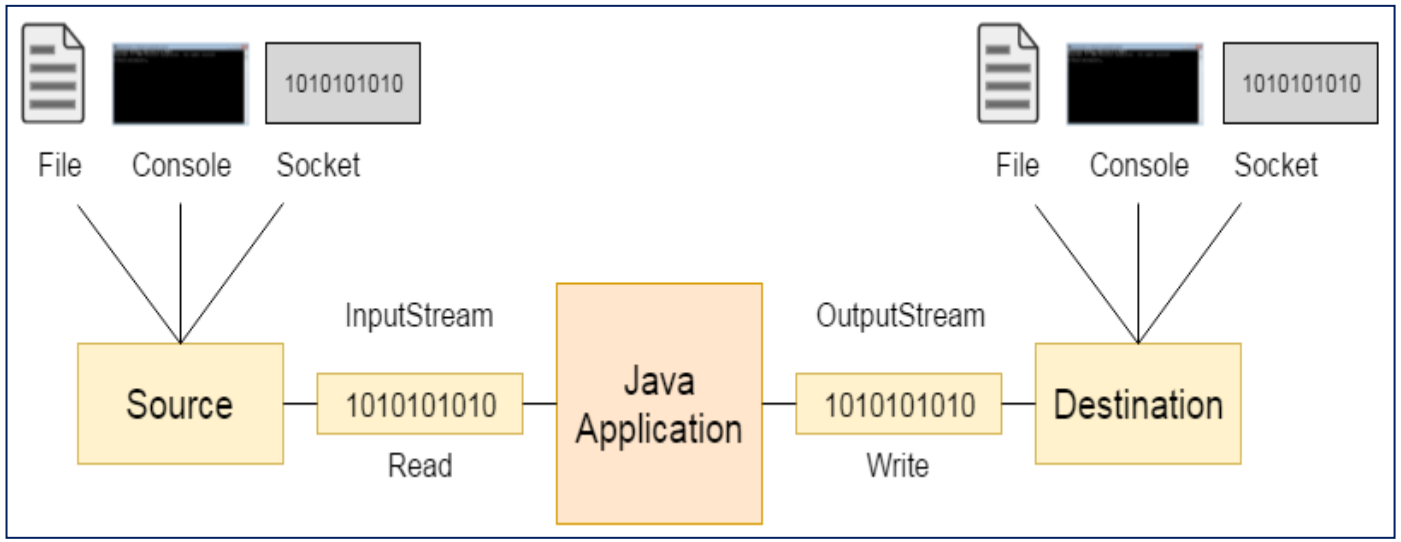
Iostream



Iostream

Streaming Data

Streaming Data



Iostream

Streaming Data VS. Random Access

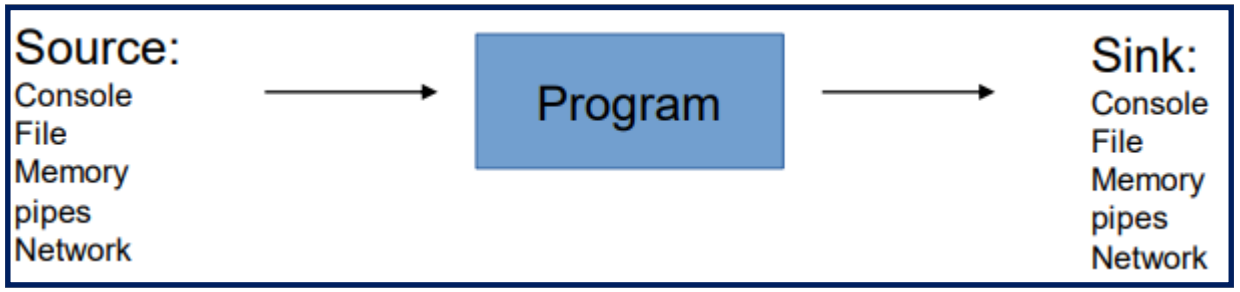
Streaming Data

- Streaming data – a stream is a sequence of data read or written, one item at a time
  - Blocking: java.io package
  - Non-blocking: java.nio package
- Random access – reading and writing data can be done in a non sequential order
  - java.io.RandomAccessFile

Iostream

Source & Sink

Streaming Data

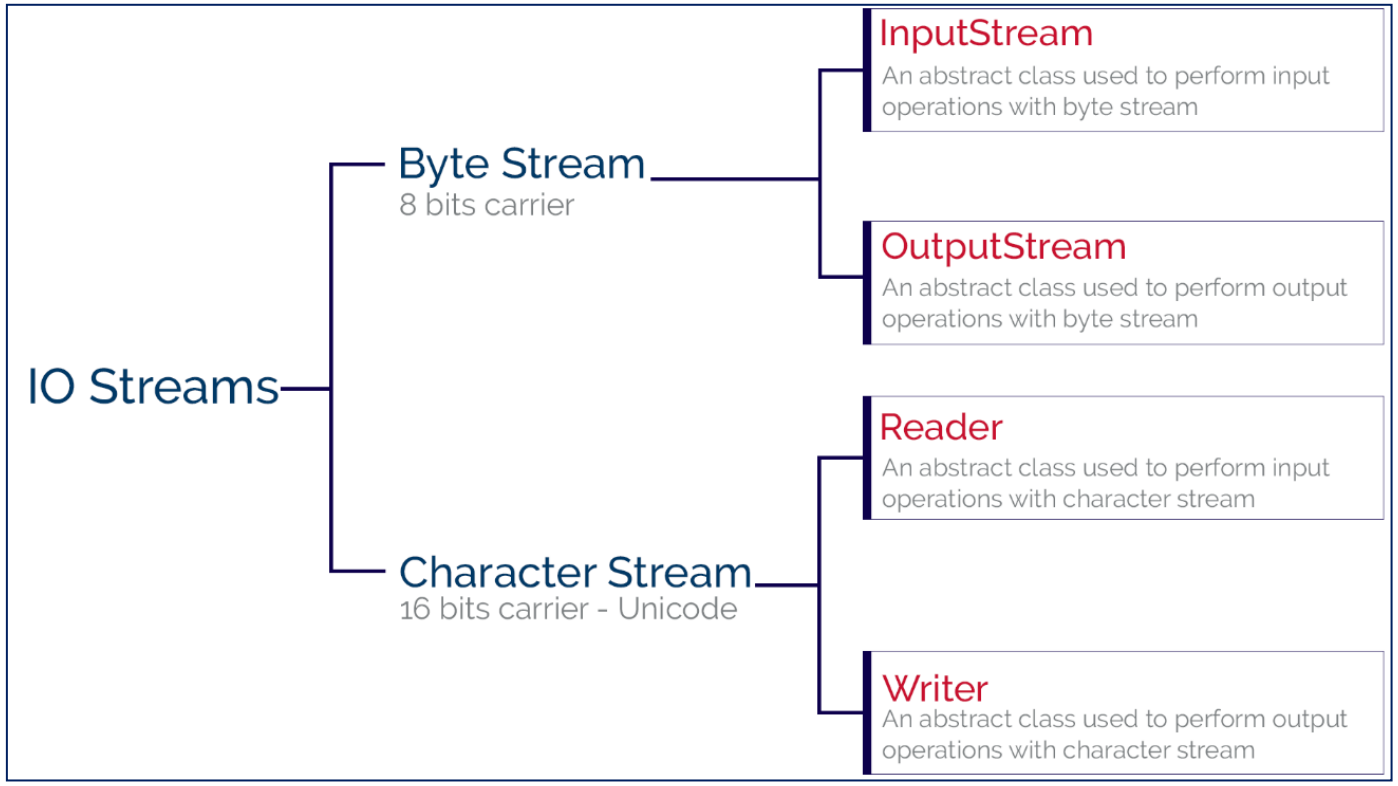


A stream can come to the program from a source or can be generated by the program to a sink. Sources and sinks are both node streams:

Iostream

Streaming Data

IO Stream



Iostream

Byte Stream VS. Character Stream

Streaming Data

The basic abstract classes for the different types of streams are:

Character streams:

- Reader
- Writer

Byte streams:

- InputStream
- OutputStream

Iostream

Streaming Data

InputStream VS. Reader

Feature	InputStream	Reader
Data Type	Byte-oriented (binary data)	Character-oriented (text data)
Superclass	InputStream	Reader
Read Method	read() (reads byte)	read() (reads character)
Buffering	BufferedInputStream	BufferedReader
File Handling	FileInputStream	FileReader
Common Use Cases	Binary files (e.g., images, audio)	Text files (e.g., plain text, JSON)
Conversion Bridging	N/A	InputStreamReader (converts byte streams to character streams)

Iostream

Streaming Data

Character Stream

Use :

When dealing with text-based data, such as plain text files, configuration files, or any file that contains human-readable characters.

Advantages:

- Character streams automatically handle character encoding and decoding, making them suitable for dealing with text in different character sets.
- They provide higher-level abstractions like Reader and Writer, which work with characters rather than bytes.

Example:

```
FileReader reader = new FileReader("textFile.txt");
FileWriter writer = new FileWriter("output.txt");
```

Iostream

Streaming Data

Byte Stream

Use :

When working with binary data, such as image files, audio files, or any non-text data.

Advantages:

- Byte streams are low-level and handle data in its raw form, making them suitable for binary data or when you need to work with the actual bytes of a file.

- They are more suitable for network operations where data is transmitted in binary form.

Example:

```
FileInputStream inputStream = new FileInputStream("binaryFile.bin");
FileOutputStream outputStream = new FileOutputStream("output.bin");
```

Iostream

Streaming Data

Byte Streaming Methods

Stream class	Description
BufferedInputStream	It is used for Buffered Input Stream.
DataInputStream	It contains method for reading java standard datatypes.
FileInputStream	This is used to reads from a file
InputStream	This is an abstract class that describes stream input.
PrintStream	This contains the most used print() and println() method
BufferedOutputStream	This is used for Buffered Output Stream.
DataOutputStream	This contains method for writing java standard data types.
FileOutputStream	This is used to write to a file.
OutputStream	This is an abstract class that describe stream output.

Iostream

Streaming Data

Character Stream methods

Stream class	Description
BufferedReader	It is used to handle buffered input stream.
FileReader	This is an input stream that reads from file.
InputStreamReader	This input stream is used to translate byte to character.
OutputStreamReader	This output stream is used to translate character to byte.
Reader	This is an abstract class that define character stream input.
PrintWriter	This contains the most used print() and println() method
Writer	This is an abstract class that define character stream output.
BufferedWriter	This is used to handle buffered output stream.
FileWriter	This is used to output stream that writes to file.

Iostream

IO endpoints

Streaming Data

Main IO endpoints (sources and sinks):

File System:

- FileReader / FileWriter / FileInputStream / FileOutputStream

Networking:

- Socket / ServerSocket

Threads communication (pipes):

- PipedInputStream / PipedOutputStream

Iostream

Streams

Streaming Data

Three default accessed streams

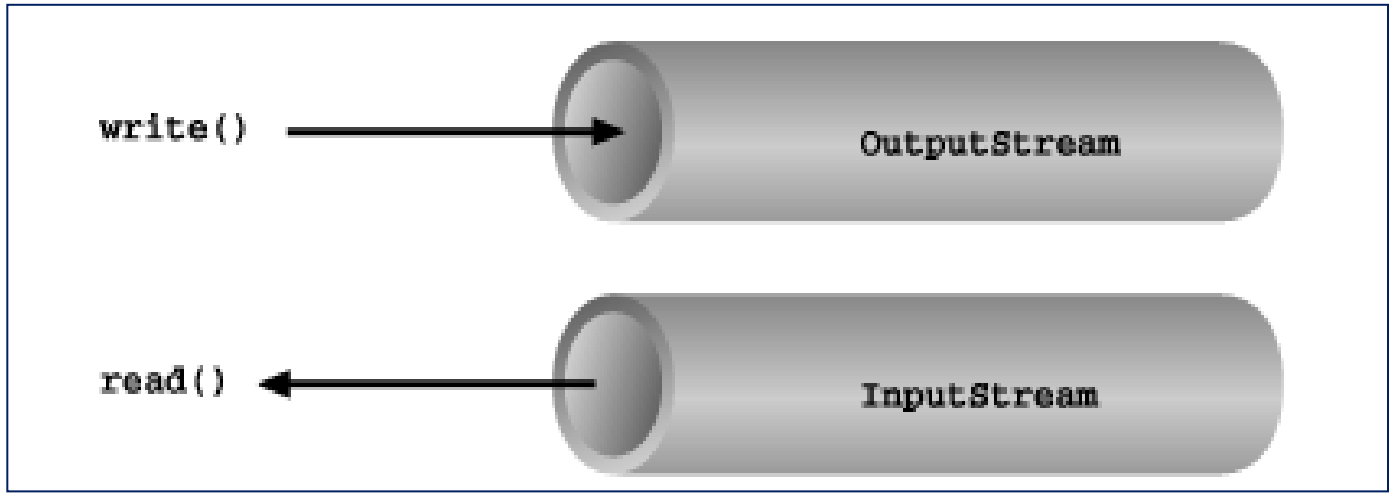
- System.in – InputStream
- System.out - PrintStream (OutputStream)
- System.err - PrintStream

Iostream

IO Stream Func

Basic input and output stream functionality

Streaming Data



Iostream

InputStream methods

Streaming Data

- int available()
- int read()
- int read(byte[] b)
- int read(byte[] b, int off, int len)
- long skip(long n)

- void close()
- void mark(int readlimit)
- boolean markSupported()
- void reset()

Iostream

Streaming Data

OutputStream methods

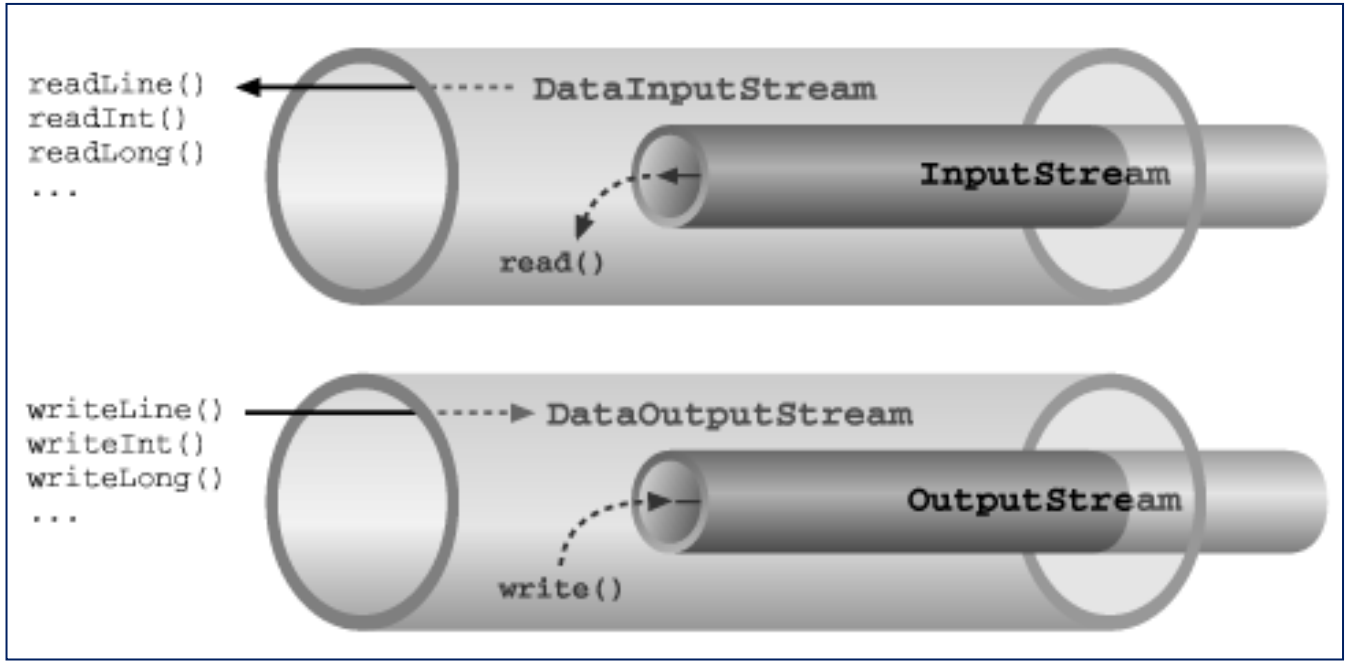
- void write(byte[] b)
- void write(byte[] b, int off, int len)
- void write(int b)
- void flush()
- void close()

Iostream

IOStream Func

Basic input and output stream functionality

Streaming Data



Iostream

Filter Streams Decorators

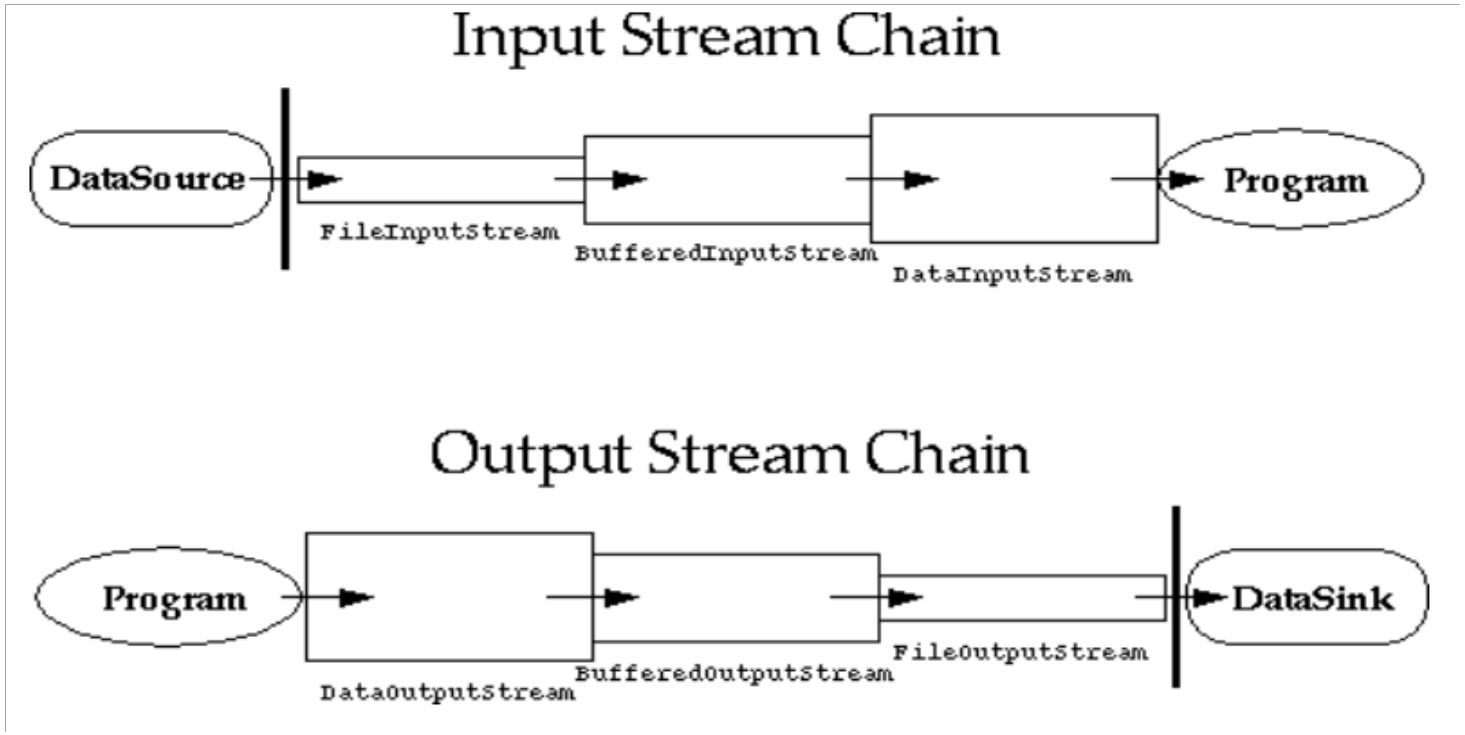
Stream Decoration

1. Stream created over another stream
2. Subclasses of:
   - FilterInputStream
   - FilterOutputStream
   - FilterReader
   - FilterWriter
3. used frequently to decorate IO into:
   - Text
   - Primitives
   - Objects

Iostream

Chaining Decorators

Stream Decoration



Iostream

Switch Stream Type

Switch between binary <-> text IO

Stream Decoration

InputStreamReader:
Converts bytes read from an InputStream into characters using a specified charset.

OutputStreamReader:
Converts characters to bytes according to the character coding before writing

Iostream

Text Decorators

Stream Decoration

BufferedReader / BufferedWriter

Provides Line based reading and writing:

- Reading:

BufferedReader.readLine() returns String (or null on end of file).

- Writing:

BufferedWriter.write(String line) & BufferedWriter.newline()

Iostream

Stream Decoration

Primitives Decorators

DataInputStream / DataOutputStream

Provides primitive based reading and writing:

Reading:

- readUTF() : String
- readInt() : int
- readBoolean() : boolean ....

Writing:

- writeUTF(String)
- writeInt(int)
- writeBoolean(boolean)

Iostream

Stream Decoration

Object Decorators

ObjectInputStream / ObjectOutputStream

All objects must implement Serializable
Provides Objects based reading and writing:

Reading:

- readObject() : Object

Writing:

- writeObject(Object)

Iostream

Serialization

Serialization

- Only the object's data are serialized
- Data marked with the transient keyword are not serialized.
- Serialization stores the state of an object to a file
- Serializable objects can be stored and loaded from object streams:
  - ObjectInputStream
  - ObjectOutputStream

Iostream

Random Access

Random AccessFile

Constructors:

- RandomAccessFile(File file, String mode)
- RandomAccessFile(String name, String mode)

Mode is usually ”r” or ”rw” .

file pointer can be read and positioned:

- int skipBytes(int)
- void seek(long)
- long getFilePointer()

Iostream

Stream Tokenizer

StreamTokenizer

A class reads the stream character by character.

Each of them can have zero or more of the following attributes:

white space, alphabetic, numeric, string quote or comment character.

types of characters:

- Word characters: ranges like ‘a’ to ‘z’ and ‘A’ to ‘Z
- Numeric characters: 0,1,...,9
- Whitespace characters: ASCII values from 0 to 32
- Comment character: /
- String quote characters: ‘ and “

Iostream

Thank You !!