Application Program Development

APD545

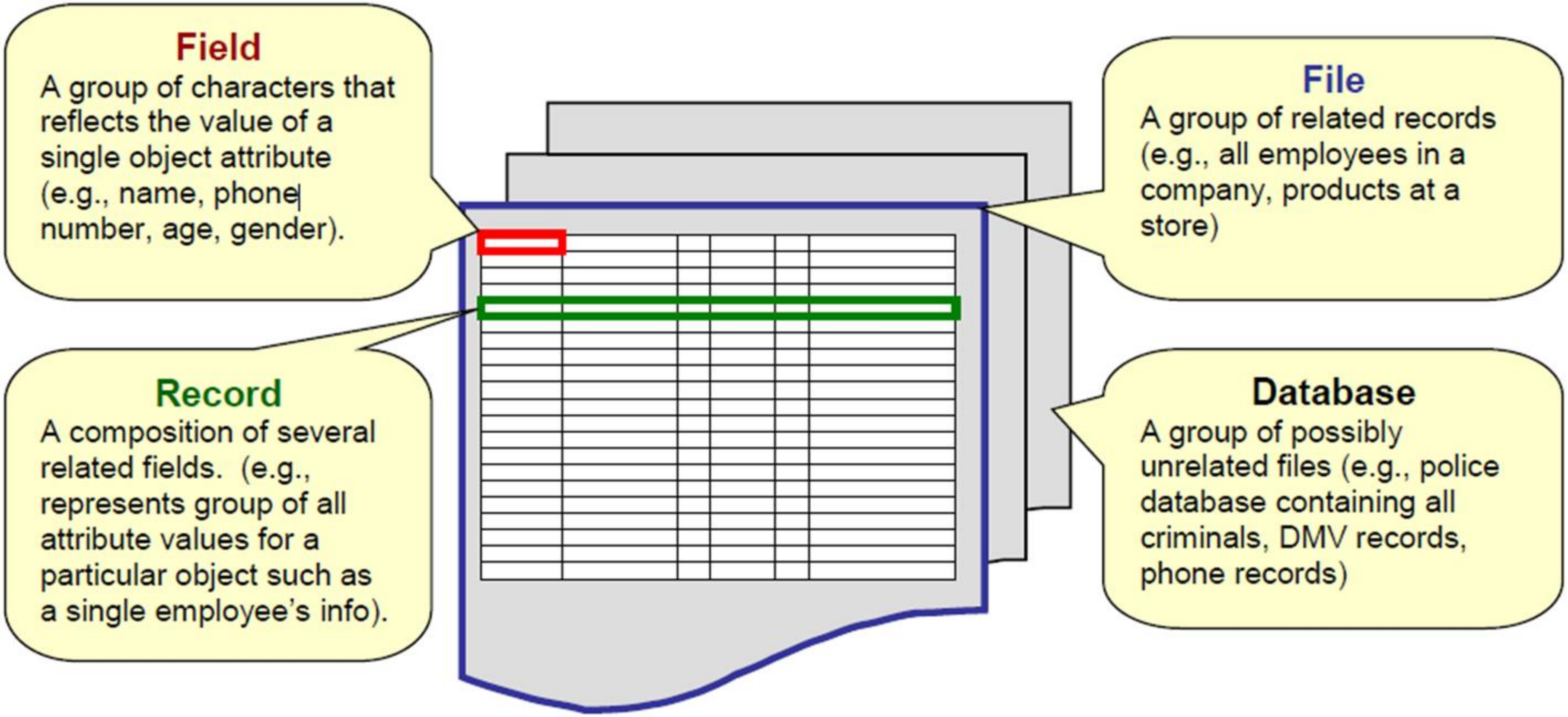
Instructor: Maryam Sepehrinour
Email: Maryam.Sepehrinour@SenecaPolytechnic.ca

Seneca

Outcomes

- ✓ Java Files and Streams

Files



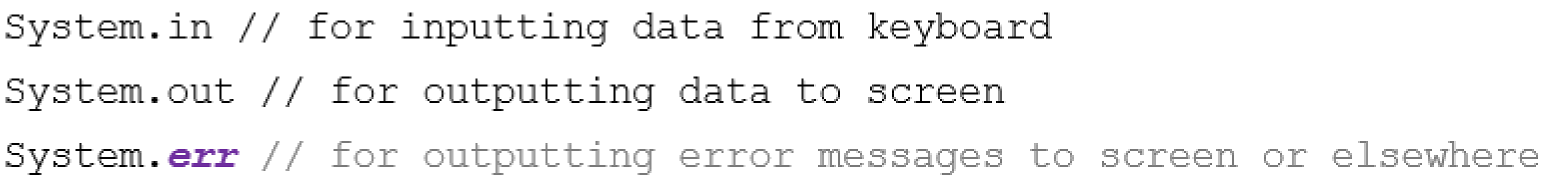
Field
A group of characters that reflects the value of a single object attribute (e.g., name, phone number, age, gender).

Record
A composition of several related fields. (e.g., represents group of all attribute values for a particular object such as a single employee's info).

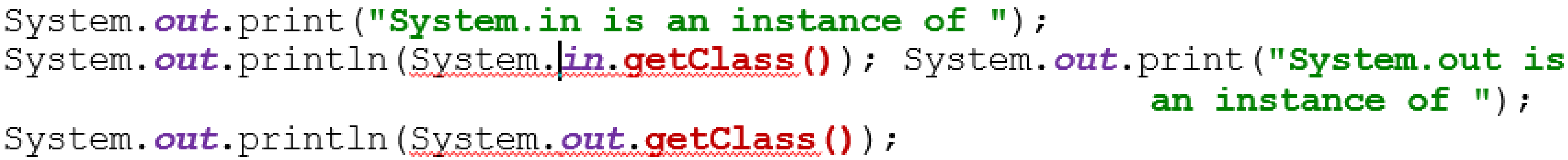
File
A group of related records (e.g., all employees in a company, products at a store)

Database
A group of possibly unrelated files (e.g., police database containing all criminals, DMV records, phone records)

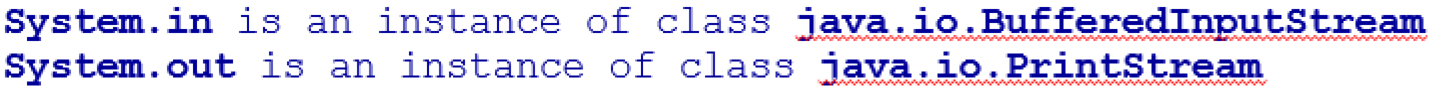
I/O



```
System.in // for inputting data from keyboard
System.out // for outputting data to screen
System.err // for outputting error messages to screen or elsewhere
```

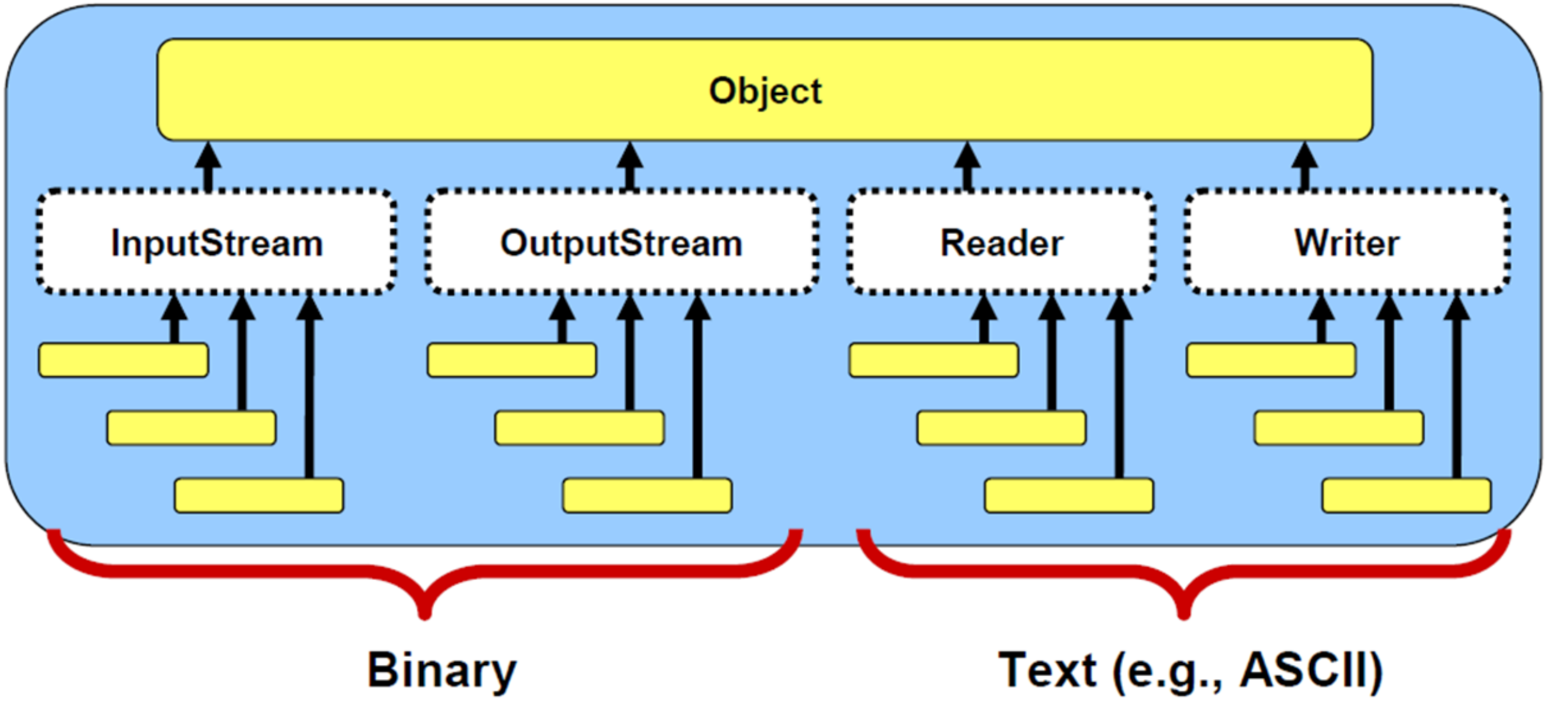


```
System.out.print("System.in is an instance of ");
System.out.println(System.in.getClass()); System.out.print("System.out is
                                                              an instance of ");
System.out.println(System.out.getClass());
```



```
System.in is an instance of class java.io.BufferedInputStream
System.out is an instance of class java.io.PrintStream
```

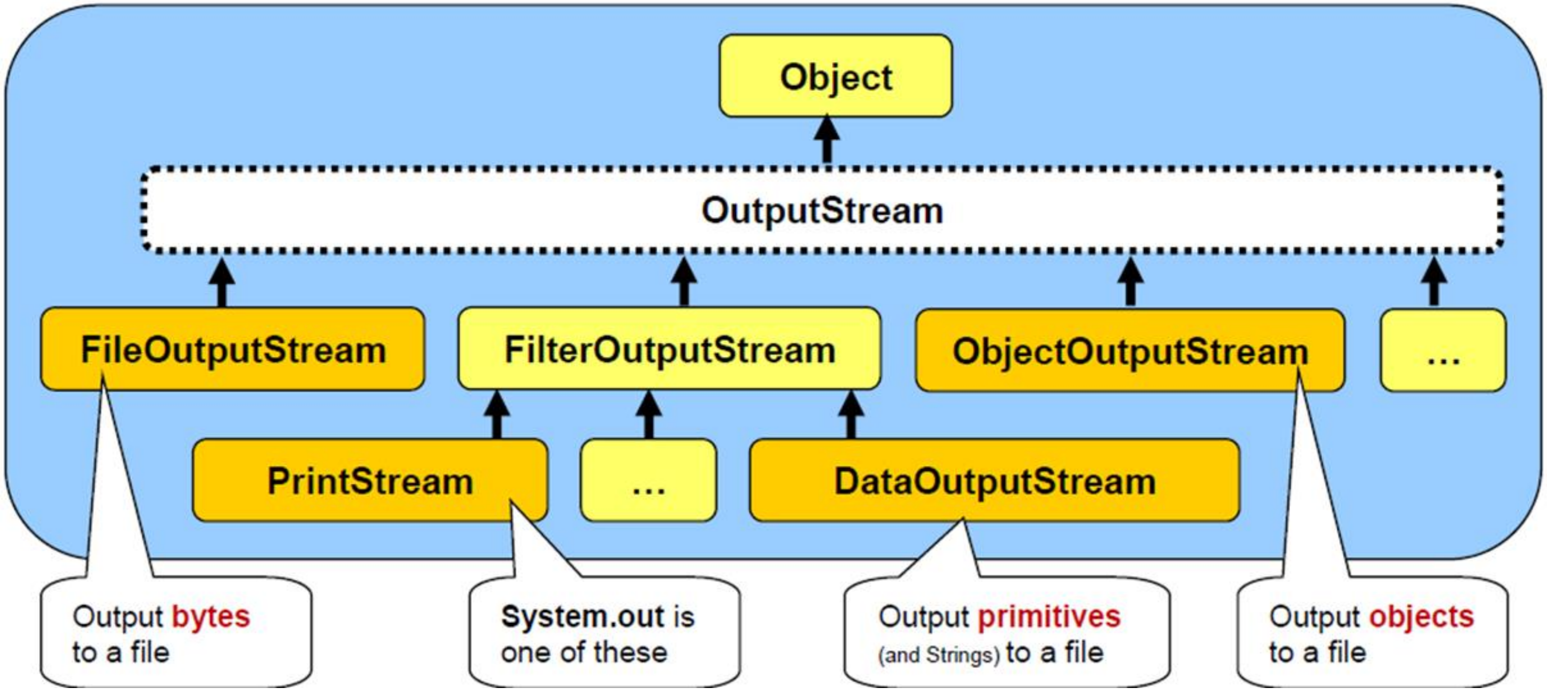
Streams


Object
InputStream
OutputStream
Reader
Writer
Binary
Text (e.g., ASCII)

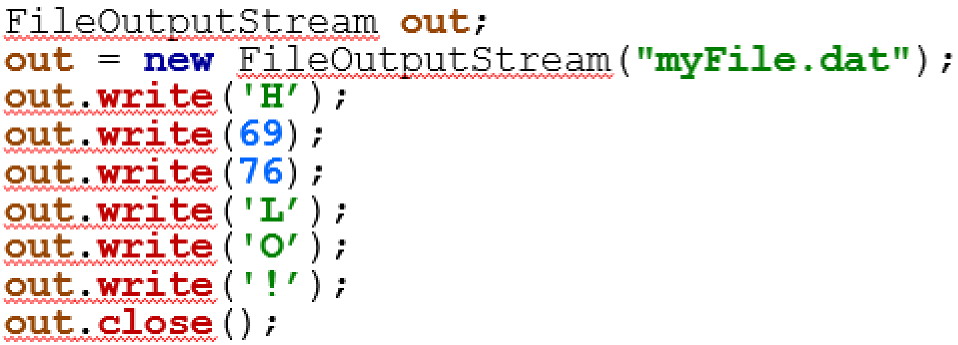
Streams

Byte Stream	**Character Stream**
Deal with “Raw Data”	Deal with “Character Data”
Byte by Byte (1 Byte -- 8-bits)	Character by Character (1 char – 16 bits)
Coder responsibility to convert bytes to character	No worries for the coder
Does not always handle Unicode correctly	Handle Unicode appropriately

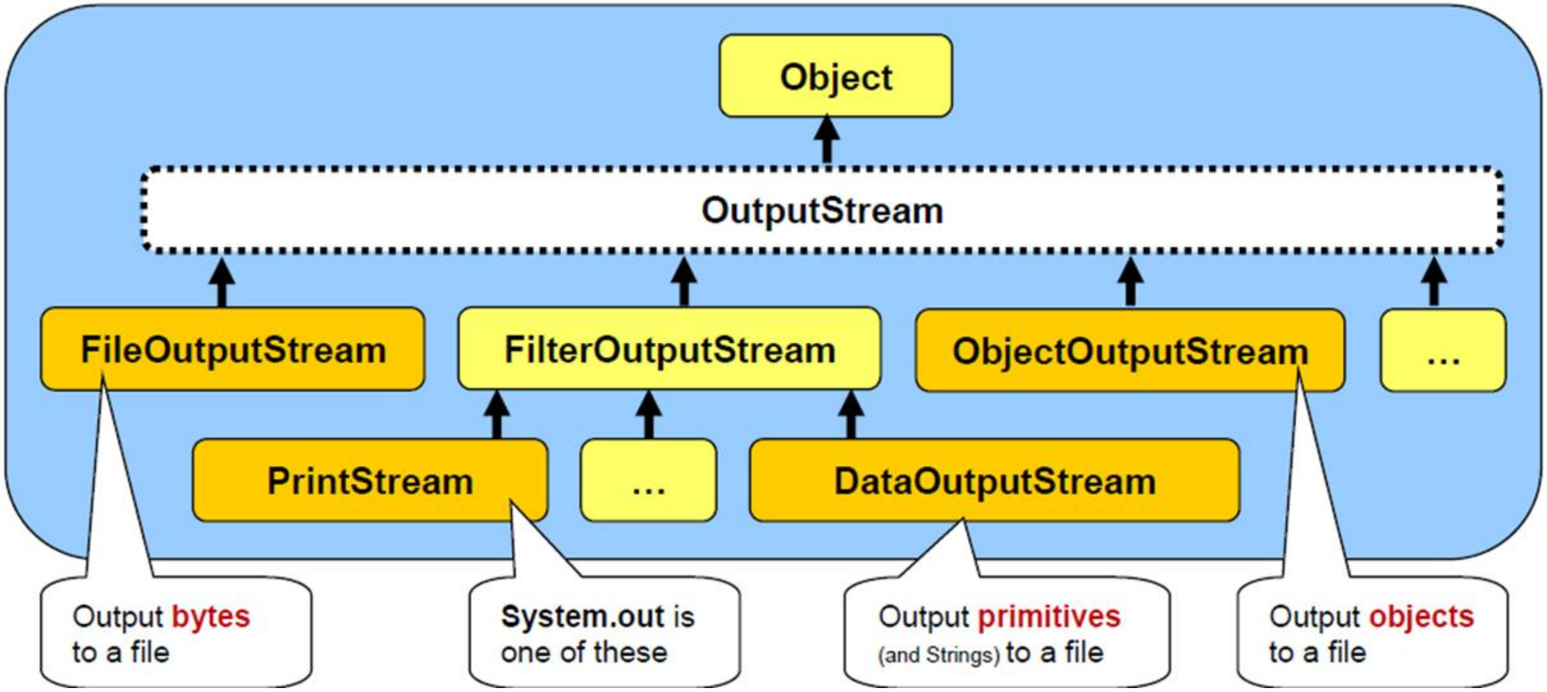
Streams


Object
OutputStream
FileOutputStream
FilterOutputStream
ObjectOutputStream
…
PrintStream
…
DataOutputStream
Output bytes to a file
System.out is one of these
Output primitives (and Strings) to a file
Output objects to a file

Streams



```
FileOutputStream out;
out = new FileOutputStream("myFile.dat");
out.write('H’);
out.write(69);
out.write(76);
out.write('L’);
out.write('O’);
out.write('!’);
out.close();
```



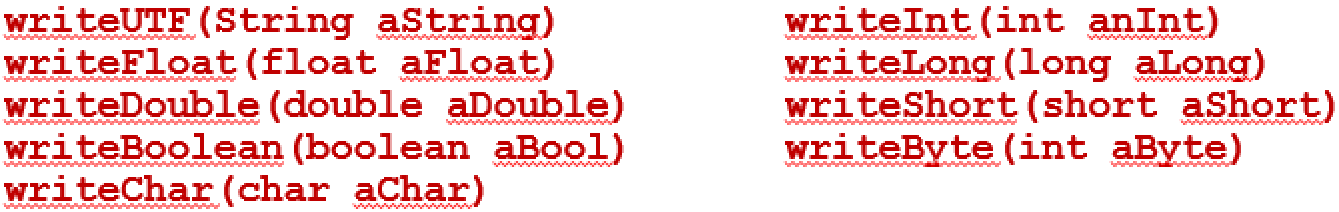
Output bytes to a file

System.out is one of these

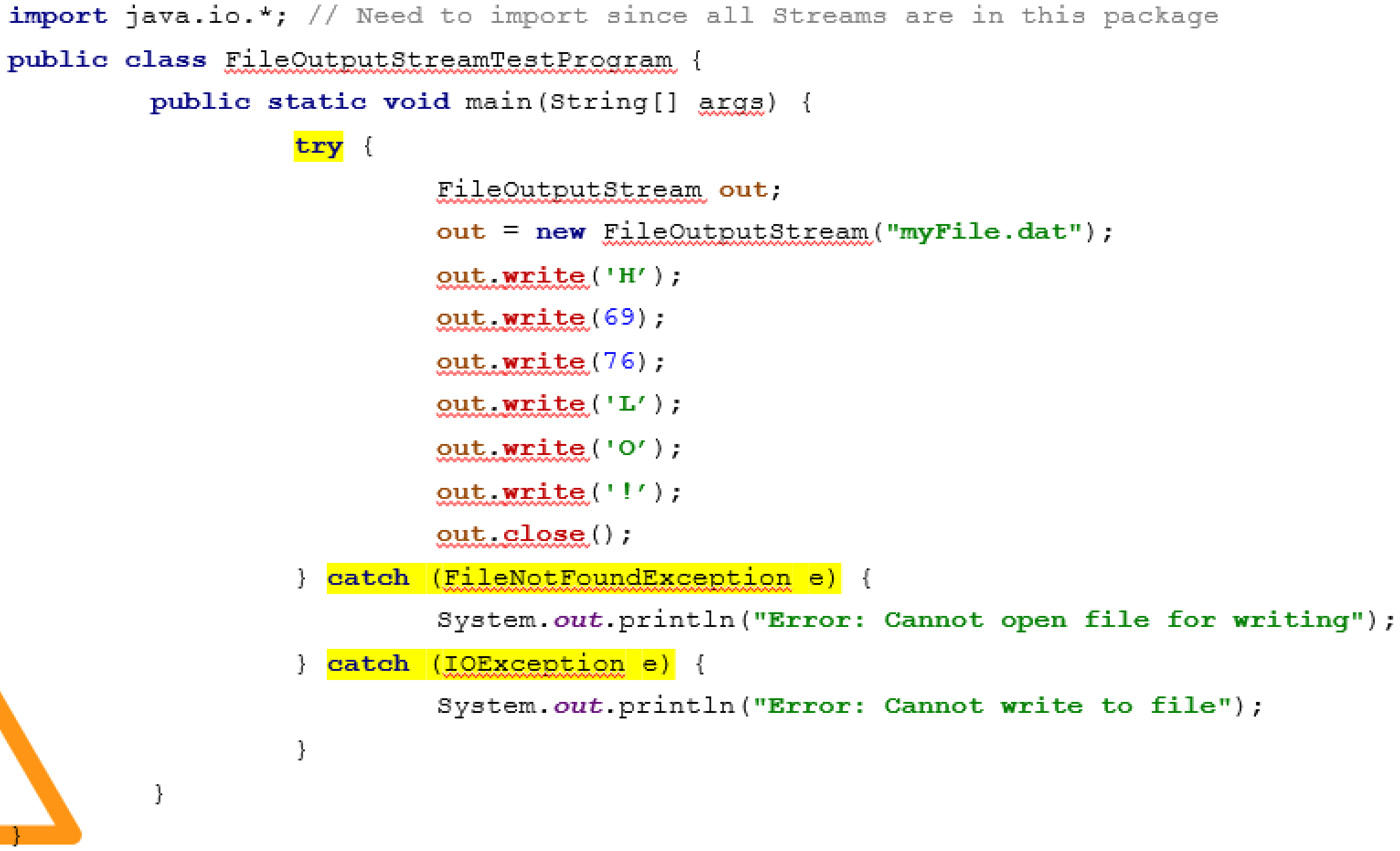
Output primitives (and Strings) to a file

Output objects to a file

Streams

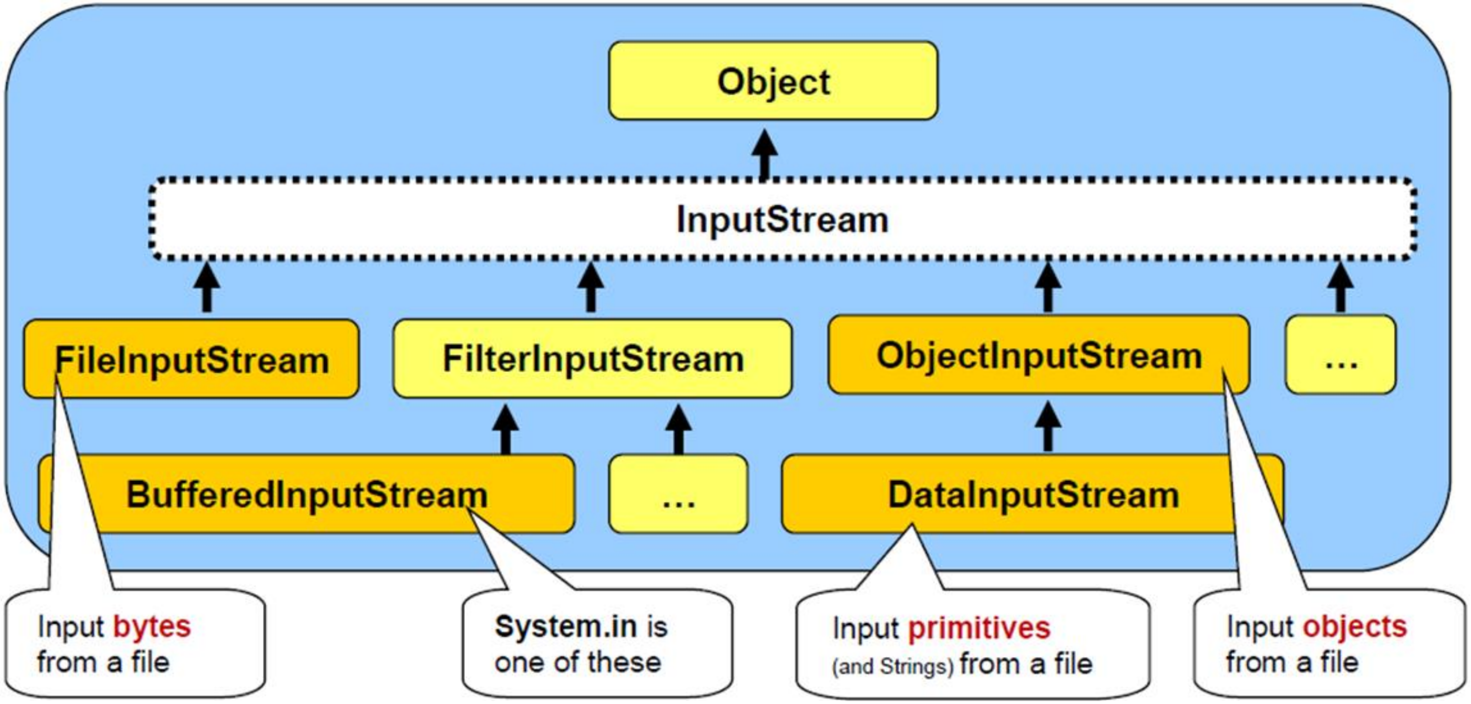


```
writeUTF(String aString)          writeInt(int anInt)
writeFloat(float aFloat)          writeLong(long aLong)
writeDouble(double aDouble)       writeShort(short aShort)
writeBoolean(boolean aBool)       writeByte(int aByte)
writeChar(char aChar)
```

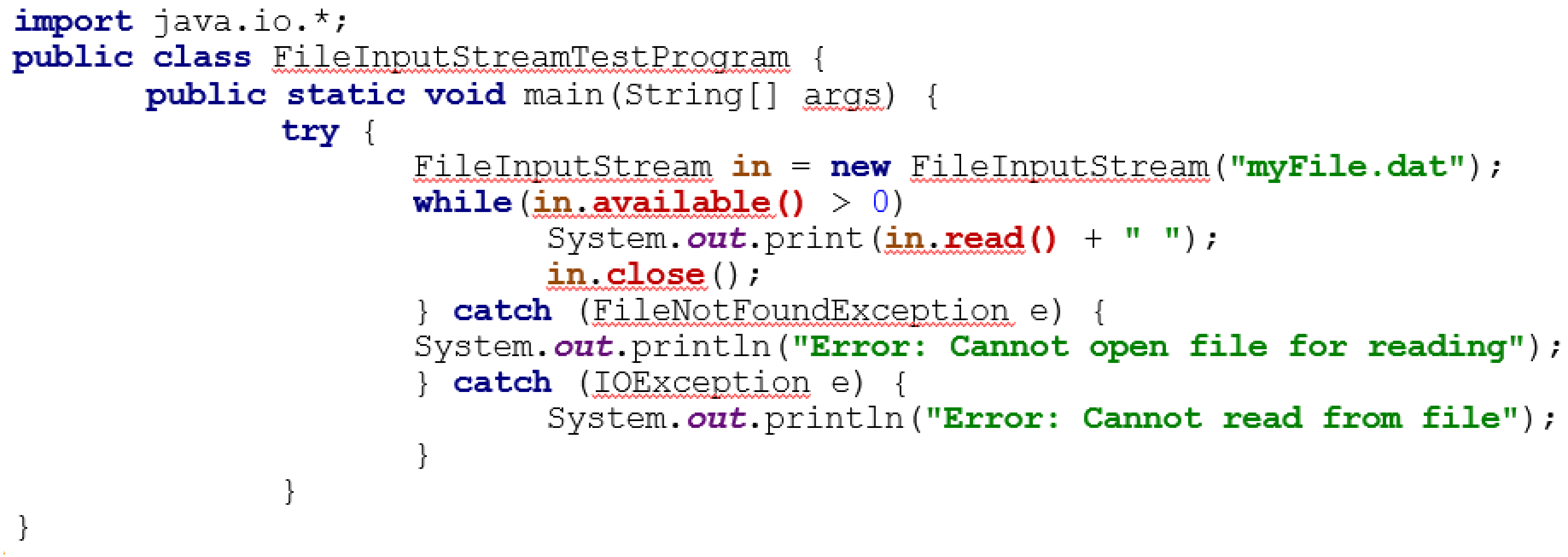


```
import java.io.*; // Need to import since all Streams are in this package
public class FileOutputStreamTestProgram {
        public static void main(String[] args) {
                try {
                        FileOutputStream out;
                        out = new FileOutputStream("myFile.dat");
                        out.write('H’);
                        out.write(69);
                        out.write(76);
                        out.write('L’);
                        out.write('O’);
                        out.write('!’);
                        out.close();
                } catch (FileNotFoundException e) {
                        System.out.println("Error: Cannot open file for writing");
                } catch (IOException e) {
                        System.out.println("Error: Cannot write to file");
                }
        }
}
```

Streams

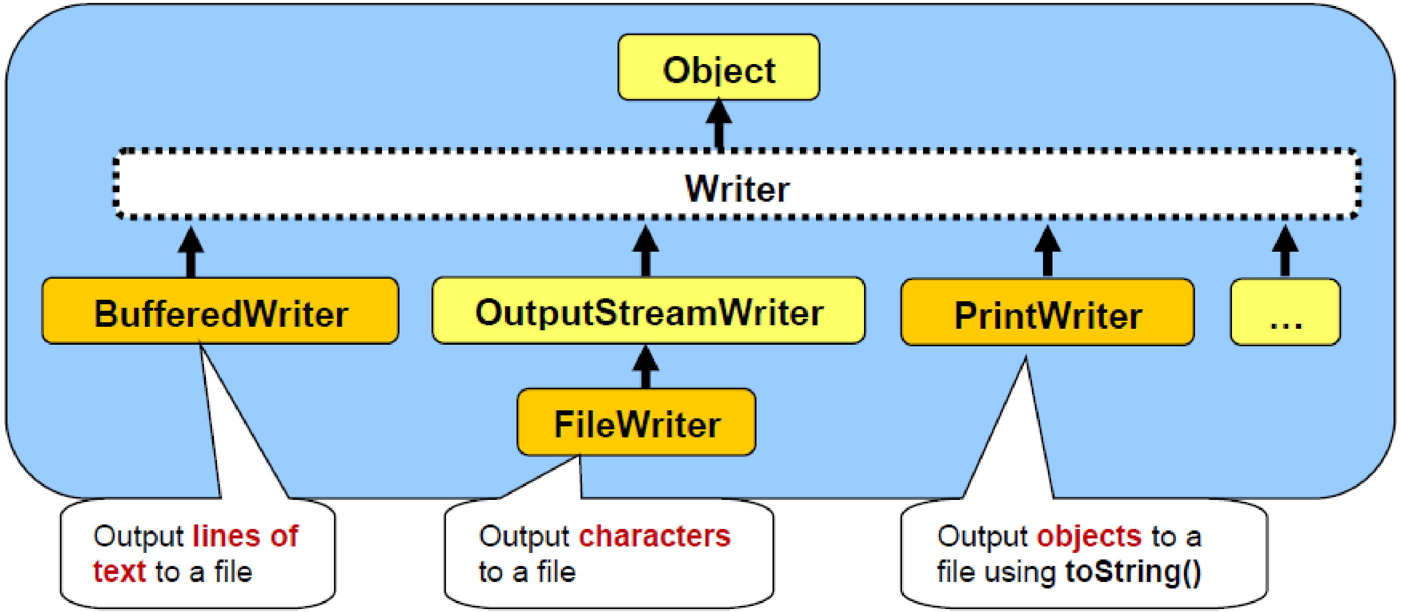

Object
InputStream
FileInputStream
FilterInputStream
ObjectInputStream
…
BufferedInputStream
…
DataInputStream
Input bytes from a file
System.in is one of these
Input primitives (and Strings) from a file
Input objects from a file

Streams



```
import java.io.*;
public class FileInputStreamTestProgram {
       public static void main(String[] args) {
              try {
                     FileInputStream in = new FileInputStream("myFile.dat");
                     while(in.available() > 0)
                            System.out.print(in.read() + " ");
                            in.close();
                     } catch (FileNotFoundException e) {
                     System.out.println("Error: Cannot open file for reading");
                     } catch (IOException e) {
                            System.out.println("Error: Cannot read from file");
                     }
              }
}
```

Streams

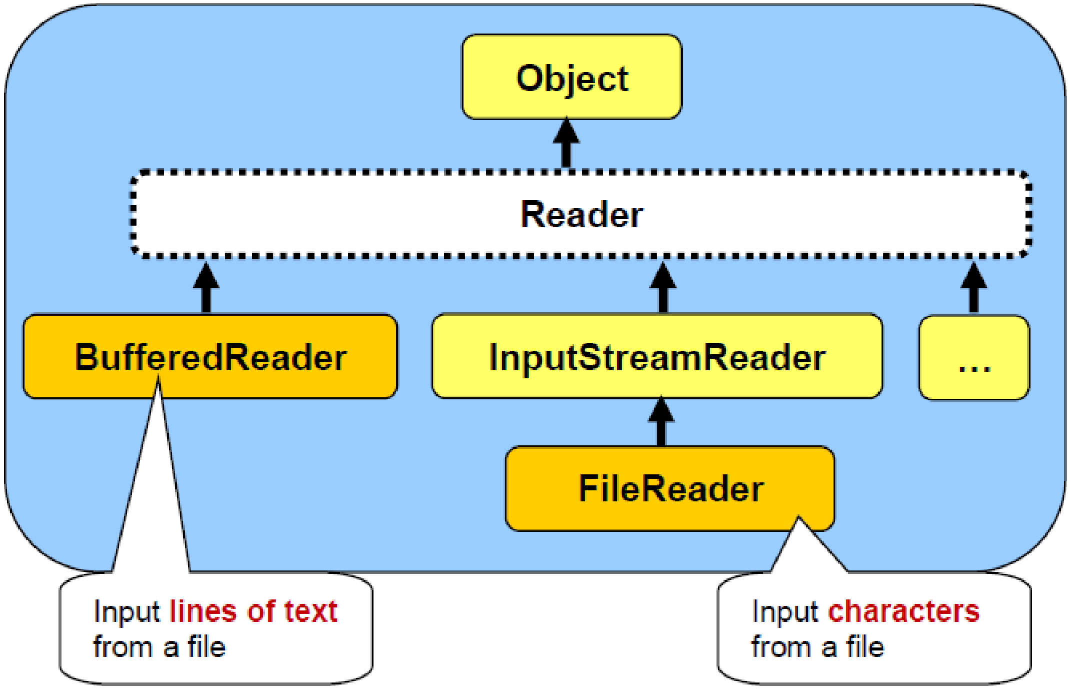


Output lines of text to a file

Output characters to a file

Output objects to a file using toString()

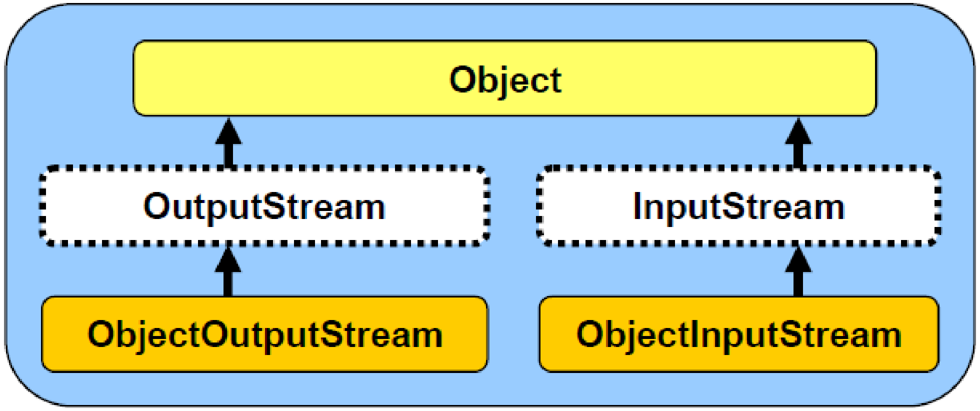
Streams


Object
Reader
BufferedReader
InputStreamReader
…
FileReader
Input lines of text from a file
Input characters from a file

Streams & Error Handling!

```
FileInputStream in = null;
try{
      in = new FileInputStream(filename); //open stream
      // read data
      }catch (FileNotFoundException ){
            …
      } finally{
            try{
                  if(n != null)
                        in.close();
                  }catch (IOException e){ …}
      }
```

Streams
Reading/Writing Whole Objects



- *Serialization is the process of breaking down an object into bytes.*

- In order to be able to save an object to a file using the **ObjectOutputStream**, the object must be ***serializable*** (i.e., able to be serialized…or reduced to a set of bytes).

```
public class BankAccount implements java.io.Serializable {
...
}
```

Thank you!