----- PROGRAMMING LANGUAGES FOR BEGINNERS ----- AND THE GLOBAL CHALLENGE

by Borge Christensen, COMAL founder
(reprint from COMAL CATALYST #2 with permission)

The first time I heard the word "basic" was in 1946. I met a fisherman one day who told me that he had just returned from England. It appeared that he was one of the many Danish seamen who had escaped to England shortly after Denmark was invaded in 1940, and had come to serve in the English navy. At the time, when I met him, I had been learning English in school for about a year, and I was eager to know, how it was to come to England just like that. What about the language! Did he speak English when he came over? Or how did he learn it? He answered my many questions by telling me that they were all offered a course in something called "Basic English". As I asked him, what that was like, he gave me an answer that I did not fully understand then. "It was a whore of a language. It could be used to cover your immediate needs, but there was no real pleasure in it".

A NEW MIRACLE ----- Soon after that the word entered my English vocabulary in its general meaning, and it went on to be just another common, useful, decent English word until 1971. At that time our mathematics department got a minicomputer installed. A Data General Nova 1200. A new miracle. We were all very proud and it was even mentioned in the local newspaper. Since I had taken an Algol course at the university, and was therefore considered to be the local expert - nobody knew that the course had only been a very short one, and that I remembered very little about Algol or computers in general - I was asked to take care of the "recent device" and maybe even teach some of the students to use it. And then I met the word "BASIC" again in a more specific context. And spelt with capitals! After a year and several "miles" of BASIC programs, I realised that something was rotten in the state we were in. Very often I found it quite hard to find out what was going on in the students programs, and especially the faulty ones, of course.

DEFICIANCIES ----- At first I blamed my own lack of profound knowledge for the whole misery, but gradually I dared to think that this marvel of a new language - that had come from THE STATES - might have some deficiencies. My suspicions were confirmed as I started to talk about it to some of my colleagues at the department of informatics, university of Aarhus. They told me straightforward that BASIC was disaster, and that its success was totally undeserved and due to the fact that no other language was generally available on the small computers used in elementary education. One of them, Benedict Lofstedt, took real interest in my problem - it is often very difficult for a teacher to get university reasearchers interested in educational problems - and he advised me to read a book that had just come out. Its title was SYSTEMATIC PROGRAMMING, and it was written by one Niklaus Wirth.

HUMAN INTERFACE ----- I bought it, started to read it, and after a month I knew that the folks from Aarhus were right. BASIC IS a disaster and to a very large extent to blame for the bad programming that was going on not only in our place, but in most other colleges and highschools. But what could be done? Pascal - the language submitted in SYSTEMATIC PROGRAMMING - was only implemented on few large mainframes and used exclusively in academic circles. In those days very few people anticipated the status Pascal has today. And after all, the language is not all there is to it. The ENVIRONMENTS of the language are of crucial importance, too. What we today might call "human interface". The Basic on our computer - DG XBASIC - was fully interactive, supported by a good multi-user system, and included very useful file handling. We could not do without all that.

After having considered all this, I came back to Benedict Lofsted and he now suggested that we design a few but powerful extensions to the language and system we had, and that we should use as much of the ideas from Pascal as possible. It was quite obvious, what was most urgent: LONG variable names, a GLOBAL IF-THEN-ELSE structure (NOT the one line IF-THEN-ELSE that comes with any silly BASIC nowadays), REPEAT- and WHILE- loops, a multi-branching CASE structure, and NAMED subroutines. During the following six months the design was finished. I coined the work COMAL (COMmon Algorithmic Language) to name the extensions we had planned, and in June 1974 we started to implement the facilities, mentioned above, on the Nova computer her in Tonder. The first versions of COMAL were launched in February 1975. We were now able to write programs like this:

```
IF TRY<3 THEN
  PRINT "NO, TRY AGAIN"
ELSE
  PRINT "NO, THE CORRECT ANSWER IS";RESULT
  PRINT "TYPE THAT."
ENDIF
```

I have chosen this example because the displayed IF-THEN-ELSE structure was the one that made COMAL popular very fast. The early versions of COMAL allowed IF-THEN-ELSE branches to be nested to a depth of only four, but it is not very often you need more, and it is much easier to do it the COMAL way that by applying GOTO statements. You do not have to go much deeper than two levels to get BASIC programs that are at least hard to read, and most beginners will get lost in the labyrinth of BASIC statements that implement more than three nested branchings. I ought to mention that the line indentation shown in the example is done automatically by the COMAL interpreter.

IMPACT OF COMAL ----- The impact of COMAL has been stronger than we had guessed that it would be. The students write much better programs in COMAL than they used to do in BASIC, and - even more important - their programs have become

READABLE. At the beginning our ideas were rejected by quite a lot of teachers. In general students were much faster to see that a good tool had come into their hands. Sometimes when I came out to talk about COMAL I got the impression that I had started some kind of a religious war and not just invented some useful improvements of a programming language that might be discussed in terms of expediency and effectiveness. Gradually the attitude changed in favor of the concept, however, and since 1977/78 it has become very difficult to sell a BASIC computer to a school in Denmark. Most teachers will ask the salesman to come back again some other day with a computer that can run COMAL.

A NEW VERSION ----- In 1979 I defined a new version of COMAL to be implemented on a microcomputer. This version - COMAL-80 - was further improved by a working group with members from The Technical University of Copenhagen, the University of Roskilde, and representatives of several Danish manufactures of microcomputers. COMAL-80 includes such facilities as parameter passing (both value and reference), local variables, and recursiveness. It has been implemented on the Commodore CBM-8032 and 4032 microcomputers, the Commodore 64, and two Danish made ones, the RC700 and the ICL COMET (I wonder if ICL, England, knows that they have in fact a very good COMAL running in Denmark?).

THIRD AND SAD CHAPTER ----- Happy ending, is it not? Well, not quite. I'm sorry that I have to write patient reader a third and sad chapter about "basic". A few days ago I met the word again in an important and fatal context. I got a paper from England titled "The BBC Microcomputer. Outlined specifications of the BASIC language interpretter". The paper does not reveal its author(s), and so far that is the only trace of good taste I can find in it. It is most laudable that this glorious institution, BBC, intends to teach the Englishmen about computers and programming. And of course those responsible for such a worthy enterprise know that it must be carefully prepared and that both hardware and software goes with it. But are they fully aware of the crucial importance of the programming language that may be used by hundreds of thousands of viewers. Not only for programming purposes, but to a still higher degree for COMMUNICATION OF IDEAS.

CONSEQUENCES ----- Programs for computers are of increasing importance and more are being written every day. These programs are meant to be used for something. They are going to DO something which in most cases immediately influences peoples welfare in the broadest sense. Economy, health, culture. It has been said that the computer is a tool to extend the human brain in much the same way as the steam and combustion engine radically extended the potential of the human body. I think it is more precise to say that the computer extends our linguistic potential. Man has always been able to use the language to trigger off movements and changes in his environment. But it is new that we are now able to leave the traces of our language in tools which then, maybe long after the words have fallen and could be far away from the place where they were spoken, translate the message into action. The consequence may be new knowledge, new potentials, new wealth, but also disasters that may leave a long trace in our history. At the beginning the word was there...

FUTURE JOBS ----- If one considers what language means to man in general, in our intercourse with each other, in our understanding of the world and ourselves, it is almost incredible to watch the improvidence with which we have designed and used programming languages. As the number of problems grow, it will be of growing importance that we can READ each others programs. The message from one person to another will also in this field be of major importance over the communication with the computer. In his book THE GLOBAL CHALLENGE the french author Servan-Schreiber says, in the chapter about the new information technology, "No industrial country will be able to survive the global revolution, if it does not create FUTURE JOBS with this revolution as a starting point. The necessary readjustment consists in leaving the outdated pure commercial competition, the aim of which is the capture of markets, and which has lasted 30 years and is played out. Instead we must institute a new competition that is based upon the EDUCATION OF MAN, the training of brains, of the ability to create..."

WHERE BBC COMES IN ----- And this is where BBC and its computer course comes in. In England you have a great tradition for good programming. Names like C.A.R. Hoare and Elliot computers come to mind. And the rest of Europe. Where were languages like Algol and Pascal invented? This is where we are much better than the Americans and the Japanese. Try to compare the miseries of Fortran and Cobol, not to speak of PL/1, the greatest scandal since the tower of Babel - with the elegance and power of Pascal, and you know what I mean. This is where WE can compete. And this is the field that matters in the futue!

But instead of seeing this, BBC unconsciously wants you to ape after the Americans, in a field where they are definitely bad, and where you could be very good. But we have to use BASIC, the cowards yell. Anybody uses BASIC. A hell of a lot of programs have been written in BASIC. Now, take it easy. That a lot of programs have been written in BASIC doesn't matter. They are very few compared with the immense amount that are going to be written in the future. And BASIC is not used by anybody. A lot of people are using BASIC, admittedly, but they are not the most important ones. And they are a minority compared to all those who will use computers in the future. In Denmark we have learned that people will turn away from BASIC if only they get something better.

FOR AND AGAINST ----- But BASIC is the only language that can be put into small and cheap

microcomputers, is another argument. Not true. The early versions of COMAL only took up 12% more storage that the BASIC interpreter, and we had not thrown away the GOTO, GOSUB, RETURN, ON GOTO, which we could easily have done without, and which in fact have not been in use since 1976. On page four of the heroic paper about BBC-BASIC it says - about the extensions to a BASIC which is neither flesh nor good red herring - "These extensions should be avoided in simple programs that are intended to be used on a variety of machines". Among the extensions is alot of the same hopeless jumble that characterizes the rest of the design, but one or two important ideas are hid in between, namely those of LOOPS and NAMED SUBROUTINES.

MEET THE GLOBAL CHALLENGE ----- And that kind of extensions should NOT be avoided in simple programs. On the contrary. Beginners should be urged to use such facilities, and at the same time be kept away from using the GOTO which is a very advanced tool only to be used by well trained and very good programmers. The whole concept has to be carefuly redesigned, if BBC is not going to spread a mortal disease among the next generation of British programmers and users of computers. Meet the global challenge, where you are able. Where did British motorcycles go? And how are British steel, ships and cars these days? Is programming and control of computer systems to go the same way! Don't let it happen. BbC, think again!

Yesterday Today credits:

Len Lindsay: Editor
Frank Hejndorf: Captain COMAL Circleman art
Wayne Schmidt: Calvin the COMAL Turtle art
Bill Nissley: Draw Poker Screen Image art
Angus Quinn: Inside Cover Circle People art
G Raymond Eddy: Video Cartoon art

Articles and Programs by:
Richard Bain
Jack Baldrige
Borge Christensen
Glen Colbert
Captain Buzz COMAL
Garrett Hughes
Debra Ruth Junk
Dick Klingens
Len Lindsay
Kevin Quiggle
Robert Ross
David Stidolph
Colin Thompson

N58----- COLOR COMBINATIONS -----
The Commodore 64 computer has some color combinations that yield unreadable text on some TV's. So we would like to know what color combinations are unreadable on YOUR system, and which are BEST. Photocopy the chart below. Then RUN the color program. Fill in the chart with your judgement of the color combination readability on YOUR SCREEN (be it black & white, green, amber, or color). Use the following rating: 3=Excellent, 2=Good, 1=OK, 0=Poor. Send the chart to COMAL TODAY, 5501 Groveland Ter, Madison, WI 53716. We will compile all charts submitted and come up with a composite chart. Do it today!

```
BACK'COLOR:=0 // hit any key for next set
REPEAT // hit STOP key to STOP
 BACK'COLOR:=(BACK'COLOR+1) MOD 16
 BACKGROUND BACK'COLOR
 PRINT CHR$(147) // CLEAR SCREEN
 FOR PENS:=0 TO 15 DO
  PENCOLOR PENS
  PRINT "PENCOLOR";PENS;"BACKGROUND";BACK'COLOR
 ENDFOR
 WHILE KEY$=CHR$(0) DO NULL // WAIT FOR KEY
UNTIL TRUE=FALSE // forever
```

3=Excellent 2=Good 1=OK 0=Poor

PENCOLOR

BACKGROUND	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	BACKGROUND
0	0																0
1		0															1
2			0														2
3				0													3
4					0												4
5						0											5
6							0										6
7								0									7
8									0								8
9										0							9
10											0						10
11												0					11
12													0				12
13														0			13
14															0		14
15																0	15

PENCOLOR

3=Excellent 2=Good 1=OK 0=Poor

My TV / Monitor is a: