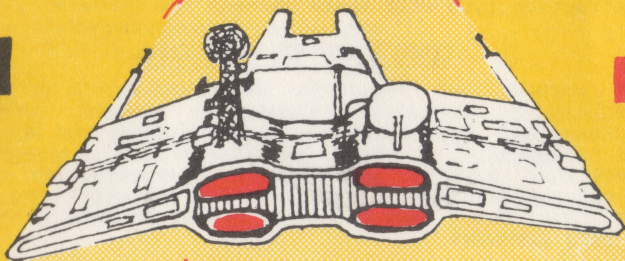# BASIC FOR BEGINNERS

NEW RUN NORMAL END

HELLO CONT PRINT

AUSTRALIAN
EDUCATIONAL
SOFTWARE
FROM

Computer Cognition

A do it yourself course in programming in

APPLESOFT BASIC

BY C. LEGG

COMPUTER COGNITION PRESENTS

BASIC TUTORIAL

This is a course in programming the Apple II computer in APPLESOFT BASIC. Although the course teaches this particular language, it can be used by anybody who wishes to learn programming in BASIC, as there are many similarities between the different forms of BASIC.

The course consists of 7 lessons and is designed to be self instructive. That is, the computer will explain, test give feedback and remedial instruction where neccessary. With these facilities, the computer becomes both an object of instruction (you will learn about the computer), and a medium of instruction (you will learn through the computer).

LIMITATION OF LIABILITY

THE AUTHOR, THE PUBLISHER AND DISTRIBUTORS OF THIS PROGRAM DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, REGARDING THE PROGRAM AND ITS DOCUMENTATION, INCLUDING BUT NOT LIMITED TO ITS PERFORMANCE, ACCURACY, MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE, AND ASSUME NO RESPONSIBILITY OR LIABILITY TO THE PURCHASER OR ANY OTHER PERSON OR ENTITY WITH RESPECT TO ANY LIABILITY, LOSS OR DAMAGE CAUSED OR ALLEGED TO BE CAUSED BY THIS SOFTWARE.

APPLE COMPUTER MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING THE ENCLOSED COMPUTER SOFTWARE PACKAGE, ITS MERCHANTABILITY OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.

GETTING STARTED

To begin your lesson, you need to insert your diskette into drive 1 of the computer, face up. On one side of the diskette, all the tests are stored, and all the lessons are on the other side.

Turn the computer on at the switch on the back. The computer will "boot up" by instructing the disk drive to read appropriate disk file, and after a few seconds, the screen will display the main menu.

LESSON

If you select the lesson side of the diskette, the screen will
display all of the lessons in the course. Each lesson includes
questions at each section of the program, to monitor student
progress, and to give remedial instruction or hints where
necessary. At the end of each lesson is a revision test which
further monitors pupil progress and provides instruction if
any error is made by the student.

At the end of each lesson, the student is returned to the
lesson menu so that you can decide what to do next.

The lesson menu looks like this:

. . . .

BASIC TUTORIAL BY COMPUTER COGNITION

WRITTEN BY COLIN LEGG

LESSON

1) LESSON 1
2) LESSON 2
3) LESSON 3
4) LESSON 4
5) LESSON 5
6) LESSON 6
7) LESSON 7
8) DEMONSTRATION OF COURSE

TYPE THE NUMBER FOR YOUR SELECTION.

. . . .

<u>TEST</u>

If you select the test side of the diskette, the screen now
displays all of the tests in part one of the course. In test
mode, the computer will allow you to complete any desired
test, and will automatically revert to teaching or lesson mode
within any program if you make an error during the test.
Following each test, you are automatically returned to the
test menu, so that you can make your choice of what to do
next. One of the options in the test menu is to go to the main
menu.

The test menu screen looks like this:

. . . .

BASIC TUTORIAL BY COMPUTER COGNITION

WRITTEN BY COLIN LEGG

1) TEST 1
2) TEST 2
3) TEST 3
4) TEST 4
5) TEST 5
6) TEST 6
7) TEST 7
8) DEMONSTRATION TEST

TYPE THE NUMBER FOR YOUR SELECTION.

. . . .

PRACTICE EXAMPLES

****

As you work through this course of instruction, you will find
that you need very little help from this manual. The course is
designed to enable you to interact with the computer, in a
fully interactive mode, so that the computer will provide most
of the instruction and practice required.

The course will require the learner to do some additional
programming without the BASIC TUTORIAL disk in operation.
These examples are usually given in the course, but they are
also given in this manual. It is an idea to consult this
manual after each lesson to take a look at the practice
programs for you to do some homework on.

After each lesson, turn the computer off, remove the diskette,
and then turn the computer on to do the practice exercises.
The disk light will come on and while the disk is spinning,
press the CTRL key, and while holding it down, press RESET.
Then do the practice exercises from in this manual.

DISKETTE REPLACEMENT

The diskette which accompanies this manual contains programs
which are ·all copyright. Any copying of the programs on the
disk is illegal. Anyone found with an illegal copy of the
programs will be prosecuted.

This manual may be copied for the purposes of group study of
the BASIC TUTORIAL.

In the event that the diskette becomes damaged, PLEASE return
it for replacement, with a cheque for $10.00 to:

<div align="center">

Computer Cognition
P.O. Box 2164
North Parramatta NSW 2151
AUSTRALIA

</div>

LESSON ONE
OBJECTIVES

After completing lesson one, the learner will be familiar with the Apple II keyboard and special function keys, and will be able to ring the Apple bell, execute a control-c command and correctly use the esc key to clear the screen with an esc-@ keystroke.

PRACTICE EXAMPLES

1) Press the ESC key, and then press the SHIFT key and P together to clear the screen. This keystroke is called ESC-@.

2) Press the CTRL key and while holding it down, press the G key. this keystroke is called CTRL-G and it rings the bell inside the Apple.

3) Once you have pressed the ESC key you will be able to move the cursor around the screen. The keys which direct the cursor movement are: IJKM.

4) The reset key can be used when you are not running a program. New Apple II computers can be set up to disable the RESET key if not used in conjunction with the CTRL key.

5) Practice using all or any of the keys on the keyboard, when not running a program, as trial and error are very good teachers. You can discover for yourself many things about the computer through being confident enough to experiment.

6) Notice the difference between the zero and the letter "O", and that there is a numeral one on the keyboard. Practice using these keys until you can recognise the difference between a zero and a "O".

7) How many lines are on the screen at once?

8) How many characters can you fit along any one line?

9) Now that the lesson is over, press the reset key and see if your Apple needs the CTRL key to be held down as well.

10) Clear the screen and move the flashing cursor to the bottom right hand location on the screen, keeping the rest of the screen blank.

EXTENSION

1) Type PRINT 3+2 and press RETURN.
2) Type   ? 3+2 and press RETURN.
3) Type   ? "3+2" and press RETURN.
4) Type ?"3+2=";3+2 and press RETURN.
5) **Is there a difference** between the question mark **and the** word "PRINT"?
6) What is the role of the quotation marks?

LESSON TWO
OBJECTIVES

The learners will know the meaning of the following words and will be able to select the correct word from a multiple choice list.

        BASIC
        INTERPRETER
        IMMEDIATE, CALCULATOR OR MONITOR MODE
        DELAYED OR PROGRAM MODE

The learner will be able to correctly use the following commands:

        RUN
        CONT
        PRINT (for strings)

PRACTICE EXAMPLES

1) Type NEW and press RETURN.
2) Type RUN and press RETURN.
3) Type PRINT "COMPUTER COGNITION IS GREAT" and press RETURN.
4) Type HELLO and press RETURN.
5) Type PRINT "COMPUTING IS REALLY GREAT" and press RETURN.
6) Type HOME and press RETURN.
7) Press the RETURN key seven times.
8) Enter the following program into the computer:

    NEW
    10 PRINT "HELLO"
    20 GOTO 10

9) Type RUN and press RETURN.
10) Press CTRL-C
11) Type LIST and press RETURN.
12) Type RUN and press RETURN.
13) Press CTRL-C
14) Type CONT and press RETURN.
15) Press CTRL-C
16) If you have not had enough then go to 1.
17) END.

EXTENSION

1) What does the command "NEW" do?
2) You can break out of a loop by pressing CTRL-C. What other method is there?

LESSON THREE
OBJECTIVES

The learner will be able to correctly use the question mark as an abreviation for the print command. The following PRINT command applications will also be taught:

> FLASH
> SEMICOLON (;)
> COMMA (,)
> NORMAL
> INVERSE
> SPC

The following concepts are included:

> Line numbers
> Mathematical signs + - * / $\wedge$

PRACTICE EXAMPLES

1) Type FLASH.
2) Type PRINT "COGNITION IS THOUGHT".
3) Type NORMAL.
4) Type 10 PRINT"HELLO".
5) Type 20 PRINT "GOODBYE"
6) Type RUN
7) Type PRINT "1+2" and press RETURN.
8) Type PRINT 1+2 and press RETURN.
9) Use the computer as a calculator to answer the following problems:

> 3/1
> 10/3
> 9/3
> 12+6/3
> 4*8
> 2$\wedge$2
> 2/3
> 3$\wedge$2
> 3$\wedge$3
> 3$\wedge$3*2+6
> 5+2*3$\wedge$3

10) Type in the following program:

```
NEW
10 FLASH
20 PRINT "THIS IS INSTRUCTIVE"
30 NORMAL
40 PRINT "THIS IS NORMAL AND INSTRUCTIVE"
50 END
```

EXTENSION

1) Write a program which will fill the screen with the word "hello", printed in inverse.
2) Type SPEED = 1, and then RUN your program.
3) What is the normal Apple speed?
4) How do you print a line of space onto the screen?

LESSON FOUR
OBJECTIVES

After completing lesson four, the learner will be able to use the mathematically correct order of operations to solve mathematical calculations involving the following signs:

ʌ
*
+
-
/
()

and will also be able to use string and numeric variables in their correct places. The commands HOME and REM and the logical counter, c=c+1 will be able to be used appropriately.

PRACTICE EXAMPLES

1) Write a program which will print your name on the screen 5 times.

2) Write a program which will print the following questions , calculate the answers, and print the results:

2+3*7
2ʌ2+2*8
5+7*2+9 2
(2+3)*(3+3)ʌ2

3) Edit the program above so that the screen is cleared before the maths are performed.

4) Write a program which will clear the screen and then print a sentence 10 words long.

5) Have the computer evaluate the following:

   3*Aʌ2 + 2*A - B   if A=2 and B=1

EXTENSION

1) Run the following program:

```
NEW
10 A=0
20 B=0
30 A=A+1
40 B=B+1
50 PRINT B,A
60 IF B<10 THEN GOTO 40
70 IF A<4 THEN GOTO 20
80 END
```

2) In immediate mode, type:

```
PRINT A
PRINT B
```

3) Does the computer hold the values of variables in memory
after a program has run?

4) Run the following program:

```
NEW
10 A$ = "WORDS"
20 B$ = "MUSIC"
30 C=C+1
40 PRINT A$
50 IF C<10 THEN GOTO 30
60 PRINT B$
70 GOTO 60
```

5) Run the following program:
```
NEW
10 A$ = "APPLES"
20 A = 0.2
30 B$ = "BANANAS"
40 B = 0.1
50 C$ = "CARROTS"
60 C = 0.3
70 D$ = "DRIED FIGS"
80 D = 0.5
90 PRINT 10,A$,10*A
100 PRINT 5,B$,5*B
110 PRINT "TOTAL",,10*A+5*B
```

LESSON FIVE
OBJECTIVES

After completing lesson five, the learner will be able to properly use the following commands:

list
goto
gosub
return
and will therefore be able to put loops into programs.

PRACTICE EXAMPLES

1) Write a program which will display onto the screen, all the numbers from 1 to 10

2) Write a program to display all the even numbers between 10 and 100 inclusive.

3) Write a program to print your name onto the screen 100 times.

4) Write a program which will calculate the square roots of all the numbers from 1 to 12, and print the results in a table.

5) Write a program to print a table of all the squares and cubes of all the numbers from 1 to 12

6) Write a program which will clear the screen and then display your name the same number of times as your age last birthday. (If you were 21 last birthday, have the computer display your name 21 times).

7) Run the following program:
```
    NEW
    10 HOME
    20 INPUT "WHAT IS YOUR NAME ?";A$
    30 PRINT "HELLO ";A$;
    40 goto 30
```

8) NEW
```
    10 HOME
    20 INPUT"HOW OLD ARE YOU?";A
    30 INPUT"WHAT IS YOUR NAME?";A$
    40 PRINT "SO ";A$;"YOU ARE ";A;" YEARS OLD ARE YOU?"
```

9) Run the following program which uses ⟨⟩ to signify not equal to.

```
    NEW
    10 HOME
    20 INPUT "WHAT IS YOUR AGE";A
    30 IF A ⟨ ⟩ 30 THEN PRINT "YOU ARE NOT 30 YEARS OLD"
    40 IF A=30 THEN PRINT "I WAS THIRTY WHEN I WROTE THIS
PROGRAM."
    50 PRINT "SIGNED A. PROGRAMMER."
```

EXTENSION

1) Enter and run the following program:

```
NEW
10 REM THIS IS A PROGRAM TO DEMONSTRATE SUBROUTINES
20 GOSUB 1000
30 ?"YOU ARE NOW AT LINE 30"
40 GOSUB 200
50 ?"YOU ARE NOW AT LINE 50"
60 GOSUB 200
70 ?" LINE 70"
80 END
200 REM DELAY SUBROUTINE
210 ?"THIS IS A SUBROUTINE TO MAKE"
220 ?"THE COMPUTER COUNT TO 300"
230 D=D+1
240 IF D<300 THEN GOTO 230
250 D=0
260 RETURN
1000 REM A NONSENSE SUBROUTINE
1010 ?"THIS IS A NONSENSE SUBROUTINE"
1020 ?"WHICH IS CALLED SUBROUTINE 1000"
1030 RETURN
```

2) Modify the program above by trial and error until the delay subroutine creates a delay of 5 seconds.

3) Add extra lines to the program so that the screen is cleared at the beginning of the program, and there is at least one line of space between each screen display of text.

4) Write a program to print out all the cards in a pack, allowing one line for each card. the answer will start like this:

```
ace hearts
2   hearts
3   hearts
etc
```

LESSON SIX
OBJECTIVES

After completing lesson six, the learner will be able to succesfully include the following commands into programs, and obtain results which meet syntax rules and output information as required:

INPUT
INT.

The command NEW will be used by the learner whenever required to clear the memory of BASIC PROGRAMMES.

PRACTICE EXAMPLES

1) Write a program which will allow the operator to enter a number at the keyboard, and have the computer respond with information as to wether the number is odd or even.

  (hint: int(3/2)<>3/2: int(4/2)=4/2)

2) Write a program which will divide any two numbers which the operator enters, and determine if the result is larger or smaller than the sum of the two numbers.

3) In a program which asks the operator to input the dimensions of a box, have the computer calculate the volume and print the results.

4) Write a program which will ask the operator to enter his or her name at the keyboard, and responds by filling the screen with that persons name."

LESSON SEVEN
OBJECTIVES

After completing lesson seven, the learner will be able to generate random numbers in any range, and use this to randomise output of programs.

PRACTICE EXAMPLES

1) Write a program which will select a random card from a pack of 52 cards.

2) write a number guessing game with the following features:

- replies with much too low, too low, a bit low, a direct hit, a bit high, too high,or much too high.

- allows the operator to nominate the number of chances required to guess the number.

- gives more points for earlier answers.

- gives a bonus if the operator requests 4 chances and wins.

- allows for 1 or 2 players.

3) Using the routine in question 1, create a card guessing game which tells you if you have selected the the right coloured card, and on the next guess, if it is the correct suit, and then use a similar format to Q2 to tell if the card is too high or too low.

4) Create a routine to play blackjack or 21, so that the computer is the dealer.

WHAT NEXT?

Now that you have completed the seven lessons, you will have
been exposed to many of the elementary commands which are used
in BASIC programming. There are few commands used in the
programming of these lessons which havent been' explained in
the course. Although you now have some experience with BASIC,
you will only gain real programming skills after many hours of
using the commands which you have learned. An excellent idea
would be to find a copy of THE APPLESOFT TUTORIAL, which is a
book included with the sale of each Apple II computer, and
work your way through it. This book is written in a friendly
style and can be reccomended.

ANSWERS TO QUESTIONS

There are no answers to questions included with this manual.
Each task in programming can be answered in so many ways that
no attempt has been made to try to give you answers. The real
challenge in programming comes from the creative and artistic
manner in which individuals tackle particular problems.

FOR

APPLE II or IIe

48k with DOS 3.3
OR COMPATIBLE
CAN BE BACKED UP