FUNDAMENTAL CONCEPTS IN PROGRAMMING

When you do programming, you create a program (software) that does what you want.

A program (software) is basically a list of instructions that tell the computer what to do.

You write instructions using a programming language that your computer system can understand.

Kinds of Software Programs

There are two (2) kinds of software programs: (1) stand-alone and (2) built-in (add-in or imbedded). A program is so classified based on whether it stands alone or is built into another program.

For example, Notepad (a text editing program) is a stand-alone program because it is a stand-alone file named *notepad.exe*. By contrast, the programs you create when doing MS Word programming are often built-in (add-on) programs in MS Word.

Components to a Software Program

There are two (2) components to a software program: (1) the interface and (2) the code. The interface is what you see and interact with. The codes are the lines of instructions you write to control your program.

Parts of Software Programming

There are two (2) important parts of software programming to consider: (1) data and (2) algorithms.

Data means the kinds of facts you decide to work on i.e. to process by means of your algorithm.

Algorithms are the logical steps or procedures that your data passes through to get processed. Before actual coding, you can describe an algorithm using pseudocode.

Important Elements of Software Programming

Two very important elements of software programming are variables and statements.

Variables. Variables are programming elements used to hold data for easier manipulation in your software program. Each variable should have a name. There are rules and guidelines of best practice on how to name variables. For example, you cannot use a "reserved word" (a word that belongs to the vocabulary of a programming language) to name a variable, and you should give a variable a descriptive name to that you and others can easily know what data it represents, etc.

Statements. Statements are expressions/instructions, etc. to tell the software program to do things when the program is run. For example, the conditional "if… then" statement forms a logic branching that directs program execution. It tells the program to do something if a certain condition is met.