**CS173: Intermediate Computer Science**

**Reading 5**

Name: \_\_\_\_\_\_\_\_\_Daniel Lee\_\_\_\_\_\_\_\_\_\_\_

Read the assigned pages below from our course textbook. Complete the responses to the questions in this document and then save as a docx or pdf file. Submit your work by the assigned deadline on the Canvas course page or in class. Responses may be neatly handwritten or typed. **Put your name at the top!**

Readings: From the course textbook please read Chapter 8. You can skim over the parts on GUI design.

You should come away with understanding:

* how functions are declared and defined
* the different ways variables are passed to functions
* software engineering techniques like assert

**1) In the principle of *functional decomposition* explain the role of the *interface* vs the *encapsulation*.**

The role of the interface is to provide a formal description of what a subprogram does and how we communicate (invoke) with it.

The role of the encapsulation is to hide a module implementation in a separate block with a formally specified interface.

**2) What is the difference between a function’s *arguments* and its *parameters*?**

A function’s arguments are the items listed in the call to a function.

On the other hand, a function’s parameters are the variables declared in the function heading and they are the internal view of the function interface.

**3) Some people think it is ok to write function definitions without function declarations. In what specific situation is it impossible to write function definitions without using at least one function declaration?**

If there is a situation where there are two functions and they call each other, it is impossible to write function definitions without using at least one function declaration because based on the order of those two function definitions, the preceding function will invoke the following function, which will eventually generate a message that the following function is undeclared.

**4) What is a *local variable* and why is it referred to as local? Answer both parts!**

A local variable is a variable declared within a block and not accessible outside of that block.

The reason why it is referred to as local is because they are accessible only within the block in which they are declared.

**5) What is a good style choice for naming functions?**

A good style choice for naming functions is to make it sound like an imperative verb or have an imperative verb as part of it to make it sound like a command to the computer.

**6) Explain the difference between a parameter that is passed-by-value from one that is passed-by-reference. Explain both the syntactic differences and also the semantic ones.**

A parameter that is passed-by-value is a parameter that receives a copy of the value of the corresponding argument and the syntax is simply stating the data type name and the variable name just like the default parameter declared in the function heading.

On the other hand, a parameter that is passed-by-reference is a parameter that receives the location (memory address) of the caller’s argument, and it is declared by adding an ampersand (&) to the data type name.

**7) What is the purpose of the *assert()* statement in a c++ program? Also give an example of how it might be used in the following function:**

The purpose of the assert() statement in a c++ program is to catch errors in the program for debugging by a highly visible error message.

int compute ( int array[], int size )

{

// we expect size to have a value > 0

assert(size > 0);

...

return 0;

}