Clark Chambers
Homework 2
CS 121

Program 1:

```
#include <iostream>
#include <fstream>
#include <iomanip>

using namespace std;

int main(int argc, char ** argv) {

       // vars
       float celc;
       float faren;
       float kelvin;

       // Files
       ifstream in_file("temps.txt");
       ofstream out_file("temp_table.txt");

       // header
       out_file << left;
       out_file << setw(20) << "Celsuis";
       out_file << setw(20) << "Fahrenheit";
       out_file << setw(20) << "Kelvin" << endl;


       while(in_file.good()) {
              // Process each line of file and stuff it up in there.
              in_file >> celc;
              faren = (9/5) * celc + 32;
              kelvin = celc + 273;

              out_file << setw(20) << celc;
              out_file << setw(20) << faren;
              out_file << setw(20) << kelvin << endl;
       }

       // Clean it up, boy.
       in_file.close();
       out_file.close();

       return 0;
}
```

Program 2

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

using namespace std;

int main(int argc, char ** argv) {

        // Vars
        string title;
        string author;
        float price;
        char y_n = 'Y';

        ofstream out_file("book_database.txt");
        // init stream
        out_file << fixed << setprecision(2) << setfill('.');

        cout << "Welcome to the cool-dude book-storing neato program!" << endl << endl;

        while(y_n == 'Y' || y_n == 'y') {

                // Get book info
                cout << "Enter Book Title: ";
                getline(cin, title);

                cout << "Enter Author: ";
                getline(cin, author);

                cout << "Enter price: ";
                cin >> price;

                // Flush buffer
                cin.clear();
                cin.ignore(numeric_limits<streamsize>::max(), '\n');

                // Store that stuff in a file.
                out_file << left << setw(40) << title // Title
                               << setw(20) << author      // Author
                               << right << setw(7) << "$" << price // price
                               << endl;

                cout << endl;
                cout <<  "Yo. You want another book? (Y/N): ";
                cin >> y_n;

                // Flush buffer
                cin.clear();
                cin.ignore(numeric_limits<streamsize>::max(), '\n');
        }

        out_file.close();
        return 0;
}
```

Program 3

```
#include <iostream>
#include <string>
#include <algorithm>
#include <cctype>
#include <vector>
#include <sstream>

using namespace std;

int main(int argc, char ** argv) {

        // vars
        int num_words = 0;
        int pos = 0;
        int num_chars = 0;
        string sentence;
        string working_copy;
        char char_to_find;

        // prompt
        cout << "Yo. Enter a sentence. We're gonna get weird: ";
        getline(cin, sentence);

        // Just use a stringstream
        stringstream ss;
        ss.str(sentence);
        for(string word; ss >> word; num_words++);


        working_copy = sentence;
        // NOTE(clark): Apply the lambda to everything in the string.
        for_each(working_copy.begin(), working_copy.end(),
                [](auto &c) {
                // Force standard namespace to avoid C conflicts.
                c = std::toupper(c);
        });


        // Get weird with it.
        cout << endl << "Here's how many characters there are in that: " << sentence.size() <<
endl;
        cout << "Here's how many words there are in that: " << num_words << endl;
        cout << "Here's uppercase: " << working_copy << endl;

        // Prompt for substring stuff.
        cout << endl << "Give me a position: ";
        cin >> pos;
        // Flush buffer
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');

        // Prompt for substring stuff.
        cout << "Give me a length: ";
        cin >> num_chars;
        // Flush buffer
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
```

```cpp
    if(pos + num_chars < sentence.size() && pos >= 0) {
        cout << "Here's your substring: " << sentence.substr(pos, num_chars) << endl;
    }
    else {
        cout << "Nice try bucko. No segfaults for you." << endl;
    }

    cout << endl << "Give me a character: ";
    cin >> char_to_find;
    // Flush buffer
    cin.clear();
    cin.ignore(numeric_limits<streamsize>::max(), '\n');
    cout << endl;

    // I'm just going to stuff the positions in here.
    vector<int> char_positions;

    // Show the sentence
    cout << sentence << endl;

    int temp_counter = 0;
    // Process the string.
    for_each(sentence.begin(), sentence.end(),
        [&](auto &c) {
            if(c == char_to_find) {
                cout << '^';
                char_positions.push_back(temp_counter);
            }
            else {
                cout << '~';
            }
            temp_counter++;
        });

    // Print where everything was.
    cout << endl << "Positions the character was at: ";
    // Print the stuff!
    for_each(char_positions.begin(), char_positions.end(),
        [](auto &i) {
            cout << i << " ";
        });
    cout << endl;

    system("pause");

    return 0;
}
```

Program4

```
#include <iostream>
#include <fstream>
#include <string>
#include <cctype>
#include <algorithm>

using namespace std;

int main(int argc, char ** argv) {

        ifstream in_file("UAH_sample.txt");
        ofstream out_file("UAH_next.txt");

        int found = 0;
        int word_count = 0;

        // Read in the file
        string line;
        string working;
        string bigfile = "";
        while(in_file.good()) {
                getline(in_file, line);

                // replace UAH with UAHuntsville
                found = line.find("UAH");
                while(found != string::npos) {
                        line.replace(found, 3, "UAHuntsville");
                        found = line.find("UAH", found + 1);
                }

                out_file << line << endl;

                bigfile += line;
        }

        // reopen the bad boy
        in_file.close();
        in_file.open("UAH_sample.txt");
        while(in_file >> working) { word_count++; }

        cout << "Number of words: " << word_count << endl;

        in_file.close();
        out_file.close();

        system("pause");

        return 0;
}
```