CS 16 5/23

# Midterm Overview

* 7a was the hardest problem, Lawton graded it personally
	+ 7a was about comparing times
	+ Solution (one way): if(a.hour > b.hour) return a is later; else if(a.hour == b.hour && a.minutes > b.minutes) return a is later; else a is not later than b.

# Helper Functions

* Recall from the last lecture’s notes that these are used to encapsulate more information, so you only have to deal with the problem abstractly.
* There is an example of a helper function included with my notes from the last lecture.

# Recursion

* Base case: problem of size 0 or 1.
* Recursive case: Solve a piece of the problem and subsequent variations of it until you have solved the whole problem.

Example of Recursion:

#include <iostream>

#include <string>

using namespace **std**;

string **reverse**(string *a*){

 int len = a.**length**();

 string recurse;

 if(len == 1){

 return a.**substr**(0,1);

 }

 recurse = a.**substr**(0, len - 1); *// This is the rest of the string except the last element*

 return a[len-1] + **reverse**(recurse);

}

int **main**()

{

 string m = "Hello World!";

 string reversed;

 cout << "Before reversing the string: " << m <<'\n';

 reversed = **reverse**(m);

 cout << "After reversing the string: " << reversed <<'\n';

}

Output:

Before reversing the string: Hello World!

After reversing the string: !dlroW olleH

# Strings(C++ and C)

* C-Strings
	+ Type char \*
	+ Terminate with ‘\0’
	+ Declaration of a C-String
		- Char s[] = “Hello this is a C-String”;
		- Char s[] = {‘S’,’t’,’a’,’r’,’t’,’\0’}
			* Notice that we include the ‘\0’ character in the bracketed version.
			* The double quotes version (Hello this is…) implicitly adds the ‘\0’ character for you
* C++ Strings
	+ Type string
	+ Can be indexed into like an array
	+ Use name\_of\_string\_variable.length() to get the length of your string
	+ Lots of useful functions in the c++ string library to help with string processing
		- Isalpha(), isalnum(), substr() …