# Local & Global Identifiers

Procedures are often referred to as sub-programs. Just as programs have their own variables and constants, procedures can also have their own variables and constants. When you define an identifier in a procedure it can only be used from the point where it is defined until the end of that procedure. When you define an identifier outside a procedure it can be used anywhere in the program after it has been defined (including procedures with some exception we'll see later). Identifiers defined inside procedures are known as *local* identifiers. They can only be used in the procedure, so they are local to the procedure. Identifiers defined outside procedures are known as *global* identifiers. They can be used throughout the program so they are available "globally" to the program.

Here is a procedure that modifies the global variable number. Notice we can access it from either inside the procedure or outside.

|  |  |
| --- | --- |
| |  | | --- | | **var** number : **int** := 0  **procedure** countCalls  number := number + 1  **put** "The procedure has been called ", number, " time"..  **if** number **not**= 1 **then**  **put** "s"..  **end** **if**  **put** "."  **end** countCalls  *% main program*  **put** "Calling procedure. Currently number is ", number, "."  countCalls  **put** "Calling it again with number now ", number, "."  countCalls  **put** "One more call with number = ", number, "."  countCalls | |

It produces the following output:

|  |
| --- |
| Calling procedure. Currently number is 0.  The procedure has been called 1 time.  Calling it again with number now 1.  The procedure has been called 2 times.  One more call with number = 2.  The procedure has been called 3 times. |

Here is an example of a procedure with some local variables. This procedure will ask the user for three (a constant we could change) values and compute the mean of those values.

|  |  |
| --- | --- |
| |  | | --- | | **procedure** findMean  *% local variables for finding mean*  **var** number, sum : **int** := 0  **var** mean : **real**  *% local constant, just change this 1 value to read a different # of values*  **const** HOW\_MANY := 3    **put** "I'm going to ask you for ", HOW\_MANY, " values."  **put** "I will find the mean of those numbers"  *% note i is local too, but it can only be used in the for loop*  **for** i : 1 .. HOW\_MANY  **put** "Enter value # ", i  **get** number  sum := sum + number  **end** **for**  mean := sum / HOW\_MANY  **put** "The mean of your ", HOW\_MANY, " numbers is ", mean, "."  **end** findMean  *% Program starts here*  findMean  **put** "That was fun lets do it again!"  findMean | |

Here is a sample run of the program:

|  |
| --- |
| I'm going to ask you for 3 values.  I will find the mean of those numbers  Enter value # 1  100  Enter value # 2  300  Enter value # 3  200  The mean of your 3 numbers is 200.  That was fun lets do it again!  I'm going to ask you for 3 values.  I will find the mean of those numbers  Enter value # 1  70  Enter value # 2  74  Enter value # 3  73  The mean of your 3 numbers is 72.333333. |

Things can get a little more complicated if you have local identifiers with the same name as global identifiers. For instance, suppose a procedure has a variable named count and that there is also a global variable with the same name. The local variable will be used in the procedure and the global one is not visible there. The global one will be used everywhere else. Here is an example:

|  |  |
| --- | --- |
| |  | | --- | | **var** number : **int** := 10 *% a global variable*  **procedure** locally  **var** number : **real** := 3.1415 % *a local variable*    **put** "In the procedure number = ", number  **put** "What would you like to change number to"  **get** number  **put** "At the end of the procedure number = ", number  **end** locally  *% program starts here*  **put** "The program starts and the global number is ", number  locally  **put** "Now that we are out of the procedure our global number is still ", number | |

An example run of the program is:

|  |
| --- |
| The program starts and the global number is 10  In the procedure number = 3.1415  What would you like to change number to?  **-9999.99**  At the end of the procedure number = -9999.99  Now that we are out of the procedure our global number is still 10 |

It is generally best to avoid using global variables in procedures whenever possible (and it is almost always possible). This will prevent a procedure from accidentally changing a variable that you were using for some other purpose elsewhere in your program. When you avoid using global variables in procedures it also helps avoid confusion if there are local and global variables with the same name.