Object-Oriented Programming

# Files I/O

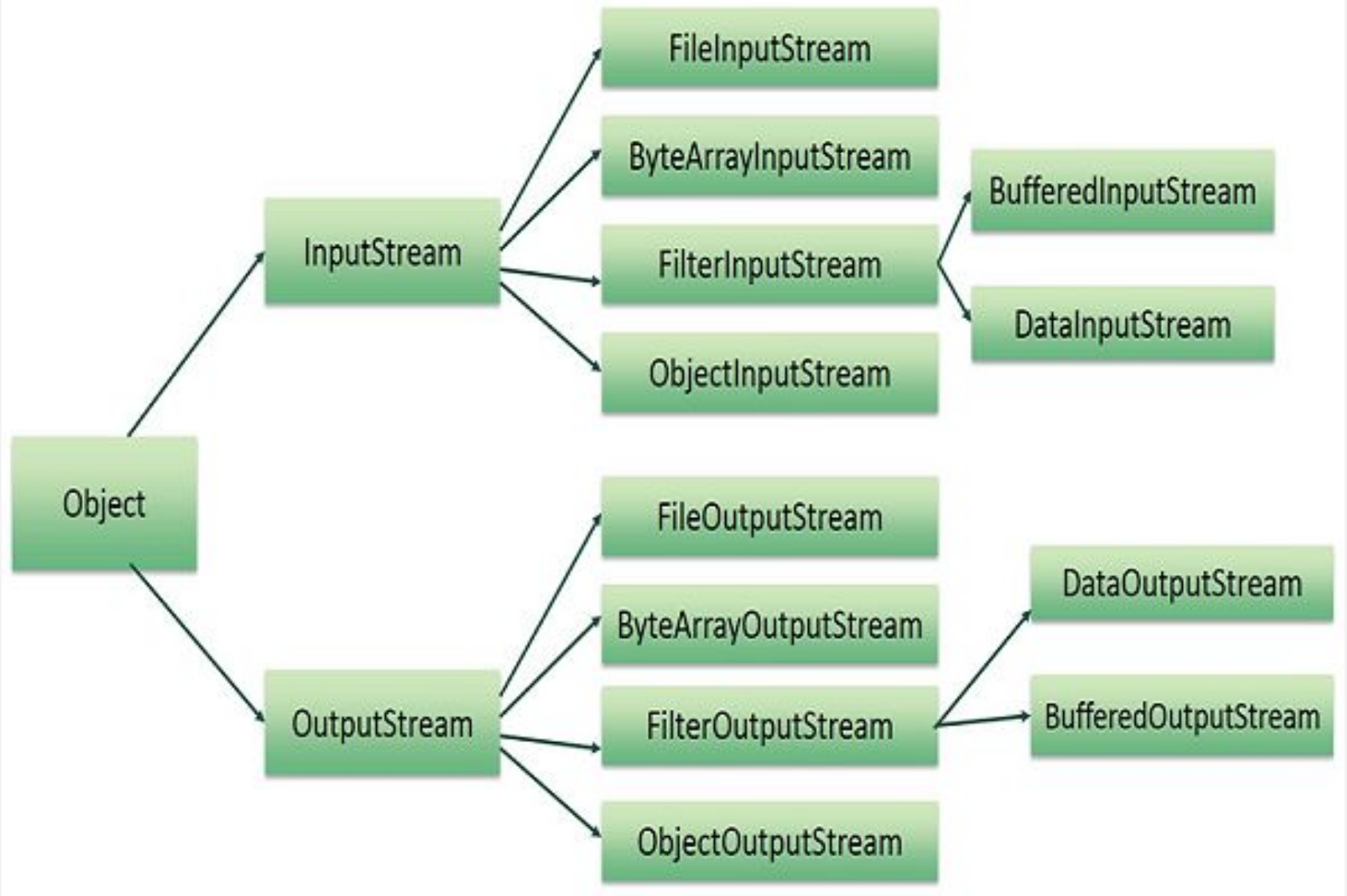
# Outline

- Stream
  - InputStream
  - OutputStream

# What is a stream?

# Stream

- A sequence of data
- In Java, a stream is composed of bytes
- Two types:
  - InputStream - used to read data from a source
  - OutputStream - used to write data to a destination


Object
InputStream
FileInputStream
ByteArrayInputStream
FilterInputStream
ObjectInputStream
BufferedInputStream
DataInputStream
OutputStream
FileOutputStream
ByteArrayOutputStream
FilterOutputStream
ObjectOutputStream
DataOutputStream
BufferedOutputStream

# Input

# FileInputStream

- Part of the java.io package
- Reads bytes of data from a file

```
FileInputStream fis = new FileInputStream ("source.txt");
int j;
while ((j = fis.read()) != -1)
    System.out.print ((char) j);
fis.close ();
```

*Take note: this block of code requires exception handling*

# BufferedReader

- Part of the java.io package
- Extends class Reader
- Reads text from character-based input stream

```
BufferedReader br = new BufferedReader (
                        new FileReader ("source.txt"));
System.out.println (br.readLine ());
br.close ();
```

*Take note: this block of code requires exception handling*

# Questions?

# Output

# FileOutputStream

- Part of the java.io package
- Writes bytes of data to a file

```
FileOutputStream fout = new FileOutputStream ("dest.txt");
fout.write (100);
String strSchool = "DLSU";

byte[] b = strSchool.getBytes (); // converts to byte array
fout.write (b);

fout.close ();
```

*Take note: this block of code requires exception handling*

# BufferedWriter

- Part of the java.io package
- Extends class Reader
- Writes text to a character-based output stream

```
BufferedWriter bw = new BufferedWriter (
                        new FileWriter ("dest.txt"));
bw.write ("Save data here");
bw.close ();
```

*Take note: this block of code requires exception handling*

# Questions?

# Next meeting…

- SOLID Principles
- Last graded exercise
- Some MP reminders

Keep learning…