

# CSE20212 Fundamentals of Computing II

## Spring 2008

Lab handout for the weeks of April 14 and 21

**Objectives:** In this final lab, you will demonstrate your ability to use STL maps, begin to look at concepts from computational geometry, and learn about design patterns.

### General activities for these weeks:

1. Write a C++ program that prompts for a filename on `cout`, reads the file name from `cin`, opens the file for reading, and does the following. The program must read each word from the file into a C++ string, stopping when end-of-file is encountered. It must print the number of words read from the file to `cout`. As it reads, it should use an STL `map` to build a word count for each unique word, ignoring punctuation marks and letter case. After the file is read, the program must create a file `counts.txt` containing each unique word in the file along with its count. After writing the file, your program must enter an infinite loop in which it asks the user for a query word and then retrieves the word count for that query word. The query word should have its punctuation characters removed and its uppercase letters changed to lower case. The program must handle “gracefully” a query string that is empty, a query string that consists of only punctuation, and a query string that does not match. Retrieve the file <http://www.usconstitution.net/const.txt>, remove all text at the start of the file prior to “We the People...” and report the count of the following words:

AmErIcA!  
state  
+Election+  
&^%@#\$!!  
IRISH

Hints: the code you developed for the Trie lab might be useful.

2. Read the articles entitled “Minimum Spanning Tree”, “Kruskal’s algorithm”, and “Prim’s algorithm” in Wikipedia.
3. Run the program at right to get four random points in the plane. Plot the points on graph paper and construct an MST by hand using Prim’s algorithm. Show your work (show what happens at each iteration). The weight of an edge is its length (Euclidean distance between the vertices).

```
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;

int main(int argc, char *argv[]) {
    srand((unsigned)time(0));

    for(int i=0; i<4; i++) {
        double x = (1.0*rand())/(1.0+RAND_MAX);
        double y = (1.0*rand())/(1.0+RAND_MAX);
        cout << x << " " << y << endl;
    }
}
```

### In-lab activities for the weeks of April 14 and 21:

1. Demonstrate your project as described in the project handout.

### Post-Lab (due 4/28 as described below)

1. (20 points) Turn in your word-count program and its results on the US Constitution to your dropbox by the end of the day (11pm).
2. (15 points) Turn in your hand-drawn MST on 4 vertices at the start of class (9:35am).

3. (15 points) Read the Wikipedia articles on Design Patterns and the Model-View-Controller pattern (and additional references as necessary). Submit to your dropbox by 11pm a document that contains the following.
  - a. A brief definition (in your own words), of what a design pattern is.
  - b. Is the STL iterator an instance of a design pattern? If so, which one?
  - c. Suppose you retrieve a video from YouTube and play it on your computer, using Windows Media Player, QuickTime, or an embedded player that the website provides. This is an instance of a Model-View-Controller pattern. Explain what the Model is, what the View is, and what the Controller is.