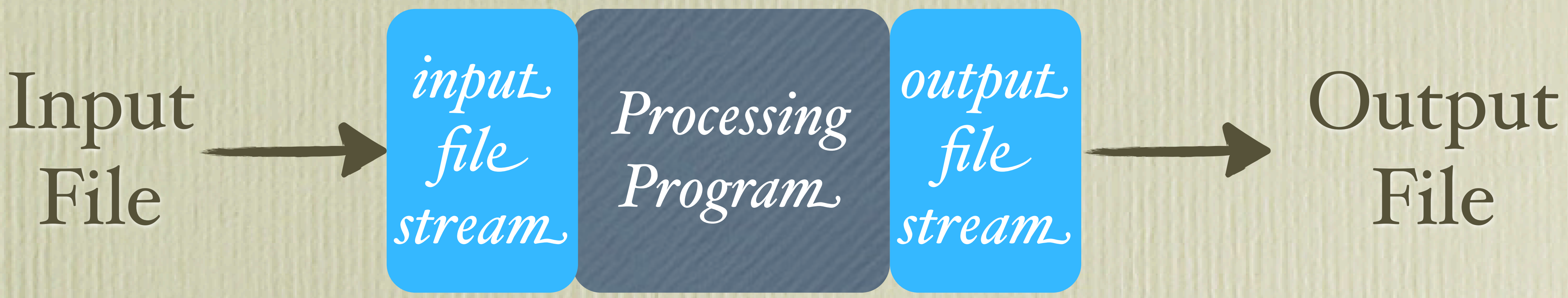
File Processing in C++

File Processing



Input File

input file stream

Processing Program

output file stream

Output File

File Streams

- The program must create a file stream *object* in memory in order to work with the file

- A file stream object is an object that is associated with a specified file (called a *stream* because a file can be thought of as a stream of data).

- File streams work similar to cin and cout objects. Hence a stream of data may be sent to a file stream, that writes the data to a file or data may be read from a file into a file stream object into variables.

File Stream Data Types

- ofstream   Output file stream. You create an object of this type when you want to create a file and write data to it.
- ifstream   Input file stream. You create an object of this type when you want to open an existing file for reading data.
- fstream   File stream. Objects of this data type are used to open files for reading, writing, or both.

File Processing

This code shows an example of opening a file for input:

Declare an object of type ifstream

input string, in this case the file name

```
ifstream  infile;
infile.open(“grades.dat”)
```

method

This code shows an example of opening a file for output:

```
ofstream  outfile;
outfile.open(“grades.out”)  //note the different file extension
```

More File Processing

- All open files must be closed. To close a file (in this case an input file):

```
infile.close( );
```

- All open files must be closed. To close a file (in this case an output file):

```
outfile.close( );
```

Writing Data to Files

- You know how to use the stream insertion operator << in a cout. It works the same way with files:

```
outfile << “Hello World” << endl;
```

OR

```
outfile << “My GPA is ” <<  GPA << endl;
```

Reading Data From Files

- The operator >> (the stream extraction operator) also can read data from a file:

```
infile >> credits;

        OR

infile >> grades;
```

Note: in both cases credits and grades MUST be declared!

Note the Power of >> and <<

- The stream extraction operator << and the stream insertion operator >> also return values of True and False. Hence they can be used in **if** and **while** loops.   For example:

```
while(infile1 >> grades && infile2 >> credits)
```