In the past week, we focused on Static Variables and Member variables. From the learning resources

and activities, I have learned helpful concepts surrounding the aforementioned focus.

Firstly, I understood that static methods can be called without creating an object for the class where

the method resides. The class name is enough to call the method, that is, the caller must first to the

class then using the dot (.) operator, point to the method name. For example, ClassName.method(),

unlike ClassName object = new ClassName, then object.method() – which is typical with void

methods. I also observed that static methods require a return type (statement).

Like other methods, I have come to understand that static methods have to be defined before they

are called. Method definition requires specification of the access modifier (public or private), static or

void, method name, and parentheses with or without parameters. For example, public static int

add(int a, int b){} – between the two curly braces (method body) goes the implementation (function

code). About parameters, I have learned that there are formal and actual parameters. Considering

the add(int a, int b) method, int a and int b are formal parameters in that they are set to accept

values passed by actual parameters (literal, expression or variables) to use in the implementation of

the method. Note that if the method is defined to accept integers in the parameter, then the caller

must supply integers. Additionally, parameters can be static member variables, these are like

constants declared in the class, depending on their declaration, they can be public or private.

For the method to be used, the caller (a class, object, another method, or recursively, the same

method) calls the method to action. Since this is a static method concept, the method is called by

pointing the class where it resides. This necessitates the concept of Black-Box, my understanding of

this concept is that the code or implementation of the method is hidden from the caller. The caller

only gets to see the impact of the implementation and is not interest in the code. For example, when

operating a printer (machine), the user is less or not interested in how the printer prints, but is

interested in the output. The understanding here is that the Black-box received input and then it

processes it (how it does it, it does not matter) and then output. This concept gave me a lot of ideas

as I will be working on projects in Java.

Comprehensively, methods are subroutines that make the implementation of a program more

efficient. Coupled with the skills I have gained in the previous Units, I feel more comfortable meaning

code in Java Programming Language, especially the Object-oriented aspect. In fact, the assignment,

was one of such motivations.

This week I have received positive feedback from peers, at the same time, going through and

accessing my peers’ work was very good and challenging.

In this week’s activates, I learned about more complex control pathways, including the switch statement,

and the try and catch statement. The try and catch statement was interesting. At first it seemed

counterintuitive to me, to try to catch an exception, rather than just code the program not to produce

the exception in the first place. However; upon researching some examples online, the utility of it

became a bit clearer.

I feel that the tools gained during the last few weeks are just the building blocks of much more complex

algorithms and combinations of code, that will allows us to start building more complex, highly

functional programs. I am excited for this, and hope I can keep up with my already busy schedule.

On that note, I actually feel that I am not keeping up with the content as well as I could be, and could go

through the practice examples in the textbook, and the material, more thoroughly. I am not having any

difficulty with the programming assignments so far, however; they seem to be quite simple, and only

incorporate two to three concepts discussed in the chapter. This is unfortunate as I am a much more

kinesthetic learner. If I simply read about programming in the textbook, but don`t apply it, I fear I will not

remember or learn much of the concepts.

I hope to do a better job of studying in the coming weeks. However; it will be a challenge as I will be

backpacking through Europe for two weeks. I will at least make sure to keep on top of the reading, but

may not be able to do as much programming during the time.

I hope to start seeing the potential of the Java language. So far, it seems a great deal more complicated

than Python, but not actually capable of much more. However; I know that Java as a language is

considered to be one of the best. So I will keep an open mind.

Total Words (400)