**Explain to another newbie how to apply algorithmic design and data structure techniques in developing structured programs.**

The algorithm is the step-by-step procedure that defines the output. How that data is organized is the data structure. You want your algorithm to be clear and unambiguous. Algorithms need to terminate after a finite set of steps. They should have clearly defined inputs and outputs to obtain the desired results. Your algorithms in developing structured programs should function independently of other programming code and called on as needed.

**Are some algorithms and data structure designs better than others?** **If so, explain why one design would be used before another design would be used.**

The answer to this question is simple, *YES*, but which ones are better, depend entirely upon what is being accomplished. There are two goals within a computer program concerning algorithms: “to design an algorithm that is easy to understand and code, and design an algorithm that makes efficient use of a computer’s resources” (Shaffer, 2013). While the answer to the question is simple, determining which are best is more complicated, as there is any number of influences and possible solutions for whatever task is to be accomplished.

With that in mind, we turn to the efficiency of an algorithm regarding the task at hand. In the world of computer science, algorithmic efficiency is measured in two different complexities, time complexity, and space complexity.

Time complexity is defined as “the number of instructions which a program executes during its running time” (Ziegler, 2004). Time complexity is an important factor because we want a program to execute as quickly and efficiently as possible.

Space complexity is precisely what it sounds like, the amount of space required to contain and execute the data.

**References**

Shaffer, C. (2013, March 28). *Data Structures and Algorithm Analysis*. Edition 3.2. Retrieved from: http://people.cs.vt.edu/~shaffer/Book/JAVA3elatest.pdf

Ziegler, J. (2004). *Time Complexity, space complexity, and the O-notation*. Retrieved from: http://www.leda-tutorial.org/en/official/ch02s02s03.html