Topic: For this module’s discussion board assignment, select at least two (2) of the topics below. In your writing, be sure to explain the what, how, and why of the selected topics. If necessary, provide code examples to future illustrate your thoughts.

1. Field Encapsulation
2. Objects with Methods and Methods Receiving Instances
3. A Collection of Object Instances
4. The "this" Reference
5. Variable Scope

Encapsulation is the act of enclosing data within an object. The object can be a class, method, attribute, etc. This is a fundamental concept in object-oriented programming that is used extensively to control how and when data is accessed. Field encapsulation specifically refers to the act of encapsulating an attribute within a class. For example, if you have a private attribute that is only accessible using a getter method you have created a field encapsulated object. Field encapsulation is very common as it helps programmers restrict how and when data can be modified from outside the Class construct. Think of it using this example from the book:

*“Consider, for instance, building a computer system. Your personal computer has many components—a CPU, memory, disk, motherboard, fan, and so on. Each component can be viewed as an object that has properties and methods. To get the components to work together, you need to know only how each component is used and how it interacts with the others. You don’t need to know how the components work internally. The internal implementation is encapsulated and hidden from you. You can build a computer without knowing how a component is implemented.” (Liang, 2020)*

Variable scope is another fundamental object-oriented programming concept. Variable scope is the term used to describe when and where a variable can be accessed. Consider the difference in when/where you can access class vs local variables. A class variable can be accessed from anywhere in that class, while the local variables can only be accessed from within the method they are declared in. This can go even deeper. For example, if you declare int I = 1 in a *for* loop, that variable will only exist within the *for* loop and not be accessible outside of it. Understanding variable scope is crucial for knowing when and how to declare variables.

Reference:

Reference:

Liang, Y. D. (2020). Introduction to java programming and Data Structures. Pearson Education, Inc.

Response 1:

Anitha,

I try to avoid creating multiple variables with the same name. The *this* modifier can help clarify, but I still find it hard to understand. Using your example:

public void setName(String name)

I would rather do it like this:

public void setName(String newName)

I still use the *this* modifier when assigning (Eg. this.name = newName), but that is mostly out of habit.

Christopher,

Variable scope was a concept that got me bashing my head against the wall when I first learned Java in undergrad. I have found that clean code can help a lot with keeping track of what scope each variable has. Naming standards, such as putting a l in front of local variables, can help as well.