CS 368 Announcements
Wednesday, April 17, 2013

Program p4 - due Monday, April 22

Last Time

- start Ch. 9 (Input and Output)
- console I/O
- overloading >>
- error states
- file I/O

Today

- continue Ch. 9 (Input and Output)
- string class
- C strings
- C I/O
- manipulators

Next Time

- Ch. 7 (Templates)
- templated functions
- templated classes

File I/O Example (from fileIO.cpp)

```
ifstream inFile("inFile.txt");
if (!inFile) {
    cerr << "Unable to open inFile.txt for reading" << endl;
    return 1;
}

ofstream outFile("outFile.txt");
if (!outFile) {
    cerr << "Unable to open outFile.txt for writing" << endl;
    return 1;
}

char ch;
while (inFile.get(ch)) {
    switch (ch) {
      case 'a':  case 'A':
      case 'e':  case 'E':
      case 'i':  case 'I':
      case 'o':  case 'O':
      case 'u':  case 'U':
         outFile << '*';
         break;
      default:
         outFile << ch;
    }
}

inFile.close();
outFile.close();
```

`string` Class

In `string` library (`#include <string>`)

Overloaded support for `==` `!=` `>` `>=` `<` `<=` `=`

C++ strings are mutable

Useful member functions:

Recommendation:

- use **`string`** library
- convert to and from C-style as needed
  but do all string operations using **`string`** library

C-Style Strings

C-style string = an array of characters ending in `'\0'`

Library support in `cstring` (need to `#include <cstring>`):

```
int    strlen( const char *str )

char * strcpy( char *LHS, char *RHS )

char * strcat( char *LHS, char *RHS )

int    strcmp( const char *LHS, const char *RHS )
```

File I/O Example (from fileIO.cpp)

```
int countLines(string & fileName) {

    ifstream myFile(fileName.c_str());

    if (myFile) {

        int count = 0;
-------------------------------------------------------------
        //VERSION 1: C style strings
        // uses a char array and member function getline()
        char line[100];
        while (!myFile.getline(line, 100).eof()) {
            count++;
            cout << "Line " << count << ": " << line << endl;
        }
-------------------------------------------------------------
        //VERSION 2: C++ style strings
        // uses a string object and free function getline()
        string line;
        while (!getline(myFile, line).eof()) {
            count++;
            cout << "Line " << count << ": " << line << endl;
        }
-------------------------------------------------------------
        myFile.close();
        return count;

    } else {
        cerr << "Unable to open " << fileName
             << " in countLines" << endl;
        myFile.close();
        return -1;
    }
}
```

C-Style I/O

Standard I/O library (inherited from C): `#include <cstdio>`

Console I/O

- To write to standard output, use

- To read from standard input, use

File I/O

- To write to a file, use

- To read from a file, use

Never mix I/O libraries in a single program!

Example

```
#include <cstdio>

using namespace std;

int main() {
    int x = 1234;
    double y = 8763.1415;
    char str[20] = "cs368";



    printf("x is %d, y is %f, str is %s\n", x, y, str);




    printf("%-20d %11.6f %s\n", x, y, str);




    printf("%-20d %11.6f %s\n", 97215, 12.34,
           "November");


    printf("Enter a number and a string: ");


    scanf("%d %s", &x, str);


    printf("x is %d, str is %s\n", x, str);
}
```

Formatting Output using Manipulators

```
#include <iostream>
#include <iomanip>
#include <string>

using namespace std;

int main() {
    int x = 1234;
    double y = 8763.1415;
    string str = "cs368";

    cout << "x is " << x << ", y is " << y <<", str is "
         << str << endl;




    cout << left << setw(20) << x << " "
         << right << setw(11) << setprecision(6)
         << fixed << y << " "
         << str << endl;




    cout << left << setw(20) << 97215 << " "
         << right << setw(11) << setprecision(6)
         << fixed << 12.34 << " "
         << "November" << endl;

    cout << "Enter a number and a string: ";
    cin >> x >> str;
    cout << "x is " << x << ", str is " << str << '\n';
}
```