CMSC 341 — Data Structures — Section 01 — Fall 2016

String Parsing Examples

Here are some examples of using the >> operator to parse strings.

Example 0:

Let's see what happens when we redirect a file into a C++ program that reads in 3 strings.

```
/* File: input0.cpp

   Testing file I/O using >> with strings.

*/

#include <iostream>
#include <string>
#include <stdlib.h>

using namespace std ;

int main() {
   string str1, str2, str3 ;

   cin >> str1 ;
   cin >> str2 ;
   cin >> str3 ;

   cout << "str1 = '" << str1 << "'\n" ;
   cout << "str2 = '" << str2 << "'\n" ;
   cout << "str3 = '" << str3 << "'\n" ;
}
```

Download: input0.cpp

The sample run shows that all white spaces are skipped including new line characters. Even reading past the last word is OK. In the last run, `str3` is the empty string.

```
linux1% g++ input0.cpp
```

```
linux1% cat data1
  hello   blue    world
linux1% ./a.out < data1
str1 = 'hello'
str2 = 'blue'
str3 = 'world'
linux1%

linux1% cat data2
Hello     World
   good
3.14159
linux1% ./a.out < data2
str1 = 'Hello'
str2 = 'World'
str3 = 'good'
linux1%

linux1% cat data3
abc def
linux1% ./a.out < data3
str1 = 'abc'
str2 = 'def'
str3 = ''
linux1%
```

Example 1:

In the next example, we take one of the strings and convert it to a floating point vlaue using the `atof()` function. (See sample run.) The `atof()` function is fairly robust and returns a 0 when the string cannot be turned into a float. (Try changing the data file yourself.)

```
/* File: input1.cpp

   Testing file I/O using >> with strings.

*/

#include <iostream>
#include <string>
#include <stdlib.h>   // so you can use atof()

using namespace std ;

int main() {
   string str1, str2, str3 ;
   string str4 ;

   cin >> str1 ;
   cin >> str2 ;
   cin >> str3 ;
```

```
   cin >> str4 ;

   cout << "str1 = '" << str1 << "'\n" ;

   // can use [] with strings to retrieve characters
   cout << "str1[1] = '" << str1[1] << "'\n" ;

   cout << "str2 = '" << str2 << "'\n" ;
   cout << "str3 = '" << str3 << "'\n" ;

   cout << "str4 = '" << str4 << "'\n" ;

   // convert to float
   float f = atof( str4.c_str() ) ;
   cout << "f = " << f << endl ;

}
```

Download: input1.cpp

Example 2:

The next example shows that the `getline()` function can be used to read in an entire line of text. (See sample run.) The Countries file has a line of headings that we don't want to keep.

```
/* File: input2.cpp

   Testing file I/O using >> with strings.

*/


#include <iostream>
#include <string>
#include <stdlib.h>

using namespace std ;

int main() {
   string str1, str2, str3 ;
   string str4 ;

   getline(cin, str1) ;   // read entire line
   cin >> str2 ;

   cout << "str1 = '" << str1 << "'\n" ;
   cout << "str2 = '" << str2 << "'\n" ;

}
```

Download: input2.cpp

Example 3:

In this example, we learn how to open a file for reading. (See sample run.)

```
/* File: input3.cpp

   Testing file I/O using >> with strings.

*/
#include <iostream>
#include <string>
#include <stdlib.h>   // so you can use atof()
#include <fstream>    // so you can use ifstream

using namespace std ;

int main() {
   string str1, str2, str3 ;
   string str4 ;

   ifstream ifile("data2") ;   // open file for reading
   getline(ifile, str1) ;      // use ifile instead of cin
   ifile >> str2 ;             // use ifile instead of cin

   cout << "str1 = '" << str1 << "'\n" ;
   cout << "str2 = '" << str2 << "'\n" ;

   ifile.close() ;
}
```

Download: input3.cpp

Example 4:

In the next example, we read in 9 fields from the second line of the input. (See sample run.)

```
/* File: input4.cpp

   Testing file I/O using >> with strings.

*/
#include <iostream>
#include <string>
#include <stdlib.h>
#include <fstream>

using namespace std ;

int main() {
   string firstLine ;
   string str1, str2, str3, str4, str5, str6, str7, str8, str9 ;
```

```
   ifstream ifile("2013WorldBankEducationCensusData.txt") ;
   getline(ifile, firstLine) ;   // toss first line

   ifile >> str1 ;   // read 9 fields
   ifile >> str2 ;
   ifile >> str3 ;
   ifile >> str4 ;
   ifile >> str5 ;
   ifile >> str6 ;
   ifile >> str7 ;
   ifile >> str8 ;
   ifile >> str9 ;

   cout << "str1 = '" << str1 << "'\n" ;   // print 9 fields
   cout << "str2 = '" << str2 << "'\n" ;
   cout << "str3 = '" << str3 << "'\n" ;
   cout << "str4 = '" << str4 << "'\n" ;
   cout << "str5 = '" << str5 << "'\n" ;
   cout << "str6 = '" << str6 << "'\n" ;
   cout << "str7 = '" << str7 << "'\n" ;
   cout << "str8 = '" << str8 << "'\n" ;
   cout << "str9 = '" << str9 << "'\n" ;

   ifile.close() ;   // put away file when done
}
```

Download: input4.cpp

Example 5:

Then, we put the printing in a loop. (See sample run.)

```
/* File: input5.cpp

   Testing file I/O using >> with strings.

*/

#include <iostream>
#include <string>
#include <stdlib.h>
#include <fstream>

using namespace std ;

int main() {
   string firstLine ;
   string str1, str2, str3, str4, str5, str6, str7, str8, str9 ;

   ifstream ifile("2013WorldBankEducationCensusData.txt") ;   // open file
   getline(ifile, firstLine) ;    // toss first line
```

```
   while(true) {    // keep reading until done

      ifile >> str1 ;   // read 9 fields
      ifile >> str2 ;
      ifile >> str3 ;
      ifile >> str4 ;
      ifile >> str5 ;
      ifile >> str6 ;
      ifile >> str7 ;
      ifile >> str8 ;
      ifile >> str9 ;

      if ( ifile.eof() ) break ;  // done?

      cout << "str1 = '" << str1 << "'\n" ;     // print country name

   /*   cout << "str2 = '" << str2 << "'\n" ;   // skip printing rest of data
      cout << "str3 = '" << str3 << "'\n" ;
      cout << "str4 = '" << str4 << "'\n" ;
      cout << "str5 = '" << str5 << "'\n" ;
      cout << "str6 = '" << str6 << "'\n" ;
      cout << "str7 = '" << str7 << "'\n" ;
      cout << "str8 = '" << str8 << "'\n" ;
      cout << "str9 = '" << str9 << "'\n" ;
   */
   }
   ifile.close() ;   // put away file when done
}
```

Download: input5.cpp

Example 6:

We look for "N/A" in the input. (See sample run.)

```
/* File: input6.cpp

   Testing file I/O using >> with strings.

*/

#include <iostream>
#include <string>
#include <stdlib.h>
#include <fstream>

using namespace std ;

int main() {
   string firstLine ;
   string str1, str2, str3, str4, str5, str6, str7, str8, str9 ;
```

```
   long int population ;

   ifstream ifile("2013WorldBankEducationCensusData.txt") ;
   getline(ifile, firstLine) ;

   while(true) {
      ifile >> str1 ;
      ifile >> str2 ;
      ifile >> str3 ;
      ifile >> str4 ;
      ifile >> str5 ;
      ifile >> str6 ;
      ifile >> str7 ;
      ifile >> str8 ;
      ifile >> str9 ;

      if ( ifile.eof() ) break ;

      cout << "country = '" << str1 << "'   " ;

      // use atol() to convert to long
      population = atol(str2.c_str()) ;

      cout << "population = " << population << endl ;

      // use == for string comparison
      if ( str3 == "N/A" ) cout << "   str3 has no value\n" ;

   /*   cout << "str2 = '" << str2 << "'\n" ;
      cout << "str3 = '" << str3 << "'\n" ;
      cout << "str4 = '" << str4 << "'\n" ;
      cout << "str5 = '" << str5 << "'\n" ;
      cout << "str6 = '" << str6 << "'\n" ;
      cout << "str7 = '" << str7 << "'\n" ;
      cout << "str8 = '" << str8 << "'\n" ;
      cout << "str9 = '" << str9 << "'\n" ;
   */
   }
   ifile.close() ;
}
```

Download: input6.cpp

Example 7:

In this version, we store a -1 where there was an "N/A". (See sample run.)

```
/* File: input7.cpp

   Testing file I/O using >> with strings.

*/
```

```
#include <iostream>
#include <string>
#include <stdlib.h>
#include <fstream>

using namespace std ;

int main() {
   string firstLine ;
   string str1, str2, str3, str4, str5, str6, str7, str8, str9 ;
   long int population ;
   float litRate ;

   ifstream ifile("2013WorldBankEducationCensusData.txt") ;
   getline(ifile, firstLine) ;

   while(true) {
      ifile >> str1 ;
      ifile >> str2 ;
      ifile >> str3 ;
      ifile >> str4 ;
      ifile >> str5 ;
      ifile >> str6 ;
      ifile >> str7 ;
      ifile >> str8 ;
      ifile >> str9 ;

      if ( ifile.eof() ) break ;

      cout << "country = '" << str1 << "'   " ;

      population = atol(str2.c_str()) ;

      cout << "population = " << population << "   " ;


      // check if field is N/A. Store -1 if so.
      //
      if ( str3 == "N/A" ) {
         litRate = -1.0 ;
      } else {
         litRate = atof(str3.c_str()) ;
      }

      cout << "literacy rate = " << litRate << endl ;


   /*   cout << "str2 = '" << str2 << "'\n" ;
      cout << "str3 = '" << str3 << "'\n" ;
      cout << "str4 = '" << str4 << "'\n" ;
      cout << "str5 = '" << str5 << "'\n" ;
      cout << "str6 = '" << str6 << "'\n" ;
      cout << "str7 = '" << str7 << "'\n" ;
```

```
      cout << "str8 = '" << str8 << "'\n" ;
      cout << "str9 = '" << str9 << "'\n" ;
   */
   }
   ifile.close() ;
}
```

Download: input7.cpp

Example 8:

This next version is a little bit more robust. It can deal with the situation where some lines may have more words than others. It uses `getline()` to read an entire line of input from the file. Then it converts the string into a stringstream. Finally, the stringstream can be used like `cin` to get separate words.

We did not encounter any issues trying to read 11 words from each line, even though each line only has 9 words.

(See sample run.)

```
/* File: input8.cpp

   Testing file I/O using >> with strings.

*/

#include <iostream>
#include <string>
#include <stdlib.h>
#include <fstream>
#include <sstream>   // so you can use istringstream

using namespace std ;

int main() {
   string oneLine ;
   string str1, str2, str3, str4, str5, str6, str7, str8, str9 ;
   string str10, str11 ;
   long int population ;
   float litRate ;

   ifstream ifile("2013WorldBankEducationCensusData.txt") ;
   getline(ifile, oneLine) ;

   while(true) {
      getline(ifile, oneLine) ;  // read entire line

      // cout << oneLine << endl ;

      if ( ifile.eof() ) break ;

      // Convert string to stringstream
      // new one each loop!
      //
```

```
        istringstream istrm(oneLine) ;

        istrm >> str1 ;    // read from istrm
        istrm >> str2 ;
        istrm >> str3 ;
        istrm >> str4 ;
        istrm >> str5 ;
        istrm >> str6 ;
        istrm >> str7 ;
        istrm >> str8 ;
        istrm >> str9 ;
        istrm >> str10 ;   // extra read attempts OK
        istrm >> str11 ;


        cout << "country = '" << str1 << "'    " ;
        population = atol(str2.c_str()) ;
        cout << "population = " << population << "    " ;

        if ( str3 == "N/A" ) {
           litRate = -1.0 ;
        } else {
           litRate = atof(str3.c_str()) ;
        }

        cout << "literacy rate = " << litRate << endl ;


    /*   cout << "str2 = '" << str2 << "'\n" ;
        cout << "str3 = '" << str3 << "'\n" ;
        cout << "str4 = '" << str4 << "'\n" ;
        cout << "str5 = '" << str5 << "'\n" ;
        cout << "str6 = '" << str6 << "'\n" ;
        cout << "str7 = '" << str7 << "'\n" ;
        cout << "str8 = '" << str8 << "'\n" ;
        cout << "str9 = '" << str9 << "'\n" ;
    */
    }
    ifile.close() ;
}
```

Download: input8.cpp

Example 9:

In this final version, we store each country's name in a vector of strings. At the end of the program, we loop through the vector and print out each country's name. (See sample run).

```
/* File: input9.cpp

   Testing file I/O using >> with strings.
```

```
*/

#include <iostream>
#include <string>
#include <stdlib.h>
#include <fstream>
#include <sstream>
#include <vector>   // so you can use vector

using namespace std ;

int main() {
   string oneLine ;
   string str1, str2, str3, str4, str5, str6, str7, str8, str9 ;
   string str10, str11 ;
   long int population ;
   float litRate ;
   vector<string> names ;

   ifstream ifile("2013WorldBankEducationCensusData.txt") ;
   getline(ifile, oneLine) ;

   while(true) {
      getline(ifile, oneLine) ;

      // cout << oneLine << endl ;

      if ( ifile.eof() ) break ;

      istringstream istrm(oneLine) ;   // new one each loop!

      istrm >> str1 ;
      istrm >> str2 ;
      istrm >> str3 ;
      istrm >> str4 ;
      istrm >> str5 ;
      istrm >> str6 ;
      istrm >> str7 ;
      istrm >> str8 ;
      istrm >> str9 ;
      istrm >> str10 ;
      istrm >> str11 ;


      cout << "country = '" << str1 << "'   " ;
      population = atol(str2.c_str()) ;
      cout << "population = " << population << "   " ;

      if ( str3 == "N/A" ) {
         litRate = -1.0 ;
      } else {
         litRate = atof(str3.c_str()) ;
      }
```

```
         cout << "literacy rate = " << litRate << endl ;

         names.push_back(str1) ;     // add country name to vector


      /*   cout << "str2 = '" << str2 << "'\n" ;
         cout << "str3 = '" << str3 << "'\n" ;
         cout << "str4 = '" << str4 << "'\n" ;
         cout << "str5 = '" << str5 << "'\n" ;
         cout << "str6 = '" << str6 << "'\n" ;
         cout << "str7 = '" << str7 << "'\n" ;
         cout << "str8 = '" << str8 << "'\n" ;
         cout << "str9 = '" << str9 << "'\n" ;
      */
      }
      ifile.close() ;


      // print out country names from vector
      //
      cout << "\n\n\n**** Vector test  *****\n" ;

      int n = names.size() ;
      for (int i = 0 ; i < n ; i++) {
         cout << names[i] << endl ;
      }


   }
```

Download: input9.cpp
