

Lab 7

In order to get credit for the lab, you need to be checked off by the end of lab. You can earn a maximum of 3 points for lab work completed outside of lab time, but you must finish the lab before the next lab. For extenuating circumstances, contact your lab TAs and Jennifer Parham-Mocello.

(2 pts) Worksheet

In small groups, trace the following code. What will each piece of the code do or print? (Show your work, **they will all compile**).

```
(1)    int a(int i){
        return i + a(i-1);
    }
    cout << a(4);
```

```
(2)    int b(int i){
        if( i == 0 )
            return 1;
        return i * b(i-1);
    }
    cout << b(4);
```

```
(3)    int c(int i){
        if( i > 3 )
            return 1;
        return i + c(++i);
    }
    cout << c(0);
```

```
(4)    int c(int i){
        if( i > 3 )
            return 1;
        return i + c(i++);
    }
    cout << c(3);
```

In small groups, answer the following question from your conceptual point of view, and as a class with your **TA, justify these concepts using code**.

1. Compare and contrast references and pointers in C++.
 - a. How they are created?
 - b. How to access another place in memory?
 - c. Can we access the memory location of pointers/references?

(2 pts) Practice Recursion

The Fibonacci numbers are F_0 is 0, F_1 is 1, F_2 is 1, F_3 is 2, F_4 is 3, F_5 is 5, F_6 is 8, etc.

$$F_{i+2} = F_i + F_{i+1} \text{ for } i = 2, 3, \dots; \text{ where } F_0 = 0 \text{ and } F_1 = 1$$

Write a recursive function called, **fib_recurs()**, that takes one parameter, n, of type int, and returns the nth Fibonacci number, F_n .

Show your TA a trace through the solution for n=5.

Fibonacci Application: You're standing at the base of a staircase and are heading to the top. A small stride will move up one stair, a large stride advances two. You want to count the number of ways to climb the entire staircase based on different combinations of large and small strides. For example, a staircase of three steps can be climbed in three different ways: via three small strides or one small stride followed by one large stride or one large followed by one small.

How could you apply the Fibonacci number series to this problem? What are the base cases for counting the ways you can climb stairs by going one stair or two stairs at a time? How many different ways can you climb 4 stairs? 5 stairs?

(3 pts) Pointers in functions. Design first (1.5 pts), then write code (1.5 pts)

To help you practice functions and pointers, you will write a short program that asks the user to enter a string (**get_string()**), makes a copy of the string, takes the copy and changes all non-space letters to dashes (**set_replace_string()**). The program then gets a letter from the user to search in the original string and replace the dashes in the copy string with the letter found, returning the number of letters found (**get_search_replace()**).

You should design first...

You can have more functions, but you must have at least the three below with the EXACT function prototypes.

Each of your functions should have the proper function headers/descriptions

```
void get_string(string *);
void set_replace_string(string, string *);
int get_search_replace(string, string &);
```

Write the function headers/descriptions for each of the functions above, as well as all the functions you create! This includes information about parameters, return values, and pre/post conditions.

Design – Give as much detail as possible for the main function and all the functions above.

- Why do the function prototypes have the specific parameter types on them?
- Why do the functions have void or a return value?
- How do all the functions interact together?

Testing – Provide testing values with expected results.

- What do you plan to use as bad values? Good values?
- What do you expect to happen with bad values? Good values?

If any of your functions are more than 10 lines of code, what can you do to make it smaller? If you are having difficulty thinking of how to make it smaller, then ask a TA to help you!!!

STOP!!! Get checked off by a TA before beginning to implement. This will help with logic and function mistakes.

Now, INCREMENTALLY write your functions and program!!!!

Show your completed work to the TAs for credit. You will not get points if you do not get checked off!