Strings (Sections 4.4,7.4)

- char word[20];
- cstring
- Strings;
- sentence.cpp revisited.

Homework

Problem: Compute the average and standard deviation of a set of n integer test scores, where the user is asked to input n and then the n test scores.

A partial solution is here. Complete it and email it to me by class time on Thursday, November 5.

Storing Words in Arrays

- char word[28]; Why 28?
- void copy_word(char first[],char next[],int MaxSize); MaxSize required.
- A character array that ends in the character with value 0 is called a cstring.
- 

- There is a library (a collection of functions) that manipulates cstrings,
#include <cstring>
- But there is a nicer way to handle arrays of characters

Example

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
  string s("Town"), s2;
   s2=s; //copy s into s2;
 cout << s2 << “=” << s <<endl;
 s[3]='s';
 cout << s2 << “!=” << s << endl;
return 0;
}
```

String Functions

```
#include <iostream>
#include <string>
using std::cin;
using std::cout;
using std::string;
int main()
{
                                        // DEFINE
    string  str1;                       // empty string
    string  str2 = "greetings";         // init using 1 arg constructor
    string  str3("Earthling");          // init using 1 arg constructor
    cout << "str1: " << str1 << "\n";   // OUTPUT
    cout << "str2: " << str2 << "\n";
    cout << "str3: " << str3 << "\n";
                                        // ASSIGN
    str1 = str3;                        // assignment, str1's value
    cout << "str1: " << str1 << "\n";   //       is a copy str3's value
                                        // INPUT
    cout << "Enter your name: ";
    string  str4;                       // empty string
    cin >> str4;                        // input into string
    cout << "str4: " << str4 << "\n\n";
                                         // CONCATENATE
    string  hey = str3 + ' '             // note char and string literal
                + str4 + ", "
                + str2 + "!\n";
    cout << hey << '\n';                 // OUTPUT
    return 0;
```

String Functions

```
///  string example: relational operators
///

#include <iostream>
#include <iomanip>
#include <string>
Using namespace std;

int main()
{
    // Prompt and input 2 strings.
    cout << "Enter two strings: ";
    string  str1, str2;
    cin >> str1 >> str2;

    // Output stream manipulator, output true or false.
    cout << boolalpha;

    // Compare the stirngs that were input.
    cout << str1 << " <  " << str2 << ": " << (str1 <  str2) << "\n";
    cout << str1 << " <= " << str2 << ": " << (str1 <= str2) << "\n";
    cout << str1 << " >  " << str2 << ": " << (str1 >  str2) << "\n";
    cout << str1 << " >= " << str2 << ": " << (str1 >= str2) << "\n";
    cout << str1 << " != " << str2 << ": " << (str1 != str2) << "\n";
    cout << str1 << " == " << str2 << ": " << (str1 == str2) << "\n";

    return 0;
}
```

String Functions

```
///  C++ sting, find() example
///

#include <iostream>
#include <string>
using namespace std;

int main()
{
    // Prompt, input
    cout << "Enter two strings: ";
    string first, second;
    cin >> first >> second;

    // Search for second string in the first string
    int idx = first.find(second);

    // Check result
    if (idx != string::npos) //string::npos is the value in strings
                             //meaning failure
        cout << second << " occurs in " << first << " at index " << idx << "\n";
    else
        cout << second << " doesn't occur in " << first << "\n";

    return 0;
}
```

String Functions

```
///  C++ sting, substr() example
///

#include <iostream>
#include <string>
using namespace std;

int main()
{
    // Prompt, input
    cout << "Enter a string, a start index, and a length: ";
    string  str;
    int start, length;
    cin >> str >> start >> length;

    // Check for length error
    if (start + length > str.size())
    {
        cout << "Substring specified exceeds the string length, exiting.\n";
        return 1;
    }

    // Get the substring
    string substring = str.substr(start, length);

    // Output the substring
    cout << "Substring is: " << substring << "\n";

    return 0;
}
```

Example

Problem: Read in a sentence of words and print out the first word alphabetically in the sentence, and its position in the sentence. The sentence is complete when the user types a “.” solution