# Doing Mathematics in a Logo Environment

*by William Fisher and Jim Jones*
*California State University, Chico*

*This article, which was originally submitted to the Western Educational Computer Conference in 1988, has been edited a bit to fit in our publication.*

For the past two years the authors have offered an experimental course at California State University, Chico that has integrated the computer as a principal tool in the doing of mathematics. Specifically, we offered an undergraduate course in geometry that utilizes the computer language Logo. The first task we gave our class (who were mathematics teachers, but had no previous experience with computers) was to write procedures that drew triangles, pentagons, etc. The results were quite predictable but they did allow the students the opportunity to become familiar with the ease of programming in Logo.

```
TO POLY.EXPERIMENT :ANGLE
FORWARD 50
RIGHT :ANGLE
POLY.EXPERIMENT :ANGLE
END
```

*POLY.EXPERIMENT 144*

*Figure 1*

## First Investigation

Our students began experiencing the process of doing mathematics by looking at the following question: What happens if we start with a specific angle and repeat the sequence of commands FORWARD and RIGHT as many times as we need to make a completed figure? Here we immediately begin to tap the recursive power of Logo by introducing the procedure POLY.EXPERIMENT (figure 1.)

The way this program is written, the procedure will run forever. If you enter POLY.EXPERIMENT 90 the procedure will draw a square and continuously retrace the square until you interrupt the procedure. But what about 50, 89 or 35/17? Will these create polygons too? Or will some inputs create shapes that never close? This was the beginning of the first investigation for our class. A report of what then transpired can only be a collective summary of individual events. Our class was divided into pairs, each pair at a computer. Each group was allowed to pursue their own avenue of attack with planned times for pairs to get together to compare notes. Since the participants had a mathematics background, most groups were already familiar with the fact that the angle sum of the interior angles of a simple n-gon was (n-2)*180. They also realized that the turtle would turn a total of 360 degrees after completing a polygon. Hence, they reasoned (incorrectly) that the only inputs that would create a simple n-gon would the divisors of 360. It should be mentioned that this analysis was done with the aid of the computer; the computer was then used as a testing ground: divisors of 360 were inputted and regular simple n-gons were drawn. Soon after, every group started to produce examples of figures like these that closed and the angle input was not a divisor of 360, and we needed terminology to talk about them. Most of the class were sure they were not polygons (because they had crossings.) Actually quite an interesting discussion about definitions took place before everyone was happy with the name of "star polygon" (non-simple regular polygon.) Hence the figure would be called a star 5-gon. Now we had a new set of questions to pursue...Is this the only star 5-gon we can create,...if not, what other inputs create star 5-gons...and more generally, for any given *n*, how many different n-gons can be produced and what inputs produce simple n-gons and what inputs

*See Environment... page 11*