Week 4: Discussion

By: Clinton E. Kline  
11-2-2021

Instructor: Quentin Miller

CIAT – ASD102

Abstract

Here is a list of object-oriented programming terms (<http://en.wikipedia.org/wiki/List_of_object-oriented_programming_terms>). Select 5 terms you have learned thus far and explained how you used them since you started class.

**Keywords**: C++, Programming, pseudo-code  
  
Week 4: Discussion

Cohesion – as our aforementioned source – Wikipedia – states, Cohesion is the degree to which the elements inside of a module belong together. That seems super vague to me, so I’ve read further, and I think my best perhaps slightly less vague interpretation right now would be:

Cohesion: the simplification and compartmentalization of the methods inside of a class by making each method as functionally basic and as independent as possible in order to promote the maintainability, modularity and reusability of the methods.

I think each of us uses cohesion in different and ever-evolving ways. I notice that as I’ve learned more and more about programming over the last year that the benefits of simplification and compartmentalization present themselves often. With each app I write I find myself trying to avoid things like repetition and redundancy. I try to require the minimum number of parameters possible for each of my methods, and I have certainly benefited from the reusability this practice provides as I have managed to scrape together a small library of basic and common functions, some of which I have used many times. I would also add that cohesion is very similar to some other words on our list of terms, Coupling, which also refers to how interconnected and dependent the different components of a class are on each other with the distinction that coupling refers to practice on a smaller scale. Rather than looking at the relationships between modules, coupling is used to describe what happens inside of a single module. Also related is the term Modular Programming which is more synonymous to cohesion than coupling may be.

Recursion – This is a concept that still seems somewhat mystical to me at times. Recursion refers to writing a method so that it refers back to itself in some way. There are other much more in depth and complicated ways to use recursion, but the most common way I use recursion is to loop a function back over itself when I want to accept multiple entries or repeat a function several times.

Mutable variable – Essentially a mutable variable is a variable that can modified after creation, as opposed to an immutable variable or constant which cannot be changed. I use mutable variables much more often than immutable variables, but they both certainly both have their place and serve purposes. I typically use immutable variables to represent things like restrict output to certain ranges or to assign a numeric value to a variable in order to create a constant value like pi or some other value that will remain the same throughout your program.

Void type – Void type functions are functions that don’t return a value. Thinking back I think I use these functions more than any other function type. I find them especially handy for sending data from one function to one or more other functions, printing output to a user interface or file, or changing the value of a variable in the case of incrementation.

Member variables - member variables listed at the top of the file, just inside of the class but outside any methods, yet are available to all methods inside of the class. These variables can be shared with an unlimited number of functions, all while retaining the same source value. That value can be manipulated and changed throughout the rest of the program, but each reference to it is referencing the same data from the same memory location.

References:

1. Wikimedia Foundation. (2021, June 17). *Index of object-oriented programming articles*. Wikipedia. Retrieved November 3, 2021, from https://en.wikipedia.org/wiki/Index\_of\_object-oriented\_programming\_articles.