```
//  file: filename_test.cpp
//
//  Test program to use C++ strings to make filenames in various ways
//
//
//  Programmer:  Dick Furnstahl  furnstahl.1@osu.edu
//
//  Revision history:
//      20-Feb-2004  original version, for 780.20 Computational Physics
//      27-Oct-2005  added system("PAUSE") for C++ seminar
//
//  Notes:
//   * compile with:  "g++ -Wall -o filename_test filename_test.cpp"
//   * note that we include <string> for strings and <sstream> for
//      building string streams
//   * building string stream objects is just like doing output using
//      iostram and iomanip, with <<'s and manipulators
//
//*************************************************************************

// include files
#include <iostream>             // cout and cin
#include <iomanip>              // manipulators like setprecision
#include <fstream>              // file input and output
#include <string>               // C++ strings
#include <sstream>              // C++ stringstream class (can omit iostream)

using namespace std;            // we need this when .h is omitted


//*************************************************************************
int
main (void)
{
  // first a little test of string input from the console
  //  notice that we don't specify the length of the string
  string test_string = "";      // initialize test string to null

  while (true)                  // do forever (until we hit a "break" statement)
    {
      cout << "Enter a single word (\"quit\" to quit): ";                         // note the \"

      // you can input strings using cin just as with any other data type
      cin >> test_string;

      if (test_string == "quit")         // comparing strings with ==
        {
          cout << "Ok, we'll move on . . ." << endl << endl;
          break;                // jump out of the while loop
        }

      // strings are output just like other data types
      cout << "You entered: " << test_string << ", didn't you?" << endl;
    }


  //*********************************************
  // second, we'll concatenate two strings using "+"
  string first_string = "", second_string = "";

  cout << "Enter two words (separated by a space or a return): ";
  cin >> first_string >> second_string;

  cout << "Combined we get " << first_string + second_string << endl << endl;


  //*********************************************
  // next, open a file with a fixed string
  string filename1;             // declare the string
```



```
  filename1 = "test.out";           // test name for the file

  cout << "Try opening the file " << filename1 << " for output." << endl;

  // open an output file stream
  //  [you could combine these as: ofstream file1 (filename1.c_str());]
  ofstream file1;
  // use .c_str() to convert the string filename1 to char * type
  file1.open (filename1.c_str());

  file1 << "This is a test of " << filename1 << endl;
  file1 << "The length of the filename is " << filename1.length();

  file1.close();                // close the output file

  cout << "Look at " << filename1 << " to see if this worked." << endl <<
    endl;


  //*********************************************
  // next, create a string with the stringstream class
  ostringstream filename_stream;        // declare a stringstream object

  int i = 3;
  // you can load the string stream just like output streams
  filename_stream << "test_stream" << i << ".out";
  // use .str() to convert to a string
  string filename2 = filename_stream.str();

  cout << "The output filename is " << filename2 << endl;

  ofstream file2;               // now for a filename
  file2.open (filename2.c_str());       // use .c_str() to convert to a char *

  // [note: you could combine the c_str and str methods into one call]

  file2 << "This is a test of putting the number " << i
    << " in this filename!" << endl;

  file2.close();                // close the output file
  cout << "Look at " << filename2 << " to see if this worked." << endl <<
    endl;

  // now try a double
  filename_stream.str ("");     // clear the string stream
  double x = 3.14159;           // more digits than we plan to use
  filename_stream << "test_stream" << "_x" << setprecision (2) << x << ".out";

  filename2 = filename_stream.str();    // switch names
  cout << "The output filename is " << filename2 << endl;
  file2.open (filename2.c_str());       // use .c_str() to convert to a char *

  file2 << "This is a test of putting two digits of the float " << x
    << " in this filename!" << endl;

  file2.close();                // close the output file
  cout << "Look at " << filename2 << " to see if this worked." << endl <<
    endl;

  system("PAUSE");  // prevent the console window from closing
  return (0);
}
```