File Input/Output

File Input/Output, Reading

We will have occasion to use text files in this course, so it is important to know the correct way to interface with disk files. While there are other code structures that work just as well, there are plenty that don't work so well except on certain compilers under certain conditions. So follow this convention in this course, because it works!

|  |  |  |
| --- | --- | --- |
| #include <**fstream**> using namespace std;  int main( ) {   ...   ifstream fin;    fin.open("**data.txt**");   if (fin.good( ))   {     ...     fin.close( ); // done with the file     ...   } } |  | #include <**fstream**> using namespace std;  int main( ) {   ...   ofstream fout;    fout.open("**data.txt**");   ...   fout.close( ); // done with the file   ... } |

Actually, the names "fin" and "fout" are just valid C++ identifiers and can named as you wish in your assignments in this course. But avoid using the "fstream" object type with text files -- that's for binary files. Use "ifstream" and "ofstream" -- that's "eye-eff-stream" and "oh-eff-stream".

Also, anything you use with cin and cout work *exactly the same* with fin and fout (that's "eff-in" and "eff-out").

Outputting To An Excel-Compatible File

It's easy to create a minimally-featured XLS file. Simply name the file with the extension .xls, and in the fout statements use tabs and line feeds like this to separate columns and rows:

  fout << "Name\tAge\tGPA\n"; // column headings  
  fout << name << '\t' << age << '\t' << gpa << '\n';

Note the 3 variables in the 2nd fout statement, and the single quotes around the \t (tab) and \n (line feed). Since there's only one char in each of these text sequences, they can be chars instead of double-quotes strings -- slightly more efficient.