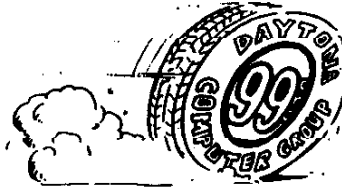


**THE DAYTONA 99'ers**  
Post Office Box 4594  
South Daytona, Florida 32021



---

**A users group dedicated to the Texas Instruments brand of home computers.**

---

**MESSAGE FROM THE PRESIDENT...**

Can you believe it's nearly Spring already? It seems like just yesterday that Christmas was just around the corner and Spring was a long way off. Oh well, it is nice to have some warmer weather ahead...but don't let your computer get rusty at the beach, O.K.? Well, even if you don't take your computer to the beach with you, let's not let it get rusty sitting at home either!

I sure hope that everyone will be able to make it to this month's meeting which, again, will be on Thursday, March 22nd, same time, same place. This will be the last month that our normal schedule will be affected by previous scheduling of our meeting room. Our April meeting will be on the third (3rd) Thursday of the month as will all of the succeeding month's meetings.

We will begin accepting dues at this month's meeting or by mail as we are about to finalize the club's organization status thanks to the diligent efforts of one of our members, Chuck Hargraves. Chuck has devoted a lot of his time since the last meeting in researching our situation and determining what needed to be done. This brings me to another point...that of the fact that I was very happy with the response I received to my pleas for help last month. We had nearly a dozen of our members volunteer to take on some of the club's operational duties and, believe me, it is much appreciated. To all of you who did volunteer last month, I haven't forgotten you! We'll be getting in touch with you to do some honest to goodness work just as soon as we get ourselves a little more completely organized.

Also at this month's meeting we will hold an election of new officers which is a very important task to accomplish. Our new officers will have the responsibility (with help from steering committee members) to formulate our club's bylaws which will be a necessary part of the paperwork that will have to be submitted to complete our organization filing status. We will present a slate of officers for your approval and/or we will accept nominations from the floor prior to the election. If there are no additional nominations the slate of officers will be approved and will begin to take over necessary duties beginning in April. I would like to see as many people nominated or volunteer as we have slated since we are not trying to shove any particular people down your throats, so to speak. We simply need some dedicated members who can devote some of their time and energies toward the evolution of our club. So, be thinking of your candidate hopefuls or be bold and volunteer yourself.

Before I go, I'd like to thank Chuck Hargraves again for his efforts. Not only has he taken on the responsibilities mentioned earlier, but he and his wife have also taken on the very large responsibility of getting this newsletter out to you. If others like Chuck and his wife will get involved in the coming months we should be in store for a worthwhile and productive year.

I should go for now, but remember this month's meeting date and come with some good suggestions for the coming year's activities.

**See you soon!!**

Note: At last month's meeting we had enjoyed a demonstration of the FORTH programming language as released by Texas Instruments in their TI FORTH package. We had hoped to include an article concerning the package in this month's newsletter but time got the best of us. So, instead, to make up for the absence I'm including a second installment of the series which I started two months ago on the UCSD p-System and the Pascal programming language. Hopefully, you'll see the FORTH article next month.

In my first article on this subject I briefly introduced TI's implementation of the UCSD p-System. Since we meant to have an article here on a different programming language for the TI (specifically, FORTH) I'll postpone more discussion of the UCSD p-System operating system and concentrate on an introduction to the Pascal programming language. TI supplies a powerful version of Pascal with the UCSD p-System known, appropriately, as UCSD Pascal. It supports all of the special features of the 99/4A such as speech and graphics including sprite animation. However, like many other versions of Pascal and UCSD Pascal, it may also be used primarily as Standard Pascal. In other words, learning Pascal in any computer environment would, in nearly all respects, allow you to immediately begin programming on the TI. For those of you who might be interested I have heard that there are still hardware and software packages available here and there making the p-system and Pascal available. Hopefully I'll find out exactly where soon.

If you have ever heard the expression "spaghetti code" you'll know that it refers to the kind of programs that are usually produced while using such a language as BASIC. Very rarely can you find a BASIC program that doesn't contain umpteen different GOTO statements. It seems like every time a minor change must be made to a BASIC program the easiest way to accomplish it is by simply adding in the new or changed statements and connecting everything together with as many GOTOs as necessary to make it all work again. If, sometime later, you tried to understand the logic flow within the program you probably found yourself drawing all kinds of lines around the program trying to follow all those GOTOs. Hence, the name "spaghetti code".

This is precisely why a brilliant gentleman by the name of Nicklaus Wirth in Germany developed the Pascal programming language several years ago. He envisioned the possibility of having available a language that could be used as an introduction to computer programming while "forcing" the student to structure the programs into a logical group of related "procedures" and "functions" that did not rely on GOTO statements to handle the changes in the flow of program execution. What emerged was a powerful, elegant, yet easy to learn programming language that has been growing in popularity ever since.

Naturally though, Pascal is not a cure-all for the "spaghetti code" blues with every feature that you could possibly want and no flaws in any feature provided. It, like all other programming languages, has its share of drawbacks and flaws. Fortunately, the language has enough advantageous features to offset these drawbacks for the most part. Of course I can't begin to list all of the features of the language in this article but I will begin to point out some of the major differences that tend to "force" the structured approach to programming that I mentioned.

One of the first things that a BASIC programmer notices about Pascal is the fact that every (and I do mean every) variable must be declared before it can be used by any part of the program. This has the effect of requiring the programmer to go through a little bit more deliberate approach to assigning names to variables than does BASIC where we sometimes simply pull a variable out of mid-air to satisfy the requirements of our code at that moment. Not only must Pascal variables be declared before they can be used, the "type" of each variable must also be specified and sometimes even the size of the variable. By "type" I mean simply what kinds of information will be stored in each variable. For instance, if I needed a variable to store a number, I would need to specify whether or not the number was to be an integer (i.e.- a whole number) or a real (i.e.- a number with a decimal). Likewise, if I needed a variable to store alphabetic or other "character" data, I would need to specify that variable as being of type "Char". Here's a very short Pascal "declaration" section which declares three variables of the above types:

```
VAR
    INT      : Integer;
    DECIMAL  : Real;
    SINGLE_CH : Char;
```

It should be noted that the variable "SINGLE\_CH" can only contain a single character at any time. If I needed a variable to hold a group of characters, such as a name, I would need to specify that the variable is of type "array". In standard Pascal, all character related data is handled either individually as with the "char" declaration or in groups by placing individual characters in successive elements of an array. This may seem terribly awkward at first but, as I'll point out later, there are certain advantages to be gained by such requirements.

While on the subject of variables, you might have noticed the names which I used to declare the above variables. This is a very nice feature of Pascal in that you may assign a name of up to 255 characters (more or less depending on the version of the language being used) to any variable. This allows a much more descriptive name to be used and prevents us from having to use such cryptic names as "A1" or "MN". In Pascal we could use names like "Address\_1" or "My\_Name". Of course, in many present-day versions of BASIC longer, more descriptive names are also allowed but, being an interpreted language for the most part, BASIC is much more memory sensitive in the use of long variable names.

Just a moment ago I mentioned the need to specify the type of each variable and we agreed that it was somewhat of an awkward requirement. Well let's look a little deeper at the kinds of "types" that are available for our variables in Pascal. What if I told you that in your BASIC program you would need to have available a group of variables that you would also need to handle as a single unit. For example, we want to keep information stored on disk containing many "records" that consist of a person's name, address (keeping street, city, state, and zip separately), and telephone number. One common method in BASIC to handle such a task would be to use six different variables and assign the appropriate value to each variable. But then, when we want to write the whole group to disk as one record, we would be forced to set up a loop that would write each variable successively to the disk in order to keep the record intact. As you might have guessed, what I'm leading to is the fact that Pascal offers a much more efficient way of handling this problem.

Without going into a lot of detail right now, the following declaration in a Pascal program would set up my variables for the necessary processing:

```
TYPE
  Person_Rec = Record
    Name   : Array [1..20] of Char;
    Street : Array [1..30] of Char;
    City   : Array [1..30] of Char;
    State  : Array [1..2]  of Char;
    Zip    : Array [1..5]  of Integer;
    Phone  : Array [1..12] of Integer
  End;

VAR
  Person : Person_Rec;
```

This illustrates one of the outstanding features of Pascal. In the above declaration I developed my own "type" of variable by specifying the different types of all of the information that would be stored in that variable. I could have made my type definition more readable by previously defining the different arrays that are used and give them different names. That way I would only have to define them once and any time I wanted to use that type of array in another definition all I'd have to do is specify its name in the new definition. More on this idea later. Generally speaking, Pascal allows the programmer to actually define his/her own special "data types". In this case I defined a new data type (in the section beginning with the Pascal reserved word "TYPE") called Person\_Rec which itself contained two different data types; arrays of characters and arrays of integers. Needless to say, it would be extremely difficult at best to even simulate this kind of data structure in BASIC. But, getting back to the original problem, now all I have to do to get this whole record on the disk is to use the following Pascal statement:

```
WriteLn (<<filename>, Person);
```

Other than my declarations to set up my disk file which is referenced by <filename> in the above statement, Pascal takes care of moving all the information of each field in the record (i.e.- Name, Street, etc.) out to the disk. Later I'll expand even more on the ideas of records and other "user definable" data types and this is just one of the many advanced features of Pascal.

For now, though, I'll let you think of some of the interesting possibilities and next time we'll delve a little deeper into "records" and also introduce you to Pascal "procedures" and "functions". Until then, I hope you enjoyed this and...happy computing!!!

WHAT KIND OF THINGS GO ON IN SCHOOL THESE DAYS? Students at Pine Trail Elementary School in Ormond Beach are working on TI computers both in the classroom and in a computer lab provided by the PTA. Students come to the lab before and after school to work on computer programs for home as well as school. The students have some formal computer instruction but most of what they do is learned by doing just as many of you have learned. Students from as low as first grade.

most of what they do is learned by doing just as many of you have learned. Students from as low as first grade, are working each day on programs that give them pleasure as well as learning something of the world of computers. Maybe you should look into some of the things your child's school is doing. If you feel it should be more, then talk to the person at the school who can help. That person will not be the same from school to school. It is up to you to find them. When you do find someone at the school who is willing to put in some time then get them some

someone at the school who is willing to put in some time then get them some help and some computers and put them to work. Most schools are now looking forward with a computer plan for their school. Maybe you could learn more about that and get your PTA in on helping to get something going. In any case, if you don't do something then perhaps no one else will either. GO FOR IT.

by Jay Langley

FLASH Wayne Stevens of Pine Trail Elementary School wins third place in the Tomoka Regional Science Fair, 1984. Wayne, is a member of the computer club at Pine Trail School. He went up against 7th, 8th, and 9th graders to win his 3rd. place honors. Wayne is in the 6th. grade.

The program that Wayne presented for his project was a math drill program. It used Extended Basic with speech. He added lots of color to the program to keep it interesting. The same program took first place in his school

He added lots of color to the program to keep it interesting. The same program took first place in his school level fair. Teachers and parents were very proud of this up and coming programmer?

Students, why not get started on a project now for next year? Wayne told me that he is ready to rest for a few days then he plans to start work on his next effort.

by Jay Langley