

Special Topics in C++

November 15, 2010

Specialized I/O

Opening User-Specified Files for Reading

Example 1

```
1 ifstream inFile( "data3.dat" );
2 if( !inFile ) {
3     cout << "Error opening file." << endl;
4     exit(1);
5 }
6 cout << "File opened!" << endl;
7 // Now read data from file ...
8 inFile.close();
```

Run Example 1

- ▶ Inflexible, but it works.
- ▶ Opens file on variable declaration.
- ▶ Note the !inFile to check failure.

Opening User-Specified Files for Reading

Example 1

```

1 ifstream inFile( "data3.dat" );
2 if( !inFile ) {
3     cout << "Error opening file." << endl;
4     exit(1);
5 }
6 cout << "File opened!" << endl;
7 // Now read data from file ...
8 inFile.close();

```

Run Example 1

- ▶ Inflexible, but it works.
- ▶ Opens file on variable declaration.
- ▶ Note the !inFile to check failure.

Example 2

```

1 ifstream inFile;
2 string filename;
3 cout << "Enter filename:_" << flush;
4 cin >> filename;
5
6 // Using member-function manipulation
7 // of inFile
8 inFile.open( filename.c_str() );
9 if( !inFile ) {
10     cout << "Error opening file." << endl;
11     exit( 1 );
12 }
13 // Now read data from file ...
14 inFile.close();

```

Run Example 2

- ▶ This example uses member functions!
- ▶ Deferred opening of file with .open().
- ▶ Use .c_str() member function of string.

Recovering from (User) Input Errors

```
6 /**
7  * prompts for and reads an int, checks the input value to be sure it
8  * is within [minval,maxval], and returns the value if all is well.
9  *
10 * use maxval<=minval to disable range checking.
11 *
12 * this function recovers the input stream from typos.
13 */
14 istream& getInteger( istream& is, const string& prompt,
15                     int minval, int maxval, int& input )
16 {
17     // as long as cin is readable
18     while( is ) {
19         cout << prompt << ": " << flush;
20         if( is >> input ) {
```

[Edit Source](#)[Run](#)

Special Windows Topics

Sleeping in Your Application

```
unsigned int Sleep( unsigned int msec ) ;
```

Pauses the application for the specified number of milliseconds.

Providing msec as a positive int type works.

Sleep is Windows specific and requires `<windows.h>`.

See `#include <cstdlib>` and `sleep()` or

`#include <unistd.h>` and `usleep()` (microsecond sleep) on Unix.

[Run Program](#)

[Edit Program](#)

Clearing the Screen & Colored Text

```
64 // clear console
65 system( "cls" );
66 int lines = 0;
67 string linedata;
68 while( getline( infile , linedata , '\n' ) ) {
69
70     // show pages
71     cout.width(5);
72     cout << ++lines << " ";
73     cout << linedata << endl;
74
75     if( lines % MAXLINES == 0 ) { //
76         cout << "c=continue,q=quit..." << flush;
77         char cmd;
78         cin >> cmd;
79         if( cmd == 'q' ) break; // exit
80         // clear console
81         system( "cls" );
82         // change display color
83         randomcolor();
84     }
85 }
```

[Edit Source](#)[Run](#)

Scanning for Data Files

```

45 string fname_from_list;
46 while( getline( infile , fname_from_list , '\n' ) ) {
47     /**
48      * for each file name in the list...
49     */
50     ifstream f( fname_from_list.c_str() );
51     if( !f ) {
52         cout << "Strange:_" << fname_from_list <<
53             "_could_not_be_opened." << endl;
54         // close is not required after an open error
55     } else {
56         // reading to '\0' retrieves the entire contents
57         string filedata;
58         getline( f, filedata , '\0' );
59
60         // generates a table of bytes per file
61         cout.width(12);
62         cout << filedata.length() << "_" <<
63             fname_from_list << ":__" <<
64             filedata.substr(0,32) << endl;
65
66         // close
67         f.close();
68     }
69 }
70

```

[Edit Source](#)
[Run](#)

finis