

Input / Output Using Files

Last updated 12/5/18

Input / Output - files

- Reading and writing files uses the stream capability already discussed (Input / Output Basics)
 - 4 Steps
 - Create a file stream object
 - Assign the file system file to the object (open the file)
 - Use `>>` or `<<` to stream data to/from the file
 - Close the file
 - `#include <fstream>`

Input / Output - files

- Create a file stream object
 - 3 data types for file stream objects
 - ofstream – output file stream (writing)
 - ifstream – input file stream (reading)
 - fstream – input and output stream

- Defining a file stream object

```
ofstream my_outputfile_object;
```

```
ifstream input_file_object;
```

```
fstream data_file_object;
```

Input / Output - files

- Opening a file stream object - output
 - `object_name.open("name of file system file");`

```
ofstream my_out_file;  
my_out_file.open("datafile_1.dat"); // datafile_1 will be in the  
// project directory in eclipse
```

```
ofstream my_out_file;  
my_out_file.open("C:\\some\\other\\directory\\data_file.dat");  
// note \\ to include the \
```

If the file already exists it will be opened

If the file does not exist, it will be created and opened

Input / Output - files

- Opening a file stream object - input

```
ifstream my_input_file;  
my_input_file.open("datafile_1.dat"); // datafile_1 must be in the  
// project directory in eclipse
```

```
ifstream my_input_file;  
my_input_file.open("C:\\some\\other\\directory\\datafile.dat");  
// note \\ to include the \
```

The file must already exist

Input / Output - files

- Reading and writing data to a file object

```
// prepare file for output
ofstream my_out_file;
my_out_file.open("datafile_1.dat");
// output 10 names from an array
for(i=0; i<10; i++){
    my_out_file << name_array[i] << endl;
}
...
// prepare file for input
ifstream my_input_file;
my_input_file.open("C:\\some\\other\\directory\\datafile.dat");
// read in 10 values and store in an array
for(i=0; i<10; i++){
    my_input_file >> data_array[i];
}
```

endl only required if you want names on separate lines in the file

Reading – reads up to the end of the line and then starts at the next line. This implementation requires each value to be on a separate line.

Input / Output - files

- Closing a file object

```
// prepare file
ofstream my_out_file;
my_out_file.open("datafile_1.dat");
// output 10 names from an array
for(i=0; i<10; i++){
    my_out_file << name_array[i] << endl;
}
my_out_file.close();
```

Input / Output - files

- More sophisticated file R/W operations will be introduced later
 - Reading multiple pieces of data from a single line
 - ...

Input / Output - files

- Try this

Write the integers 0-19 out to a file

Read the values back in and print them

Input / Output - files

- Try this

```
/*
 * file_io.cpp
 *
 * Created on: Dec 4, 2019
 * Author: johnsontimoj
 */
//
// try this file for reading and writing to a file
//
//
//
#include <iostream>
#include <fstream>
using namespace std;

int main(void){

    //
    // Open file
    //
    ofstream data_file;
    data_file.open("./data.dat");

    //
    // writing data out
    //
    //
    int i;
    for(i=0; i<20; i++){
        data_file << i << endl;
    } // end for

    //
    // Close file
    //
    //
    data_file.close();
}
```

```

//
// Open file
//
ifstream data_file_in;
data_file_in.open("data.dat");

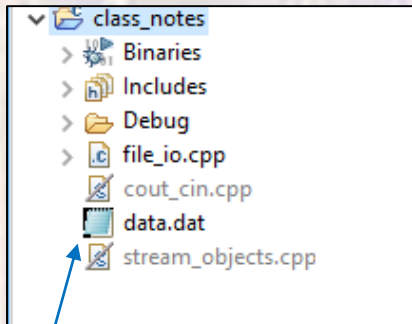
//
// reading data in
//
//
int j;
int val;
for(j=0; j<20; j++){
    data_file_in >> val ;
    // print value
    cout << val <<endl;
} // end for

//
// close file
//
//
data_file_in.close();

return 0;
} // end main
```

Input / Output - files

- Try this



file created and placed in project

```
data.dat - Notepad
File Edit Format View Help
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
```

data file

```
<terminated> (exit value: 0) class_notes.exe [C:
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
```

run results