The National LOGO Exchange

ISSN 0734-1717

Volume 1 Number 5 FORWARD 100! January 1983

YPLA Releases Turtle's Sourcebook

To say that The Turtle's Sourcebook is a practical guide to LOGO in the classroom is to understate the case. Authors Jim Muller and Donna Bearden of the Young Peoples' LOGO Association, and Kathleen Martin of the University of Dallas have put together an excellent book of over 100 pages of concepts, background information, activities (both on-computer and off), worksheets, and a list of references for further study.

It features detailed instructions throughout for all versions of LOGO now available, thus giving the reader what is needed, in addition to a feel for the differences between each version.

But do not get the idea that it is a step-by-step one-two-three book. Not at all. To do so would be to defeat one of LOGO's purposes: to give children the opportunity to take charge of their own learning. Instead, the authors give specific suggestions on how to foster an attitude of exploration in the children. "The element of discovery cannot be over-emphasized," they point out.

After an introduction to computers and programs in general, the turtle graphics capabilities of LOGO are explored. Teaching turtle control is emphasized with several suggestions for movement activities.

Next come procedures and several interesting activities and worksheets. Variables are then added to increase the level of sophistication. Recursion is perhaps the most powerful concept introduced.

The authors then give many ideas for putting everything together to construct more complex projects. Testing for conditions and incrementing variables add to the fun. The worksheets in this section are particularly interesting.

The last chapter deals with the list processing capabilities of LOGO. Several of the word and list operations are covered, with many examples. Complete procedure listings of a hangman game and a toolkit of list processing utility procedures is included.

continued on page 8

Turtle Folders Help Third Graders

by

Bonnie Rifkin

Third graders certainly can be LOGO lovers. I'd like to share some of the successful techniques as well as problem areas I have experienced wih this age group.

The particular students of whom I speak are third graders at a private school in Cambridge, Massachusetts, learning Apple LOGO.

My time with the students is very limited: two one-half hour sessions per week. When I am not there, the students do have access to their disks and the computers for additional experimentation. I had to make the LOGO time quality time.

Each student has his or her own turtle folder. The first page of the folder consists of a list of what the student can do in LOGO (draw a square, write a procedure, etc.). Some skills can be completed "alone" and some "with help" and checked off accordingly. This is a constantly growing list and a ready reminder to the students that they are progressing.

Also in the folder is a daily note of encouragement and praise from me, summarizing what was accomplished in the last session and commenting on problems and successes. All worksheets and written procedures are also kept in the folder.

Before students begin their session for the day, folders are passed out. Students look for additional suggestions in their checklists and read their notes. The folders have become very personal to each child and are a good means of positive reinforcement.

I also have found task cards to be most helpful. Even though I deal with no more than two students at a time, these cards could be a life saver to a teacher with a "normal" situation involving many children all at once.

continued on page 2

The National LOGO Exchange, copyright 1983 Posy Publications, a part of The Posy Collection, all rights reserved. Published monthly September through May, $25 per year, $45 for two years, mailed FIRST CLASS from Charlottesville, VA. Add $5 per year for addresses outside the United States. Single copy price $3. Opinions expressed by the authors are not necessarily those of The National LOGO Exchange. Address all editorial correspondence to the editor at The National LOGO Exchange, Post Office Box 5341, Charlottesville, VA 22905. ISSN 0734-1717

Editor......................Tom Lough

What A Time!

I have just returned from the MECC '82 conference in Minneapolis. What a time! It was uplifting to encounter so many upbeat educators all together in one place!

The theme of the conference was "Sharing a Decade of Experience." MECC has been operating in Minnesota since 1972, and has accumulated a wealth of experience and information. They have developed a respected line of software, and have caused Minnesota to be recognized as one of the leading states in the field of educational computing.

They shared their experiences generously, and provided many worthwhile activities, including addresses by Steve Jobs and Alan Kay.

LOGO caused quite a stir! Every LOGO workshop, every LOGO presentation was filled to capacity! In fact, Marcia Horn's last session required opening up an additional adjacent room -- and it was still standing room only! The word about LOGO's incredible potential and the quality of Marcia's presentation had spread like wildfire.

If you have ever doubted whether you are doing the right thing by getting involved with LOGO, you would have become convinced by that crowd!

The experiences I had at MECC will influence me the rest of my life. I heartily recommend that each one of you take time, make time, to go to a computing conference.

Here is one coming up which promises to be a real knockout: the National Educational Computing Conference to be held in Baltimore, Maryland, June 6, 1983. Billed as NECC '83, it has many interesting activities scheduled.

For further information, write to Doris Lidtke, Towson State University, Baltimore MD 21204.

FORWARD 100!

Turtle Folder continued

My set of cards begins with very easy activities, such as "Find out how many steps the turtle takes to go across the screen." and "How many turtle steps are there in an inch?"

Other cards give suggestions and challenges for ideas of simple figures to try to reproduce or alter, such as stars, triangles, and sketches of more complex figures, such as a tree.

The task cards help in several ways: (1) For the child who never knows what to draw, is unmotivated, or needs structure, the cards give a specific activity and take away the anxiety of "You have twenty minutes. What can you draw on the screen?" (2) Developing task cards forces the teacher to formulate a sequence in his/her mind. (3) The cards can be used as a kind of worksheet in situations requiring the teacher to work exclusively with one student while the others remain purposefully occupied. (4) Most importantly, the activities can be springboards that will give students ideas for further personal projects.

Additionally, I have noticed that children of this age group often have difficulty visualizing on paper where the turtle will go. Thus, I initially encourage my students to plan their projects right on the computer in the immediate mode.

The first time I did this, the figures they were designing eventually appeared on the screen. But when the children discovered that their drawings could not be saved directly (they were not yet procedures), frustration and tears followed.

Then we realized that, if they worked in pairs, with one writing down the commands that the other typed in, things went better. After they got the drawing they wanted, they simply went back and wrote it as a procedure, copying from what the partner had recorded.

When they decided they did not want to enter their commands twice, they realized the value of planning on paper.

In summary, turtle folders and task cards have been helpful to me. I am still working on ways to help my third graders plan projects.

Bonnie Rifkin is an educational consultant, teacher, and an intern at the Technical Education Research Center (TERC) in Cambridge, MA.

TIPPS for TEACHERS

by

Steve Tipps

From Turtling to Programming

In the first phases of working with LOGO, learners control turtle movements and discover the turtle world. Beginning LOGO-ers explore the turtle's environment using the distance (FD, BK) and direction (RT,LT) instructions. The turtling explorations become the foundations for programming.

Perhaps "programming" actually begins with the first turtle movement; control is certainly the primary idea in programming. Programmming, however, suggests multiple actions performed in a delayed manner often utilizing some efficient shortcuts.

Indirectness is the best general description for programming. Turtling is usually done with only a few actions performed immediately and specifically. Turtling in the immediate mode builds understandings important in programming the turtle for more interesting complicated actions.

Helping children move from turtling to programming is the issue raised in a letter from Steve Matthias, principal of Washington Center Elementary School in Fort Wayne, Indiana. His question may be the most crucial for all who hope that LOGO will fulfill its Papertian promise. In response to Steve's question, I'll share two examples of guiding students toward programming.

The students in a middle school exploratory class were familiar with turtle commands, pen and background colors, and REPEAT. Working with REPEAT serves as an introduction to a shortcut way of executing a sequence of actions. To facilitate working with REPEAT, the teacher had also introduced a procedure using TO (space) to enter the editor and CTRL-C to define. Although all of the children had mechanical knowledge about making procedures, none had used procedures in very purposeful ways. They were still turtling around the screen and making colors change.

WILL'S STOP SIGN

While exploring REPEAT, the students made squares, circles, and other regular polygons. One boy, Will, drew an octagon which, in turn, inspired him to draw a stop sign. He worked for two days on writing S, T, O, and P within the octagon. Only OCTAGON was being drawn as a procedure. An indecipherable string of FD, BK, LT, RT, PU and PD was used to complete the figure. All of these actions were performed in the immediate mode; trial and error finally brought about reasonable sizes and positions.

The question was posed whether other parts of the picture had names which could be defined as procedures. No response to the question came at first. But, on the next stop sign, he said, "The first thing I need is an S, then a T, then an O, then a P." In naming the parts of the project, Will had changed the nature of his effort and had moved appreciably toward programming.

Will went to the editor with TO S, described S and defined it, returned to graphics, drew OCTAGON, drew S, and found that he had forgotten to position it! He returned TO S, included positioning moves, returned to graphics, drew OCTAGON and S satisfactorily, and then went on to work out T, O, and P. By moving back and forth between graphics and the procedure editor, he got his stop sign.

Will had five procedures which he could name separately to form his design. The teacher then asked if he could teach the turtle how to do all five things with one instruction. After a few moments, Will responded that he could teach the turtle how TO STOPSIGN, and created a master procedure:

```
TO STOPSIGN
   OCTAGON
   S
   T
   O
   P
   HIDETURTLE
   END
```

The teacher introduced SAVE "WILL.STOPSIGN to record the file on disk. The next day, Will READ the file, called his procedures and began to use them in new and different ways. The T procedure used in a REPEAT instruction made a kind of railroad track all over the screen. O and P were rotated and created some interesting effects. S did not do much. But each of the parts of STOPSIGN lead to new explorations.

The trial and error decisions which had been made were important to the completion of the project. Letter sizes and positions were already developed; had they not been, the process would have been important to complete concurrently. Will had defined a project, worked through a solution, and begun to build a repertoire of procedures to explore and elaborate upon.

continued on page 4

DAVID's ZIG AND STAR

On another day, the teacher had proposed the task of making zigzags. This is an exercise which I consider excellent for consolidating distance and direction control with a REPEAT statement. Several observations lead me to this recommendation. First, making regular polygons is so familiar to children that doing it does not demand full understanding of distance and direction. Second, regular polygons use only LEFT or RIGHT while zigzag designs require both. Finally, zigzags are so variable that they generate many projects.

The students were confident enough with the screen turtle that they immediately began putting things into REPEAT's. A number of designs resulted, but neither zigs nor zags. As the teacher moved from group to group, suggestions were made to draw or walk through a zigzag and record the instructions for exactly what happened. As students talked through the figure, they found a pattern of FD, turn, FD, turn, with the turns being in different directions. Some of the groups took the pattern immediately to a REPEAT while others did the pattern step by step.

After isolating the pattern, the groups went different ways. Directions and distances were not equal in all the patterns, resulting in many different zigzags. David settled on a zig pattern which was wide-angled. His zig was composed of:

```
FD 20 LT 75 FD 20 RT 75
```

After typing the pattern into repeat statements several times, David was challenged by the teacher to make a procedure having only those commands in it. With reluctance, David defined TO ZIG. He simply saw no purpose to doing it.

But when he returned to immediate mode and began experimenting with ZIG in REPEAT statements, he soon found how much easier having ZIG was than retyping even four simple commands again and again. He explored several things before stumbling onto:

```
ZIG RT 75
ZIG RT 75
ZIG RT 75
ZIG RT 75
ZIG RT 75
```

which made a star with one point crossed rather than pointed. The RIGHT 75 came from the angle in ZIG rather than any particular plan. The pattern David found led to the command

```
REPEAT 5 [ ZIG RT 75 ]
```

then, to adjust the angle,

```
REPEAT 5 [ ZIG RT 70 ]
```

and finally to

```
REPEAT 5 [ZIG RT 72 ]
```

which caused a star to be drawn.

The teacher suggested that David might waant to make a STAR procedure. He did this without as much reluctance! The star was then drawn and rotated and moved about the screen. One of the most impressive displays was

```
REPEAT 40 [ STAR RT 9 ]
```

When David decided that the star was not big enough, he first tried to change STAR, then ZIG. He had recognized that the size was controlled by the FD command and that FD was in ZIG rather than STAR. He changed the size of STAR several times by editing ZIG.

MOVING FROM TURTLING TO PROGRAMMING

The vignettes are offered both because they are excellent examples of guiding children toward programming. A number of people would find too much intervention in the teacher's approach. The examples are offered instead because they are realistic portrayals of a teacher's struggle not to infringe on exploration and also to move the children toward some power with LOGO.

Students get bored or frustrated if they don't have enough power to do more than random turtling. Defining procedures which become new objects gives learners more to explore. The mechanics of REPEAT and of defining procedures are not as important as the kind of thinking which is encouraged by appropriate use of them. The teacher asked questions and made suggestions aimed at improving problem solving with programming. Some of the strategies being developed were:

1. Moving back and forth from immediate and editing mode. After students learn about editing, they tend to start by defining a procedure which they expect to work automatically rather than working things out in graphics first. (See Bonnie Rifkin's article in this issue.) The teacher also reinforced body syntonicity and drawn models for working out ideas.

2. Defining small and understandable segments of code. OCTAGON, S, T, O, P, STOPSIGN, ZIG, and STAR are coherent sets of instructions.

continued on page 6

MICROWORLDS

by
Glen Bull

How to Use LOGO

Some of the characteristics of LOGO have been discussed in previous columns. The interactive, modular nature of LOGO makes it particularly suited for the introduction of programming concepts. An interactive language provides faster feedback, which accelerates the learning curve.

A modular language makes it possible to break large programming problems into smaller parts. In addition, LOGO has a number of other features which makes code generally more readable. That is not to say that the same sorts of things can not be accomplished in other programming languages; just that it is more convenient in LOGO.

PROGRAMMING CONCEPTS

The suitability of LOGO for introduction of programming concepts creates a temptation to use it for just that. Some predictions suggest that many of the smokestack industries such as steel will be lost or substantially reduced by the end of the decade.

If so, the middle of the job market could drop out, leaving the unskilled jobs at the lower end and a plethora of technological positions at the other end. The result might be the curious phenomenon of numerous unfulled jobs coexisting with high unemployment.

Given this forecast, the expense of introducing LOGO in the schools might be justified on the basis of economic self-preservation. A student who can program with facility in LOGO, using its advanced capabilities such as list processing, should be able to transfer the same concepts to most other common programming languages. A student who can program is also likely to have an increased probability of finding a job.

It is true that the programming position will not necessarily require knowledge of LOGO, but neither will it necessarily require knowledge of other languages commonly used to introduce programming. For example, many professional programming positions involve use of COBOL, a language which is not commonly used as an introduction to programming - nor should it be. Acquisition is more important than knowledge of the mechanics of a particular language.

Having noted this, I must confess that I am dubious about the programming rationale as a justification for getting LOGO into the schools. The economic arguments are strong, and are likely to become stronger by the end of the decade.

However, carried to extreme cases, I have nightmarish visions of elemen- tary teachers racing to see which one can teach the most LOGO commands by the end of the school year. A rigid curriculum for the "proper" way to introduce these commands would be developed. Teachers and children would be forced to march in lock-step fashion through a specified content at each grade level. In short, LOGO could be used for the exact opposite of the purpose for which it was developed.

LOTS OF OTHER USES

There are a number of uses to which LOGO can be put, in addition to its potential value as an introduction to programming concepts. I recently reviewed several elementary school science texts. A typical chapter included a picture of the center of the earth with concentric circles of brown, yellow, and red. As I recall, when I was a child at that grade level, my class constructed clay volcanoes with baking soda in the middle to simulate an eruption.

The difficulty with such an approach to science is that it is authoritarian. The child is told to believe the center of the earth is molten because the teacher or an even more distant author says so. There is no way for the child to personally verify the truth of the statements made. Experiments may be trivial, or possibly preselected to demonstrate only evidence favorable to the interpretation selected. For example, I suspect that most chidlren are aware that baking soda in a well of clay proves very little about the contents of the center of the earth. It is fun, but also irrelevant.

The most powerful aspect of the scientific method is the effective means it provides for thinking about how the world works. An important component of this method is hypothesis testing and personal verification. The difficulty with the textbooks I reviewed was the limited number of opportunities for children to convince themselves that these methods work.

Part of the reason for this lack of opportunities is economic. The equipment required for children to engage in personal verification would be too expensive. The time required to supervise each child would also be prohibitive, since true science involves quite a bit of backtracking and false trails. I suspect that even
continued on page 6

if time and expense were no object, specialized "science education" equipment would be developed which would dictate prescribed paths each child was to take.

In this educational environment, LOGO can be used to validate the teacher's authority. There are certain principles which are proven problem-solving strategies. There are a limited number of opportunities in a classroom environment for children to demonstrate to themselves that these principles work.

EXPLORATION

A task in LOGO selected by the child precludes the possibility that the teacher conspired to select the single example in the universe which will yield a particular set of results. The child can try strategies suggested by the teacher and compare them with approaches suggested by other children.

The computer system acts as an impartial arbiter in a democracy of ideas. An incorrect algorithm will not run solely because of an assertion that it should work. The fact that teachers are occasionally incorrect about problem solving strategies only lends credence to the majority of times in which suggested avenues prove fruitful.

One difficulty with using LOGO in this way is the problem of access. The child needs time to fumble around and convince herself that an inefficient strategy will not work. This may cease to be a problem as families opt for a computer in place of an encyclopedia.

Another difficulty is the higher level of skill required for this approach. Anyone can march a group of children through a prescribed sequence of programming instructions. No special skills are required - just follow the manual and grade the exercises provided. Do not allow the students to work on unauthorized projects, or else limit this type of work to special unsupervised activities by brighter students.

Teachers who allow children to use LOGO to convince themselves that certain styles of thinking are effective will come closer to creating true scientists. The fact that these children will gain a solid grasp of programming concepts along the way will be a mere side effect of the process.

Glen Bull is a professor at the University of Virginia, and teaches LOGO courses at both the graduate and undergraduate level.

---

3. Naming of the sets of code with meaningful labels. Each of the names is descriptive of the procedure. This is a feature which sets LOGO apart from most other languages.

4. Using procedures in many different ways. Combinations of procedures with other procedures and commands emphasize the modular and independent nature of procedures, encouraging problem solving in small chunks.

The teacher did NOT do several things. First, although the "magic" number 360 appeared several times in the course of the projects, the teacher did not stop and give a lecture about the Total Turtle Trip Theorem. During STAR, the teacher asked what 5 times 72 was and got the answer 360. Questions about SQUARE and CIRCLE also got the answer 360, but no recognition that this was something which could be used for planning.

Second, the teacher did not dissuade the students from putting positioning moves inside the procedures. Although not a good practice in the long run, at this time the students did not seem ready to contend with the complication. Need for generalizable procedures will come with new projects. For instance, spelling POTS or SPOT with S, T, O, and P would be possible but not neat. The teacher might suggest these words, knowing the problems encountered would point to problems of positioning within procedures.

I often ask students if PC and BG should be put into procedures or left out. The discussion leads to the determination that there are no absolute answers -- only situational ones. "If you put the color inside the procedure, it will always be that color." The issue of position is analogous. Eventually, recognition of what you always want to happen and what you want to be able to set will support the idea of variables.

Finally, the teacher did not introduce variables, even though changing the star size would have been much easier with variables. Variables provide much power; but many things are possible with the power at hand. LOGO does not need to be approached in a rapid lock-step fashion. Exploration at the level of procedures can be very challenging. Not only that, doing things in a clumsy way makes the short cut more understandable and appreciated.

continued on page 8

Can the Turtle Draw a Sine Wave?

It is fun to give presentations on LOGO to teachers and other interested people. Most of the time, everyone gets very excited and a bit awestruck when they realize the power of the language.

Occasionally, a member of the audience will let the imagination soar and think of a completely unexpected application for LOGO.

That happened to me recently. After my presentation, I was taking questions about what had just been demonstrated. Then up went the hand!

"Can the turtle draw a sine wave?"

Everyone got quiet and looked at me. Hem. Haw. To tell the truth, that had never entered my mind. So, let's follow our own advice!

"I don't know," I said. "But I think it could. That certainly is an intriguing question!"

I couldn't get the idea out of my mind. Later, after I was able to think about it some more, a few ways to start came to me.

How would I walk in a sine wave? Well, it is just going forward a little and turning a little within a certain range. But how much to turn?

From my etch-a-sketch days, I recalled that I made something like a sine wave by rotating one knob at a constant rate while I "rocked" the other back and forth. In fact, that rocking was really a periodic input, just like the sine wave output.

Then I began to understand. The turtle would have to turn according to the derivative of the sine wave! (The derivative is a measure of the rate of change of a function, and is an important part of the calculus.)

This came as a surprise, but, then again, the more I explore with LOGO, the more depth I discover. Lots of other people are reporting similar experiences.

Since the listings furnished are far from perfect, why don't you try to develop your own independently?

The EXPANSION in SINE is just a scaling factor which controls the size. I had wanted to be able to control both the height and the width separately. But I have not yet gotten a good grasp on all the scaling concepts. EXPANSION controls both the height and width in these listings.

STEP is the amount the angle is changed for each point on the sine curve. I found a good compromise between speed and smoothness with values between 5 and 10 degrees.

the turtle is turned to an initial heading of COS :STEP. Then, in the SKETCH procedure, it goes forward an amount determined by the angle and the values of EXPANSION and ANGLESTEP. This amount is not strictly correct, but it gives a nice result.

Then the angle is increased and the turtle is turned as a function of the cosine of the angle. (The cosine is the derivative of the sine.) Note the need for the absolute value of the cosine. Can you figure out why? Try it without using ABS!

Here is what SINE .1 5 looks like:

You can do much better than this listing, I'm sure! If the "sine bug" bites you and you work up a procedure which is satisfying, send it in to share with everyone! Good luck!

MIT VERSION

```
TO ABS :VALUE
    IF :VALUE<0 OUTPUT -:VALUE
    OUTPUT :VALUE
    END

TO SINE :EXPANSION :STEP
    DRAW
    RT ATAN COS :STEP 1
    PU SETX -100 PD
    MAKE "ANGLE 0
    SKETCH :EXPANSION :STEP
    END

TO SKETCH :EXPANSION :STEP
    FD :EXPANSION * :STEP *
3.14159 * ABS COS :ANGLE / 180
    MAKE "ANGLE :ANGLE + :STEP
    RT ((90 - HEADING) - ATAN
COS :ANGLE 1 )
    SKETCH :EXPANSION :STEP
    END
```

For the LCSI version, put brackets around OP -:VALUE in ABS, change DRAW to CS, and change ATAN COS :ANGLE (or :STEP) 1 to ARCTAN :ANGLE (or :STEP).

FORWARD 100!

Tipps continued

Projects are the motivation for programming: having something you're interested in doing is essential. Turtle graphics provide a set of projects for children and adults other than accounting and engineering ones. The teacher gave the children a lot of freedom in deciding projects. The ZIGZAG exercises are more skill builders than a single project per se. Both STAR and STOPSIGN came from exploration.

Guiding chidren to programming allows them to complete their projects with skills which grow with them toward solution of more complex problems. Guidance has alays been the most challenging teaching role. Finding the right combination of intervention (See "Hold Your Horses" November NLX) and benign neglect will be the main task for LOGO teachers -- and the most fun.

Steve Tipps is a professor at the University of Virginia, and conducts LOGO workshops for teachers throughout the eastern United States.

Turtle's Sourcebook continued

Lastly, three appendices include a treatment about using the edit mode, a LOGO resource list, and a look at computer languages.

If you have been looking for a teacher-oriented guide to introducing LOGO to your classroom, we recommend The Turtle's Sourcebook.

The book, 20 worksheets ready for duplication, and a demonstration disk in the appropriate LOGO version come packaged together for $29.95, plus $3.00 shipping and handling.

Mail your orders to Young Peoples' LOGO Association, 1208 Hillsdale Drive, Richardson, TX 75081. Be sure to specify your LOGO version.

Get Your NLX ABC's!

Many NLX-ers have received their LOGO ABC's already. In case you missed the invitation in the December NLX on page 2, to get your listing of how to draw all the ABC's with the Apple computer and any of the LOGO versions (please specify), send a self-addressed stamped envelope to NLX ABC's, Box 5341, Charlottesville, VA 22905.

Learning with — — — — LOGO

Have you ordered your copy of Dan Watt's new LOGO book from McGraw-Hill yet? If so, you may want to check your order.

We have been informed recently that there will be two separate books published, in a manner similar to those of Abelson's (See December NLX). One title, "Learning with LOGO," has appeared in many notices to date. However, what the notices did not say was that this is the title for the MIT (Terrapin/Krell) version of LOGO. If you have the LCSI LOGO (Apple LOGO), you should order the second title, "Learning with Apple LOGO."

The LCSI title has not received much publicity. This may cause some inconvenience to those of you with LCSI who have already ordered "Learning with LOGO" in advance of the publication date. If you act immediately, it may be possible to change your order before you receive the wrong book.

Release of "Learning with LOGO" is scheduled for sometime in January, 1983. According to information given us, "Learning with Apple LOGO" will not be released until two weeks later.

Many LOGO people are waiting eagerly for Watt's books. His writing style is excellent and his reputation in the LOGO field is outstanding. We may have another LOGO classic on our hands very soon.

LOGO to Pascal?

Two educators recently announced a plan to consider teaching a high school course in Pascal. However, they will be teaching their students LOGO for the first six weeks! They hope to investigate whether LOGO will provide an easy entry into Pascal, a high threshold language.

The course will be general in nature, and will not necessarily be designed for advanced placement candidates.

They would like to hear from other interested educators, particularly those with both Pascal and LOGO experience.

Write to:

Robert D. Nelson
Richfield Public Schools
70th St. & Harriet Ave. South
Richfield, MN 55423

Charles Schlimpert
Lutheran High School
2222 North Santiago Boulevard
Orange, CA 92667