Assignment: Week 4 Articulate: *Abstraction*

Course: Programming with Classes (CSE 210; Section A5)

Professor: Eric Carter

Submission Due Date: Saturday, May 14

Submitted By: Andrew Palmer

Student ID: 886695731

**What is abstraction and why is it important?**

To understand what is meant by the term *abstraction* a look at its root word is helpful. In one definition of *abstract* we find that it is “a summary of the contents of a book, article, or formal speech” (Oxford online dictionary, 2023). Similarly, *abstraction* implies that a more complex idea or set of ideas is taken and made simpler through a process of summarization.

In computing, typical examples of this concept is in the use of classes and functions. For example, a class might contain a large number of member variables, and then also define several functions related to that class, even pinpointing needed parameters, etc. All of this occurs within the class itself, which is its own program. For example in this week’s programming assignment, I have a class entitled *entry.cs,* and it contains a function called *WriteEntry*.

The call of the class function is when the benefit of abstraction is seen. Simply put, rather than having to either rewrite the entire class, and/or to memorize and reiterate all the variables that went into that function, a summary command can be made that effectively represents or summarizes what was previously stated. This then makes for a simpler and cleaner computer code to read, as well as more easily understood and less error prone.

To drive home this point, let’s look at one other quick example of abstraction: the simple naming of variables. Suppose we wanted to introduce a string variable called *person*, and the string associated with that variable initially was to be ‘carl.’ The coding to do so would be: *string person = ‘carl’.* However, we wouldn’t need to rewrite that every time we wanted to reference *person*. In fact, because of the principle of abstraction, we could simply write *person*, and the computer would already know it was a string variable, and that (unless we updated it), it would be ‘carl’ in value.