Lord DefCo' Guide to programming in QBASIC

By: The DefCo Press
2000

Vol. 1 Issue 3

<u>* Disclaimer ***</u>**

This document is herein presented without warranty of any sort, any damage done to you or your system whether intentional or consequential from the use of this information is your responsibility. The DefCo press will not be held responsible. Further reading of this disclaimer establishes that you have read it and that you will abide by it. This document is the sole property of the DefCo press, however, you may distribute it as long as it is left intact, complete and unchanged; and that this disclaimer is left unaltered. If you want to use this document for a different purpose than self-education, please contact the DefCo press for information on our releases.

THANK-YOU,

Lord DefCo

=-

Lecture: Program control

Right now, if you had to write a program to display “Hello” on the screen you would have to type the print command 20 times! That may not seem like much, but it increases the size of your program, and also if you need to change what it says, you would need to go to each statement and change each one individually, slow. Now we will institute things oftentimes referred to as program controls. If statements and Definite loops are the most basic of these types. In definite loops, the program has control over how many times the loop will cycle before terminating. The command used to implement this is FOR. You will learn more about FOR later in this text. The IF statement is what allows the program to make decisions based on fact, and also to make basic comparisons.

These commands allow the programmer to change the direction of the program statement execution form straight down to whatever suits the requirements for the program best. For clarification, as of now, if you type a statement and then two more, the one you typed first would execute then the next and the next, but with program control, you could set the first to skip the second statement or vice versa.

Command Summary:

Print – Will display text enclosed in “” ‘s on the screen, also displays the value of numeric and string variables.
Rem – Makes any statement after the command appear as a comment (not executed) use comments to remind you of what a variable holds, why a loop is used ect.
Cls **-** Statement that clears the screen of all text, graphics and other things.
End – Statement that brings the execution of a program to a stop
Let – Assigns a value to a variable
For – Prototype FOR <variable> = <Low number usually 1> to <High range number> This command will execute the commands from it to the NEXT statement as many times as <High range number>**-**<Low range number> An optional tag is step <value by which to progress> STEP 2 means count by two and so on. By default this loop will count by one.

Next – Tells the compiler where the for loop ends, usually you would put the variable used in the loop after the next i.e. For x = 1 to 10
Next x

If <condition> [=,<,>,<=,>=] then <action> is the most basic form of the If styatement. The then statement must be on the same line as the If.

Input “Prompt”; or, <variable>

Lesson: Input of numbers

In this lesson, you will learn how to allow the user of your programs to change values in your program to make it more useful. The most basic type of variable is the numeric. To use a numeric variable, you need not declare it. Simply use the command INPUT followed by the variable name. *NOTE, that it the user enters letters, the program will crash, right now you don’t know how to avoid this, just assume for now that the user will always enter a number.* Notice when you run the program, it stops and waits for the user to type something. Also note the question mark was added to the end of whatever you typed as a prompt. Until later, this will ALWAYS happen so just get used to it. If you are confused, look at the sample code at the end of this file to see it in action.

Sample code

```
REM This program will demonstrate the principals of program control
CLS
PRINT “WELCOME TO COUNT4U”
INPUT “TO WHAT NUMBER DO YOU WANT ME TO COUNT (Positive integer only please)”;num
IF x< 1 then END ‘ There is a better way to do this I will show you in a later issue
FOR x = 1 to num step 1
  PRINT x :REM the loop actually leaves the value in the variable during the iteration of the loop
NEXT x
```

Advanced project:

Use the FOR loop to count backwards and see if you can get input to allow more than one variable to be entered at a time. Hint use commas in the input statement as well as during input. Good luck.

Contact Information

As always, if you need help, please contact me at the email address below. Please send copies to your friends, post it on your BBS or place it (gently) on your web site. If you put it on a web site, please tell me. How, you ask? Well I am about to tell you.

To contact the DefCo press, email:

ATTN: Lord DefCo press
Innsoft@yahoo.com