**Interface Purpose:**

The INTERFACE statement is the first statement in an interface block. The interface block is a powerful structure that was introduced in FORTRAN 90. When used, it gives a calling procedure the full knowledge of the types and characteristics of the dummy arguments that are used inside of the procedure that it references. This can be a very good thing as it provides a way to execute some safety checks when compiling the program. Because the main program knows what argument types should be sent to the referenced procedure, it can check to see whether or not this is the case. If not, the compiler will return an error message when you attempt to compile the program. Also with just a slight variation, it can then be used to create generic functions for use inside of a program unit.

You begin the interface block with the INTERFACE statement which should be placed right after all of the non executable statements (program, use, implicit, type declarations, etc.) in a program unit. Note that the intent of the dummy arguments should also be included.

To extend the use of the interface block for defining generic functions requires only a few small steps beyond what you have already learned. First, instead using just the plain interface statement, you use an interface statement that includes the name you wish to give to the generic procedure.

**Difference between subroutine and function:**

The purpose of a function is to take in a number of values or arguments, do some calculations with those arguments and then return a single result. You will want to use a function if you need to do a complicated calculation that has only one result which you may or may not want to subsequently use in an expression. Subroutines, on the other hand, can return several results. However, calls to subroutines cannot be placed in an expression. In the main program, a subroutine is activated by using a CALL statement which include the subroutine name followed by the list of inputs to and outputs from the subroutine surrounded by parenthesis. The inputs and outputs are collectively called the arguments. A subroutine should perform an operation. Thus, properly used, subroutines tend to be longer and more complicated than functions.