IO in Java

Byte Streams

- byte streams performs input and output of 8-bit bytes
- byte stream classes are descended from InputStream and OutputStream
- files are one type of type streams

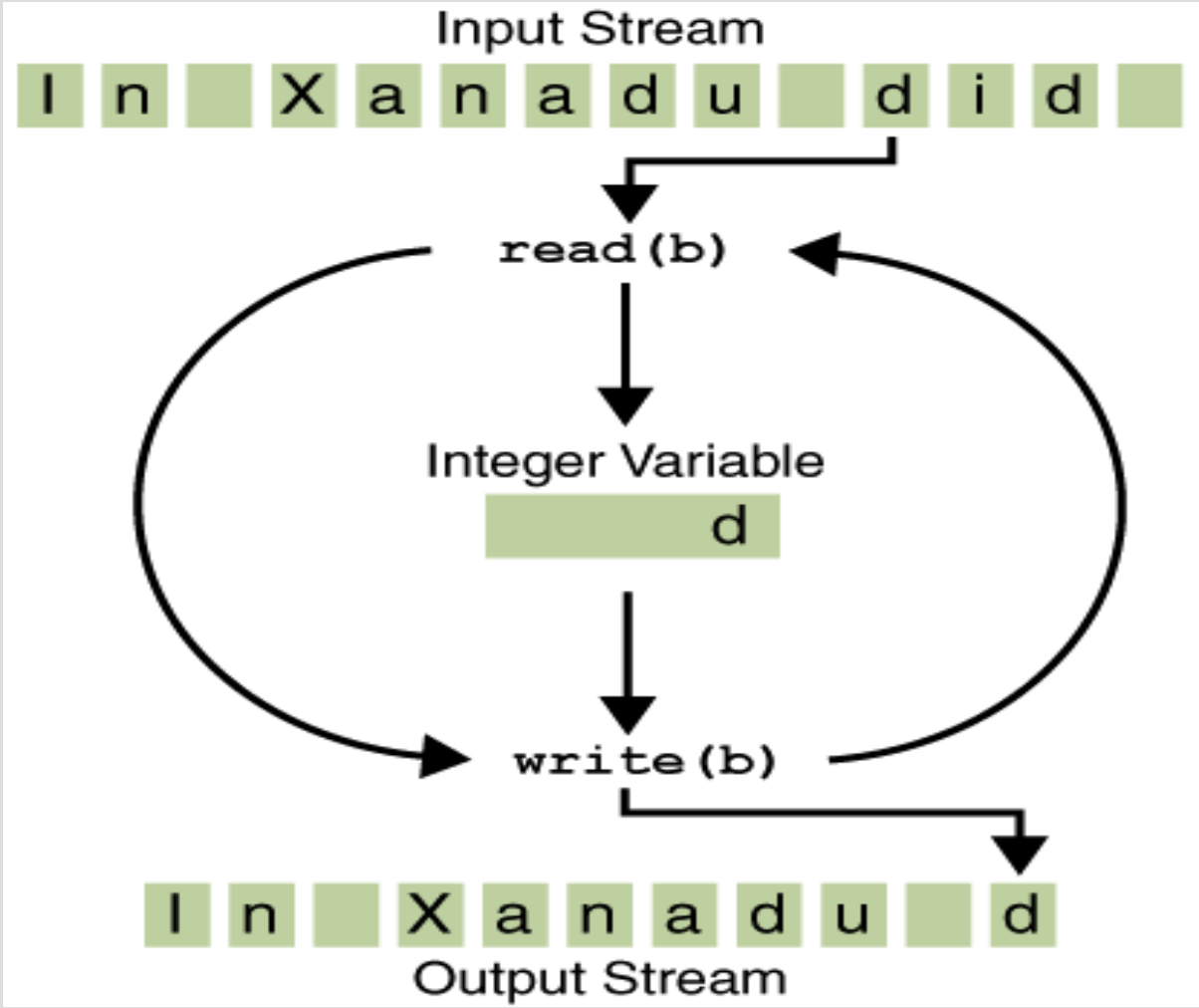
File IO

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class CopyBytes {
    public static void main(String[] args) throws IOException {
        FileInputStream in = null;
        FileOutputStream out = null;
        try {
            in = new FileInputStream("xanadu.txt");
            out = new FileOutputStream("outagain.txt");
            int c;

            while ((c = in.read()) != -1) {
                out.write(c);
            }
```

File IO

```
        } finally {
            if (in != null) {
                in.close();
            }
            if (out != null) {
                out.close();
            }
        }
    }
}
```

File IO


Input Stream
I n X a n a d u d i d
read(b)
Integer Variable
d
write(b)
I n X a n a d u d
Output Stream

Buffered Streams

unbuffered I/O handle read and write directly by underlying OS:
such request often triggers disk access, network activity ...

buffered I/O reads data from memory and writes first to memory:
most likely do read/ write when buffer is empty/full

Buffered Streams

```
inputStream =
    new BufferedReader(new FileReader("xanadu.txt"));
outputStream =
    new BufferedWriter(new FileWriter("characteroutput.txt"));
```

Command Line I/O

Two ways: Standard Streams and Console

Standard Stream:
- Feature of many OS
- By default, I/O to the display
- Java supports three Standard Streams:

Standard Input, Standard Output, Standard Error

Command Line I/O

Two ways: Standard Streams and Console

Console:
- More advanced alternative
- a single, predefined object of type Console

Get Number Input

```
package numinput;
import java.io.*;

public class Main {

    public static void main(String[] args) throws IOException{
        String x;
        Float f;
        BufferedReader stdin = new BufferedReader( new InputStreamReader( System.in ) );
        try{
            x = stdin.readLine();
            f = Float.parseFloat(x);
        }finally{
            stdin.close();
        }
    }

}
```

Reference

Sun' Java Tutorial
http://java.sun.com/docs/books/tutorial/