



This is the first of a four-part series for the beginner programmer and those who want to know more about the BASIC language. In this series we will cover many of the commands used in the BASIC programming language as it pertains to the Color Computer.

First, we will cover some of the commands used and what each command does.

### CLS

The CLS command clears the screen of any information that is being displayed. It is especially useful when you want to clear data from the screen to display new information.

Various forms of CLS are CLS $x$ , where  $x$  is a number value between 0 to 8. The number will set the color of the screen. But when typing or printing of data onscreen occurs, the data will still appear black on a light green screen. If no number follows the CLS command, the screen will remain light green but all data will be removed from the screen display.

Table 1 shows the color that will be displayed with the value of  $x$ .

**Table 1**

X	Color
=	=====
0	Black
1	Green
2	Yellow
3	Blue
4	Red
5	Buff (Off-White)
6	Cyan (Blue-Green)
7	Magenta (Purple)
8	Orange

*Dave Ostler is an IC layout designer and the systems manager for a CAD main-frame system. He teaches CAD and electronics at Gailford Technical Community College. Dave is married and has three children, Avis, Chuck and Erik.*

# BASIC for Beginners

## Lesson 1

By David W. Ostler

Try this on your computer. Enter the CLS command and add a number directly behind it. Remember that the syntax (the way that the command has to be for the computer to recognize it) has to be correct. (Example: CLS0, CLS1, CLS2, CLS3, CLS4, CLS5, CLS6, CLS7, CLS8, CLS.)

Enter Listing 1 to see what types of screens your computer can generate by using the CLS command.

## INPUT

The INPUT command allows you to enter variables into the program. These variables may be string variables (A\$, B\$, C\$, CS\$, etc.) or integer variables (A, B, C, AE, etc.). This command is used where more than one character needs to be entered as a variable or can be used as a place to pause a program. Please note that this command will not allow entry of any punctuation such as commas, periods, colons or semicolons.

Proper syntax for this command is INPUT ;x, INPUT "ENTER YOUR NUMBER HERE";x, for integer variables or INPUT "ENTER YOUR TEXT HERE";x\$ for string variables.

Other commands that also allow

variable entry are LINEINPUT and INKEY\$. These commands will be covered later in this series.

## PRINT

The PRINT command allows printing of variables and text strings or forces the cursor down the screen for proper screen printing. This command can also send information to various devices that are connected to your computer via the RS-232 (Serial I/O) plug or to the disk via the disk controller.

It can also be combined with other characters to print, to the screen or to the devices mentioned above, in a particular way.

The proper syntax for this command is PRINT, PRINT "THIS STRING ON THE SCREEN" or PRINT xxx, where xxx is a specifier of where to send the data or how to print the data when it is sent. Table 2 covers some of the specifiers and their use; for more information, refer to your manual on the PRINT command and its various forms.

Of course, there are variations of these commands and other commands not covered here. Refer to your manual for more information on variations and their uses.

Table 2

Specifier	Use
=====	===
PRINT#-1	Sends data to the tape drive
PRINT#-2	Sends data to the printer
PRINT@XXX	Prints data at a particular place on the screen
PRINTTAB	Moves the cursor to the tab location
PRINTUSING	Prints data in a specified format.

## FOR/NEXT

The FOR/NEXT command can be used to set up a delay or force looping for data calculation or conversion.

Loops are classified as two types: conditional and unconditional. Conditional loops have, just like the phrase says, conditions that must be met for the loop to continue and go on to the rest of the program. Unconditional loops will loop and go on to the rest of the program no matter what. The FOR/NEXT commands used in the programs

# BEST

WE'VE CHOSEN THE BEST OF OVER 600 PROGRAMS (OVER 5 YEARS OF ACCUMULATING FINE SOFTWARE), AND PACKAGED THEM FOR YOU. 10 TO 12 PROGRAMS EACH PACKAGE. COLOR COMPUTER I, II OR III. SPECIFY TAPE OR DISK. ONLY \$29<sup>95</sup> EACH PACKAGE!

### #1 Home Mgmt I

Budget  
Checkbook Balancer  
Cost of Living  
Tincalc Spreadsheet  
Electronic Datebook  
Account Manager  
Stock Market  
Word Processor  
Lottery Analyst  
Coco Database  
Coco Terminal  
Bartender

### #4 Adventures

Treasures of Barsoom  
Killer Mansion  
College Adventure  
Coco-Terrestrial  
Escape  
Zector Adventure  
Skid Row  
Quest  
Naughts  
Haunted House

### #7 Utilities I

Cassette Merge  
Coco Monitor  
Tape Analysis  
ML to Data  
High Text Mod.  
Program Packer  
Easy Basic  
Key Repeat  
Full Screen Editor  
ROM Copy  
Basic RAM

### #2 Home Mgmt II

Video Cassettes  
Home Product Eval.  
Elec., Gas & Water  
Baseball Manager  
Car Manager  
Ham Radio Log  
Home Inventory  
Personal Directory  
Recipe Machine  
Disk Labeler  
Password Scrambler  
Disk Dir Print

### #5 Business Helper

Workmate  
Word Processor  
Spread Sheet  
Calendar  
Accounts Receivable  
Accounts Payable  
Income Property  
Mail List  
Small Business Helper  
Stock Charting  
Job Log  
Asset Manager

### #8 Utilities II

Disk Dir Protector  
Dir. Pack & Sort  
Disk Zapper  
Roll Out  
Doss Boss  
Disk Backup  
51'24 Editor  
51'24 Screen  
Autocopy  
Fastsort  
I/O Error Ignorer  
Text Screen Print

### #3 Education

Flash Card  
Spanish Lessons  
Typing Tutor  
Creativity Test  
Anth. Football  
Cost of Living  
Math Tutors 1,2  
Trigonometry Tutor  
Typing Game  
Word Tests  
Talking Alphabet  
Clown Dunk Math

### #6 Games

Trek  
Galactic Conquest  
Warlords  
The Power Sword  
Steps  
Robot Bomber  
Force Field  
Rat Attack  
Caterpillar Cave  
Meteor

### #9 Machine Lang. Tut.

Basic Compiler  
ML Tutorial Pt. 1  
ML Tutorial Pt. 2  
ML Tutorial Pt. 3A, 3B  
ML Tutorial Pt. 4  
ML Tutorial Pt. 5  
ML Tutorial Pt. 6  
ML Tutorial Pt. 7  
ML Tutorial Pt. 8  
MLT Dictionary  
Coco Technical Look  
Coco Technical Look Pts. 1-3



**29<sup>95</sup> EACH SET**  
★ Special This Month ★  
Buy 2 Packages and get 1  
FREE



T & D Subscription Software • 2490 Miles Standish Dr. • Holland, MI 49424 • (616) 399-9648

## THE SOFTWARE HOUSE

A DIVISION OF DATAMATCH, INC.

DS/DD  
PROGRAMMERS  
DISKS



WOW!

10 FOR \$ 5.95  
50 FOR \$28.95  
100 FOR \$49.95

W/SLEEVES, LABELS, W.P. TABS  
MADE IN USA! CERTIFIED ERROR FREE.  
5 YEAR WARRANTY

FLIPPY DISKS 10/\$8.95

FACTORY PUNCHED - USE BOTH SIDES

PRINTER RIBBONS

EPSON MX/RX/FX 70/80	\$5.00 Ea.	6/\$28.00
RED, GREEN, BROWN, BLUE	\$6.00 Ea.	4/\$22.00
GEMINI 10/10X/SG10	\$2.00 Ea.	DOZ/\$22.00
GEM/OKI COLORS	\$3.00 Ea.	4/\$10.00
OKIDATA 80/82/92/93	\$2.00 Ea.	DOZ/\$22.00
C.I.TOH.NEC 8023, APPLE		
DMP/IMAGEWRITER	\$6.00 Ea.	6/\$34.00
RED, GREEN, BROWN, BLUE	\$6.50 Ea.	4/\$25.00

ALL ITEMS 100% GUARANTEED

Add \$2.50 S/H in U.S.A. - Canada Add \$3.50 + \$1.00/LB  
Michigan Residents Add 4% Sales Tax  
Send Check/Money Order Payable to:

**THE SOFTWARE HOUSE**

9020 HEMINGWAY, REDFORD, MI 48239

(313) 937-3442



Send Card Number & Exp. Date

Min. Charge Order \$20.00

for this part of the series are conditional. They are used to put timing delays into the programs.

The proper syntax for this command is `FOR x = a TO b STEP c:NEXT x`, where *x* is the variable dependent upon the loop, *a* is the value (which can be constant or variable) that determines the starting point of the loop, *b* is a value (which can also be a constant of a variable) that determines the end point of the loop, and *c* tells the computer how many value units to count before checking the loop. The `NEXT x` statement forces the loop back to the count. When the count is equal to the value of *b*, the program will proceed out of the loop.

### INKEY\$

The `INKEY$` command questions the keyboard for keystrokes, then checks them against the conditional loops set up within the program to see if the conditions are satisfied. Or it can be used with other commands to pause the program or select values. One use that most of us are familiar with is "PRESS ANY KEY TO CONTINUE".

The proper syntax for this command is `A$=INKEY$`, where *A\$* is the variable string selected to be interpreted by the `INKEY$` command.

### END

This command marks the ending point of the program. When the `END` command is acted upon, the program terminates and you are placed back into the BASIC operating system.

The proper syntax for this command is `END`. This command requires no variables or other conditions for the command to be executed.

### Variable Explanation

This is a numerical value or text string, which can change within the program to allow entry of desired information. This information can be generated within the program or entered by the keyboard. This information may then be manipulated to return values or data information. Examples of variables are found in the listings.

#### Looking at Listing 1

Line 10 clears the screen, moves the cursor down the screen two lines and prints the text "THIS IS A CLS COMMAND".

Line 20 sets up a timing delay so that the screen will not flash on to the next screen before you have a chance to observe it. Please note that the loop will

count from 1 to 1000, and count by ones, before the program will proceed. The `NEXTx` command tells the program that the count has been registered and to continue counting to 1000.

Lines 30 through 180 perform the same functions as those described above, but the `CLS` command has the values attached to it, so that will change

*"Integer variables will not allow any entry other than numbers."*

the screen as directed. (As an example, Line 30 prints the text "THIS IS A 'CLS 1' COMMAND"; Line 40, a `NEXTx` command; Line 50 prints "THIS IS A 'CLS 2' COMMAND"; Line 60, a `NEXTx` command; and so on.)

Line 190 clears the screen, moves the cursor down the screen by two lines and prints the text "THIS DEMO SHOWS THE 'CLS' COMMAND IF YOU WISH TO SEE THIS DEMO AGAIN PRESS (A) ELSE ANY OTHER KEY TO END THIS PROGRAM".

Line 200 sets string variable *A\$* so that the keyboard will set *A\$* equal to the the key pressed.

Line 210 tests string variable *A\$* and checks to see if any key has been pressed; if no key has been pressed, then the program will loop back to Line 200. If a key has been pressed, then the program will proceed to the next program step. This is a conditional loop.

Line 220 tests string variable *A\$* to see if the key pressed was the A key. If the A key is pressed, the program is forced back to Line 10 to start the program again at that point. If the key pressed is not the A key, the program will then proceed on to the next program step.

Line 240 clears the screen and then ends the program.

#### Looking at Listing 2

Line 10 clears the screen, moves the cursor down the screen one line and then prints the text "ENTER YOUR NAME". It also allows the entry of string variable *A\$* from the keyboard. Note: The `INPUT` command will not allow entry of variables with any form of punctuation.

Line 20 moves the cursor down the screen two more lines and then allows the entry of integer variable *A*. Note: Integer variables also will not allow any entry other than numbers, any other type of notation such as commas for

thousands or dollar signs to indicate dollars. It will allow the entry of periods. Periods are treated as decimal points.

Line 30 clears the screen, moves the cursor down the screen two lines and prints the text "WORKING . . . .".

Line 40 does the mathematical manipulation of integer variable *A* (subtracts the integer variable *A* from 87) and sets integer variable *B* equal to that manipulation. It then sets up a timing delay to simulate that the computer is working on the problem. Please note that the loop will count from 1 to 1000, and count by ones, before the program will proceed. The `NEXTx` command tells the program that the count has been registered and to continue counting to 1000.

Line 50 clears the screen, prints string variable *A\$* (your name entered in Line 10), directly following it prints the text string "YOU ARE", directly following it prints integer variable *B*, and directly following that the text string "YEARS OLD". So the entire line reads "your name YOU ARE 35 YEARS OLD".

Line 60 moves the cursor down the screen two lines and prints the text "ANOTHER (Y/N)".

Line 70 sets string variable *B\$* so that the keyboard will set string variable *B\$* equal to the the key pressed. It then tests string variable *B\$* and checks to see if any key has been pressed; if no key has been pressed, then the program will loop back to Line 70. If a key has been pressed, then the program will proceed to the next program step.

Line 80 tests string variable *B\$* to see if the key pressed was the Y key. If the Y key is pressed, the program is forced back to Line 10 to start the program again at that point. If the Y key is not the key pressed, then the program will proceed to the next program step.

Line 90 tests string variable *B\$* to see if the key pressed was the N key. If the N key is pressed, the program is forced to Line 110 to end the program. If the N key is not the key pressed, then the program will proceed on to the next program step.

Line 100 forces the program to go back to Line 70 (this is an unconditional loop). This loop ensures that only the Y and N keys are able to control the program action. Any other keys cause the program to go back to the `INKEY$` routine in Line 70.

Line 110 clears the screen and forces the program to terminate and place control back into the BASIC operating system.

## PROGRAMMING EXERCISES

### Exercise 1

Utilizing the methods presented in listings 1 and 2, write a program that will take any number you enter as a variable and multiply that number by 2.

Note: The figure that your computer uses for multiplication is the asterisk (\*). Example:  $4 \times 3 = 12$  is how we would write it on paper.  $4 * 3 = 12$  is how the computer needs it.

### Exercise 2

Write a program that will allow you to enter your name, street address, city, state, ZIP and phone number as different string variables.

Variables can be any two-character combination. You can use any number of variables within a program in any combination of numerical and/or alpha characters. The only limitation is that you may run out of memory in the computer if you use too many.

The solutions to the programming exercises will be printed in the next installment of this series. Note that there may be many different ways in programming to obtain the same result; in this series we will attempt to use the most straightforward method for simplification.

*(Questions about this tutorial may be directed to the author at 901 Ferndale Blvd., High Point, NC 27260. Please enclose an SASE when writing.)* □

#### Listing 1: CLS

```
10 CLS:PRINT:PRINT"      THIS IS A
   'CLS' COMMAND"
20 FORX=1TO1000STEP1:NEXTX
30 CLS1:PRINT:PRINT"      THIS IS
   A 'CLS1' COMMAND"
40 FORX=1TO1000STEP1:NEXTX
50 CLS2:PRINT:PRINT"      THIS IS
   A 'CLS2' COMMAND"
60 FORX=1TO1000STEP1:NEXTX
70 CLS3:PRINT:PRINT"      THIS IS
   A 'CLS3' COMMAND"
80 FORX=1TO1000STEP1:NEXTX
90 CLS4:PRINT:PRINT"      THIS IS
   A 'CLS4' COMMAND"
100 FORX=1TO1000STEP1:NEXTX
110 CLS5:PRINT:PRINT"      THIS IS
    A 'CLS5' COMMAND"
120 FORX=1TO1000STEP1:NEXTX
130 CLS6:PRINT:PRINT"      THIS IS
    A 'CLS6' COMMAND"
140 FORX=1TO1000STEP1:NEXTX
150 CLS7:PRINT:PRINT"      THIS IS
    A 'CLS7' COMMAND"
160 FORX=1TO1000STEP1:NEXTX
170 CLS8:PRINT:PRINT"      THIS IS
    A 'CLS8' COMMAND"
180 FORX=1TO1000STEP1:NEXTX
190 CLS:PRINT:PRINT" THIS DEMO S
   HOWS THE 'CLS' COM-   MAND IF YO
   U WISH TO SEE THIS   DEMO AGAIN
   PRESS (A) ELSE ANY   OTHER KEY
   TO END THIS PROGRAM"
200 A$=INKEY$
210 IFA$=""THEN 200
220 IFA$="A"THEN 10
240 CLS:END
```

#### Listing 2: VARIABLE

```
10 CLS : PRINT : INPUT "ENTER YO
   UR NAME";A$
20 PRINT : INPUT "ENTER YEAR OF
   YOUR BIRTH";A
30 CLS : PRINT : PRINT "WORKING.
   ....."
40 B=(87-A) : FOR X = 1 TO1000 S
   TEP 1 : NEXT X
50 CLS : PRINT : PRINT A$ ; : PRI
   NT " YOU ARE "; B ;: PRINT "YEAR
   S OLD"
60 PRINT : PRINT "ANOTHER (Y/N)"
70 B$=INKEY$ : IF B$ = "" THEN 7
   0
80 IF B$ = "Y" THEN 10
90 IF B$ = "N" THEN 110
100 GOTO 70
110 CLS : END
```

## THIS IS IT. \*THE HOT NEW PROGRAM FROM BOILING SPRINGS.\*

POLYTINT will enable you to recolor your PMODE3 and PMODE4 images in 16 beautiful colors of your choice. Coloring is quickly and easily done by a friendly new approach. Your new colored pictures are saved in only three to six granules.

SYSTEM REQUIREMENTS. COCO 3. DISK DRIVE. RGB MONITOR.

ORDER POLYTINT FROM: **Boiling Spring Lakes Software,**  
P.O. Box 2536 B.S.L., Southport, NC 28461  
Telephone (919) 845-2881 NC Residents Add 5%.



Money Order or Check \$17.50 Plus \$1.50 Postage and Handling.

