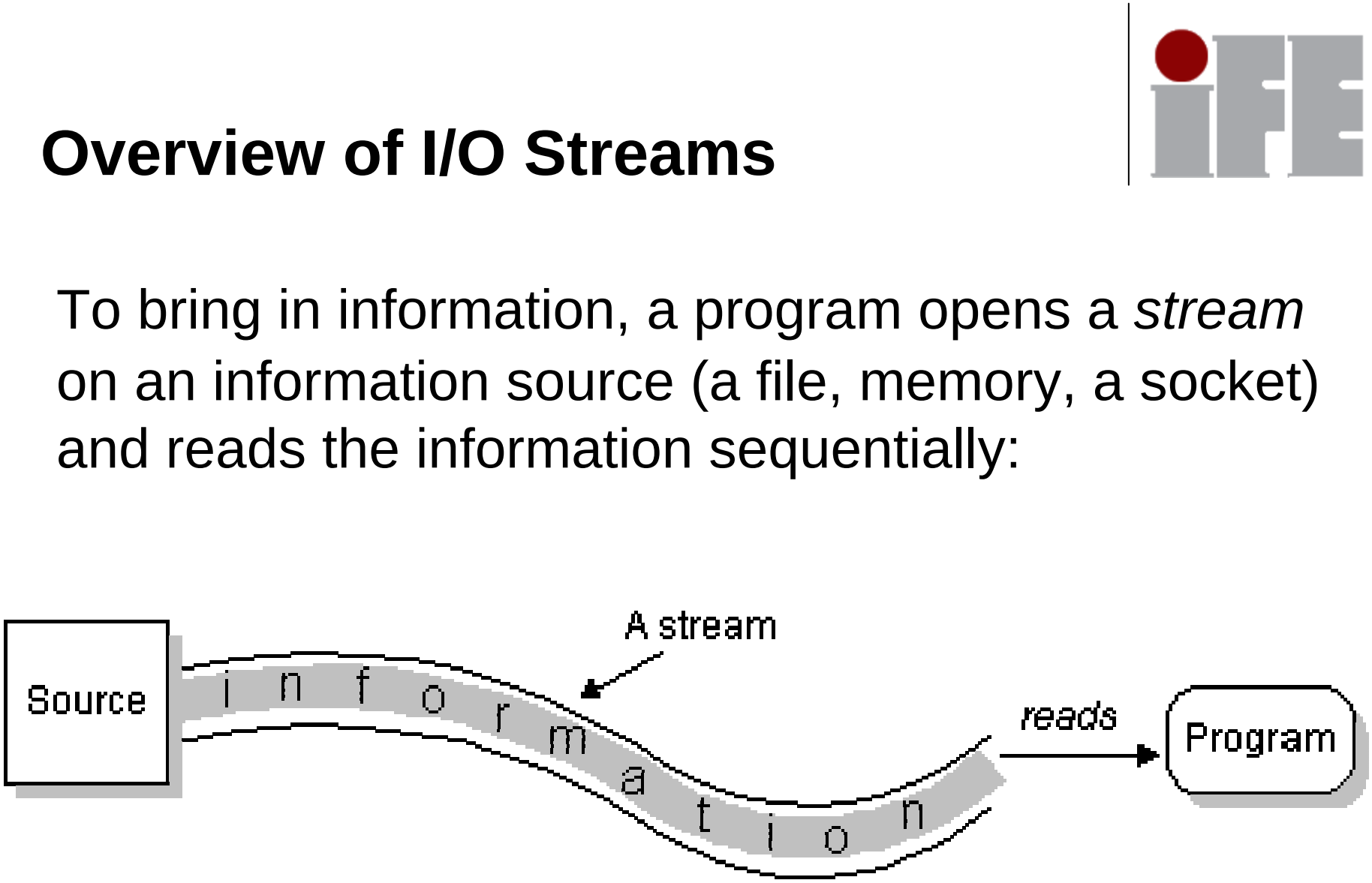
Fundamentals of Programming

Laboratory 10

OOP & I/O Streams

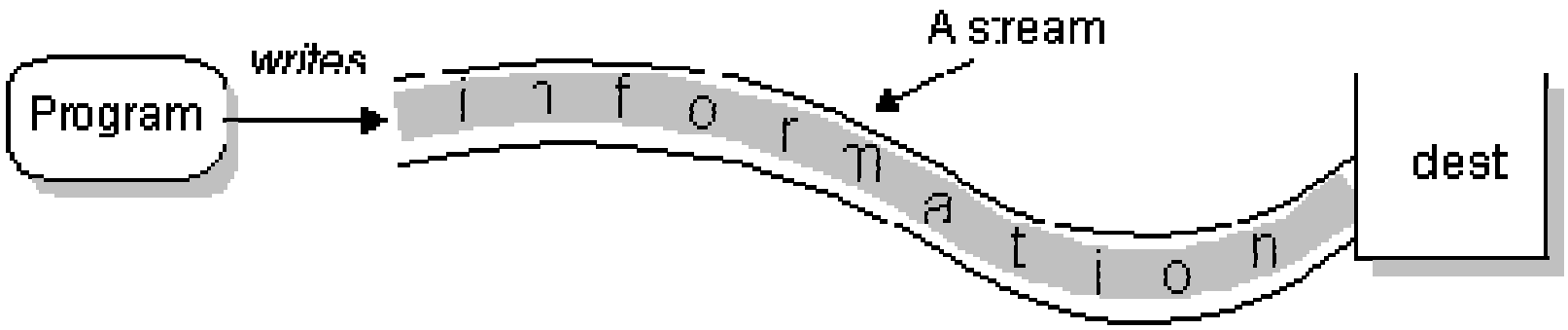
Overview of I/O Streams



To bring in information, a program opens a *stream* on an information source (a file, memory, a socket) and reads the information sequentially:

Overview of I/O Streams

Similarly, a program can send information to an external destination by opening a stream to a destination and writing the information out sequentially:



Overview of I/O Streams

The algorithms for sequentially reading and writing data are basically the same:

Reading:

open a stream
while more information
read information
close the stream

Overview of I/O Streams

The algorithms for sequentially reading and writing data are basically the same:

Writing

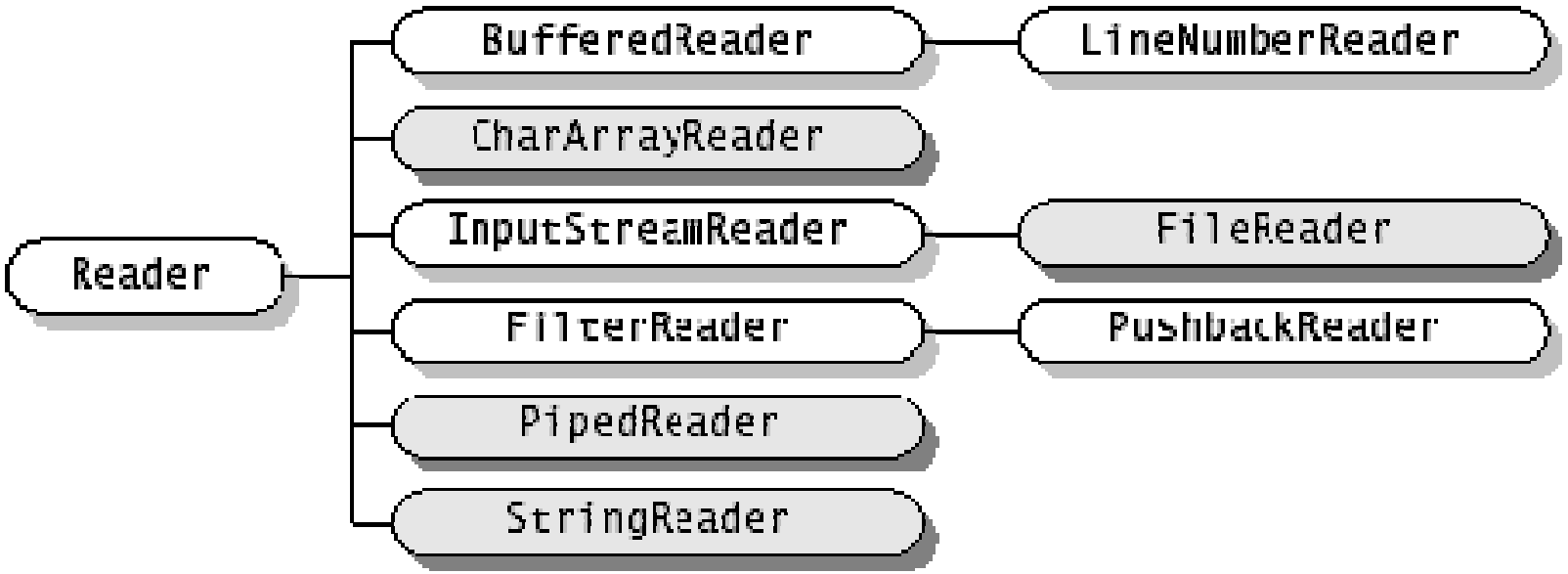
open a stream
while more information
write information
close the stream

Overview of I/O Streams

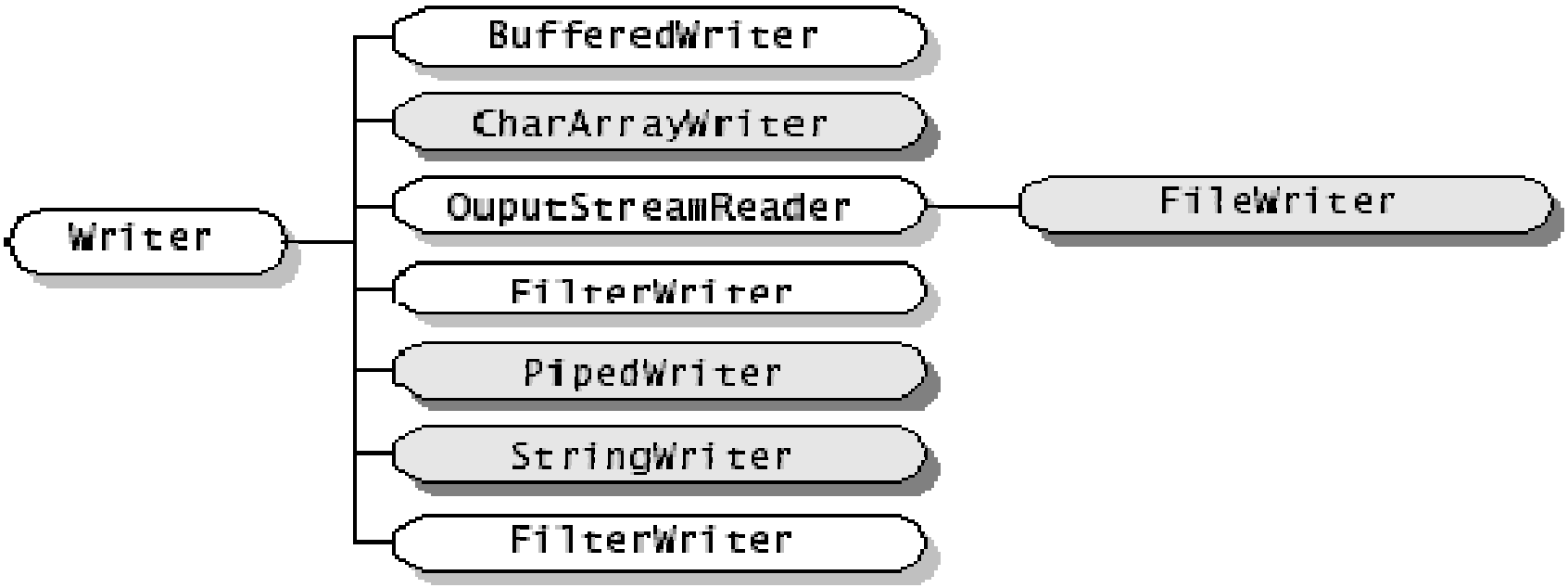
The **java.io.*** package contains a collection of stream classes that support these algorithms for reading and writing.

The stream classes are divided into two class hierarchies, based on the data type (either characters or bytes) on which they operate.

Character Streams



Character Streams



Access to input/output:

Import the necessary components:

```
import java.io.*;
```

Any method that uses the filereader or filewriter must add the phrase **`throws IOException`** at the end of the header

Reading methods:

set up the basic input stream:

```
FileReader fr = new FileReader (“prices.txt”);
```

to read next character

```
fr.read ();
```

to close the basic I/O stream object

```
fr.close();
```

Writing methods

set up the basic output stream

```
FileWriter fw = new FileWriter("prices.txt");
```

buffer it for faster output

```
BufferedWriter bw = new BufferedWriter(fw);
```

this provides a set of useful methods

```
PrintWriter pw = new PrintWriter(bw);
```

```
pw.print("Hello World");
```

Appending information to existing files

```
FileWriter objName = new FileWriter(String
file_name, boolean append);
```

If append is true - start from the end of the file

If append is false – create a new file (overwrite existing)

Programming exercise 1

Write a program to print an invoice for bought goods. Attributes of a program are arrays: *prices*, *units* and *product_names. „Java T-shirt", "Java Mug", "Java Pin"*

Program result is:

19.99	12	Java T-shirt
9.99	8	Java Mug
15.99	13	Java Pin

PROGRAMMING EXERCISE

Programming exercise 2

Write a program to copy one text file (*invoice.txt*) to another (*out.txt*)

PROGRAMMING EXERCISE

Programming exercise 3

Add “Hello java” text to the end of “out.txt” file.

PROGRAMMING
EXERCISE

Advanced exercise 1

Implement a pair of classes, one Reader and one Writer, that count the number of times a particular character, such as ‘e’, is read or written to.

The character can be specified when the stream is created. Write a program to test your classes.

PROGRAMMING EXERCISE

Advanced exercise 2

Write class Logger. Class method *log (String text)* writes text to the log.txt file with a current date and time. New logger text begins from a new line. Add this class to your existing program (for example program from lab no. 9 for calculating figure areas) and use it to log important program elements.

File elements example:

…

2008.05.27 10:50:45 object triangle: calculating area;

2008.05.27 10:50:46 object circle: calculating area;

…

PROGRAMMING EXERCISE

Graphical materials

HOMEWORK EXERCISE

ORAL EXERCISE

BOARD EXERCISE

PROGRAMMING EXERCISE