As discussed previously, a FORTRAN compiler has built within it certain intrinsic functions. These functions are built in because of the frequent use they find in calculation procedures. A programmer may find it useful to construct other functions which may be used repeatedly in a given program. The function may involve one line of calculation procedure or several lines. It may be desired to have just one value of the function or a whole array of values. These procedures may be accomplished with functions and subprograms.

There are four types of programming entities that come under the category of functions and subprograms. These are:

1. Intrinsic or built in functions
2. Statement functions
3. Function subprograms
4. Subroutines

The built in function has been discussed in another tutorial and will not be repeated here.

**Statement functions.** When one value of a variable is desired and only one line of computation procedure is required, a statement function can be used. All statement functions are placed at the beginning of a program before any executable statement. The form of the statement function is as follows:

\[ A(\text{list}) = B \]

A is the name of the function and is named by the rules used for variable names. If A begins with a letter from I-N, then the function type is an integer. The 'list' must contain the dummy variables that will be used in the calculation. B is a valid FORTRAN mathematical expression containing various operands and variables (including the dummy variables). An example of a statement function is:

\[ PV(X,Y,N)=X/(1.+Y)^N \]

PV is the name of the function. It will take on a real value since it begins with the letter P. It contains three dummy variables, X, Y, and N. The expression on the right side of the equal sign uses the three dummy variables in a mathematical operation. Now suppose that a program used the above function in the following sequence of steps:

\[ \text{DO 3 I = 1, 20} \]
\[ 3 \quad \text{SUM = SUM + PV(CF(I),RATE,N)} \]
When the program comes to the DO loop, everytime statement number 3 is executed, a value of PV will be calculated with the current values of CF(I), RATE, and N assuming the position in the defining mathematical expression for X,Y, and N, respectively. Note that the variable type must always be in agreement between the function statement and its use within the program. For example, X and Y in the function definition are both real variables, then CF(I) and RATE both need to be real variables. Statement functions are unique to a given program.

Function subprograms. The function subprogram is very similar in concept to the function statement except that it involves a much more complicated set of calculations so that more than one line of instruction is needed. The function subprogram still returns to the main body of the program only one value of a variable. The subprogram is a separate program-like entity which is placed after the END statement of the main program. It contains its own variables, statement numbers, DIMENSION statements, etc. as required. The last statement is an END statement and nearly always the next to last statement is a RETURN. The RETURN simply says - return the execution to the main program. If more than one function subprogram is being used for a program, then they simply follow one after the other immediately after the END statement of the main program.

The first statement in a function subprogram defines the function name and type, i.e. real or integer.

FUNCTION NAME(X,Y,Z)

Again X,Y, and Z are dummy variables used in the calculation procedure. As an example of a function subprogram, consider the calculation of the average of a set of numbers.

FUNCTION AVE(X, N)
DIMENSION X(N)
SUM = 0.0
DO 10 I = 1, N
   SUM = SUM + X(I)
10  CONTINUE
AVE = SUM/N
RETURN
END

You should note several things. The independent variables are sent down in an argument list. In the above case, the argument list consists of a dimensioned real variable X and an integer variable N. Since X is a dimensioned variable, it must be dimensioned in the subprogram. In the case of a subprogram, the number of dimensions can be given using a variable name (N) instead of a number, as long as that number is an integer and is in the argument list. The name of the function
(AVE) must be the name of the variable which you are calculating. You must put in a RETURN and an END. Each such subprogram can be written before or after the main program or other subprograms, but it cannot be in the middle of a main program or other subprogram. Once this function subprogram is written, it is accessed by using its name. For example, suppose you wanted to calculate 3/2 of the average in the main program. You could do it as follows.

```
PROGRAM CALC
  DIMENSION Y(25)
  OPEN (UNIT = 12, FILE = 'NUM.DAT', STATUS = 'OLD')
  READ (12, *) M
  DO 10 I = 1, M
    READ (12, *) Y(I)
  10 CONTINUE
  CLOSE (UNIT = 12)
  TTHAV = 3.0/2.0*AVE(Y, M)
```

Notice that the variable names do not have to be the same in the main program as they are in the subprograms. They must, however, be in the same order and be of the same type. That is, Y must be a real dimensioned variable and M must be an integer. A dimensioned variable must be declared as such in both the main program and the subprogram. Function subprograms can be as complex as needed. Some are thousands of lines long. They can contain DO LOOPS, IF, THEN, ELSE blocks, and may even call on other subprograms that you have written. They cannot call on themselves, however. The name of the function must be real (i.e., start with a-h, o-z) if you wish to calculate a real number such as an average. It must be an integer (start with i-n) if you wish to calculate an integer. You can of course override the default values if you declare the name of the function as real or integer in both the main and the subprogram.

**Subroutines.** Subroutines are used when it is desired to return to the main program more than one value of a variable or more than one variable. The basics of a subroutine are very similar to the function subprogram. There are two main differences.

1. There is no value associated with the name of a subroutine which means there is no significance to the first letter in the name. Otherwise the naming of a subroutine follows the procedures for the function.

2. A subroutine is called from the main program with a CALL statement. The format for the CALL is as follows:

   (more will come later on Subroutines)