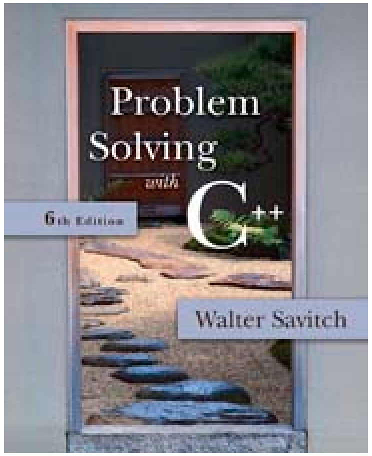
APS105: Lecture 13

Wael Aboelsaadat

wael@cs.toronto.edu

http://ccnet3.utoronto.ca/20079/aps105h1f/

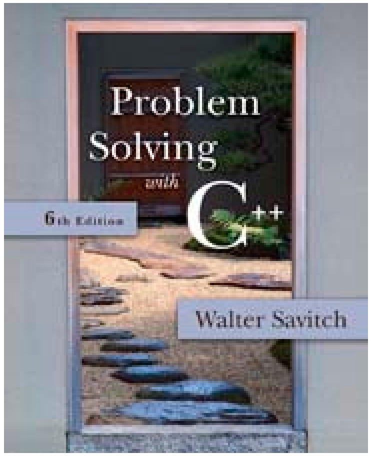
Acknowledgement: These slides are a modified version of the text book slides as supplied by Addison Wesley



Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Chapter 6

I/O Streams as an Introduction to Objects and Classes



Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Overview

6.1 Streams and Basic File I/O

6.2 Tools for Stream I/O

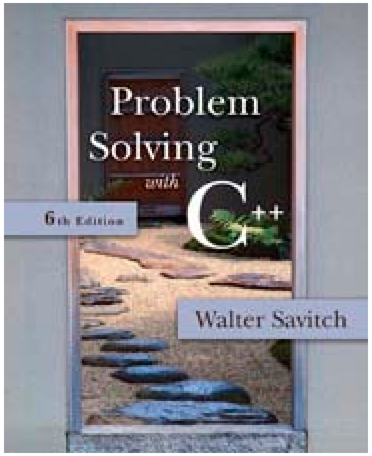
6.3 Character I/O

6.4 Inheritance

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

6.1

Streams and Basic File I/O



Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

I/O Streams

- I/O refers to program input and output
    - Input is delivered to your program via a stream object
    - Input can be from
        - The keyboard
        - A file
    - Output is delivered to the output device via a stream object
    - Output can be to
        - The screen
        - A file

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Objects

- Objects are special variables that
  - Have their own special-purpose functions
  - Set C++ apart from earlier programming languages

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Streams and Basic File I/O

- Files for I/O are the same type of files used to store programs
- A stream is a flow of data.
  - Input stream: Data flows into the program
    - If input stream flows from keyboard, the program will accept data from the keyboard
    - If input stream flows from a file, the program will accept data from the file
  - Output stream: Data flows out of the program
    - To the screen
    - To a file

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

cin And cout Streams

- cin
  - Input stream connected to the keyboard
- cout
  - Output stream connected to the screen
- cin and cout defined in the iostream library
  - Use include directive: #include <iostream>
- You can declare your own streams to use with files.

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Why Use Files?

- Files allow you to store data permanently!
- Data output to a file lasts after the program ends
- An input file can be used over and over
  - No typing of data again and again for testing
- Create a data file or read an output file at your convenience
- Files allow you to deal with larger data sets

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

File I/O

- Reading from a file
  - Taking input from a file
  - Done from beginning to the end (for now)
    - No backing up to read something again (OK to start over)
    - Just as done from the keyboard
- Writing to a file
  - Sending output to a file
  - Done from beginning to end (for now)
    - No backing up to write something again(OK to start over)
    - Just as done to the screen

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Stream Variables

- Like other variables, a stream variable…
  - Must be declared before it can be used
  - Must be initialized before it contains valid data
    - Initializing a stream means connecting it to a file
    - The value of the stream variable can be thought of as the file it is connected to
  - Can have its value changed
    - Changing a stream value means disconnecting from one file and connecting to another

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Streams and Assignment

- A stream is a special kind of variable called an object
  - Objects can use special functions to complete tasks
- Streams use special functions instead of the assignment operator to change values

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Declaring An Input-file Stream Variable

- Input-file streams are of type ifstream

- Type ifstream is defined in the fstream library
  - You must use the include and using directives

```
#include <fstream>
using namespace std;
```

- Declare an input-file stream variable using

```
ifstream    in_stream;
```

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Declaring An Output-file Stream Variable

- Ouput-file streams of are type ofstream
- Type ofstream is defined in the fstream library
  - You must use these include and using directives

```
#include <fstream>
using namespace std;
```

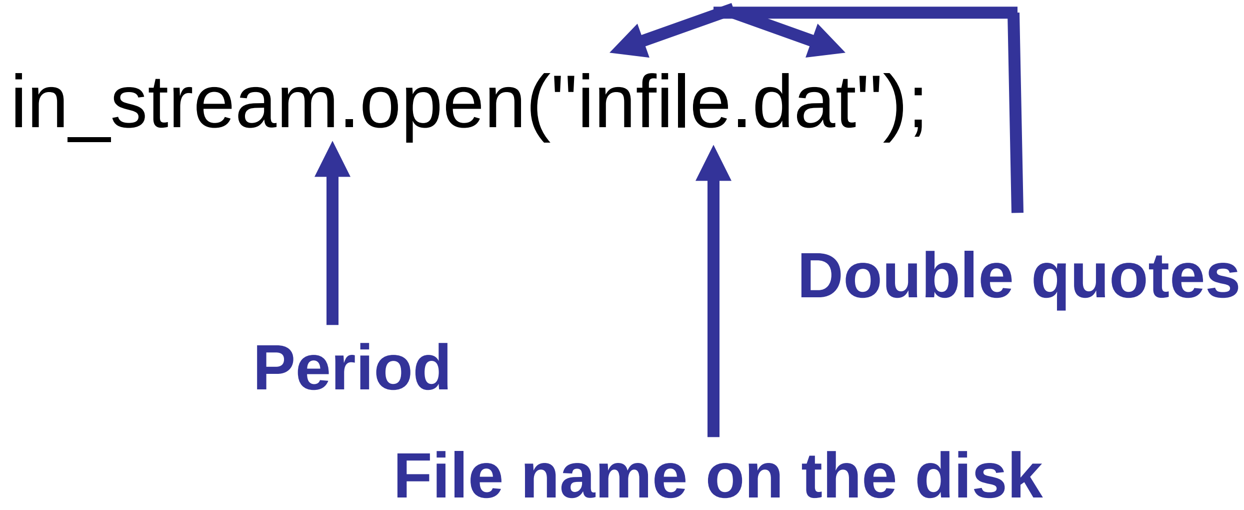
- Declare an input-file stream variable using

```
ofstream    out_stream;
```

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Connecting To A File

- Once a stream variable is declared, connect it to a file
  - Connecting a stream to a file is opening the file
  - Use the open function of the stream object



```
in_stream.open("infile.dat");
```

Double quotes

Period

File name on the disk

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Using The Input Stream

- Once connected to a file, the input-stream variable can be used to produce input just as you would use cin with the extraction operator
  - Example:

```
int one_number, another_number;
in_stream >> one_number
          >> another_number;
```

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Using The Output Stream

- An output-stream works similarly to the input-stream

  -
    ```
    ofstream out_stream;
    out_stream.open("outfile.dat");

    out_stream << "one number = "
                   << one_number
                   << "another number = "
                   << another_number;
    ```

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

External File Names

- An External File Name…
  - Is the name for a file that the operating system uses
    - infile.dat and outfile.dat used in the previous examples
  - Is the "real", on-the-disk, name for a file
  - Needs to match the naming conventions on your system
  - Usually only used in the stream's open statement
  - Once open, referred to using the name of the stream connected to it.

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Closing a File

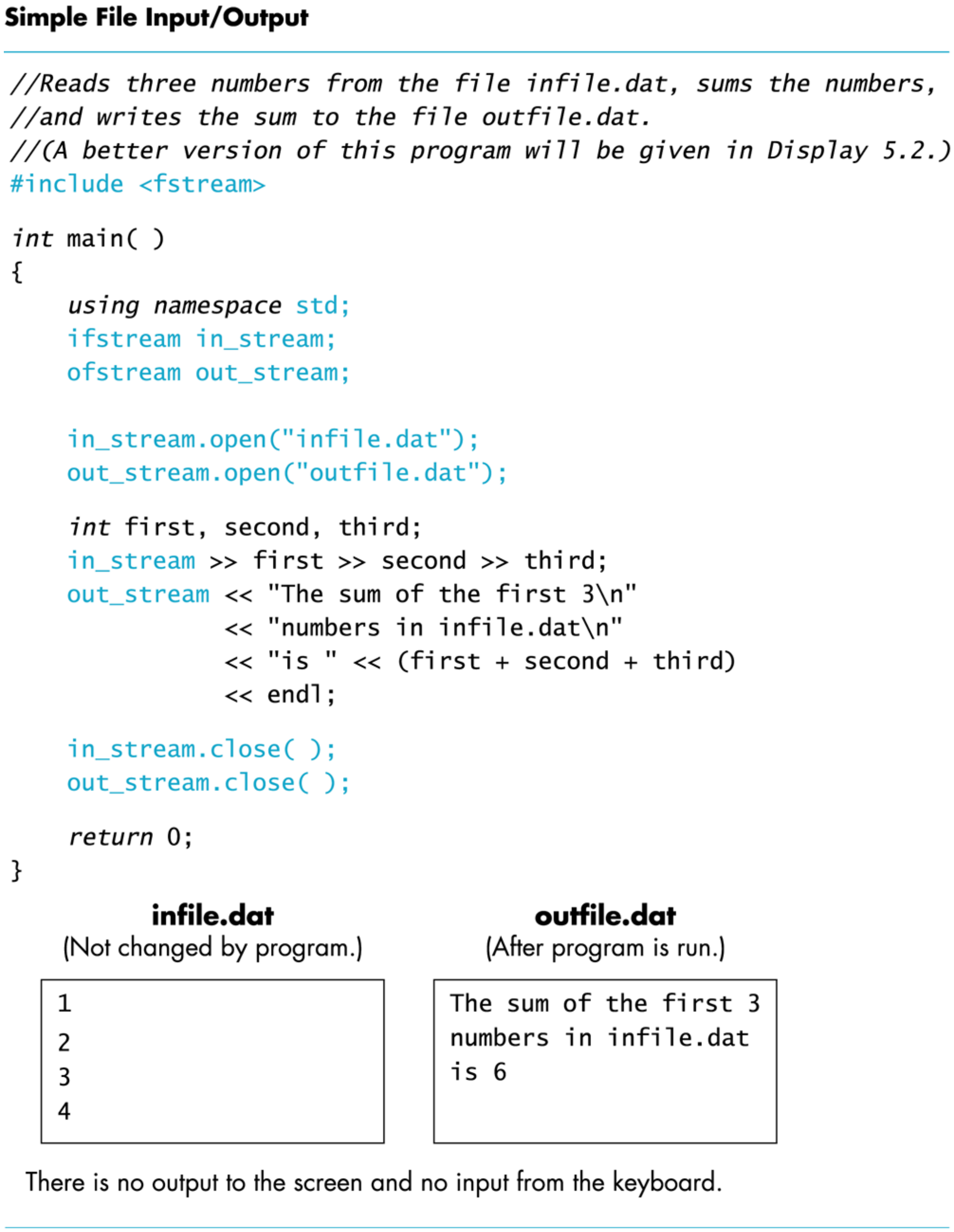
- After using a file, it should be closed
  - This disconnects the stream from the file
  - Close files to reduce the chance of a file being corrupted if the program terminates abnormally
- It is important to close an output file if your program later needs to read input from the output file
- The system will automatically close files if you forget as long as your program ends normally

Display 6.1

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Display 6.1

Back Next



Simple File Input/Output

```
//Reads three numbers from the file infile.dat, sums the numbers,
//and writes the sum to the file outfile.dat.
//(A better version of this program will be given in Display 5.2.)
#include <fstream>

int main( )
{
    using namespace std;
    ifstream in_stream;
    ofstream out_stream;

    in_stream.open("infile.dat");
    out_stream.open("outfile.dat");

    int first, second, third;
    in_stream >> first >> second >> third;
    out_stream << "The sum of the first 3\n"
               << "numbers in infile.dat\n"
               << "is " << (first + second + third)
               << endl;

    in_stream.close( );
    out_stream.close( );

    return 0;
}
```

infile.dat
(Not changed by program.)

```
1
2
3
4
```

outfile.dat
(After program is run.)

```
The sum of the first 3
numbers in infile.dat
is 6
```

There is no output to the screen and no input from the keyboard.

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley