//Illustrates output formatting instructions.

//Reads all the numbers in the file rawdata.sat and writes the numbers

//to the screen and to the file neat.dat in a nearly fornatted way.

#include <iostream>

#include <fstream>

#include <iomanip>

using namespace std;

void make\_neat(ifstream& messy\_file, ofstream& neat\_file,

int number\_after\_decimalpoint, int field\_width);

//Precondition: The streams messy\_file and neat\_file have been connected

//to files using the function open.

//Postcondition: The numbers in the file connected to messy\_file have been

//written to the screen and to the file connected to the stream neat\_file.

//The numbers are written opne per line, in fixed-point notation (that is, not in

//e-notation), with number\_after\_decimalpoint digits after the decimal point;

//each number is preceded by a plus or minus sign and each number is in a field

//of width field\_width. (This function does not close the file.)

int main()

{

ifstream fin;

ofstream fout;

fin.open("rawdata.dat");

if (fin.fail())

{

cout << "Input file opening failed.\n";

exit(1);

}

fout.open("rawdata.dat");

if (fout.fail())

{

cout << "Output file opening failed.\n";

exit(1);

}

make\_neat(fin, fout, 5, 12);

fin.close();

fout.close();

cout << "End of program.\n";

return 0;

}

//Uses iostream, fstream, and iomanip:

void make\_neat(ifstream& messy\_file, ofstream& neat\_file,

int number\_after\_decimalpoint, int field\_width)

{

neat\_file.setf(ios::fixed);

neat\_file.setf(ios::showpoint);

neat\_file.setf(ios::showpos);

neat\_file.precision(number\_after\_decimalpoint);

cout.setf(ios::fixed);

cout.setf(ios::showpoint);

cout.setf(ios::showpos);

cout.precision(number\_after\_decimalpoint);

double next;

while (messy\_file >> next)

{

cout << setw(field\_width) << next << endl;

neat\_file << setw(field\_width) << next << endl;

}

}