Learning Curve

This fictitious but typical example shows how LOGO encourages a group of children to solve problems they haven't encountered before.

```
TO CURVE
REPEAT 80
  FORWARD 1
  RIGHT 1
END
CURVE
```

'If we want petals then we need to draw a curve.'
'But the turtle always moves in straight lines.'
'What if we made it move a short distance, turn just a little bit, then move forward again, and so on — that would be like a curve.'
'OK, the smallest distance is one, and the smallest angle is one. Let's try doing that eighty times.'

```
CURVE
CURVE
```

'There, that's just what we want.'
'Two of them will make a petal — let's try it.'

```
TO PETAL
CURVE
RIGHT 90
CURVE
END
PETAL
```

'That's nothing like a petal — what happened?'
'It just carried on from the last curve — we should have told it to go in another direction.'
'But how much do we make it turn?'
'Let's try ninety — that often works.'
'And let's make a new word — PETAL — that will save time.'

```
TO PETAL
CURVE
RIGHT 100
CURVE
END
PETAL
```

'That's better, but ninety wasn't enough — what shall we try next?'
'Let's try and work it out, instead of guessing.'
'Yes — remember we learnt that if the turtle goes right round anything it turns through a total of three hundred and sixty.'
'Well, we know it turns through eighty on the first curve, so it must turn through eighty on the way back — that makes one hundred and sixty.'
'Leaving two hundred to be turned at the point of the petal.'
'No, because to get back to the position it started from, it would need to turn round at the other end too.'
'So we should try half of two hundred.'

```
PETAL
PETAL
PETAL
PETAL
```

'Great — four of those will make a flower.'

```
TO FLOWER
REPEAT 4
  PETAL
  RIGHT 10
END
FLOWER
```

'That's not much good, it's lop-sided.'
'We forgot to put in any turns between the petals.'
'But why didn't it draw them on top of each other, then?'
'Because when it's drawn one petal, the turtle is facing one hundred to the left of where it originally started.'
'So the petals are turned round by one hundred to the left each time.'
'That looks about right — what we want is ninety, so let's add in a right turn of ten between each petal.'

```
FLOWER
RIGHT 180
FORWARD 100
RIGHT 180
PETAL
```

'At last! Turn the turtle round and we can draw the stalk.'
'One hundred ought to be long enough.'
'Let's have a leaf on the end — it can be the same as a petal.'
'But remember the angle this time — the turtle needs to be turned right round.'

LIZ DIXON

a powerful device with which children can learn the basic concepts of spatial relationships, taking them up to advanced geometry.

Control of the turtle, however, is only one small application of LOGO, but because it is the most visually interesting it is the most publicised aspect. Of greater importance is the concept of building up simple ideas into more sophisticated ones, and conversely the breaking down of large problems into smaller problems of a kind that have previously been tackled.

These processes can be clearly seen in the imaginary conversation of a group of children learning to instruct the turtle to draw a flower (see box). They start off with only three available commands: FORWARD — which moves the turtle forward by a specified amount; RIGHT — which turns the turtle through a specified angle; and REPEAT, which repeats the lines indented in the program a specified number of times.

From these fundamental ideas the children first construct a 'tool' — a program — for drawing a curve (TO CURVE . . . END). This whole sequence can now be called up simply by typing CURVE. Similarly, after experimentation and further learning, a PETAL command is defined, which makes use of the CURVE command. Eventually a command FLOWER, which will draw the complete picture, is developed.

LOGO is not the only language to incorporate such structures (another is FORTH — see page 150), but it is the only one designed to be used by young children. It does away with many of the formalities and procedures associated with programming in other languages. Indeed, the aim was that the child shouldn't be aware that he is programming a computer — only that he is solving a problem.

In some learning situations, the pupil does not even get involved at this level. The teacher sets up a series of powerful tools using LOGO, which all relate to a particular subject or area of knowledge. The child is then allowed to explore the subject using the tools and discover it for himself. These areas are called 'microworlds' — limited environments in which the computer is used to simulate something in the real world or some area of knowledge.

The best example of a microworld is probably the LOGO model of Newtonian physics. Though Newton's First Law states that without the influence of external forces a body will continue to move in a straight line at a constant speed, young minds observe that in the real world everything slows down. This causes a blockage to learning. Using LOGO, however, a microworld can be set up in which everything behaves in true Newtonian fashion, and with the aid of tools to push objects around the screen, children soon learn all three of Newton's laws for themselves.

LOGO is a powerful concept well worth learning about with a home computer. Floor turtles are available but are not cheap. Versions of LOGO that use screen turtles are becoming available for several popular home computers.