A Simplified Guide to Computer Programming in BASIC

CHR$, ENDPROC, ABS, SPC, SIN, TAB may sound double-Dutch to you. It probably is double-Dutch. But it's the kind of double-Dutch that computers like to talk in. Unfortunately,the moron that invented computers didn't speak very good English—but he spoke damn good double-Dutch! And thats what you have to learn to do - speak another 'language'.

It's quite possible to learn computer programming in a couple of weeks or it may take a lifetime. It all depends on how 'fluent' you wish to be. Unfortunately, the story doesn't stop there. Each computer has its own different 'dialect' and you've got to learn the right one. However, there are basic similarities and this is what this article will try to explain.

OK! So you've got a computer and you've managed to decode your four million page manual to figure out how to turn the thing on. You wait there patiently for it to suddenly talk to you and answer all the questions that you've always wanted to ask, such as: What is Life? You'll be there 'till a cow jumps over the moon because contrary to common belief computers are a bit ignorant. In fact they're downright stupid! Indeed, instead of the computer telling you what to do, you have to tell the computer. You've got to list your commands precisely, and you musn't miss a single step. Otherwise the computer will just cough and then say that it doesn't know what you are blabbering on about (or rudder words to that effect!). For example, if you type in

```
10 PRINT "HELLO"    (and press RETURN or
                     ENTER key, and do this at
                     the end of each command)
20 STOP
RUN
```

the computer will write out the word HELLO on the top left hand corner of the screen. Notice that each command needs a number and by international convention it's usual to begin with 10 and then to continue with subsequent multiples of [illegible]! It's called a line number.

RUN is the command that tells the computer to start execution of your program, and it must be typed in at the end of every program.

```
10 PRINT "HELLO" (Press RETURN or ENTER key)

20 GOTO 10
30 STOP
RUN
```

GOTO means exactly what it says—GO TO line number 10.

When you typed in RUN the computer should have printed HELLO on the screen (just as before) and then, having dealt with this, your first command (line number 10), it went on to your second command (line number 20) which of course told it to go back to the first command and print HELLO again. The 64 thousand dollar question is: When does the computer get to line number [illegible] telling it to stop what it's doing)?

Got it?

No?

Well have another think.

That's right! It never gets to line number 30 and that's why you get HELLO printed so many times on your screen. This program is called a loop in computer double-Dutch jargon because the computer keeps going round and round in the program, never getting to the end.

If you got that question right you can congratulate yourself. You're now one step above a moron in terms of computer programming. Welcome to the rat race!

To get rid of everything on the screen type in CLS (or HOME if you've got an Apple or PRINT [illegible] if you've got a VIC or PET computer).

Ask a computer to choose an [illegible] twenty:

```
10 LET A=INT(RND*20+1) [illegible]
20 PRINT A
```

```
30 STOP                    (Press RETURN or ENTER)
RUN                              "  "  "  "  "  "
```

Note:

If you've got a VIC or PET change RND to RND (1) in line number 10.

" " " " BBC or Acorn Electron change RND to RND (1)

" " " " Apple then " " " "

" " " " TRS-80 " " " " RND (0)

The gobbledegook in brackets in the above program simply tells the computer to choose a random (RND) number between 1 and 20. Try putting in different numbers.

The INT is short for integer; otherwise the computer may quite rightly choose the number 7.86436!

LET A = ... tells the computer to put the random integer between 1 and 20 that it has chosen in a memory bank called A.

You should know enough computer programming by now to be able to figure out the rest of the program.

Got it?

Yes?

Good!

Now break out the champagne; you're two steps ahead of a moron in terms of computer programming!

Let's alter the above program so that you have to guess what the number is.

```
10 LET A=INT (RND*20+1) (Press RETURN or
                             ENTER)
20 PRINT "WHATS
     YOURE GUESS?"               "
30 INPUT B                       "
40 IF A=B THEN PRINT
   "WELL DONE"                   "
50 IF A<>B PRINT
   "WRONG-THE COR-
   RECT ANSWER WAS            "; A
60 STOP
RUN
```

Line 10 tells the computer to choose a random number between 1 and 20 and keep it in a memory bank called A.

Line 20 tells the computer to print everything within the speech marks (quotes), i.e., WHATS YOURE GUESS?

Now comes the exciting bit! You have to guess what the number A is. (Take a guess and don't forget to press the RETURN or ENTER key once you've had your guess.)

Line 30 tells the computer to wait for your guess and then put your guess in a memory space called B.

Line 40 tells the computer to see if its random number A is the same as your guess B, and if it is so it will print WELL DONE, otherwise it will do nothing and go on to line 50.

Line 50 tells the computer to see if its number A is less than or more than (i.e. not equal to) B and if this is the case then to print everything that's mentioned within speech marks. Notice the semicolon after the speech marks and then the letter A. This tells the computer not to go to line 60 until it has printed the random number that it called A in place of the letter A after the semicolon. Got it? Have another read.

You've now climbed the first three steps (perhaps the hardest of them all) that everybody has to climb before becoming proficient in computer programming. You now have the basis of programming under your belt and, believe me, it's surprising what you can do with just learning these three moron-proof steps in double-Dutch. Read on!

There is one other thing you should know. In the very first program shown to you the computer was made to loop an infinite number of times printing HELLO on the screen each time it did so. However, there is a way to control the number of times the computer loops, and it is quite useful to be able to do this while, for example, setting limits to the number of tries a player can have on a game. Take a look at this:

```
10 FOR A=1 TO 3          (Press RETURN or
                           ENTER key)
20 PRINT "HELLO"               "
30 NEXT A                      "
40 STOP                        "
RUN                            "
```

It is better to explain lines 10 and 30 together. Anything between these lines will be repeated three times so that in this example the word HELLO will be put on the screen three times. Why three? Well the number of times the computer will loop between lines 10 and 30 is dictated by the last number in line 10. For example, if line 10 was changed to

```
10 FOR A=1 TO 10
```

then the computer will print HELLO ten times.

Let's play a game in which you have to guess a random number that the computer has chosen in say three tries.

```
LIST
 10   CLS
 20   PRINT "IN THIS GAME GUESS A NUMBER
      THAT THE COMPUTER HAS IN MEMORY IN
      THREE TRIES. THE NUMBER IS BETWEEN 1
      AND 20."
 30   PRINT
 40   PRINT "PREPARE TO BE BEATEN."
 50   PRINT
 60   PRINT
 70   PRINT
 80   LET A=INT (RND*20+1)
 90   PRINT "I HAVE CHOSEN A NUMBER. GUESS
      IT IN THREE TRIES."
100   FOR C=1 TO 3
110   INPUT F
120   IF F < A PRINT "TOO LOW"
130   IF F > A PRINT "TOO HIGH"
140   IF F=A GOTO 210
150   NEXT C
160   PRINT
170   PRINT "BAD LUCK"
180   PRINT
```

```
190  PRINT "THE CORRECT ANSWER WAS  ";A
200  GOTO 220
210  PRINT "THE CORRECT ANSWER-WELL
     DONE"
220  STOP
```

***Note:**
If you've got an Apple then change CLS to HOME in line number 10
" " " " VIC PET "   " PRINT CHR$(147)

Try to first go through the program assuming that you didn't get the answer and then follow the commands in the program assuming that you did get the correct answer.

If you can do this you have uncovered the fundamentals of computer programming. This is the basis on which all programming is based and you should know enough to be able to write similar programs.

By now you are probably wondering how a computer produces games such as space invaders and moving ships on the screen. Computer animation, as it's called, is very difficult and varies enormously from computer to computer. So it's really not possible to talk about it in a general sense when so many of the readers are likely to have different computers.

Further, computer animation is rarely done in the language called BASIC (the one used in this article) because it's a very slow language and would result in very slow moving targets and missiles. It's usually written in something called machine code which I advise you to keep well away from until you've at least mastered BASIC!

But just to show you that animation isn't always that difficult here's a program which shows a moving 'ship'. Its very simple and works on the basis of putting a symbol (which represents the ship) in a position, say X, and then placing a blank space in this position X. Finally, the symbol is placed a little way further to the right (say X+1). Repeating this many times, very quickly, will give the illusion of a moving ship.

Use this program which illustrates the point if you have a Sinclair Spectrum:

```
 10 CLS
 20 LET X=0
 30 FOR A=1 TO 30
 40 PRINT AT 10, X; ">=="
 50 GOSUB 100            (line 50 makes the
                          computer jump to line
                          100)
 60 PRINT AT 10,X; "    "
 70 LET X=X+1
 80 NEXT A
 90 STOP
100 FOR B=1 TO 15        (line 100-110 is a
                          delay sequence, i.e.,
                          dictates the speed of
                          the ship) .
110 NEXT B
120 RETURN               (line 120 makes the
                          computer return to
                          line 50)
```

If you have a VIC, PET, BBC, Acorn Electron, Apple or TRS-80, use this program:

```
 10 CLS*
 20 LET X=0
 30 FOR A=1 TO 35
 40 PRINT TAB (X,10)">=="
 50 GOSUB 100
 60 PRINT TAB (X,10)"    "
 70 LET X=X+1
 80 NEXT A
 90 STOP
100 FOR B=1 TO 300
110 NEXT B
120 RETURN
```

*If you are using an Apple, change CLS to HOME and if you are using a VIC or PET computer, change CLS to PRINT CHR$(147).

Space invaders is fascinating in terms of programming but it's so complex and way beyond the scope of this article. It is perhaps interesting to just note that to your computer your average space invader is just a bunch of complex mathematical equations.

But that's another story .... □