DOCUMENT RESUME

ED 290 461 IR 013 117

AUTHOR	Nolan, Pat; Ryba, Ken
TITLE	Assessing Learning with LOGO.
INSTITUTION	International Council for Computers in Education, Eugene, Oreg.
REPORT NO	ISBN-0-924667-31-1
PUB DATE	86
NOTE	88p.
AVAILABLE FROM	International Council for Computers in Education, University of Oregon, 1787 Agate Street, Eugene, OR 97403-1923 (1-4 copies $12.50 each, prepaid).
PUB TYPE	Guides - Non-Classroom Use (055) -- Reports - Research/Technical (143)
EDRS PRICE	MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS	Check Lists; *Cognitive Development; *Cognitive Processes; *Computer Assisted Instruction; Elementary Secondary Education; Microcomputers; *Models; *Programing; Programing Languages; *Self Evaluation (Individuals); Worksheets
IDENTIFIERS	*LOGO Programing Language

ABSTRACT

The first is a series of booklets which present a new model for assessing and developing the thinking processes in which learners engage as they work at each Logo level, this booklet focuses on the method for assessing learning at the levels of basic Turtle commands, repeats, and procedures. It contains all the necessary materials--checklists, assessment worksheets, and activities--for developing the six main thinking processes at each of these three levels. The thinking processes to be developed are identified as coding, exploration, prediction, analysis and planning, creativity, and debugging. The methods and activities have been especially designed to highlight the role of the educator as a facilitator of learning who guides students to reflect on their own thinking as they come into contact with powerful ideas at the beginning levels of Logo. All of the ideas and methods presented here can be applied with almost any version of Logo on any type of computer. While the reading age of the items is aimed at the upper primary and junior high levels, the activities, assessment tasks, and checklists were designed to be easily adapted for use at any educational level with both child and adult learners. (EW)

About the Authors

Pat Nolan is a Senior Lecturer in Education at Massey University, Palmerston North, New Zealand. His main professional interests are in the fields of educational sociology, curriculum development and evaluation, organizational analysis, and education with computers. His M.A. and Ph.D. training in educational sociology has in recent years been extended to include research and publications on the development and evaluation of computer learning systems in diverse education settings.

Ken Ryba lectures on computers in education and special education in the Department of Education at Massey University, Palmerston North, New Zealand. He has conducted several research projects to evaluate the effects of computer education on the learning and behavior of handicapped people. His professional interests are in the areas of cognitive development, computer education for preservice and inservice teachers, and computers as an intervention tool in special education

This booklet is published by the International Council for Computers in Education, a non-profit, tax-exempt professional organization. ICCE is dedicated to improving educational uses of computers and to helping both students and teachers become more computer literate. ICCE publishes *The Computing Teacher*, a journal for teachers and teacher educators. It also publishes over 25 booklets of interest to educators.

The booklet prices given below are for prepaid orders. On other orders a $2.50 processing charge will be added.

QUANTITY	PRICE (U.S. $)
1-4 copies	$12.50 each
5-9 copies	$11.25 each
10-99 copies	$10.00 each
100+ copies	$8.75 each

Place your orders with:

INTERNATIONAL COUNCIL FOR COMPUTERS IN EDUCATION
University of Oregon
1787 Agate St.
Eugene, OR 97403-1923
(503) 686-4414

ISBN 0-924667-31-1

Assessing Learning with Logo

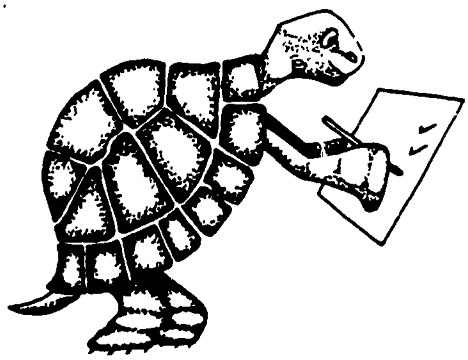
Table of Contents

Preface

This is the first in a set of three booklets which describe how to assess learning with Logo. They present a new model for assessing and developing the thinking processes in which learners engage as they work at each Logo level. Based on Seymour Papert's idea of the "educator as anthropologist," the model puts together Logo's twin programming and educational agenda. It does this by providing a set of practical methods which teachers, inservice trainers, curriculum developers, researchers, special educators and students can use to assess the development of such specific thinking processes as coding, exploration, prediction, analysis and planning, creativity, and debugging.

The first booklet presents the method for *Assessing Learning With Logo* at the levels of basic Turtle commands, repeats and procedures. It contains all the necessary materials—checklists, assessment worksheets and activities—for developing the six main thinking processes at each of these three levels. The methods and activities have been especially designed to highlight the role of the educator as a "facilitator of learning." In this role educators guide students to reflect on their own thinking as they come into contact with powerful ideas at the beginning levels of Logo.

Author's Note: Some of the examples in this book were written with Apple Logo I in mind. However, all of the ideas and methods presented here can be applied with virtually any Logo version on any type of computer. Also, while the reading age of the items is aimed at the upper primary and junior high levels, the activities, assessment tasks and checklists were designed to be easily adapted for use at any educational level with both child and adult learners.



Chapter 1
A Model for Assessing Learning With Logo

Why Assess Learning With Logo?

Over the past few years, educators have become increasingly interested in using the computer as a student-controlled learning tool. A lot of attention has been given to the view that children should be able to program the computer and use it to help develop their thinking skills. This perceived need for computer education and suitable learning tools has led educators all over the world to the "discovery" of the computer language Logo. Originally developed under the direction of Seymour Papert at the Massachusetts Institute of Technology more than 18 years ago, Logo use has grown at an incredible rate. During 1982 alone, Logo use in schools increased from a few sites to hundreds and then thousands of classrooms all over the country. Since that time it has continued to gain popularity as a computer learning tool for people of all ages.

Following on the wave of Logo's popularity as an "object for children to think with," a large number of programming manuals and activities books were written with the aim of explaining to users how they might go about learning to program with Logo. The best of these manuals, by such authors as Harold Abelson (1982), Shirley Torgerson and Mary Kay Kriley (1984), Daniel Watt (1984), The Minnesota Educational Computing Corporation (1982) and The Apple Corporation (1983), to name but a few, are thoroughly consistent with Logo's educational philosophy. With the aid of these manuals, people of all ages and abilities, but especially children and adolescents, are learning how to program.

Good as they are, however, few of the books and manuals have given much attention to the systematic assessment of the nature and effects of learning with Logo. This is doubtless because of the tendency in the early days of Logo to focus on programming per se, to develop teaching strategies and to create a pool of activities (games, puzzles, simulations) necessary to ensure that Logo could be successfully implemented. Even the book *Mindstorms*, by Seymour Papert (1980), does not explain, beyond the main concepts of procedural thinking and debugging, how to systematically develop the learning-assessment approach to Logo. As Papert (1980) puts it: Students and teachers alike should regularly step back from their work in order to:

- Reflect on what it is they have been attempting to do;
- Appraise the quality of their Logo products and designs; and
- Review progress with the acquisition of Logo programming and problem solving skills and abilities.

Not to make this assessment regularly as an in-built part of working with Logo is, we believe, to run the risk of bypassing the main objectives that Logo was invented to accomplish: to provide individuals with new possibilities for learning, thinking and growing emotionally as well as cognitively (Papert, 1980). This is not to say, of course, that the recreational value of Logo should be denied or even downplayed. It is often the case with the creation of games and in playing with Logo that children make exciting discoveries, begin to develop a sense of fun and infuse a sense of purpose into their learning. When, however, the use of Logo is motivated by a larger educational purpose—i.e., the development of new ways of thinking and the creation of new knowledge—then a more systematic and comprehensive approach to the assessment of learning is required than presently available in the existing Logo learning manuals. It was in recognition of this requirement that we saw the need to develop a Model for *Assessing Learning With Logo*.

Assessing Learning With Logo incorporates both a programming and an educational agenda. In putting these two agendas together, recognition is given that learning to program with Logo, while both enjoyable and rewarding, is but a means to an end: the development of mind and the powerful high-level thinking necessary if this development is going to happen. It is helpful to think of the programming dimension as including the content of Logo (e.g., basic Turtle commands, repeat statement, procedures) and the educational dimension as including the specific thinking processes (coding, predicting, analyzing) required to learn the content. When students working with Logo begin to move from one programming level to the next, they bring into play various kinds of thinking which enable them to process Logo's programming content. The thinking processes and the content interact with one another as the learner progresses step by step through Logo's programming levels.

According to Papert, the main thinking processes that learners must develop and use if they are to make progress with Logo are:

- Exploration, which includes such key sub-processes as coding, experimenting, predicting, hypothesizing and model-building;
- Procedural thinking, the process of solving problems by doing analysis and planning; and
- Debugging, the process whereby a learner becomes highly skilled at identifying and correcting mistakes.

Assessing Learning With Logo incorporates all of the above processes, but in doing so separates coding and prediction and adds one other—creativity. The six resulting processes—coding, exploration, prediction, creativity, analysis and planning, and debugging—represent the main thinking processes that learners typically employ in their Logo programming endeavors. These processes are represented, either implicitly or explicitly, in all of the main popular Logo manuals.

A problem with many of the Logo manuals, however, is that they discuss the development of thinking skills in very general terms without regard to the type and complexity of programming operations involved. To overcome this problem, *Assessing Learning With Logo* was designed so that the six specific thinking processes would correspond with each programming level. In this way, it is possible to assess a learner's thinking processes in relation to the Logo level at which s/he is working.

By relating the educational and programming agendas to one another, *Assessing Learning With Logo* provides a way of specifying the exact type of thinking processes that a learner engages in at each Logo level. This is done by using a set of assessment tasks, thinking skills checklists, and activities which are provided for each of the main programming levels. The six processes are, in effect, the concrete educational objectives that individual learners can actually achieve. Once achieved at each Logo level they may be developed further in subsequent programming and/or applied to solve problems in other learning contexts (e.g., written compositions, music, science).

By thinking of programming and the development of intellectual abilities in this means-end way, we can grasp the real educational significance of Logo. To this end, *Assessing Learning With Logo* reflects both a technically sound programming agenda and an educationally sound learning agenda.

Who Can Use *Assessing Learning With Logo*?

The educational agenda of *Assessing Learning With Logo (ALWL)* is intended to support everyone who uses Logo, including: teachers, curriculum developers, researchers, inservice trainers, special educators, and students of all ages and ability levels. Rather than providing a precise set of methods, the aim is to provide you with a framework to develop a set of educational objectives which coincide with Logo's programming levels.

Teachers can use *ALWL* to measure the progress of individual students and to tailor learning programs to meet individual needs. The checklists and assessment tasks provide a systematic and ongoing record of learning with Logo. By focusing on the thinking processes at each programming level, *ALWL* can provide meaning and direction to teachers and students alike. Teachers can also apply *ALWL* methods to measure the overall progress of groups and classes of learners who are working with Logo. This can be done by focusing on the development of specific thinking processes (e.g., analysis and planning, prediction) or by using the methods to help devise a special-purpose curriculum for the classroom.

Inservice Trainers can use *ALWL* to illustrate how Logo can be used to develop specific thinking processes at each Logo level. The benefit of using the assessment methods in this way is that it encourages teachers to focus on the educational and programming aspects of Logo. The progress of teachers as learners can be monitored with *ALWL* assessment tasks and checklists.

Curriculum Developers can use *ALWL* to prepare teaching and learning materials which are aimed directly at the assessment and development of students' thinking processes rather than at the instruction of programming content per se. Used in this way, *ALWL* provides a method for translating Logo's programming operations into a sequence of concrete educational objectives at each programming level. It then becomes possible to design relevant tasks which can be used for observing learners' progress as they solve problems and move, step by step, through the various stages of Logo programming.

Educational researchers can apply *ALWL* as a set of tools for measuring learning outcomes of groups and individuals. *ALWL* offers a systematic method for collecting data and analyzing the progress of individuals' learning with Logo. With the increasing call for information on how Logo aids intellectual development, ALWL provides a means for monitoring and evaluating both general classroom applications and specialized projects.

Special educators can use *ALWL* as a means of tailoring programs to suit individual needs. For example, students may need to develop certain thinking processes such as coding and exploration before they are ready to engage in the planning and analysis of more complex projects. The activities and assessment tasks in *ALWL* can be used to encourage students to direct their own learning and to take responsibility for the outcome of their decisions. The sequence of thinking processes and checklists provides a step-by-step description of learning objectives. Used in this way, *ALWL* offers a method for developing Individual Education Programs (IEPs) that comply with State and Federal Laws.

Learners of all ages and ability levels can use *ALWL* as a method for assessing their own progress with learning Logo. Such self-evaluation procedures are consistent with the spirit of Logo and stress the role of the learner as being at the centre of the learning process. *ALWL* provides a "language of thinking" which learners can use to articulate their problems and to share their discoveries with others. In this way, the thinking processes reflected in Logo's programming operations become personally meaningful concepts rather than abstract ideas.

The Model

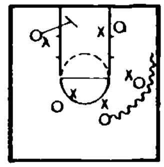
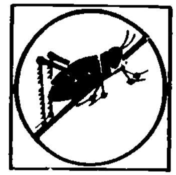
The Model for *Assessing Learning With Logo*, with both the educational and programming agendas built in, is presented in Table 1. In the table, the content or programming dimension appears on the vertical axis and contains the nine Logo (programming) levels. The process or educational dimension appears on the horizontal axis and contains the six thinking or problem-solving processes as they are referred to by Papert, Abelson, Watt and many other Logo educators. The resulting 54-cell matrix provides all of the criteria necessary for assessing students' progress as they learn to program and as they develop their thinking abilities in Logo's computer learning environment.

8

Table 1.
Model for Assessing Learning With Logo

Thinking Processes

Logo Level

Definitions Level	1. Coding	2. Exploration	3. Prediction	4. Creativity	5. Analysis & Planning	6. Debugging
						
A Basic Turtle Commands	Identifying and abbreviating Turtle commands and organizing command combinations	Changing commands and command values to alter size and shape of design	Estimating distances and degrees of turn in Turtle path	Creating geometric shapes and designs using all the basic Turtle commands	Identifying all the elements and steps to construct patterns, geometric shapes and designs	Interpret error messages and correct simple syntax and design faults
B Repeat Commands	Simplifying command lists using common factors	Experiments by changing repeat values for a given design	Estimating number of repeats to draw a particular pattern	Creating complex geometric designs using the repeat concept	Recognizing the part repeated in design	Detecting and correction of errors in repeat operations
C Defining Procedures	Combining basic Turtle and repeat commands to compose a new command	Using a procedure with other commands, e g , rotation of a square	Specifying the outcome of a given procedure	Creating a procedure from a "Turtle trip" previously solved, e g , defining a maze solution	Breaking down procedures into constituent parts	Locating errors in a procedure from comparisons of visual and coded output
D Editing and System Operating Procedures	Converting file maintenance tasks into operating system terms, i e , saving, loading, and erasing procedures	Changing command terms and values used in making procedures	Nominating outcomes of altered procedures and procedure combinations	Modifying an existing procedure to create a new one, using the "Edit" mode	Redesigning a procedure through identification and reconstruction of procedure components	Using the Editor to correct errors in a defined procedure, e g , adding, deleting, changing specific parts
E Sub-Procedures & Super-Procedures	Combining procedures to create a "Super-Procedure"	Experiments by adding, deleting and reordering procedures	Estimating sizes and proportions of and relationships between elements in a design	Building a project by using the super-procedure concept	Devising a method for integrating sub-procedures into a super-procedure	Identifying and correcting sequencing errors
F Inputs/Variables	Replacing a fixed value with a variable statement	Observing the effects of changing the values of outputs to variables	Estimating the outcome of varying the relationships between variables	Applying the concept of variables for controlling sizes and proportions of procedures and sub-procedures	Determining the type and location of variables required to control elements of a program	Identifying incorrect placement, sequencing and syntax of variable statements
G Recursion and Conditional Statements	Inserting a new command which causes a procedure to repeat a set of commands and/or increases or decreases an input value of a variable	Altering the form, value and location of a recursive statement	Describing the final outcome which will result from a specific form of recursion	Applying the concept of recursion for the control and regeneration of a new procedure	Determining the form, sequence and location of a recursive statement within a procedure	Identifying and correcting errors in the placement, syntax and sequencing of recursive statements
H Using Numbers, Words and Lists	Inserting commands into procedures that enable words and lists to be specified	Trying various list processing combinations, e g FIRST, LAST, BUTFIRST, to observe the effects on output	Specifying the logical relationship between input and output	Applying the concepts of numbers, words, lists to name and use sets of information in projects	Identifying the steps needed to "converse" with the computer's numbers and words	Locating and correcting errors that affect the logical operation of numbers and lists
I Writing Interactive Programs	Inserting a new command list to read input from the key board	Observing the effects of changing messages and commands in response to user-specified input	Estimating the Turtle's behavior in response to keyboard control of an ongoing process	Applying the concepts of interactive programming to design games and projects	Determining the form and sequence of procedures required in the design of an interactive program	Detecting and correcting conceptual errors within interactive procedures

There can be no doubt that by now many thousands, perhaps hundreds of thousands, have learned to program a computer with Logo, and for this reason alone Logo is one of the most popular learning tools ever invented. The important question, though, is, "Does it develop the mind as Papert intended?" Our provisional answer to this question is yes, with the qualification that this development of mind requires the input of measured and periodic help from a perceptive teacher who is competent in programming with Logo. In other words, students can indeed make substantial progress in learning to program by working on their own. Over and above this learning of programming, however, the development of the thinking processes contained in the Model requires a social context and deliberate attention in order to flourish and grow in the mind of the learner. This again is the educational, as distinct from the programming, purpose of Logo to which Papert refers.

The *ALWL* Model develops this educational purpose of Logo by providing a method for observing the extent to which a learner understands and applies the "powerful ideas" contained in the programming operations. This is done by using the assessment tasks and thinking processes checklists which correspond to each Logo level. For this purpose, it is essential to define the programming content at each level and to have definitions for each of the thinking processes in which learners engage as they interact with Logo.

The Nine Programming Levels

Logo's programming agenda, powerful yet simplicity itself, provides the structure for *ALWL*. This structure, following most of the popular manuals, goes from the simple to the complex and from the easy to the difficult programming concepts for students to learn. Each assessment level corresponds to one of the following nine discrete programming levels:

- Basic Turtle commands
- Repeat commands
- Defining procedures
- Editing and system operating procedures
- Sub-procedures and super-procedures
- Variables as inputs
- Recursion and conditional statements
- Using numbers, words and lists
- Writing interactive programs

Division of the Logo programming continuum into nine programming levels rather than the usual seven or eight made it possible to create a more orderly and easily managed system for assessing learning with Logo.

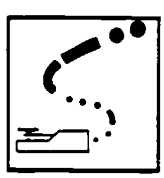
ALWL's nine programming levels are, we believe, a faithful representation of both the programming structure of Logo and the bottom up, inductive learning approach recommended by Papert. Teachers and learners may, however, choose for any number of reasons to work through the Logo levels in a sequence different from that presented here. In fact this is very likely given the great diversity of people learning and wanting to learn Logo. For instance, adults coming to Logo for the first time but with a good understanding of logic concepts and geometry may learn Logo better if the teacher were to use a top down, deductive, rather than bottom up, inductive, approach. With a group of young children, however, an approach more in keeping with that recommended by Papert may be the best way to proceed.

The Six Thinking Processes

As children begin to learn programming with Logo, they begin also to start thinking about the precise strategies that they use to make the computer do what they want. These strategies, problem-solving in nature, "turn the child into an epistomologist, an experience not even shared by many adults" (Papert, 1980, p. 19). In this way they become "active builders of their own intellectual structures." Equally important, however, the students require encouragement, guidance and support from without in order to both identify and then to consciously develop the thinking processes involved as they embark on this general process of intellectual building.

Papert's ideas concerning how children learn to "think about their own thinking processes" with Logo is akin to Piaget's notion that advanced thinking is marked by the ability to view one's own behaviour as something which can be analyzed (Piaget and Inhelder, 1969). In Logo programming, children consciously solve problems using new and often unfamiliar strategies. They learn to communicate the organization of a task (coding) and the processes for solution (exploration, prediction, analysis and planning) to one another, to the teacher and to the computer. They also learn to analyze their own thinking processes for errors and revision strategies (debugging). In addition, because Logo requires continuous invention and construction of one's own graphic projects, creativity may be engendered (Clements, 1985). Creativity can be viewed as a general form of intelligence which involves the combination of ideas and thinking strategies leading to the invention of new objects or concepts.

The definitions of the six main thinking processes which are represented here serve two purposes. First, they provide the necessary clarification of the general meaning of each concept. Second, they indicate the part that each thinking process plays in the mind of a learner as s/he begins to acquire and apply some of the powerful ideas through programming in Logo.



Coding

Coding refers to the process of translating ideas in a communication (e.g., a message, command or instruction) from one form into another equivalent form.

With the use of this process the learner experiences first hand what it means to take ordinary language messages and convert them into a form (code) that a computer can understand. At the lowest programming level (basic Turtle commands), the coding process is literal and requires mainly that students recognize direct code equivalences, write simple statements and translate Turtle commands from long into short form. As programming with Logo develops, coding accordingly becomes more complex, e.g., the communication of directions for specifying commands to be repeated, creating procedures, and combining procedures to form super-procedures.

At more advanced levels still, coding involves the communication of symbols for altering inputs to procedures (variables), specifying rules for checking, modifying and stopping procedures, and writing commands in which there is communication between computer and user. In this way, coding activity effectively establishes an interface between the programming and educational agendas of learning with Logo. On the one hand, coding provides the student with a way into programming. On the other hand, coding triggers the mind to go the next step and bring into play the full range of thinking processes upon which making progress in learning with Logo depends.

Exploration

Exploration refers to the process whereby a learner may, as long as s/he wants, play with an object—maybe a new-found toy or tool—to see how it works and what it can do.

Although exploration is but one of the six thinking processes that Logo invites novice computer learners to use, its persistent use over time will affect, much more than the others, the quality and worth of all the learning that follows. It is for this reason that Papert himself, along with many other notable Logoists (Abelson, 1982; Torgerson, 1984; Watt, 1983; Wills, 1984), stresses the importance of sustained work in this mode. Exploration can, in principle, go on indefinitely, seemingly without result. While learners are preoccupied in this mode, questions of great moment often are formed and may remain in mind as powerful but unclear symbols for a long time (Hodgkin, 1976). However, discovery will sooner or later occur.

With Logo, the forming of questions and the act of discovery are each culminations of a growing sense of pattern. Significant events will take place uncovering some structural relationship or way of performing an action that was not evident to the learner before. In this way students embark on a voyage of discovery learning with Logo and, in their own good time (though often much sooner than they or the teacher had thought), break through in their thinking from one mode to the next. This discovery learning perspective is central to Papert's vision of how, "in teaching the computer to think, children embark on an exploration about how they themselves think" (*Mindstorms*, p. 19).

Prediction

Prediction is the process whereby an individual, wanting to test out (show to be true or to be false) a hunch, hypothesis or idea, states the likely outcome of taking a specific action(s) or causing a specific event(s) to occur.

Within textbooks on scientific method, prediction is often portrayed as a complex and difficult activity to do. The prediction of previously unobserved phenomena is, of course, always a spectacular scientific achievement. Moreover, the use of prediction to test a particular scientific theory is an especially important aspect of scientific method. But contrary to popular belief, neither prediction nor the scientific method to which it belongs is the exclusive property of either scientists or even of the scientific community. Everyone, but especially young children, is a scientist at heart, and we all constantly seek explanations and regularly make predictions about the outcomes of the specific actions and events that seem to influence our everyday lives.

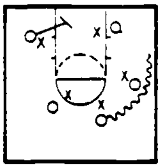
At the level of basic Turtle commands, prediction means estimating distance and turns based on an intuitive concept of what the Turtle is able to do. At higher Logo programming levels, prediction involves, for instance, estimating the effects of changing variable values in order to alter the size of a given shape or programmed design. Whatever the level may be, a prediction is made "almost knowing the answer already" (Papert, 1981). Logo's variously simple and high level programming commands are then used to structure a test whose purpose is simply to confirm or deny.

Creativity

Creativity refers to the process whereby individuals, through combining ideas and seeing relationships, design, make or invent a concept or an object in a way that is new and has personal meaning for them.

Creativity is a multifaceted process in that it incorporates such stylistic and personal qualities as ingenuity, originality and inventiveness. Creative invention, which is the quality best encouraged by Logo, can be either convergent or divergent in nature. The convergent Logo inventor typically starts with a problem that offers a variety of apparent possibilities and focuses attention down to a single likely solution. In contrast, the divergent Logo inventor starts from a problem or a goal that offers a variety of possibilities and then spreads his or her thought to look for and accept many likely solutions.

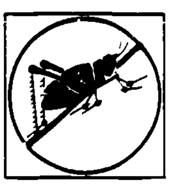
Either way, "the printed page cannot capture either the product or the process: the serendipitous discoveries, the bugs, and the . . . insights all require movement to be appreciated. . . . Something the computer affords the child is the opportunity to draw in motion, indeed to doodle and even to scribble with movement as well as with lines. Perhaps they will be learning, as they do so, to think more dynamically" (*Mindstorms*, p. 93). They may also create what are for them new and personally rewarding objects and ideas using the structured tools of creative invention that Logo's unique kind of programming language provides. This invention may amount to no more than simple geometric patterns, shapes and designs. With initial success and when valued peer and teacher praise is forthcoming, then students free up and go on to become more fluent, flexible and original thinkers as they make progress at their own individual rates. In this way, creative learning with Logo leads students to gain higher-level programming skills and inventively employ increasingly powerful ideas to achieve educational ends that they personally choose to pursue.



Analysis and Planning

Analysis and planning refer to the composite process whereby an individual subdivides a concept, problem or task into its constituent parts and then, through recombination, devises a concept implementation, a problem solution or a task completion procedure.

The distinctive feature of analysis and planning activity is that it requires explicit identification of the parts that make up the whole of a concept, problem or task. More than all of the other five thinking processes, with analytical thinking the learner must develop a conscious awareness of the intellectual task s/he is performing and know the rules for reaching a valid and workable conclusion. Here, both deductive and inductive reasoning may be applied. Deduction means reasoning from the general to the particular; the test of validity being whether or not a conclusion is consistent with the original premises. This has become popularly known as "top down" analytical thinking. Induction means reasoning which goes from the particular to the general. In programming terms, this approach is often referred to as "bottom-up" thinking. Systematic analysis and planning with Logo requires that learners use both deductive and inductive methods in combination. In this way, they "can learn more, and more quickly by taking conscious control of the learning process" (*Mindstorms*, p. 113).



Debugging

Debugging refers to the process whereby learners judge the worth of their self-made concepts, products and projects and assess the adequacy of the methods used to create them.

When defined in this way, the debugging process is a subset of evaluative thinking that constantly goes on as an integral part of our everyday lives. Significantly, however, evaluative activity within Logo's learning environment is aimed at helping learners judge for themselves the adequacy of the methods they use and the quality of the products they make. As Papert points out, debugging is the means whereby the learner becomes highly skilled at isolating and correcting mistakes that would otherwise prevent a program from working. As a result, "children learn that the teacher too is a learner, and that everyone learns from mistakes" (*Mindstorms*, p. 114). Within the assessment model, ongoing "debugging" places high value on making and correcting mistakes as an inevitable and constructive part of learning with computers, thus eliminating the perception that errors are a sign of personal incompetence. The power of debugging resides in the requirement that it makes of learners to stand off from the work in which they are engaged and practice the art of "thinking about their own thinking."

When taken together, the six thinking processes encompass the working mind of the Logo learner in action. The relationship between the processes is similar to that between the basic colors in a rainbow. There, these colors—red, orange, yellow, green, blue and purple—can plainly be seen. Between each color, however, there is a region that is neither one nor the other, but made up of both. The same is true of the Logo thinking processes. Quite often all the processes seem to be operating at once. Perhaps coding merges in with analysis and planning or some other complicated process combination occurs.

For assessment purposes, the important point for learners and teachers to know is that each process has unique elements. The general definitions describe the uniqueness of each process. Specific criteria for observing the separate thinking processes in action at each Logo programming level are presented in the chapters that follow.

How to Assess Learning With Logo

The approach that we outline here for the assessment of learning with Logo differs markedly from the psychological tests traditionally used in schools to evaluate the outcomes of school-based learning. There, assessment most frequently occurs at the end of a programme of study or a unit of school work. The process whereby students learn is, of necessity, often overlooked. In contrast, the form of assessment sympathetic with the learning goals of Logo is formative rather than summative. As such, it is consistent with the anthropological approach to assessment in education, developed by Hamilton and Parlett (1975). The main advantage of this approach is that it enables teachers to incorporate both the processes and the products, along with the social context of learning, into the ongoing assessment of students' learning.

The conduct of assessment, using this anthropological method, is, in effect, to adopt the suggestion made by Papert (1980) himself that:

> The educator must be an anthropologist. The educator as anthropologist must work to understand which cultural materials are relevant to intellectual development. Then, he or she needs to understand which trends are taking place in the culture. Meaningful intervention (with Logo and computers) must take the form of working with these trends. In my role of educator as anthropologist, I see new needs being generated by the penetration of the computer into personal lives.

Assessment using the anthropological perspective recommended by Papert seeks to discover:

1. What is the student's attitude about and point of view on computer-based learning;
2. What is the meaning that students take from their learning with Logo; and
3. With what concepts, strategies and principles do students carry out the Logo programming and problem solving tasks that are set for them, or which they set for themselves.

With traditional learning, the assessment process is usually controlled by the teacher. With Logo, however, the best

12

results seem to be achieved with students and teachers working together in groups, asking pertinent questions, discussing difficulties, diagnosing problems and sorting out specific new learning directions (Watt, 1982). Here, the stu[illegible] occupy centre stage. They reflect about their own th[illegible] and take conscious control of the learning process by a[illegible]ating and analyzing their own behaviour (Papert, 1980). In this way, both they and their teachers are able to appraise the adequacy of their programming and problem solving endeavours and then judge the necessity for modification and change. These reflections and judgments are, we believe, crucial in getting to know the kind and complexity of problems a particular student can solve and the level of computer programming achieved in his/her learning with Logo. Equally important, they are crucial in determining the type and the extent of teacher intervention necessary to ensure the personal enjoyment and s[illegible]e of accomplishment that seem to go hand in glove with a student's ongoing mastery of Logo.

Once armed with the complete model for assessing learning with Logo, the role of the educator as anthropologist can effectively be brought into play. Just as the good anthropologist must participate in a culture in order to observe its origins and trends, the Logo educator must learn to observe in precisely the same way. This assessment must be done in two main stages:

1. An exploratory stage, when the teacher seeks to establish where the learner is cur[illegible]ly at in his or her thinking and programming with Lo[illegible]
2. An inspection stage, when interventions are made, using the model to both diagnose and help students develop from their present Logo learning condition.

The Exploratory Stage of Assessment

During the exploratory stage, the teacher sets a stage where students can both play and work with a purpose. For instance, this might involve providing an opportunity for students to "play around" with the Turtle. At other times, students may be encouraged to begin with an idea of what they want to d[illegible] and a crude plan of how to do it. The teacher is then strategically placed to encourage, help and generally support students as they get on with their work. To do this job well a combination of direct observations, structured conversations and, if necessary, interviews, is an appropriate and effective way to proceed.

Seen from a distance, the teacher may at first seem like a perfect recording machine: neutral, receptive and unobtrusive. This aloof posture does not last long, however, as the Logo environment and the assessment needs of the day inevitably require a much more inquisitive stance. Perhaps the teacher will ask students in an open-ended sort of way about their intentions:

- "What are you doing?"
- "What were you trying to do?"
- "What do you like best?"
- "Is that what you wanted to do?"
- "What do you think you will do next?"

Conversations started this way enable the teacher to probe into the reasons for actions and diagnose and help resolve any learning problems a particular student might have. Furthermore, good per[illegible]nal relationships build between teacher and students, based on mutual respect and rapport.

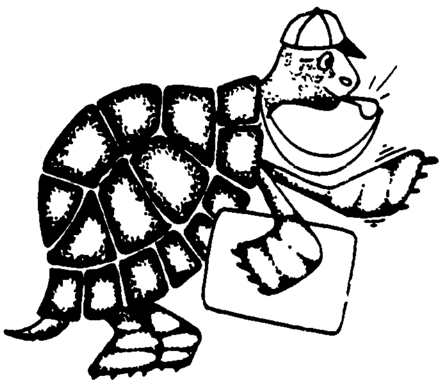
The Inspection Stage of Assessment

Once learning with Logo has started, ongoing assessment involves using the assessment model as an inspection device. The teacher applies the methods contained in the model to: identify the programming operations and thinking processes each student is using; and diagnose problems and make well-informed judgments of the progress each student has made. Here the model serves two purposes. On the one hand, it provides an overall concept of the range of learning possible with Logo. On the other hand, it provides the required framework with which to assess and develop children's thinking processes at each Logo level.

There are five separate but related parts to *ALWL* at each programming level. These include:

1. **Assessment Objectives**—These identify the six main thinking processes that learners must employ as they interact with Logo at each programming level. Each objective represents an educational goal that ongoing assessment helps students achieve.
2. **Descriptions of the Six Main Thinking Processes**—These specify the meaning of each thinking process as it comes into play at each Logo level. With each meaning clearly defined in this way, both learners and teachers can directly focus their attention on the specific processes the learner is required to master and use.
3. **Activities for Assessing and Developing Each Thinking Process**—These comprise a set of practical suggestions and assessment activities that teachers can use to establish the programming and thinking competencies students should possess at each Logo level. The activities are equally suited for use by learners working alone or groups of students who have common interests and needs.
4. **Assessment Tasks for Each Logo Level**—These comprise a structured set of assignments that provide baseline information on how a particular student employs specific thinking processes to solve programming problems at each Logo level. By arranging the assessment in this way, learners who need to further develop one or more of their thinking processes can select tasks which will help them progress beyond the particular stage of thinking and programming they are currently at.
5. **Thinking Processes Checklists**—These lists provide objective criteria that can be used to check off and record the progress particular students have made in their thinking and programming with Logo. With students' achievements recorded this way, both learners and teachers can easily construct and then regularly update a register of the learning progress each student has made.

Taken together, these five components make up the method for *ALWL*. This method, built in to the assessment chapters that follow, provides the "tools of the trade" which educators can use to observe and assess how learning with Logo extends each learner's mind.



Chapter 2
Basic Turtle Commands

A Powerful Idea

In many schools today, the phrase "computer-aided instruction" means making the computer teach the child. One might say the computer is being used to program the child. In my vision, *the child programs the computer* and, in doing so, both acquires a sense of mastery over a piece of the most modern and powerful technology and establishes an intimate contact with some of the deepest ideas from science, from mathematics, and from the art of intellectual model making. *

This mastery over the computer to which Papert refers is best achieved when the learner receives a thorough grounding in the use of Logo's basic Turtle commands. It is with these commands that all learners, irrespective of age and ability, immediately learn to program. That is, they instruct the computer to do what they want. More importantly, though, they begin almost at once to plan, predict and explore. As Harold Abelson (1982) points out, the basic commands have simple yet powerful effects, and with them it is possible to conduct "weeks of activities in programming and mathematics," exploring such questions as, "How does the shape of a POLY figure depend on the angle of input?" or simply creating intricate patterns and shapes. These initial products of programming with basic Turtle commands are indeed spectacular. But the real power of working with Logo, even at this simplest of levels, is the thinking and processing that it demands of the learner.

Accordingly assessment, at the level of basic commands, promotes programming in the masterful sense that Papert describes, but does this specifically by encouraging students to think, using the processes and strategies outlined in the assessment model. Furthermore, because of the cumulative nature of Logo, the focus of assessment is on both:

1. Successful initiation of students into learning—creating, exploring, discovering—with Logo; and
2. Early consolidation of these basic programming and thinking abilities which, thereafter, will effectively govern their learning progress in the computer environment.

In practical terms, effective intervention at the basic command level involves guiding students' learning with Logo in two separate but related ways:

1. Encouraging them to apply the full range of thinking strategies to produce personally satisfying shapes and designs; and
2. Counseling them when to move on to the next programming level or, more likely, advising them to resist this temptation when the skills and abilities for doing so have not yet been adequately practiced and mastered.

Here, the Logo learning experience is not, however, primarily "one of memorizing facts or of practicing skills. Rather, it is getting to know the Turtle, exploring what a Turtle can and cannot do. It is similar to the child's everyday activities, such as making mudpies and testing the limits of parental authority—all of which have a component of 'getting to know'. . . . While good teachers play the role of mutual friends who can provide introductions, the actual job of getting to know . . . cannot be done by a third party. Everyone must acquire skill at getting to know and a personal style for doing it" (Papert, 1980, pp. 136-137).

These styles of getting to know and using the basic Turtle commands may vary markedly, and while most individuals will progress rapidly to the next Logo level, a minority of students may not. Unlike other programming languages, however, achievement with Logo is best "measured" in terms of numbers of powerful ideas that the students experience and use, rather than numbers of programming statements memorized or the level of programming mastered (Wills, 1984). Within the basic commands, all students encounter the powerful ideas of coding, exploring, predicting, creating, analyzing, planning and debugging with Logo. These ideas or thinking processes, in their turn, provide criteria in the form of objectives against which to assess the extent and quality of students' learning at the level of basic Turtle commands.

Assessment Objectives

The following list of objectives identifies the distinct kinds of operational knowledge and thinking processes that all students, to a greater or lesser degree, experience and master during their introductory learning with Logo.

1. Basic Turtle commands are the primary means of telling the Turtle how to follow instructions (coding).
2. Basic Turtle commands can be used to make the Turtle move and behave in both random and purposeful ways (exploration).
3. Basic Turtle commands may result in both expected and unexpected outcomes (prediction).
4. Basic Turtle commands can be used either singly or in combinations (sets) to make Turtle paths, patterns and shapes (creativity).

*From *Mindstorms: Children, Computers and Powerful Ideas*, by Seymour Papert.

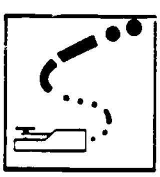
5. Basic Turtle commands, when used in programs, can stand on their own or be viewed as parts of a whole (analysis and planning).
6. Basic Turtle commands may contain errors which are readily found and ought to be corrected, thus making a program work as its author intended (debugging).

Although students' initial work using basic Turtle commands is usually conducted in private, they also enjoy getting "together with others engaged in similar activities because they have a lot to talk about. And what they have to say to one another is not limited to talking about their products: Logo is designed to make it easy to tell about (and appraise) the process of making them" (Papert, 1980, p. 180). In this way the Logo culture develops, and from the outset students' reactions and questions signify where they are at. With assessment activity already built-in, a basis exists for "more articulate, effective and honest teaching relationships" (Papert, 1980) than usually possible in educational settings of a more conventional kind.

Because assessment at the first programming level is formative rather than summative, the teacher's time should be predominantly taken up with observing and listening. Information thus gained may then be used to:

1. Diagnose each student's learning condition; and
2. Guide the essentially discovery learning process at the level of basic Turtle commands.

How to Assess the Six Thinking Processes



1. CODING

This ability in its most developed form involves the recognition of complex code equivalences and the translation of concepts from one code (e.g., ordinary language) into another (e.g., programming language). At the beginning of Logo, where the foundations are laid, coding means learning the Logo syntax and doing simple command translations. Accordingly, the main assessment aim here is appraisal of students' abilities to: memorize Logo's basic commands; apply the rules of syntax; and learn code equivalences, mainly by translating basic commands from long into short form or from English into Turtle commands. These basic skills can be developed and tested through the use of various decoding and recoding activities. The specific focus is on helping students think flexibly and openly in their coding with basic commands.

Activities to Assess and Develop Coding

You may wish to use the review items contained in this chapter, or to use materials of your own choosing, perhaps from other manuals. Match-mismatch and completion-type test formats, though less appropriate later on, are recommended as baseline assessment procedures. These procedures can be administered either at or away from the computer to both individuals and small student groups of similar ability. At the basic command level they provide the necessary data required in checking out a particular student's ability to:

1. Recall all of the basic Turtle commands;
2. Translate commands from long into short form and vice versa; and
3. Comprehend, in an elementary way, the concept of code equivalence.

Over and above these primary coding abilities, students may require explicit teacher or peer assistance in order to thoroughly master the accompanying syntax that, like balancing a bicycle, must become second nature in order to progress further with programming in Logo. Here, coding drill and practice games are an appropriate follow-up to assessment. While these types of games may include debugging elements, they not only serve to consolidate, through reinforcement, Logo's rudimentary coding skills, but do so in a climate of enjoyment and fun.

Specific examples of the suggested coding assessment activities are provided in the section that follows. It is worth noting the two complementary assessment strategies suitable for use with coding. The first requires translation of verbal or written directions into Logo commands. The second involves coding full Turtle commands in their abbreviated form. These can be done either as off-computer activities or through direct interaction with Logo. The coding review items can be presented as either informal assessment games or as more structured class activities under teacher direction. Regardless of the approach used, the underlying aim should be to provide learners with an opportunity to do a self-assessment of their own problem-solving skills. *(See worksheet on page 17)*

2. EXPLORATION

The main assessment here is monitoring and appraising students' preparedness to experiment with the new tools for procedural thinking that Logo makes available to them in the form of basic Turtle commands. Exploratory activity may be either random or purposeful depending on a student's temperament and learning style. Whatever the particular approach taken by any given student, the teacher's initial reaction should be one of approval and tacit acceptance. With teacher-student rapport thus established, the teacher then unobtrusively observes the range of commands that are used and the extent to which students:

(1) Try out specific commands to see what happens;
(2) Experiment, perhaps by substituting one command for another or simply changing command values; and
(3) Develop Turtle paths by using command combinations.

Activities to Assess and Develop Exploration

A. Simply ask students to construct a pattern or design of their own making. Observe the range of commands they are prepared to use. Judiciously ask *why*; what are their reasons for using the commands they do? When appropriate, suggest other possibilities by asking such questions as: What do you think would happen if . . . ? or, What happens to the Turtle when . . . ? or, How can the Turtle move without being seen? and so on. By asking students questions in this way, the teacher is able to accomplish three goals simultaneously:

   1. Assess where the student is at;
   2. Suggest alternative avenues for exploration; and
   3. Provide a model of the kind of questions the student should be asking her/himself.

B. Provide students with a simple program that contains a variety of commands and command values. Ask them to experiment by adding and/or deleting commands and also changing command values. Additionally, they should observe and record the effects of the changes they make. Because programming with Logo is structured and purposeful, students should be encouraged to keep their own personal Logo Logs for tracking the progress they make. Ideally, of course, this will be done using the computer itself.

C. Ask students, working with others in teams, to collectively devise command combinations of their own making, using the full range of basic Turtle commands. Following completion, these explorations might be discussed with the whole class or Logo group and then put on display. *(See worksheet on page 19)*

3. PREDICTION

This distinctive activity seems to come naturally to everyone, but especially to children who are constantly "assessing"; guessing at and estimating the outcomes of actions, others' as well as their own. At the level of basic Turtle commands, prediction ability is relatively easily assessed by simply comparing intentions (in the form of specific command sequences) with the actual products.

Activities to Assess and Develop Prediction

A. Present students with specific prediction tasks in the form of strings of basic Logo commands. Then ask them to draw the pattern or shape that the commands will produce. Testing the prediction (confirming or falsifying it) simply involves the student's keying in the command sequence correctly, then comparing the computer's output with the original drawing.

Again, keeping accurate records (just as a scientist would do) is an important habit to begin developing here. A variation of the following format may prove useful to start with:

LOGO RECORD LOG

Command List	Predicted Pattern/shape	Actual Pattern/shape

Within this suggested Log format, sufficient space should be made to include student and teacher comments and notes.

B. Repeat the assessment outlined in (A) above, but this time ask students to set their own prediction tasks.

C. Ask students, working in pairs or small groups, to do the following:

   1. Set prediction tasks for each other to complete;
   2. Discuss any falsified predictions, diagnose problems and propose solutions; and
   3. Record helpful hints.

By conducting prediction assessments in any one or all three of the ways suggested, the teacher is able to move around individuals and groups in order to observe the students in action and provide help and guidance when required. *(See worksheet on page 22)*

4. CREATIVITY

At all levels of Logo, creativity involves the use of inventive imagination. Basic Turtle commands are the initial "tools of invention" with which students begin making their own patterns, shapes and designs. The development of creative abilities with Logo is a process, however, not just of unfolding in a permissive atmosphere, but of aesthetic education. This education requires that a great deal of learning be done. Moreover, since both Logo and mathematics involve new worlds, they involve systematic instruction early on, structured by well-defined conventions and rules of procedure.

Given Logo's basic Turtle commands, however, students and especially young children will be happily active, and even progress through definite stages of accomplishment, all with minimal comments from the teacher. Yet just as natural curiosity needs to be endowed with intellect, a student's first creative efforts with Logo need the help of a discerning but appreciative teacher if this initial development is to be maintained. Here, teacher-student discussion enables students to step back from their programs and products in order to appraise the things they are making and examine the creative processes they use.

Accordingly, the assessment of creativity has three main aspects:

(1) Observing the manner in which students apply their new-found tools of invention;

(2) Appreciating the Logo products they have made or are making; and

(3) Taking notes on abilities displayed, difficulties encountered and problems resolved.

Activities to Assess and Develop Creativity

Even at the most elementary level of thinking and programming with Logo, there are seemingly endless possibilities for creative invention and growth. It should be noted, however, that students' best creative products may only emerge with the passage of time, when students have become fully familiar and confident with the new tools for invention and thinking presented to them by Logo. Consequently, the assessment of a student's creative production must also occur over time, based on the notion that, "being creative is not to have arrived at a destination, but to travel with a different view."

With these cautionary comments having been made, then assessment is primarily a matter of asking:

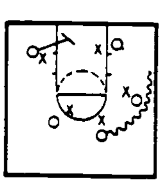
1. "What are you doing and what have you made?" While this question is deceptively simple, it is the key that opens the door to the thinking behind a student's ongoing creative production. To the extent that the situation calls for evidence of a more specific and a more immediate kind, then the teacher could ask students to:

2. "Make a pattern, shape or design of your own choosing using Logo's basic Turtle commands." Here the decision as to which and how many of the commands ought to be used should be left open to the individual student. This way, in the absence of externally imposed constraints, a better diagnosis can be made of each student's present creative capability with Logo. In addition, the basis is made for deciding the specific teaching and learning actions that might follow next.

   When ongoing assessment suggests that consolidation is required, the old adage, "Two minds are better than one" is good advice to follow. As Papert points out, children naturally follow it anyway when they spontaneously form their own informal groups in order to resolve a programming problem or discuss an exciting idea. Therefore, when the time is right, ask the students, working in pairs or in small groups to:

3. "Cooperate, combine your best ideas, and then make a pattern, shape or design of this group's own choosing."

Not only will the ensuing group activity provide the teacher with extra useful data, it will also create opportunities for students to engage in collective appraisals of their group project. This is excellent preparation for learning to be an objective critic of one's own best work. *(See worksheet on page 25)*



5. ANALYSIS AND PLANNING

Benjamin Bloom once commented in his widely known *Taxonomy of Educational Objectives* that the development of analytical reasoning is generally neglected in both elementary and high school curricula. This is not surprising, mainly because analytical thinking has been reserved for scholars. We have generally believed, erroneously, that only they have needed to give serious attention to the elements and processes of reasoning in the pursuit of their disciplines. As Papert has shown, however, even very young children know how to analyze and plan and they do this naturally, though implicitly, in their everyday living. Logo, right from the outset, explicitly helps develop these everyday, yet powerful, cognitive processes by asking the learner to:

(1) Identify the components of a design or problem;
(2) Envisage each as a separate entity; and then
(3) Put them together to make up the whole.

Once students have learned the basic Turtle commands, they begin to develop their own ideas for the creation of specific patterns and Turtle projects. These ideas can serve as the means for studying concepts of linear and angular measurement that are required to carry out more complicated projects. For example, students can draw their projects to scale using graph paper and protractors. The accurate measurement of angles and lines can then be translated into a set of Logo commands.

It follows then, that the assessment of a student's analysis and planning ability with basic Turtle commands should aim to both review and develop each student's writing and purposeful use of the three essential components of analysis and planning specified above.

Activities to Assess and Develop Analysis and Planning

A. Present students with a simple picture, pattern or shape AND a random list of more commands than needed to write a program that will produce it. Then ask them to compose the required program by:

   1. Selecting the commands they will need;
   2. Rearranging the commands into a programmable sequence: and
   3. Running the program to see if it works.

   This kind of task permits early diagnosis of each student's analysis and planning ability, thus enabling the teacher to provide encouragement and guidance when required.

B. As students create shapes and designs of their own, ask them to record, perhaps in a Logo log book designed for this purpose, the analysis and planning steps they take. With very young students this recordkeeping may not be feasible, in which case the responsibility for doing it will revert to the teacher.

C. Present students with a program written in two or more ways. Ask them to key it in and compare the two methods. Then, from inspection of both together, ask them to analyze the program and then put it back together, *but* in a way different from the original pattern, shape or design.

For practice of their analysis and planning skills, the students could be encouraged to compose, both for themselves and for each other, programs that are amenable to analysis and reconstitution in the manner outlined. In this way the teacher is able to facilitate students' initial analysis and planning with Logo through strategic observations and the assessment reports that arise out of these. *(See worksheet on page 28)*

6. DEBUGGING

With this process, perhaps more than any other, students experience the genuinely liberating character of learning with Logo in three ways: First, "when you learn to program you almost never get it right the first time." Second, "learning to be a master programmer is learning to become highly skilled at isolating and correcting bugs, the parts that keep the program from working." Third, "the question to ask about the program is not whether it is right or wrong, but if it is fixable" (Papert, 1980, p. 23).

The assessment of debugging ability is best carried out in the form of games. In these games the students search for the bugs in both student- and teacher-written programs. In this way, assessment activity itself places positive value on making and correcting mistakes as an inevitable and necessary part of learning to program and learning to learn with Logo.

Activities to Assess and Develop Debugging

At the basic Turtle command level, bugs are often syntactical in nature; i.e., spacing errors, words misspelled, command values inadequately specified or typing errors such as the confusion of letters with words. Programming bugs (e.g., wrong command, direction or input value) are also easy to identify and correct in the immediate command mode where constant and rapid feedback is available.

A. Provide students with programs containing a selection of typical bugs found in Turtle commands. These programs and the bugs they contain should range from simple to complex depending on the age and ability of each student, thus ensuring that the assessment activity is enjoyable and afford the learner a sense of accomplishment.

   A the students to type in the faulty commands and then examine these to see how many errors they can find. They should complete the task by:

   1. Recording all the errors they find;
   2. Rewriting the program; and
   3. Keying it in to see if it works.

B. As students become more able and confident, they might be asked, working in pairs or in small groups, to set and complete debugging tasks of their own. Once completed, their debugged programs should be checked by a friend or the teacher prior to keying in. The computer will show no mercy.

C. Students should constantly practice debugging with the programs they write in the normal course of ongoing learning with Logo.

As with all the other Logo processes, astute teacher observation and ongoing assessment is crucial at the level of basic Turtle commands. This will ensure a secure foundation for all of a student's future learning accomplishments with Logo. *(See worksheet on page 32)*

Name: ______________________

Date: ______________________

Checking Your Coding Skills

Basic Turtle Commands

1. What do each of these commands tell the Turtle to do? Write your answers in the space beside each command.

RIGHT ______________________
LEFT ______________________
FORWARD ______________________
BACK ______________________
SHOWTURTLE ______________________
HIDETURTLE ______________________
CLEARSCREEN ______________________
PENUP ______________________
PENDOWN ______________________

2. This set of commands is written in long form. Write the short form for each command in the right column.

Long Form	Short Form
CLEARSCREEN	
FORWARD 50	
BACK 25	
RIGHT 90	
FORWARD 25	
LEFT 90	
FORWARD 25	
BACK 50	
PENUP	
RIGHT 90	
FORWARD 25	
LEFT 90	
FORWARD 50	
PENUP	
HOME	

Now, type in the commands and draw a picture of the Turtle's path.

What does the number in RIGHT 90 tell the Turtle to do?

What does HOME mean?

Why do the commands RIGHT and LEFT need input numbers?

What does the number in FORWARD 100 tell the Turtle to do?

3. Change these plain English instructions into Logo commands.

Plain English Instructions	Logo Commands
Clear off the screen.	____________________
Go forward 25 Turtle steps.	____________________
Turn left 45 degrees.	____________________
Go forward 25 Turtle steps.	____________________
Move back 25 steps.	____________________
Turn right 90 degrees.	____________________
Go forward 25 Turtle steps.	____________________
Lift up the pen.	____________________
Tell the Turtle to go home.	____________________

Now, type in the commands to see what happens.

Draw a picture of the Turtle's path.

Name: ____________________________

Date: ____________________________

Exploring with Basic Turtle Commands

1. Pick a point on the screen and mark it with your finger. Now type in some commands to make the Turtle move to where you are pointing.

   Try pointing at different spots on the screen and see if you can move the Turtle to each of the spots.

   How many Turtle steps is it from the center of the screen to the bottom edge?

   What happens if the Turtle moves off the edge of the screen?

   How can you get the Turtle to come back?

2. Type in the following commands to see what happens.

```
CLEARSCREEN
FORWARD 50
RIGHT 90
FORWARD 25
RIGHT 90
FORWARD 25
RIGHT 90
FORWARD 25
HOME
```

Draw a picture of the design.

Now, what happens if you change RIGHT 90 to LEFT 90?

Draw a picture to show the new design.

How are the two designs the same?

How are the two designs different?

3. Try some variations of your own by:
   a. Changing the input numbers for RIGHT and LEFT;
   b. Using PENUP and PENDOWN to make a different pattern.

Draw your new design in the space below.

Draw pictures of any other designs you discovered when changing and adding commands.

Are there some other changes that you could do to make the design change yet again?

Name: ______________________________

Date: ______________________________

Checking Your Prediction Skills
Basic Turtle Commands

1. Draw a picture to show what you think these commands will tell the Turtle to do.

Commands	Draw what you think will happen	Draw what happened
A. CLEARSCREEN RIGHT 45 FORWARD 50 HOME LEFT 45 FORWARD 50 HOME		
B. CLEARSCREEN FORWARD 50 RIGHT 90 FORWARD 25 PENUP HOME PENDOWN FORWARD 25 RIGHT 90 FORWARD 25		

Now, type in the commands to see what happens.

Draw a picture to show "what really happened" in the space next to your first drawing above.

What differences are there between what you though would happen and what really happened? If there are differences, can you explain why?

2. Make up your own commands and draw a picture showing what you think will happen.

Commands	Draw What You Think Will Happen	Draw What Happened

Type in the new commands and draw a picture of the path the Turtle makes.

What differences are there between what you thought would happen and what really happened? If there are differences, can you explain why?

3. Can you think of a set of commands that will make a Turtle path to join all the * marks below?

*

*   *

*

Write your commands here, then type them in. Draw a picture to show what happened.

Commands   Draw a Picture of the Turtle's Path

What differences are there between what you thought would happen and what really happened? If there are any differences, can you explain why?

How did you estimate the number of turns and steps that the Turtle would need to make?

How did you estimate the number of degrees (input numbers) to use with your RIGHT and LEFT commands?

Assessing Learning with Logo

Name: ______________________

Date: ______________________

Using Your Imagination with Logo

Basic Turtle Commands

1. Imagine two different shapes that you would like the Turtle to draw. The shapes can be of any design and size that you choose.

   A. Now, go ahead and make the shapes.
   Use the design box below for planning and drawing your shapes. Then write a list of commands that will tell the Turtle how to draw each shape.

   Design Box for Shape 1 | Shape 1 Commands

   Design Box for Shape 2 | Shape 2 Commands

   B. Are there some changes that you now could make to each shape?
   If so, write your suggestions in the space below.

   Shape 1 Changes | Shape 2 Changes

C. Now, can you think of ways of putting these two shapes together to make a brand new design?

Ways to Combine Shapes 1 and 2

2. Make up a set of commands that puts both of your shapes together in one of the ways that you noted above.

Use the design box below for drawing and planning your new shapes. Write your commands in the space provided.

Design Box

Combined Shape Commands

Explain how you went about putting the two shapes together.

What are some ways in which you could change this combined shape design?

Suggested Changes for the Combined Shape Design

3. Use some of your best ideas to make a pattern, shape or design of your own choosing.

Use the design box below for planning and drawing your design. Write your commands in the space provided.

New Design Box

New Design Commands

Did you have any ideas before ycu started to draw?

Did you come up with any new ideas as you were drawing?

Did you have a picture of the design in your mind before you started to draw? If you did, then explain how you were thinking.

Did a picture of the design form in your mind as you were drawing? If so, explain how you were thinking.

List the main steps that you followed to make the design.

1.____

2.____

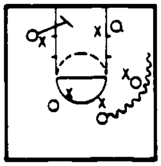
3.____

4.____

5.____

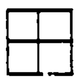
Name: ____________________

Date: ____________________



Checking Your Analysis and Planning Skills
Basic Turtle Commands

1. Think about how you would make this design.



What is the first step that you would take?

What are the next steps that you would take?

What is the last step that you would take?

How would you check to see that the design you made on the computer is the same as the one shown above?

Now, go ahead and make the design. Use the design box below for planning and drawing your picture. Then write a set of commands that will tell the Turtle how to draw your design.

Design Box | Design Commands

How many shapes are used to make up the design?

How many of these shapes can you count in the design?

Think back on how you made the design.
What step did you take first?

What steps did you take next?

What step did you take last?

How did you check whether the shape that the Turtle made was the same as the one that you drew?

Is there a way of drawing this design using fewer commands? If so, explain how.

2. Use the square parts of the design that you have already drawn to make a new pattern. You can arrange the parts any way you like but don't add any extras.

   Use the design box below for planning and drawing your new pattern.
   Write your commands in the space provided below.

   New Design Box

   New Design Commands

How many parts are there in your new design?

Think about how you made the new design.

Did you have a plan before you started to make the design?

Did you stick to the same plan from the start to the finish or did you try different ways of making the design?

Draw pictures of any other patterns you discovered while making the design.

Pictures of Other Patterns

3. Make a design of your own choosing. It can be of any size and shape that you like. Write the commands below. Use the design box for planning and drawing your work.

Design Box

Design Commands

32

Describe in your own words how you made the design.

What shape(s) did you use to make the design?

Think back on how you made the design.
What step did you take first?

What steps did you take next?

What step did you take last?

Did you have a plan before you started to make the design?

Did you stick to the same plan from the start to the finish?

What parts of the design did you try changing?

Write down any suggestions that will help other people plan new designs.

Assessing Learning with Logo

Name: ______________________

Date: ______________________

Checking Your Debugging Skills

Basic Turtle Commands

1. The student who wrote the set of commands listed below is having a lot of trouble. Help him get started by fixing the bugs. There are lots of them.

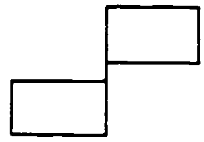
   Start from the top and type in each command line by line. If there are no bugs, then put a checkmark in the Bug Check column. If there is a bug, then write down the bug message printed by the computer in the Bug Check column. Fix each bug and write the corrected command in the Debugged Command column.

Bugged Commands	Bug Check	Debugged Commands
CLEAR SCREEN		
SHOWTOTLE		
LIFT 90		
FORWARD100		
RIGHT 90		
F 100		
RH 135		
FARWORD 140		
HIDE TOTLE		
HOME		

   How many bugs did you find altogether?

   What shape does the debugged program tell the Turtle to draw?

2. John is having a lot of troubie writing the commands to draw this shape:



   Help him find the bugs in his program by typing in each command and making any changes that are needed. Use the space next to his commands to write down any changes you make.

John's Commands	Changes You Made
FD 25 LF 90	
FD 50 LT 90	
FD 25 LT 99	
FD 50 LT 90	
FD100	
RT 90 FD 50	
RT 90 FD 50	
RH 90 FD 50	
HOME	

How many bugs did you find altogether?

Explain how you checked the design to see that it was the same as the one shown above.

3. Make up a program that contains bugs. The program can have any commands you want, but it should make a design or pattern of some sort when it is fixed.

Write both the correct commands and bugged commands below.

Correct Commands	Bugged Commands	What it is Supposed to Draw

Ask someone else to correct your bugged program. Keep a record of the changes they make.

Bugged Commands	Changes Made	What the Program Draws

Did the other person find all of your bugs?

How did they check to see whether their computer drawing was the same as the one you drew?

Write down any suggestions that you think will help other people debug their programs.

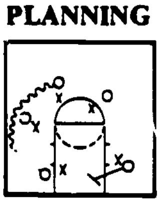
BASIC TURTLE COMMANDS
THINKING PROCESSES CHECKLIST

NAME: ______________________ **DATE:** ______________________

SCHOOL: ______________________ **CLASS:** ______________________

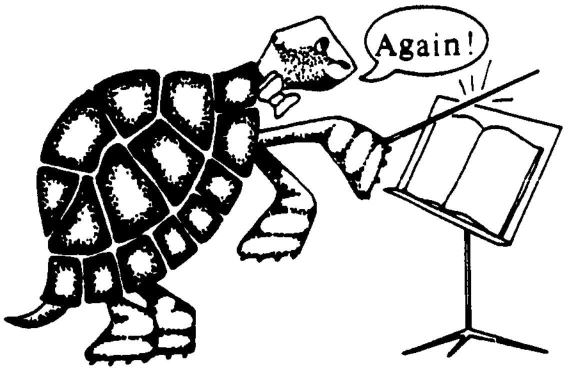
INSTRUCTIONS: Place one of the codes from the following table alongside each item in the Assessment column to indicate the development of thinking processes.

M = Learner has mastered and consistently demonstrates this process.
P = Learner has only partially mastered or inconsistently demonstrates this process.
N = Learner has not yet developed or demonstrated this process.

	THINKING PROCESSES	ASSESSMENT
CODING 	1. Remembers the full set of basic Turtle Commands, including the rules of syntax.	______
	2. Translates commands from long into short form and vice versa.	______
	3. Compiles simple programs written as lists of the basic Turtle commands and command combinations.	______
EXPLORATION 	1. Directs the Turtle around the screen using different commands and input values.	______
	2. Experiments by substituting one command for another and changing command values.	______
	3. Makes patterns using a variety of commands and command combinations.	______
PREDICTION 	1. Specifies the Turtle's behavior, bearing and movement for any single command and command combination.	______
	2. Estimates steps and turns in a given Turtle path.	______
	3. Predicts the commands and command combinations to produce specified patterns, shapes and designs.	______
CREATIVITY 	1. Constructs Turtle paths using the full range of basic commands and Logo syntax.	______
	2. Devises simple patterns and shapes by combining basic commands.	______
	3. Creates geometric shapes and designs by combining basic commands and altering command values.	______
ANALYSIS AND PLANNING 	1. Identifies all the separate basic commands required for, and composes, a simple Turtle path program.	______
	2. Identifies all the separate basic commands and command combinations for, and composes, simple pattern and shape programs.	______
	3. Identifies all the separate basic commands and command combinations for, and composes, geometric shape and design programs.	______
DEBUGGING 	1. Recognizes and corrects Logo's routine basic command and syntax error messages.	______
	2. Identifies and corrects immediate errors while making a Turtle path.	______
	3. Modifies faulty basic command programs by detecting and correcting programming errors.	______

COMMENTS

30



Chapter 3
REPEAT Commands

A Powerful Idea

> The effect of work with Turtle geometry on some components of school math is primarily relational or affective. Many children have come to Logo hating numbers as alien objects and have left loving them. In other cases, work with the Turtle provides specific intuitive models for complex mathematical concepts most children find difficult. The use of numbers to measure angles is a simple example. In the Turtle context children pick up this ability almost unconsciously. Everyone—including the few first graders and many third graders we have worked with—emerges from the experience with a much better sense of what is meant by 45 degrees or 10 degrees or 360 degrees than the majority of high school students ever acquire. *

This relational or affective development to which Papert refers occurs when learners begin to experience self-satisfaction through their own creations. Beyond simply directing the Turtle's movements around the screen a step at a time, a real sense of "personal power" is felt at the point where movements can be combined and repeated to form intricate and often spectacular visual effects.

The REPEAT command, one of Logo's most powerful ideas, provides a tool for understanding mathematical concepts that is simply not available in manual modes. This is because controlled repetition and animation offer a way of concretely demonstrating and simulating ideas that may otherwise be too abstract for learners to understand. As Papert notes, even young primary school children are capable of intuitively grasping some important ideas that lie at the heart of such formal topics as calculus, geometry and trigonometry.

Assessment at the level of REPEAT commands is aimed at encouraging learners to think systematically as they combine and repeat basic Turtle movements they have mastered at an earlier stage. Progressing in a building block fashion, the focus of assessment at this level is upon both:

1. Helping students discover the power of controlled repetition of commands with which they are already familiar; and
2. Developing further their specific thinking processes as they begin to engage in higher-order programming operations.

Intervention by the teacher at the level of repeats involves two separate but related aspects:

1. Encouraging learners to apply the full range of thinking strategies needed to make the most effective use of REPEAT commands; and
2. Helping learners reflect on their own thinking processes at each stage as they begin to explore, analyze, plan and create effects using REPEAT commands.

Here, the Logo experience becomes more conceptual and formal in the sense that learners must integrate several important ideas from mathematics and geometry. Considerable time might need to be spent working on the concept of angle and the relation of this concept to the formation of geometric designs and patterns. Working at this level requires careful observation of each learner's progress as they begin to consolidate and extend the basic programming and thinking abilities they have already mastered.

At this level there will be more variation in learners' thinking abilities than was apparent at the level of basic Turtle commands. This is because the REPEAT command requires a thorough grasp of earlier concepts. It might be observed, for example, that a learner can correctly use the syntax for REPEAT yet have little understanding of how repetition controls the sequence in which basic commands are performed by the Turtle. In this case, it might be necessary to spend time consolidating some of the previous concepts (e.g., distance, turns, state-changes) before going on to more complicated types of REPEAT operations. The essential point to remember is that a thorough assessment of each person will provide valuable information on the development of specific thinking processes. This will make it possible to tailor learning programs to meet a particular learner's needs.

Assessment Objectives

The following assessment objectives describe the main feature of each thinking process at this level:

1. REPEAT commands are a way of telling the Turtle how to combine Turtle commands and follow instructions in a particular sequence (coding);
2. REPEAT commands can be used to make the Turtle move and behave in a surprising number of ways (exploration);
3. REPEAT commands may result in the production of expected and unusual outcomes (prediction);
4. REPEAT commands can be used either singly or in combination with other REPEAT commands to make elaborate patterns and designs (creativity);
5. REPEAT commands can be used to put parts of programs together in various ways (analysis and planning): and

*From *Mindstorms: Children, Computers and Powerful Ideas*, by Seymour Papert Copyright © 1980 by Basic Books, Inc., publishers. Reprinted by permission of the publisher.

6. REPEAT commands may contain bugs which need to be located and fixed so that the program works as its author intended (debugging).

The above assessment objectives and the thinking processes to which they refer provide a language that teachers and learners can use to talk about how they are thinking as they work with Logo. This language of thinking helps students to articulate their ideas and problems better and opens doors to communication with others. The somewhat abstract and formal concepts of thinking and problem solving thus become more understandable and hence more meaningful to Logo learners and teachers alike.

How to Assess the Six Thinking Processes

1. CODING

At this level, coding involves the translation of basic Turtle movements into groups of movements which are to be repeated in a certain way. The requirement here is that through playing Turtle, learners develop insights into the ways of directing the Turtle on the screen to form desired patterns and designs. These insights are expressed in Turtle Talk as programs or equations that teach the Turtle a set of movements which are to be repeated to produce a specific effect. Here, a student who has already learned the basic Turtle commands needed to draw a square, triangle or rectangle is now ready to learn a formal method for communicating sequences of movements to be repeated. Accordingly, the main assessment aim here is to observe the student's ability to:

(1) Apply the rules of syntax for writing REPEAT commands in a form the Turtle can understand;

(2) Identify code equivalences mainly by translating common factors into a set which is to be repeated; and

(3) Translate ordinary language expressions into REPEAT commands.

This thinking process can be developed by using various decoding and recoding activities as described in the next section.

Activities to Assess and Develop Coding

Extending from the level of basic Turtle commands, match-mismatch and completion activities can be used to assess and develop skills in coding. These activities can be done either on or off computer with individuals or small groups. Here are some examples of specific activities that might be used:

A. Prepare overheads or handouts which contain REPEAT statements that are to be completed. For example:

"Complete the following REPEAT statements so that they make circles."

```
REPEAT 120 [ FD 1 RT   ]
  120 ×    = 360
REPEAT 60 [ FD 1        ]
  60 ×   = 360
```

"How many times does this statement need to be repeated to draw a square (triangle, rectangle)."

```
REPEAT      [ FD 50 RT 90 ]
REPEAT      [ FD 50 RT 120 ]
REPEAT      [ FD 50 RT 90 FD 60 RT 90 ]
```

Such coding activities can serve as concrete starting points for getting students to collaborate with one another. These types of activities can be done with paper and pencil and then checked on the computer later.

B. Have students write sets of basic Turtle commands to draw shapes. Then ask them to identify the common sets of commands and the REPEAT statement needed to make the shape.

Have students prepare a sheet which shows:

BASIC TURTLE COMMANDS	SETS OF COMMANDS	REPEAT COMMANDS

These coding activities can be done away from the computer and then checked later using Logo.

C. Ask students to translate plain English descriptions of movements into REPEAT commands.

For example, use an overhead to present the ordinary language descriptions and ask students to code the language using a Logo REPEAT command.

1. "Turn right 30 degrees."
2. "Go forward 50 steps and turn right 120 degrees."
3. "Do 2 again two more times."
4. "Turn left 30 degrees "
5. "Go forward 50 steps and turn right 90 degrees."
6. "Do 5 again 3 more times."

While structured activities should be used at the beginning, students should be encouraged to create their own drawings using basic Turtle commands and then work out the code equivalence for directing the Turtle's movements by using a REPEAT command. Regardless of the approach used, be sure to provide learners with an opportunity to do a self-assessment of their coding and problem-solving skills. *(See worksheet on page 42)*

2. EXPLORATION

The main aim here is to encourage and guide students through a variety of experiments that use the repeat concept. Some learners will naturally explore the full range of possibilities that this new-found tool makes available. Other learners will need to be introduced in a more systematic way to the kinds of exploration which can be undertaken. In any case, the essential point is to extend the exploration process so that all students can fully participate in Logo's discovery learning method. Exploration is assessed at this level by observing the extent to which learners:

(1) Experiment by changing the number of times that sets of commands are repeated;

(2) Make variations to basic patterns by altering commands and the number of times a statement is repeated; and

(3) Make modifications to geometric designs by changing the command combinations and number of repetitions.

Activities to Assess and Develop Exploration

A. Ask students to experiment by putting different numbers into given repeat statements. For example:

"Polygons are closed shapes that have equal sides and equal angles. Explore by studying what happens when you put different numbers into this statement.

REPEAT [FD 20 RT]

The rule is that the number of degrees the Turtle turns, times the number of turns, must equal 360."

At this point it is a good idea to use paper and pencil tasks that involve working out internal and external angles with protractors. For example, have students measure angles in familiar shapes and then put these numbers into REPEAT commands.

B. Following on with this basic geometry lesson, have students explore by altering commands and the number of times that a statement is repeated. For example:

"Explore this repeat command by changing the number of repeats and the number of steps and degrees. Keep a record of each REPEAT statement that you try and the drawing that the Turtle produces.

Students should be encouraged to keep their own journals and to exchange findings with one another. Students could work in groups to undertake different exploratory projects and then report back to the others.

C. Ask students, working together in teams, to explore some different ways of producing geometric patterns with repeat statements. You might give them a basic design at the start and then ask them to try changing it by altering the number of repeats and command combinations. Have students keep a record of their findings so that they can share this with others following completion of the exploration activities.

Remember that the purpose of exploration is to get students to reflect on what they are doing and how they did it. When they know how to go about exploring, then they can recognize the full range of possibilities that are open to them. Regardless of what approaches are used, the main aim is to encourage them to make up their own minds as to how to accomplish what they want to do.

You can encourage students to explore and develop their ideas by:

- Asking them to say what they intend to do:
  —"What would you like to do with this one?"
  —"Good. You decided to try using that design in a different way."
- Getting them to recognize their capabilities:
  —"What are some other changes you could make?"
  —"How does this compare with the ones you did before?"
- Giving examples of ways to think about what to do:
  —"Sometimes it is a good idea to just change one thing at a time, then see what happens."
  —"Think about the kinds of changes that you could make, then try exploring each one."

Exploratory activities are a good place to help children learn to articulate their ideas and problems. Get them to talk a lot about what they have been doing and what they found out as a result of their exploration. *(See worksheet on page 44)*

3. PREDICTION

Logo is rich in opportunities for learners to test their own ideas by making hunches, guesses, estimations, etc. The interactive medium of the computer provides a tool for developing the process of prediction in a way that is not readily available with other methods. Assessment of prediction at this level is concerned with observing the extent to which learners engage in self-directed testing of ideas by:

(1) Specifying what shapes and designs are produced by certain REPEAT command combinations;

(2) Estimating the number of sides, degrees of turn and repeated command combinations required to draw specific shapes; and

(3) Predicting the elements needed to produce interesting and complicated geometric designs.

Activities to Assess and Develop Prediction

A. At the beginning, present students with specific types of prediction tasks to ensure that they study the full range of possibilities.

Prepare an overhead or handout which asks them to predict the visual output from a particular REPEAT statement. For example:

"Draw a picture to show what you think this REPEAT statement will tell the Turtle to do."

REPEAT 6 [FD 50 RT 60]

"Now type the command into the computer to see what happens. Compare your 'guess' drawing with the drawing the Turtle made."

"Were they the same or different? If they were different, can you explain why?"

It is a good idea to make these prediction activities as interactive as possible. Prediction lends itself to exchanging ideas and students should be encouraged to talk about how they were thinking.

B. Ask students to fill in the blanks on a chart to estimate the number of sides, degrees of turn and REPEAT commands needed to produce particular shapes. For example, prepare a chart:

Picture of Shape Wanted	Number of Sides	Number of Degrees in Each Turn	REPEAT Command

Prediction activities lend themselves to discussion of students' own ideas and an analysis of how they are thinking. When surprising results occur, these can be used to study ways of thinking and Logo problem-solving approaches. These activities can be done by a whole class, in teams, or by individual students. Whatever method is used, *it is vitally important to get students to freely share their ideas.*

C. Have students, working in pairs or small groups, do the following:
   1. Make a geometric pattern that uses a REPEAT command.
   2. Ask other students or groups of students to predict the outcome of the REPEAT command used in 1.
   3. Discuss any false predictions, diagnose problems and propose alternate solutions.

By setting tasks such as those described above, the teacher is free to serve as a learning "consultant" or "facilitator." The main aim of that role is to get students to talk about their ideas and articulate their problems. Particular attention needs to be given to encouraging students who are reluctant communicators to join in the information exchange. The teacher can help ensure that this takes place by setting the stage and encouraging individual learners to communicate by:

- Inviting children to reflect on their own thinking:
  —"Ah, you're taking a few moments to work that one out."
  —"Nice one. You are studying the command combinations first."
- Asking them to explain how they are thinking:
  —"What step did you do first?"
  —"What do you want to find out?"
- Encouraging them to exchange their ideas with others:
  —"Can you tell us how you worked that one out?"
  —"Do you have any suggestions that would help us solve this one?"

The above statements reflect what the teacher sees in the children's actions and on the screen. The main aim is to get students to "think about their own thinking" and to use the language of thinking as a way to communicate their problems and achievements. *(See worksheet on page 47)*

4. CREATIVITY

The creative process emerges from a range of thinking experiences students have encountered in such areas as coding, exploring and predicting. The ability to engage in the creative process is then made possible by the "tools of invention" which Logo provides. It is important to recognize, however, that creativity does not automatically result from working with Logo. Intervention is often needed to allow the creative process to unfold as it should. Here, assessment of creativity is done by observing the way students:

(1) Use the concept of controlled repetition to construct basic shapes of their own;

(2) Devise intricate patterns and designs by employing repeated combinations; and

(3) Invent their own geometric shapes and designs by using multiple REPEAT commands.

The underlying aim of this assessment is to encourage students to appraise the things they are making and to examine the creative processes they use.

While it is true that many children can be said to be "naturally curious," it does not automatically follow that this curiosity will lead to creative forms of expression. Rather, thinking creatively is a process which needs to be intentionally developed. This requires a teacher to serve as the students' guide and companion as they begin to engage in the process of creating products of their own. Here, the teacher encourages students to become more fluent, flexible and original thinkers as they make progress in their own way with Logo. The teacher, as facilitator, thus provides a setting where students are free to develop and explore their new ideas.

The assessment of creativity has three main parts:

1. Observing the manner in which students apply their newfound tools of invention;
2. Guiding them to use their "best ideas" as the basic material for inventing new products; and
3. Helping them to articulate ideas and problems encountered as they engage in the creative process.

Activities to Assess and Develop Creativity

A. Ask students to invent their own surprises. For example:

   "Use a REPEAT command and your best ideas to invent your own surprises. Draw a picture of your most interesting creations here. Write the REPEAT command you used below the drawing."

MY INVENTION

MY COMMANDS

Use these inventions as a basis for discussion. Do this by identifying the basic ideas and how these were developed using the tools Logo provides.

B. Suggest to students that they use some of their best ideas from above as a building block for a more elaborate design—perhaps one that uses two REPEAT commands. Here, the decision as to which and how many commands to be used should be left open to the student. The teacher may need to facilitate this process by providing illustrations of the ways in which a basic idea can be extended. This should be more of an inspirational than instructional session. Students should be encouraged to share their ideas and communicate problems they encounter.

C. Have an open class discussion about thinking creatively. This could include several types of warm-up activities using a "what if" game. For example:

"What if the Turtle could only draw triangles? How could you make it draw circles and squares?"

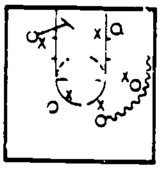
"What if the Turtle could only draw squares? What sort of interesting design could you make with square parts?"

"What if you wanted to make the Turtle look like it was moving across the screen? How could you do that using a REPEAT command?"

Encourage students to identify Logo tools which can help them think creatively. Some of these include background and pen color, hideturtle and showturtle, combining repeat statements, etc. They could then be given the task of using some of the tools to invent new products.

It is important to remember that students need the right kind of atmosphere in which to think creatively. The role of the teacher is to set the stage for students to become more fluent, flexible and original thinkers by:

1. Providing lots of time and a permissive atmosphere in which to generate new ideas;
2. Encouraging learners to try different ways of thinking and to experiment with alternative ideas; and
3. Supporting learners even when they come up with apparently silly and unusual ideas. Even the most inappropriate ideas should be greeted with some kind of enthusiasm. *(See worksheet on page 50)*



5. ANALYSIS AND PLANNING

This refers to the process which a learner engages in as s/he breaks a given task down into a logical and orderly (step-by-step) sequence to reach a defined goal. Analysis and planning can be thought of as a formal or informal process. Formally, a learner might prepare written, orderly outlines of how to approach a certain problem. Informally, analysis and planning may be unstated methods of approaches to problem solving that exist only in the mind of the learner. Informal analysis and planning is often a spontaneous process in which the learner engages as s/he works directly on each part of a problem. Logo provides opportunities to observe and practice doing both formal and informal kinds of analysis and planning

The development of analytical thinking does not usually happen automatically, however. Often learners need direct experience to help with the development of this process. Here, the role of the teacher is to find out what plan of attack a learner uses and to intervene when necessary to help him/her become systematic in working on problems. The aim is to help learners become aware of their planning skills and to teach them strategies which they can use to improve this thinking process.

Assessment at this level is concerned with observing the strategies students use to:

(1) Devise a plan for preparing a given shape or design;
(2) Identify the parts and steps needed to construct complex geometric patterns; and
(3) Describe the main steps involved in designing a project of their own.

Activities to Assess and Develop Analysis and Planning

A. Have students work in pairs on a problem (design or pattern) which requires the use of a REPEAT command. Set the stage for analysis and planning by introducing them to the four steps of problem solving.

Step One: Understanding the Problem. Have students work in pairs to talk about and walk through the problem. This verbal rehearsal is a good way for learners to become aware of their planning skills.

Step Two: Make a Plan. First have students write single commands to solve the problem. Then have them write multiple commands to solve the problem. Finally, have them write a REPEAT command to solve the problem.

Step Three: Carry Out the Plan. Have students type their REPEAT commands into the computer and then draw a picture of what the Turtle did.

Step Four: Look Back. Ask students if the Turtle did what they wanted it to. If not, have them go back and change the program.

B. Ask students to describe in their own words what steps they followed to solve the problem in A above. Use the blackboard or a handout to keep a record of the problem solving steps. For example:

"Make a plan
What step did you do first?
What steps did you do next?
What step did you do last?"

"Did you stick to the plan from the start to the finish, or did you make changes as you were working? What changes did you make?"

Explain that an important part of analyzing and planning is to change plans while working on a problem. That is, we make a plan, analyze it as we are working on the problem, and make changes that we think are needed to help improve the work.

C. Give each student a problem and ask them to make a plan for solving it. Have them keep a written record of the problem solving steps they followed (as above).

Next, have students work in pairs to explain their problem solving steps to one another.

Finally, ask each pair of students to describe in their own words what the main steps are in planning a design. Have them keep a written record:

"Write down in your own words what the main steps are in planning a design which uses a REPEAT command."

Step 1
Step 2
Step 3
Step 4

These teaching and sharing activities are valuable ways of encouraging students to reflect on their analytical thinking. First, students are required to put their ideas into an order that makes logical sense for a given task. Second, students have to talk through the planning steps with one another. This verbal rehearsal is important because it helps them articulate their problems and revise plans to suit the situation at hand.

The teacher can facilitate the analysis and planning process by encouraging students to think about what they are doing. For example:

- "I can see why you needed to change your plan from what you intended to do at the beginning."
- "Stop for a minute and think about what the next step in your plan might be."
- "If you were helping someone to make a plan, what is the first step that you would get them to do?"

Planning styles will vary among students and it is important to encourage each person to use whatever style works best for them. For example, some students will keep their own mental checklists while others talk a lot about what is happening as their plan unfolds. Some will write formal plans and stick to the order of steps they have set for themselves. Others will change their plans extensively from the outline they started with at the beginning. Regardless of what planning style is used, the main aim is to encourage the development of "systematic" analysis and planning. *(See worksheet on page 53)*

6. DEBUGGING

At this level, students become involved in detecting and fixing both surface errors in syntax and logic errors in REPEAT commands. This requires that students develop an understanding of the ways in which Logo performs a set of instructions. Such understanding is conceptual in nature because it depends upon an evaluation of several interrelated factors (command combinations, order of operations, number of repeats). This self-evaluation process requires students to compare the logic of their own thinking with the logic used by the program. As Papert points out, "by deliberately learning to imitate mechanical thinking, the learner becomes able to articulate what mechanical thinking is and what it is not" (*Mindstorms*, p. 27).

Assessment at this level involves observation of the strategies which a student employs as s/he debugs REPEAT commands by:

(1) Locating and fixing simple bugs (surface errors) in given drawings;

(2) Identifying and correcting program logic errors; and

(3) Describing the main steps involved in debugging programs.

Activities to Assess and Develop Debugging

A. Give students a bugged program and ask them to see if they can find the bugs and fix them. Have them keep a record of the bugs they found and fixed.

Prepare a handout containing the following information:

"Look at the following REPEAT commands. Each of them has bugs so they won't draw what we want them to. Your job is to find and fix each of the bugs. Keep a record of the bugs you found and how you fixed them."

REPEAT COMMAND

DRAWING OF WHAT THE TURTLE WAS SUPPOSED TO MAKE

DRAWING OF WHAT THE TURTLE MADE

BUGS FOUND   BUGS FIXED

B. Have students work in pairs to prepare a debugging checklist. This should include helpful hints that others could use as a guide for debugging programs. For example:

1. Check each command line by line.
2. Draw a picture or walk through each of the Turtle moves.
3. Underline each suspected bug.

As a class activity, prepare a master checklist using the helpful hints that each pair of students identified. This will generate a lot of discussion on types of bugs and effective ways of locating and fixing bugs. It goes without saying that bugs are an interesting and normal part of learning with Logo.

C. Ask students to:

Create a problem.
Make a plan.
Carry out the plan.
Look back to evaluate the solution.
Make changes if necessary.
Keep a record of the bugs that were found and fixed.

Have a class discussion about the different strategies for handling the unexpected results that bugs often produce. Explain the three main strategies:

1. Look back to review plans and find out where the unexpected took place.
2. Explore the bug to see what it is doing and how it works.
3. Decide what needs to be changed.

Ask students to explain how they figured out where the bugs were and how they fixed them. Explain that bugs can encourage thinking and sometimes even stimulate new ideas. Ask if anyone got "bugged" with a new idea. That is, did a bug cause them to think of a new idea and change their plan?

Use an example of your own (or one of the students) to show how a bug can lead to the invention of a new idea. Draw a picture that is produced by a command with a bug in it. Ask students to think about the following:

"How is this different from what we intended?
"What do you find interesting about this bugged design?"
"Is there a new idea that we could make use of here?"

The essential point in debugging activities is to treat bugs as a natural and logical part of thinking about thinking. In fact, debugging is the main ingredient of Logo's discovery learning approach. It is the means through which students reflect most deeply on the development of their own ideas.

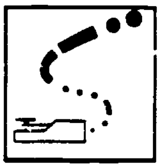
Here is a checklist which teachers can use to guide the debugging process:

1. Don't unnecessarily interrupt students when they are deeply involved in a debugging task.
2. Let students discover answers for themselves.
3. Ask questions which lead to solutions without giving away the answer.
4. Ask questions that make the student think and try alternatives.
5. Keep hands off the student's keyboard.
6. Let students talk about their problems and ideas for solutions.
7. Ask questions which clarify the problem.
8. Give students time to answer questions.
9. Ask follow-up questions.
10. Encourage students to summarize the problem in their own words as part of their answer. Stating the problem for oneself often leads to a solution. *(See worksheet on page 56)*

43

Name: ______________________________

Date: ______________________________



Checking Your Coding Skills

REPEAT Command

1. Complete the following REPEAT statements so they will make squares.

   A. REPEAT 4 [FD 50 RT   ]
   B. REPEAT   [FD 50 RT   ]
   C. REPEAT 4 [FD   RT   ]
   D. REPEAT   [FD   RT   ]

   How many times is each command repeated?
   Which two commands in the above statement are repeated?

2. Write down the sets of commands which are repeated below. Then write a REPEAT command for each set of commands. The first one is done for you.

	SINGLE COMMANDS	SETS OF COMMANDS	REPEAT COMMANDS
A.	CS		
	FD 40		
	RT 120	FD 40 RT 120	
	FD 40		
	RT 120	FD 40 RT 120	
	FD 40		
	RT 120	FD 40 RT 120	REPEAT 3 [FD 40 RT 120]
B.	CS		
	FD 40		
	LT 90		
	FD 40		
	LT 90		
	FD 40		
	LT 90		
	FD 40		
	LT 90		
C.	CS		
	FD 40		
	RT 60		
	FD 40		
	RT 60		
	FD 40		
	RT 60		
	FD 40		
	RT 60		
	FD 40		
	RT 60		
	FD 40		
	RT 60		

44

Now, type in each of those REPEAT commands to see what happens.

Draw a picture to show what each REPEAT command tells the Turtle to do.

A. B. C.

How are the three drawings the same?

How are the three drawings different?

3. Change these plain English instructions into REPEAT commands.

Plain English Instructions	REPEAT Commands
A. Go forward 25 Turtle steps. Turn right 120 degrees. Go forward 25 Turtle steps. Turn right 120 degrees. Go forward 25 Turtle steps. Turn right 120 degrees.	
B. Tell the Turtle to do the following sets of commands 4 times in a row: Go forward 50 Turtle steps. Turn right 90 degrees.	

Now type in the REPEAT commands to see what happens.

Draw pictures of the Turtle's paths here.

A. B.

Describe in your own words what REPEAT statements are used for in Logo.

What is the main advantage of using REPEAT statements?

Name: ____________________

Date: ____________________

Exploring with the REPEAT Command

1. Think cf a square.
   Now, fill in the missing number to make a square.

   REPEAT   [FD 50 RT 90]

   What number do you think?

   Try it! Draw the picture of what the Turtle made.

   What happens if the number is bigger than 4?

   What happens if the number is 1?

   Try putting some different numbers in the REPEAT statement above and see what happens. Write down anything you discover or find interesting here.

   This time think of a triangle. Fill in the missing number to make a triangle.

   REPEAT   [FD 100 RT 120]

   What number did you use?

   Try it out! Draw the shape the Turtle made.

Play around with different numbers for REPEAT.

What is the largest number you can use with REPEAT?

What is the smallest number you can use with REPEAT?

2. Type in the following command to see what happens.

   REPEAT 5 [FD 20 RT 90 FD 20 LT 90]

   Draw a picture of the design here.

Now what happens if you change RT 90 to LT 90?

Draw a picture to show how the design has changed.

How are the two designs the same?

How are the two designs different?

What happens if you change the number of REPEATs from 5 to 10?

Draw a picture to show how the design has changed.

3. Type in the following REPEAT statement.

REPEAT 4 [FD 80 RT 90 FD 10 RT 90 FD 10 RT 90 FD 10 RT 90 BK 70 RT 90]

Draw a picture of the design here.

Try some variations o. your own by:

a. Changing the size of the pattern.
b. Changing each RT 90 to LT 90.

Draw your new design in the space below.

Draw a picture of any other designs you discovered when changing and adding commands.

Are there some other changes you could do to make the design change yet again?

Name: ______________________

Date: ______________________

Checking Your Prediction Skills

REPEAT Command

1. Draw a picture to show what you think each of these REPEAT statements will tell the Turtle to do.

   A. REPEAT 5 [FD 30 RT 60]

   Draw what you predict will happen. Draw what happened.

   B. REPEAT 6 [LT 90 FD 10 RT 90 FD 10 BK 10 HT]

   Draw what you predict will happen. Draw what happened.

   C. REPEAT 7 [RT 90 FD 10 LT 90 FD 15 BK 15 HT]

   Draw what you predict will happen. Draw what happened.

49

Now, type in the commands to see what happens.

Draw a picture to show what happened in the space next to your "predicted" drawing above.

What differences are there between what you guessed would happen and what really happened?

If there are any differences, can you explain why?

2. Complete the following chart by writing in the number of sides, degrees of turn and a REPEAT statement that will make each shape.

Shape	No. of Sides	No. of Degrees For Each Turn	REPEAT Statement
SQUARE			
TRIANGLE			
RECTANGLE			
PENTAGON			
HEXAGON			
OCTAGON			

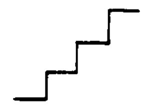
Type in each of the REPEAT statements above to see if they draw the shape you wanted.

Does the Turtle draw each of the shapes you wanted?
If not, draw a picture of what the Turtle drew and change the REPEAT statements.

How did you figure out the number of sides and degrees of turn for each shape?

How did you figure out the number of times that each set of commands would need to be repeated for each shape?

3. Can you think of a set of commands that will tell the computer how to make this pattern?



Write your REPEAT statement below, then type it in.
Draw a picture to show what happened.

REPEAT Statement: | Draw a picture to show what happened:

What differences are there between what you thought would happen and what really happened? If there are any differences, can you explain why?

If the computer's drawing is not the same as the one shown above, try changing the REPEAT statement.

How did you figure out the number of turns and steps that would need to be repeated?

How did you figure out the number of times that each set of commands would need to be repeated?

Name: ______________________

Date: ______________________

Using Your Imagination with the REPEAT Command

1. Invent your own surprising designs with REPEAT. Draw two of your best ones here. Write the commands you used below your drawing.

Design 1

REPEAT [

Design 2

REPEAT [

Can you think of some ways to change each of the above designs? If so, write your suggestions in the space below.

Design 1 Changes

Design 2 Changes

Now invent some new surprises by changing each design.

Draw a picture of your best new inventions here.

Changed Design 1

Changed Design 2

Which design do you like best? Explain in your own words what it is that you like about this design.

2. Invent some new designs which use two or more REPEAT statements. Draw your best ones here. Write the commands you used below your drawing.

REPEAT [

REPEAT [

53

Did you have any ideas before you started to draw this design? If so, what ideas did you have?

Did you come up with any new ideas as you were drawing? If so, what ideas did you come up with?

Briefly describe in your own words what this design looks like.

3. Use some of your "best ideas" to invent a new design which uses a double REPEAT statement, i.e., a REPEAT within a REPEAT statement. Draw your favorite one here. Write the commands you used below your drawing.

New Design Drawing

REPEAT [

Did you have any ideas before you started to draw? If so, what ideas did you have?

Did you come up with any new ideas as you were drawing? If so, what ideas did you come up with?

Did you have a "picture" of the design in your mind before you started to draw? If you did, then explain how you were thinking.

Did a "picture" of the design form in your mind as you were drawing? If so, explain how you were thinking.

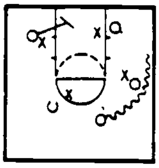
Did you figure out your design on paper first and then try it on the computer? Or did you figure out your design on the computer and then draw it on paper?

Explain what method you like to use to figure out designs.

Assessing Learning
With Logo

Name: ______________________ ________

Date: ______________________________



Checking Your Analysis and Planning Skills

REPEAT Command

1. Think about how you would make this design using a REPEAT statement.

A. **Make a plan.** Write the commands here that tell the computer what to do.

B. **Carry out the plan.** Type the commands into the computer.
Draw a picture of what the Turtle did.

C. **Look back.** Compare the design above with what the Turtle drew. If they are not the same, go back and find the bugs and make the changes.
What bugs did you find? Explain how you fixed these.

Think back on how you made this design.
What step did you do first?

What steps did you do next?

What step did you do last?

2. Think about how you would make this design using a REPEAT statement.

A. **Make a plan.** Write the steps to make the design above.

B. **Carry out the plan.** Type the commands into the computer.

Draw a picture of what the Turtle did.

C. **Look back.** Compare the design above with what the Turtle drew. If they are not the same, go back and find the bugs and make changes.

What bugs did you find? Explain how you fixed these.

Think back on how you made the design.
What step did you do first?

What steps did you do next?

What step did you do last?

3. Use some of the best ideas from the drawings you have already done to make a new design. Make any design you like but be sure to use a REPEAT statement.

Use the Design Box below for planning and drawing your new pattern. Write your commands in the Design Commands space below.

Design Box

Design Commands

Describe in your own words how you made the design.

How many parts are there in your new design?

Think about how you made the new design.
What step did you do first?

What steps did you do next?

What step did you do last?

Did you have a plan before you started to make the design?

Did you stick to the same plan from the start to the finish?

Write down in your own words what the main steps are in planning a design.

Name: ______________________

Date: ______________________

Checking Your Debugging Skills

REPEAT Command

1. See if you can find the bugs in the program below and make changes so that the Turtle knows how to draw the pattern below.

REPEAT 6 [RT 90 FD 10 LT 90 FD 10 FD 10 HT]

Carry out the plan. Type the commands into the computer.

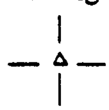
Draw a picture of what the Turtle did.

Look back. Find the bugs and make the changes so that the Turtle's drawing is the same as the above pattern.

How many bugs did you find?

How did you figure out what the bugs were?

2. This student is having a lot of trouble doing the problem below. Help him get started by fixing the bugs in his REPEAT statement.



REPEAT 3 [PD FD 10 PU FD 20 PD BK 40 PU RT 90 HT]

Carry out the plan. Type the commands into the computer.

Draw a picture of what the Turtle did.

58

Look back. Find the bugs and make the changes so that the Turtle's drawing is the same as the above pattern.

How many bugs did you find?

How did you figu. out what the bugs were?

3. Make up a REPEAT statement that contains bugs. The statement can have any commands you want, but it should make a design or pattern of some sort when it is fixed.

Write the cor. ct REPEAT statement here.

REPEAT [

Draw what it is supposed to make here.

Write the bugged REPEAT statement here.

REPEAT [

Ask another student to correct your bugged REPEAT staterr nt. Keep a record of the changes they make here.

How many bugs were there altogether?

Did the other student find all of your bugs?

Ask them how they figured out where the bugs were. Write down their answer here.

Write down the main steps that you use for finding and fixing bugs (debugging).

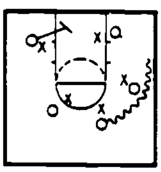
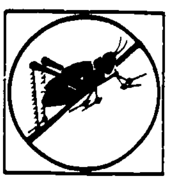
REPEAT COMMANDS
THINKING PROCESSES CHECKLIST

NAME: ____________________ DATE: ____________________

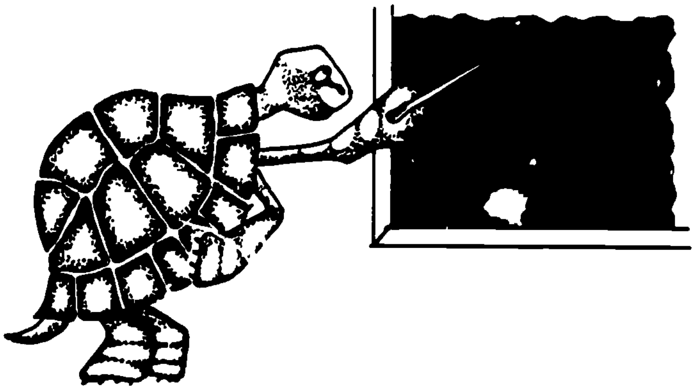
SCHOOL: ____________________ CLASS: ____________________

INSTRUCTIONS: Place the code letter which best describes the development of a particular thinking process alongside each item in the ASSESSMENT column.

M = Learner has mastered and consistently demonstrates this process.
P = Learner has only partially mastered or inconsistently demonstrates this process.
N = Learner has not yet developed or demonstrated this process.

	THINKING PROCESSES	ASSESSMENT
CODING 	1. Translates simple repetitious sets of basic Turtle commands into REPEAT commands.	________
	2. Identifies sets of basic Turtle commands which are repeated and simplifies these using REPEAT commands.	________
	3. Identifies plain English commands which are repeated and translates these using REPEAT commands.	________
EXPLORATION 	1. Experiments by changing the number of times a statement is repeated.	________
	2. Makes variations to basic patterns by altering commands and the number of times a statement is repeated.	________
	3. Makes modifications to geometric designs by changing the number of repeats and command combinations.	________
PREDICTION 	1. Specifies what shapes and designs will be produced by repeated command combinations.	________
	2. Estimates number of sides, degrees of turn and repeated command combinations required to draw particular shapes.	________
	3. Predicts the number of repeats and command combinations needed to produce geometric designs.	________
CREATIVITY 	1. Constructs own patterns using simple REPEAT statements.	________
	2. Devises own patterns or designs using combinations of two or more REPEAT statements.	________
	3. Invents own complex geometric shapes and designs by using combinations of REPEAT within REPEAT statements.	________
ANALYSIS AND PLANNING 	1. Devises a plan for the construction of a given design using a REPEAT statement and carries it through to completion.	________
	2. Identifies the parts and steps needed to construct a given pattern using a REPEAT statement.	________
	3. Devises a plan for own design using a REPEAT statement and describes the main steps in planning.	________
DEBUGGING 	1. Locates and fixes simple bugs (syntax errors) in a given drawing produced by a REPEAT statement.	________
	2. Identifies and corrects logic (programming) errors in a given drawing produced by a REPEAT statement.	________
	3. Produces a bugged program for analysis and describes the main steps for debugging programs.	________

COMMENTS



Chapter Four
Defining Procedures

A Powerful Idea

> Everyone works with procedures in everyday life. Playing a game or giving directions to a lost motorist are exercises in procedural thinking. But in everyday life procedures are lived and used, they are not necessarily reflected on. In the Logo environment, a procedure becomes a thing that is named, manipulated, and recognized as the children come to acquire the idea of procedure. . . . I have clearly been arguing that procedural thinking is a powerful intellectual tool and even suggested analogizing oneself to a computer as a strategy for doing it. *

This procedural thinking to which Papert refers is the lynch pin of the Logo computer language. Here, students are required to express their ideas in a structured and logical form as a procedure. These procedures are really models which students have constructed to explain the mechanics of how their ideas are put together. This use of the procedural model is not simply a matter of thinking in mechanical or linear terms. Rather, procedural thinking is the knowledge about a model that has been gained through creating, using and changing the procedure.

This procedural knowledge is used intuitively by children in their everyday lives through the games they play and the way they physically direct themselves through the environment. For example, even very young children seem to be able to work out planning strategies for a matching card game or know various routes to places in their community. But as Papert points out, often the same child does not apply this "procedural thinking" to formal learning of arithmetic and language. The idea of procedures as things that can be debugged is a powerful, difficult concept for many students until they have accumulated experience in working with them (Papert, 1980, p. 154).

The assessment aim at this level is to provide students with a range of experience in procedural thinking. Here, there are two specific aims:

1. Helping students discover how procedural knowledge can be used for building models as representations of their ideas; and
2. Helping students reflect on the specific thinking processes which come into play when they use the computer to name, manipulate and change their ideas.

Similarly, the role of the teacher in this assessment is to:

1. Encourage learners to apply the full range of their thinking processes in order to develop this procedural knowledge; and
2. Help students acquire an understanding of when and how to use the idea of a procedure as a model for good (efficient) thinking.

It is at this level that students are provided with opportunities to play with the idea of procedure building. At first they might use procedures to build simple shapes or to create animated patterns and designs. Later, they will use the procedure as a basic building block for creating other, more complicated, effects which bring several ideas together. Through the process of building, manipulating and changing procedures, students come to understand how they can use procedural thinking as an effective problem solving tool. The main educational feature of this tool is that it encourages students to express their ideas in a very structured and systematic way.

The role of the educator is to create the conditions for students to use procedural thinking in an enjoyable and purposeful way. Here you can help students by getting them to focus on the thinking processes they engage in with this intellectual tool. Think of a simple procedure for drawing a triangle; this is a tool which makes the construction of other objects easier. It is a tool that students use to explore and develop new ideas. By using procedural thinking as a tool, students can extend Logo to personalize their work and fit their individual needs. Here, the focus is upon encouraging students to use the idea of a procedure as part of their intuitive thinking.

Assessment Objectives

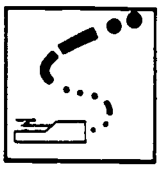
Assessment and development of the specific thinking processes at this level is guided by the following objectives:

1. Procedures are a way of defining new words which contain instructions that the Turtle uses to understand a model of an idea (coding).
2. Procedures are used as a way to experiment with different forms of instructions and command combinations (exploration).
3. Procedures are used to test and develop ideas. They may produce both expected and unexpected results (prediction).
4. Procedures are used either singly or in combination with other commands to create new and unusual effects (creativity).
5. Procedures are basic building blocks that can be put together to make larger and more complicated programs (analysis and planning).
6. Procedures are a powerful tool for finding and fixing things that don't work as they should (debugging).

*From *Mindstorms: Children, Computers and Powerful Ideas*, by Seymour Papert.

Procedures provide a language that teachers and students can use to talk about the idea of procedural thinking. They offer a concrete and meaningful way of representing abstract ideas that might otherwise be too difficult for the student to grasp. The structured and systematic nature of procedures makes it possible to focus on the specific thinking processes that students engage in as they construct models in Logo which best reflect their ideas.

How to Assess the Six Thinking Processes



1. CODING

At this level, coding involves the translation of a set of commands or instructions into a procedure (program) which Logo understands. The first requirement is that students understand the words which tell Logo to begin and end a procedure. Next, students need to know how to name a procedure and put a given set of commands into a procedure. Finally, students are required to translate plain English instructions into the form of a procedure.

There are several related concepts that need to be developed at this level of coding. First, students need to understand the process of giving meaningful names to procedures. Naming procedures can be a powerful means of personalizing a piece of work. Next, they need to recognize the distinction between working in the immediate mode (?) and the procedure writing mode (>). It is a good idea to introduce students to the procedure writing mode by having them code a set of commands between the title line TO (name) and END.

The main aim of assessment here is to observe whether students can:

(1) Remember the commands needed to leave and enter the procedure writing mode;

(2) Write simple procedures from given sets of Turtle commands; and

(3) Translate plain English statements into sets of basic Turtle commands and write these as procedures.

Activities to Assess and Develop Coding

A. Review the method for setting up a Logo procedure with students.

```
?TO PROCEDURENAME
>
>
END
```

Write a procedure together as a class activity and then have students check it by typing the commands into the computer.

Give students a set of commands which they are to put into a procedure. Use this as an opportunity to talk about the distinction between the immediate mode (?) and the procedure writing mode (>).

B. Prepare an overhead or handout which introduces students to the importance of procedure names.

"Procedure names are very important in Logo. They are:

"Commands which tell the Turtle to do something."
"New words which the Turtle understands and obeys."
"A way to help you remember what a procedure does."
"A way to help you find a particular procedure."

Do a name guessing activity. Invite students to guess the meaning of certain procedure names. For example:

BLUSQ
BIGTRI
SMREC

Show students some Turtle drawings. Ask them to think of a name for each one.

C. Ask students to work in teams to translate basic Turtle commands into procedures. For example:

"Write a procedure which uses these Logo commands."

LOGO COMMANDS	PROCEDURE
REPEAT 3 [FD 50 RT 120]	
RT 180	
REPEAT 3 [FD 50 LT 120]	

"What word tells the Turtle to draw (run) your procedure?"

"Make a picture here to show what the Turtle drew."

"Write a procedure in Logo which follows these instructions."

"Repeat the following set of commands 4 times:
Go forward 20 steps and turn right 90 degrees."

"Repeat the following set of commands 4 times:
Go forward 40 steps and turn right 90 degrees."

"Make a picture here to show what the Turtle drew."

"Describe the main steps for writing a procedure.
What do you do first?
What do you do next?
What do you do last?"

Ask students to help make a list of the main steps in writing a procedure. Use the blackboard or overhead to prepare a master list which students can copy into their workbooks.

Structured activities are ideal for introducing students to the commands and steps in procedure writing. Completion-type activities can be used at the beginning to acquaint them with conventions for naming and putting commands into procedural form. Class activities can be combined with team work later on. Most of the coding exercises can be done on paper and then checked on the computer. All of the activities presented here will serve as a useful basis for discussion of the main concepts and problems encountered. *(See worksheet on page 66)*

2. EXPLORATION

The main aim at this level is to encourage learners to use procedures as a tool for exploring. Here they begin to develop insights into the ways that procedures can be used to "program" the Turtle. Within the procedures, students can experiment by changing instruction and command combinations. Having defined procedures, they can then use these procedure names along with other commands to observe the effects produced by each combination. For instance, a learner who wants to draw a pattern with many squares or circles can explore different drawing approaches by telling the Turtle to use the procedure SQUARE in combination with the turn commands (LT and RT). The essential point of this process is to have learners engage in the full range of exploration which procedural thinking permits.

Assessment of the exploration process is aimed at observing the extent to which learners:

(1) Make variations to procedures by altering, adding and recombining commands and by changing the numbers used as inputs to procedures;

(2) Compare the outcomes of procedures which contain different command combinations and input numbers; and

(3) Experiment by using a procedure name with other command combinations to produce new drawings.

Activities to Assess and Develop Exploration

A. Remember that not all students naturally know how to explore. Much of their past school experience may not have actively encouraged free exploration and discovery. The Logo emphasis on exploration as a personal process can be encouraged by getting students to direct their own experiments.

Present students with a procedure and ask them to list the ways in which it can be changed. For example:

"Here is a procedure which draws two squares, one on top of the other. List some ways in which the procedure can be changed to make a different drawing."

PROCEDURE

```
TO SQUAREUP
>REPEAT 4 [ FD 40 RT 90 ]
>LT 180
>REPEAT 4 [ FD 40 LT 90 ]
>END
```

"Describe some changes you could make here."

B. Make a master list on the blackboard or overhead of changes that students thought of. For example:

1. Change the direction in which the squares are drawn.
2. Add more combinations of squares and turns.
3. Change the REPEAT commands so they make triangles.
4. Change the size of the squares.

Ask students to work in teams to try making any changes they like. Have them keep a record of the procedures used and the drawing they produced.

PROCEDURE	DRAWING

NOTE: You will need to explain to students that it will be necessary to use a different name for each new procedure they write. Otherwise, have them remove the first procedure from the computer memory before attempting to do it again.

To erase the procedure SQUAREUP use the erase command. Type: ERASE "SQUAREUP.

C. Ask each team to write a procedure that draws a simple shape or design.

Have the teams exchange procedures with one another.

Ask each team to try changing the procedure they were given in whatever way they like.

Have the teams keep a record of the procedures used and the drawings they produced.

Finally, ask each team to present a drawing of their most interesting procedure. Have them explain what changes they made.

Remember that the aim of these exercises is to increase students' awareness of the kinds of exploration that can be undertaken. It is good to start with a structured activity at the beginning and then provide opportunities later on for students to write and explore their own procedures. Students will vary in their approaches to exploration: Some will explore the full range of possibilities from the outset while others will need guidance to see the alternatives which are available. The activities presented here can cater to such individual differences and provide a basis for discussing the exploration process. It is important to encourage students to talk about what they did and how they did it. *(See worksheet on page 68)*

3. PREDICTION

Procedures provide a new and powerful tool to use for prediction. They also make working with Logo just a little bit more interesting and challenging. The basic aim at this level is to have students develop their skills in predicting what a given procedure will produce. At a more advanced stage, prediction involves the estimation of effects that will result from using procedures in combination with other Turtle commands. The role of the teacher is to encourage students in every possible way to engage in self-directed testing of their ideas.

Assessment of the prediction process involves observing the extent to which students can:

(1) Specify what shapes and designs will be produced by given procedures;

(2) Estimate the size and proportion of drawing which will

result from a given procedure or set of procedures; and

(3) Predict the outcome which will occur when procedures are used in combination with other commands.

Activities to Assess and Develop Prediction

A. Some students naturally engage in prediction as part of their ongoing work with Logo. Others will need guidance and encouragement to think in a more systematic way about the process of making informed predictions.

At the outset, you should encourage students to estimate what shapes will be produced from a given procedure. For example:

"Have a look at this procedure. Guess what it will draw. Then type the procedure into the computer to see if your guess was right."

```
TO GUESS1
>REPEAT 2 [ FD 50 RT 90 FD 30 RT 90 ]
>END
```

"Was your guess right?"

"How did you figure out what the procedure would make?"

B. Have students work in teams to write procedures which make simple shapes and designs.

Then have them exchange their procedures with one another. Ask them to guess what they think the procedure they have will draw.

"What do you think the procedure you were given will draw? Think first, then make a picture to show what you think it will draw. Then try out the procedure to see what happens."

PROCEDURE	WHAT I THINK IT WILL MAKE	WHAT THE TURTLE REALLY MADE

"Did you guess right the first time? If not, can you explain how your 'guess' drawing was different from the one the Turtle drew?"

Prediction activities are a good way to get students interacting with one another and sharing their ideas. When surprising results occur, these can be used to discuss ways of thinking about Logo problems. Activities can be done individually, in teams, or as a class activity. Teams are best for encouraging students to talk about their ideas and problems.

C. Ask students, working either individually or in groups, to do the following:

1. Write a procedure which uses a shape to make a geometric design.
2. Ask other students or groups of students to predict what will be produced by the procedure.
3. Discuss any falsified predictions, diagnose problems and propose alternate solutions.

Have a class discussion about methods for making predictions. Ask students to explain how they figured out what a procedure would make and what they think are the main steps in the prediction process.

Use students' responses to prepare a master list of prediction methods. For example:

1. Start from the beginning to get a picture in your mind of what the procedure is doing.
2. Do each part in turn and then put the parts together. For example, work out what a REPEAT command makes and then check how it works with other commands.
3. Walk out the Turtle commands or draw them on a piece of paper.
4. Take time to think about what the procedure is doing.

This kind of exercise helps students "think about their thinking." It gives them information they can use to work out a prediction strategy for themselves. Remember that (as research has shown) students do not automatically know how to reflect on their own thinking processes. But when they are given appropriate training, they can learn effective problem-solving methods. The teacher's role is to provide this training while at the same time encouraging students to think for themselves. *(See worksheet on page 71)*

4. CREATIVITY

The understanding of procedure provides a rich source of material which students can use to generate new ideas. It is this skill in knowing how to generate ideas which is the key to good thinking and creative problem solving both in the microworld of the Turtle and the real world of the learner. But, as is true with any skill, the ability to think creatively is something that needs to be developed. This is done by providing students with the kinds of knowledge and experience they need to build new ideas for themselves. The ability to build something with one's own ideas can offer a lot of personal satisfaction and lead to some wonderfully original work.

The procedure is the main "tool of invention" that Logo provides. It is this tool which students use for creating new objects or devices from available ideas and materials. But the ability to write a procedure in Logo does not mean that the student will automatically know how to use this as an inventing tool. Intervention by the teacher is often required to guide students in learning how to use the procedure as a tool for invention. This intervention takes the form of assessing and developing the creative process which students engage in as they:

(1) Invent their own designs and patterns by using a procedure;

(2) Devise their own patterns or designs by using a procedure in combination with other commands; and

(3) Construct geometric shapes and designs by using combinations of two or more procedures.

Activities for Assessing and Developing Creativity

A. Explain to students that ideas come through many different sources. Ideas can come from:

- Looking into your own mind and feelings;
- Observing the things that surround you; or
- Thinking about other people's thoughts and ideas.

To demonstrate the three main sources of ideas, ask students to:

"Make a picture of a shape in your mind. Draw a picture on paper to show what you are thinking. Write a procedure which tells the Turtle how to draw your shape."

THE SHAPE I HAD IN MIND

MY PROCEDURE

"This time, observe all of the things around you. Pick something that interests you and make a simple drawing to show what this is. Write a procedure which tells the Turtle how to draw the object you selected."

DRAWING OF THE THING I FOUND INTERESTING

MY PROCEDURE

B. Prepare a few procedures which draw simple shapes or designs. Present these on the blackboard or by overhead projector.

"Here are a few thoughts that I had. I want you to pick one of these thoughts and use it any way you want to make a new pattern or design."

MY NEW DESIGN DRAWING

MY PROCEDURE

C. Have an open class discussion about using procedures as "tools of invention." Ask students to share some of their best ideas from the above projects. Use the blackboard/ overhead to do an analysis of the students' drawings. Encourage them to discuss what basic part was used to create the design and how it was put together. The teacher may need to facilitate this process by getting students to talk about how they were thinking and what methods they used to build the design.

The essential point to remember is that creative thinking skills don't develop just by playing with the Turtle. Students need practice in using procedures to generate ideas and experience in thinking about how to build with ideas. The role of the teacher is to set the stage for students to not only acquire these thinking skills, but to know how to make best use of them. One important way of developing these creative skills is to encourage children to become fluent, flexible and original thinkers as they invent with Logo:

- Help children become original thinkers.

  "That's an interesting one. I've never seen anything like it before."

  "That's a new idea. Keep a copy of that so we can analyze it later."

  "That's an interesting way to put those shapes together. I never would have thought of that."

- Help children become flexible thinkers.

  "That's an interesting way of putting the parts together. What would happen if you put the parts together in exactly the opposite way?"

  "Can you think of an unusual way to make a castle from these squares?"

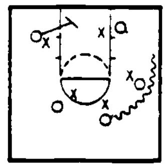
  "Good. It's nice to see you using the idea in two different ways."

- Help children become fluent thinkers.

  "Can you think of some more ideas?"

  "Now tell me what choices you have for putting these parts together."

  "Think about that one for a minute. Then tell me a way for making it." *(See worksheet on page 74)*



5. ANALYSIS AND PLANNING

Students working with Logo often engage in informal planning. They spontaneously work out problems in their heads and don't prepare any written plans or notes about the methods they are using. While this informal analysis and planning works well for some students, it is not good for everyone. Many learners need guidance and direction to learn methods of planning which are effective for them. For example, some students work best in a situation where they have detailed and orderly written plans. Many will use a few notes to outline their ideas but do little formal planning. Others will plan in a spontaneous way by following the "insights" which develop in their minds. Any one of these methods can be effective, but not for all students.

Working with procedures provides an ideal opportunity to help students improve their analysis and planning skills. This is because the structured form of procedures can encourage them to reflect in a systematic way about their thinking. But as Seymour Papert and mathematician George Polya have suggested, the general methods for planning and problem solving should be taught. Here, the role of the teacher is to help students become aware of which analysis and planning strategies are most effective for them. Teaching this self-awareness first requires a thorough assessment to determine how and to what extent students are reflecting on planning.

There are three specific aims of assessment at this level:

(1) Observe the particular kind of analysis and planning approaches that a student uses;

(2) Assess the effectiveness of these approaches in relation to the student's style of learning;

(3) Intervene when necessary to help students learn alternate methods of planning that might be more effective for them.

Remember that the process of analysis and planning does not automatically happen when students program with Logo. Very often it is necessary to guide the development of this process with attention to individual learning styles. The starting point here is to observe the strategies which students use to:

(1) Devise a plan for writing a procedure which makes a given design;

(2) Identify the main problem solving stages for making a procedure which produces a specific drawing or shape; and

(3) Devise a plan which uses a procedure to construct their own drawing and describe the main steps in planning it.

Activities to Assess and Develop Analysis and Planning

A. Prepare an overhead/handout which shows the four main steps in analysis and planning:

Step One: Understand the Problem. Make a drawing of the proposed project. Describe the main parts which are needed and how these might be put together.

Step Two: Make a Plan. Write a procedure which you think will tell the computer how to make the project.

Step Three: Carry Out the Plan. Type the commands into the computer and then draw a picture of what the Turtle did.

Step Four: Look Back. Decide if the Turtle did what you wanted it to. If not, go back and make changes.

Have students work individually or in teams to plan a project of their own. Ask them to keep a record of the four planning steps they used.

B. Have a class discussion about the steps involved in analysis and planning. Explain that having a plan is more than just following a fixed scheme. It means being systematic in making and changing plans to suit a particular situation. Here, we make a plan, analyze it as we are working on the problem, and make changes that we think are needed to help improve the work.

Ask students to help you prepare a planning checklist for the class. Ask them to describe which methods they find helpful. For example:

"Take time to think through a problem."
"Be systematic in working on problems."
"Work out a general plan of attack."
"Evaluate plans and make changes as needed."
"Try alternative ways of approaching the problem."

Encourage students to talk about methods and problems they encountered in planning their work. This verbal rehearsal is a good way for them to become aware of their planning skills.

C. Ask students to do a project of their own the four steps of analysis and planning. Have them keep a written record of the main steps they followed.

"MAKE A PLAN"

"Make a drawing of your project here."

"Explain in your own words what the main parts of the drawing are and how these can be put together."

"CARRY OUT THE PLAN"

"Write your procedure here. Then type the commands into the computer."

"Draw a picture of what the Turtle did."

"LOOK BACK"

"Did your procedure draw what you wanted the first time?"

"What changes did you need to make?"

"Did you stick to the same plan from the start to the finish?"

"Can you think of a different way to write a procedure which will make the same drawing? Write your new procedure here." *(See worksheet on page 77)*

6. DEBUGGING

The idea of procedures as things that can be debugged is a powerful, difficult concept for many children until they have accumulated experience in working with them (*Mindstorms*, p. 154). As educators, we can help by providing the conditions which students need to engage in debugging as a natural and enjoyable process. Sensitivity and care should be used to ensure that students gain an awareness of debugging as a learning tool. We are not talking about programming errors, but about the different ways in which a Turtle behaves. The Turtle may behave in ways that are either intended or unanticipated by the student. In any event, the task of the student is to understand the Turtle's behaviour and to fix or modify it as needed to produce a desired result.

Procedures are the main building blocks of Logo and the key to understanding Turtle behavior. As such, procedural thinking and the debugging process go hand in hand. These processes of thinking do not develop automatically, however. They depend instead on skilled intervention by the teacher. To be effective, this intervention needs to be guided by a careful and thorough assessment of the debugging process with each individual student. Assessment at this level involves observing the extent to which students can:

(1) Identify and fix simple bugs (syntax) in a given drawing produced by a procedure;

(2) Locate and correct logic (programming) bugs in a given drawing produced by a procedure; and,

(3) Connect the idea of debugging real life procedures with the process of debugging Logo procedures.

Activities to Assess and Develop Debugging

A. Give students a bugged procedure and ask them if they can find the bugs and fix them. Have them keep a record of the bugs they found and fixed.

Prepare a handout containing the following information:

"Look at this procedure. It has bugs which keep the Turtle from drawing what it was supposed to. See if you can find and fix each of the bugs. Keep a record of the bugs you found and how you fixed them."

86

PROCEDURE

DRAWING OF WHAT THE TURTLE WAS SUPPOSED TO MAKE

DRAWING OF WHAT THE TURTLE MADE

BUGS FOUND | BUGS FIXED

B. Do a class exercise to illustrate the ways in which bugs can be used to promote thinking. Explain that bugs are not necessarily right or wrong, bad or good. There are times when bugs provide a source of ideas for inventing something new. Finding and fixing bugs (debugging) often leads us to new ways of thinking about a problem. For example:

"Have a look at this procedure. It has some bugs that keep it from working exactly as we want."

Ask students to think about some ways they could use the procedure as it is to invent something new. Encourage them to think about;

- New ideas that come to mind.
- Ways of improving the procedure.
- Ways of changing the procedure.

Have students work individually or in teams to change the procedure any way they like. Ask them to keep a record of the changes they made.

Ask students to present their new inventions to the class and describe the changes they made.

C. Have a class discussion about the use of procedures in other areas of life. Describe some everyday activities in which people use procedures. For example, following a recipe, reading a set of instructions or giving someone directions are all procedures of a similar kind. They contain an orderly set of instructions that help people think in a systematic way. Ask students to think of situations in their own lives where they follow or make procedures of their own.

Ask students to do an exercise which helps them draw a connection between procedures in real life and procedures for the Turtle. For example:

"Think about the steps involved in doing each of the following things:

- Turning on the computer and loading a program from disk.
- Doing addition on a calculator.
- Drawing a square with paper, pencil and a ruler."

"Write a procedure which contains a set of instructions for doing one of the above. Be sure you write down all the steps in the order they should be done. Use as many steps as you need to."

```
TO
>
>
>
>
>
END
```

"Ask another student to check and debug your procedure. Keep a record of any changes made."

"Write down any changes made here."

"Did you agree with the changes they made? If not, explain why."

"Explain how fixing real-life procedures is like (the same as) fixing procedures for the Turtle."

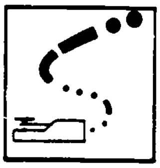
"Explain how fixing real-life procedures is unlike (different from) fixing procedures for the Turtle."

Have a class discussion about the procedure-writing exercise. Ask students to talk about their procedures and any bugs that they found. Use these examples to illustrate how bugs can occur in real-life procedures just the same way as they do in Logo procedures. Encourage students to think of bugs as a natural and enjoyable part of the learning process. *(See worksheet on page 80)*

Assessing Learning
With Logo

Name: ______________________

Date: ______________________



Checking Your Coding Skills

Defining Procedures

1. What word tells Logo that you want to write a procedure?

   What word tells Logo that you have finished writing a procedure?

   CODE NAMES
   Procedure names sometimes have lots of meaning. Guess what these names might mean. Write your guess next to the name.

   BIGSQUARE

   TRI

   SMCIR

   BLUESQ

   REDREC

2. Write a procedure which makes a rectangle using these commands:

   FD 50 RT 90 FD 60 RT 90
   FD 50 RT 90 FD 60 RT 90

   TO RECTANGLE
   (Insert Logo commands for a rectangle here.
   You may use 1 or more lines.)

   \>
   \>
   END

   Now write a procedure which uses the following commands. Put your procedure in the column next to the Turtle commands.

LOGO COMMANDS	PROCEDURE
REPEAT 4 [FD 50 RT 90] RT 180 REPEAT 4 [FD 50 LT 90]	

   Try out your procedure by typing it into the computer.
   How do you tell the Turtle to draw (run) your procedure?

68

Draw a picture to show what your procedure tells the Turtle to do.

What name did you give your procedure?

3. Write a procedure to draw a shape that has equal sides, 50 steps long.

?TO
>
>
>END

What name did you give the procedure?

How do you tell the Turtle to draw (run) your procedure?

Describe in your own words how to write a procedure in Logo.

Assessing Learning
With Logo

Name: ______________________

Date: ______________________

Exploring With Logo Procedures

1. Type in the following procedure.

```
TO TRI
>REPEAT 3 [ FD 50 RT 120 ]
>END
```

Now experiment to see what patterns you can make with TRI. Try this one to get started:

```
TRI
RT 90
TRI
RT 90
TRI
RT 90
TRI
RT 90
```

Draw a picture to show what the Turtle made.

What happens if you change RT 90 to RT 45?

What happens if you change RT 90 to LT 90?

Explore what happens when you use TRI with other commands.

Make a picture of any design you discovered when playing around with TRI.

2. Experiment to see if you can discover how to write a procedure which draws a circle inside of a square.

Start with this square:

REPEAT 4 [FD 50 RT 90]

Write the commands for your procedure here. Use as many commands as you like. Draw a picture to show what the Turtle made.

PROCEDURE | DRAWING PRODUCED BY TURTLE

TO
>
>
>
>
>
>
>END

HINT—Try drawing the square first. Then move the Turtle so that it is in the right position to begin drawing a circle inside the square.

List some ways in which you could change the above design.

1.
2.
3.
4.

Now write a new procedure which makes one of the changes you have noted above.

Remember, you will need to use a new name for each procedure you write (e.g., SQCIRCLE, SQCIRCLE1). This tells the computer to give each new procedure a name of its own.

3. Type in the following procedure.

```
TO TURNIT
>REPEAT 4 [ FD 50 BK 50 RT 90 ]
>END
```

Run the procedure by typing TURNIT and draw a picture of what the Turtle made.

71

Try making some changes of your own by:

a. Changing the REPEAT statement.
b. Changing the number of degrees in RT.
c. Adding commands.

Remember, you will need to use a different name for each new procedure you write.

Write your new procedure and draw a picture of what it makes in the space below.

PROCEDURE DRAWING PRODUCED BY PROCEDURE

Draw a picture of any other designs you discovered when changing and adding commands to your procedure.

List some more changes that you could make to the design.

1.
2.
3.
4.

72

Name: ____________________

Date: ____________________

Checking Your Prediction Skills

Defining Procedures

1. What do you think these procedures will make? Think first and then make a drawing on paper of what you think the Turtle will make.

A. ?TO TWOFLAG

```
>REPEAT 2 [ FD 80 RT 90 FD 40 RT 90 FD 40 RT 90
>FD 40 RT 90 BK 40 RT 180 ]
>END
```

Draw what you predict will happen. Draw what happened.

B. TO TRI3

```
>REPEAT 3 [ FD 50 RT 120 ]
>RT 120
>REPEAT 3 [ FD 50 RT 120 ]
>RT 120
>REPEAT 3 [ FD 50 RT 120 ]
>END
```

Draw what you predict will happen Draw what happened.

73

Now type in the procedures to see what happens.

Draw pictures to show what happened in the spaces next to your predicted drawings.

What differences are there between what you guessed would happen and what really happened?

2. Write procedures for a pentagon (five-sided shape) and a hexagon (six-sided shape).

PENTAGON

HEXAGON

Type in each of the above procedures to see what happens.

Does the Turtle draw the shapes you wanted? If not, explain why your procedure drew something different.

How did you figure out the number of steps and degrees of turn for each shape?

How did you figure out the number of times each set of commands would need to be repeated for each shape?

3. Think about what this procedure will make.

```
>TO GUESS2
>REPEAT 4 [ FD 50 RT 90 FD 20 RT 90 ]
>END
```

Draw a picture to show what you think the Turtle will make.

Type in the procedure to see what happens.

74

Now think about what will happen when we use the procedure GUESS2 with another command like this.

```
?GUESS2
?RT 90
?GUESS2
?RT 90
?GUESS2
?RT 90
?GUESS2
?RT 90
```

Draw a picture to show what you think the Turtle will make.

Type in the above commands now to see what happens.

What differences are there between what you thought would happen and what really happened? If there are any differences, can you explain why?

What do you think the Turtle would draw if we changed RT 90 to RT 45?

Draw a picture to show what you think the Turtle will make.

Type in GUESS2 and RT 45 several times to see what happens. Did the Turtle draw what you thought it would?

Name: ______________________________

Date: ______________________________

Using Your Imagination to Define Procedures

1. Think of something you can invent from these shapes.

Draw a picture to show two different things you could invent.

Invention 1 | Invention 2

Now write a procedure which tells the Turtle how to make each of your inventions.

Invention 1 Procedure | Invention 2 Procedure

Which invention do you like best? Explain in your own words how you made this invention.

2. Think about how you can make a design by joining five squares of equal size.

Write a procedure which tells the Turtle how to draw a square.

Make a design which uses the square only five times. At least one side of each square must join the side of another square. Draw your design here.

Design Using Five Squares

Tell the Turtle how to draw your design. Use your SQUARE procedure and other Turtle commands to complete the design. Keep a record of the commands you used.

3. Have a look at the things around you. Pick two things that interest you and make SIMPLE drawings to show what each one looks like.

Thing 1 | Thing 2

Write a procedure that will tell the Turtle how to make each thing.

Thing 1 Procedure | Thing 2 Procedure

77

Now can you think of some ways to put these things together to make a brand new drawing? Play around by using your procedures with other Turtle commands until you come up with an idea that you like.

Draw a picture of your best idea for combining the two things here.

Combined Things Drawing

Write down the commands which you used to tell the Turtle how to make your drawing.

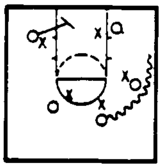
Combined Things Commands

Explain in your own words how you put the two things together.

Make drawings of any other pictures you discovered while combining the two things above.

Name: ____________________

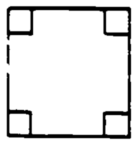
Date: ____________________



Checking Your Analysis and Planning Skills

Defining Procedures

1. Think about how you would make this design using a procedure.



A. **Make a plan.** Write the commands here that tell the computer what to do.

B. **Carry out the plan.** Type the commands into the computer.

Draw a picture of what the Turtle did.

C. **Look back.** Compare the drawing above with what the Turtle made. If they are not the same, go back and find the bugs and make the changes.

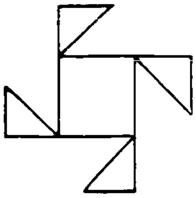
What bugs did you find? Explain how you fixed these.

Think back on how you made this design.
What step did you do first?

What steps did you do next?

What step did you do last?

2. Think about how you would make this design with a procedure.



A. **Make a plan.** Write a procedure to make the design.

B. **Carry out the plan.** Type the commands into the computer.
Draw a picture of what the Turtle did.

C. **Look back.** Compare the design above with the one the Turtle drew. If they are not the same, go back and find the bugs and make changes.

What bugs did you find? Explain how you fixed these.

Think back on how you made the design.
What step did you do first?

What steps did you do next?

What step did you do last?

80

3. Use some ideas of your own to make a new design. Make any design you like but be sure to use a procedure.

   Use the New Design space below for planning and drawing your new design. Write your commands in the New Design Procedure space.

   New Design | New Design Procedure

Describe in your own words what the new design looks like.

How many parts are there in your new design?

Look back. Think about how you made the new design.
What step did you do first?

What steps did you do next?

What step did you do last?

Did you have a plan before you started to make the design?

Did you stick to the same plan from the start to the finish?

Describe in your own words what the four main steps are in planning a design.

81

Name: ____________________

Date: ____________________

Checking Your Debugging Skills

Defining Procedures

1. This procedure is supposed to make an octagon but it doesn't work right. See if you can find the bugs.

```
?TO OCTAGON
>REPEAT 7
>FD 30 RT 45
>END
```

How many bugs did you find?

How did you figure out what the bugs were?

List the main steps you follow when debugging a procedure.
What is the first thing you do?

What things do you do next?

What is the last thing you do?

2. This procedure is supposed to make a house. Type in the commands to see what it does.

```
?TO HOUSE
>REPEAT 4 [ FD 50 RT 90 ]
>FD 50
>REPEAT 3 [ FD 40 RT 120 ]
>END
```

Draw a picture of what the Turtle made here.

82

Change the procedure so that it tells the Turtle how to draw the house properly.

Make a drawing of your new house here.

What changes did you make?

Think of some ways to improve the house drawing. Make a list of your suggestions here.

3. Think about how you would make a peanut butter sandwich. Describe in your own words how you would make the sandwich.

Now write a procedure which contains a set of instructions for making a peanut butter sandwich. Be sure to include all the steps in the order you would do them. Use as many steps as you need to.

```
?TO SANDWICH
>
>
>
>
>
>
>
>END
```

Now ask another student to check and debug your procedure. Keep a record of any changes made.

Write down any changes made here.

Do you agree with the changes made? If not, explain why.

How is fixing the peanut butter sandwich procedure the same as fixing a procedure for the Turtle?

How is fixing the peanut butter sandwich procedure different from fixing a procedure for the Turtle?

Think of other times in your life when you use procedures. List them here.

What do you think are the main advantages of using procedures?

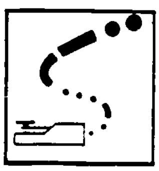
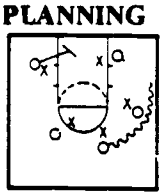
DEFINING PROCEDURES
THINKING PROCESSES CHECKLIST

NAME: ______________________ DATE: ______________________

SCHOOL: ______________________ CLASS: ______________________

INSTRUCTIONS: Place the code letter which best describes the development of a particular thinking process alongside each item in the ASSESSMENT column.

M = Learner has mastered and consistently demonstrates this process.
P = Learner has only partially mastered or inconsistently demonstrates this process.
N = Learner has not yet developed or demonstrated this process.

	THINKING PROCESSES	ASSESSMENT
CODING 	1. Remembers the commands needed to enter and leave the procedure writing mode.	______
	2. Writes simple procedures from given sets of basic Turtle commands.	______
	3. Translates plain English commands into sets of basic Turtle commands, and writes these as procedures.	______
EXPLORATION 	1. Makes variations to basic procedures by altering command combinations and input numbers.	______
	2. Compares the outcomes of procedures which contain different command combinations and input numbers.	______
	3. Experiments by using a procedure with other commands and command combinations (e.g., REPEAT statement) to produce complicated patterns.	______
PREDICTION 	1. Specifies what shapes and designs will be produced by given procedures.	______
	2. Estimates size and proportion of drawings which will be produced by given procedures.	______
	3. Describes the outcome which will result from using a procedure in combination with other commands.	______
CREATIVITY 	1 Invents own designs and patterns by using a procedure.	______
	2. Devises own patterns or designs by using a procedure in combination with other commands.	______
	3. Constructs geometric shapes and designs by using combinations of two or more procedures.	______
ANALYSIS AND PLANNING 	1. Devises a plan for writing a procedure which will make a given design.	______
	2. Identifies the main problem solving stages for making a procedure which produces a specific drawing or shape.	______
	3. Devises a plan which uses a procedure to construct own drawing and describes the main steps in planning.	______
DEBUGGING 	1. Identifies and fixes simple bugs (syntax errors) in a given drawing produced by a procedure.	______
	2. Locates and corrects logic (programming) bugs in a given drawing produced by a procedure.	______
	3. Connects the idea of debugging real life procedures with debugging Logo procedures.	______

COMMENTS

References

Abelson, H. "A Beginner's Guide to Logo." *BYTE*, August, 1982.

Abelson, H. *Apple Logo*. Peterborough: BYTE/McGraw Hill, 1982.

Clarke, V.A. and Chambers, S.M. *Thinking With Logo*. Sydney: McGraw Hill, 1985.

Clements, D.H. "Effects of Logo Programming on Cognition, Metacognition Skills and Achievement." Paper presented at the annual meeting of the American Educational Research Association, Chicago, 1985.

Hamilton, D. and Parlett, M. "Evaluation as Illumination." In Tawney, D. (Ed.). *Curriculum Evaluation Today: Trends and Implications*. London: Macmillan, 1975.

Hodgkin, R.A. *Born Curious: New Perspectives in Educational Theory*. London: John Wiley & Sons, 1976.

Minr 'a Educational Computing Corporation. *Apple Logo in ..e Classroom*. St. Paul: MECC, 1983.

Pape , S. *Mindstorms: Children, Computers and Powerful Ideas*. New York: Basic Book 1980.

Torgerson, S., Kriley, M. and Stone, J. *Logo in the Classroom*. Eugene: International Coun .l for Computers in Education, 1984.

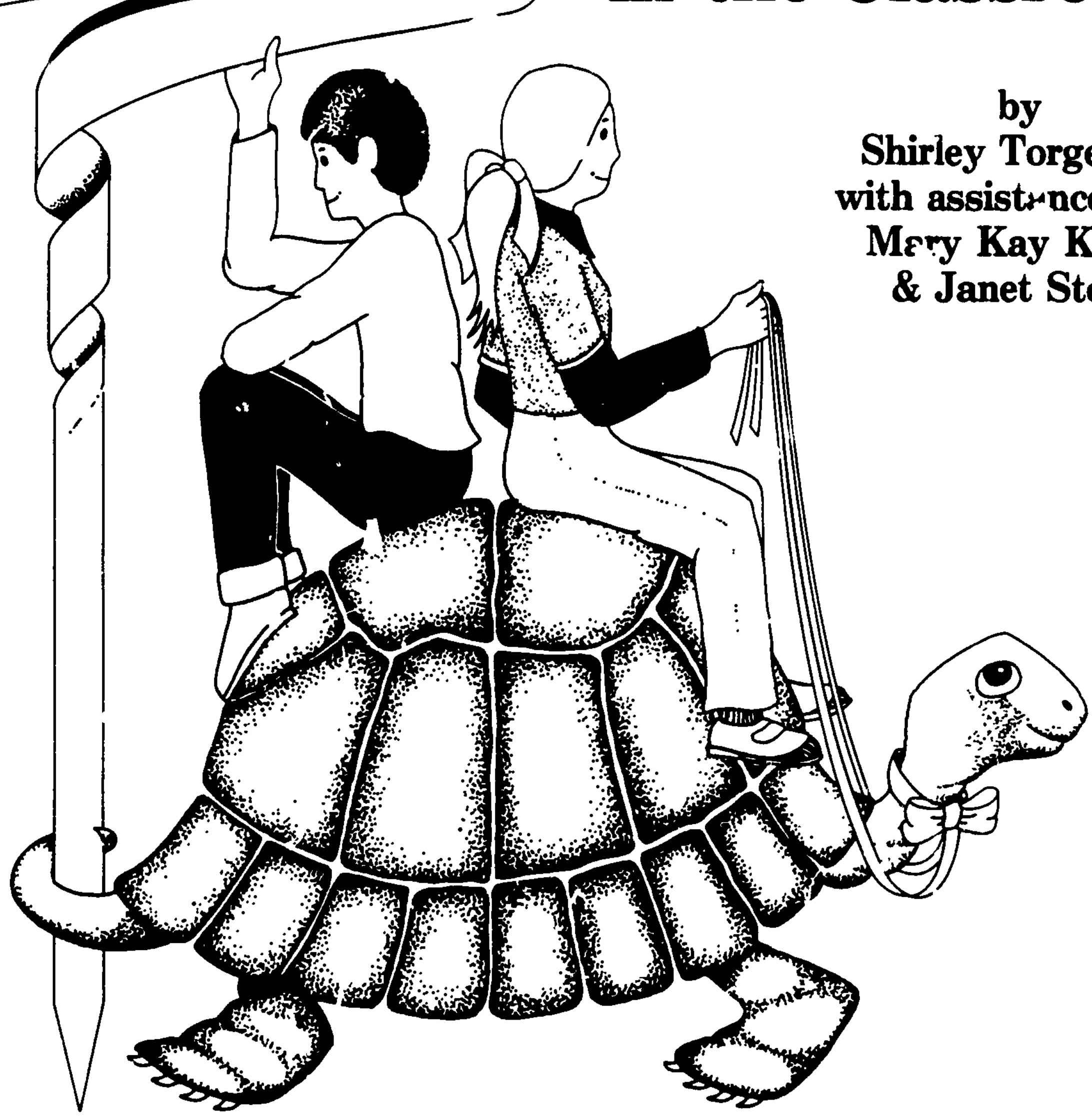
Watt, D. "Logo in the Schools." *BYTE*, August. 1982, Vol. 7, pp. 116-134.

Watt, D. *Learning with Logo*. New York: McGraw Hill, 1982.

Wills, S. "Is There Logo Beyond FORWARD, BACK, LEFT and RIGHT?" In Salvas, A.D. (Ed.), *Computing and Education—1984 and Beyond*. Melbourne: Computer Education Group of Victoria, 1984.

86

Logo in the Classroom



by
Shirley Torgerson
with assistance from
Mary Kay Kriley
& Janet Stone

Logo in the Classroom integrates Logo into your elementary curriculum. Twenty lessons were developed in a classroom setting as a response to "How can Logo work in a classroom where computers are in short supply?"

Detailed teacher information is given for each lesson along with copyable practice sheets, transparency masters and 12 charts. Useful for teacher inservice.

$13.00 (US)

International Council for Computers in Education (ICCE)
University of Oregon 1787 Agate St. Eugene, OR 97403

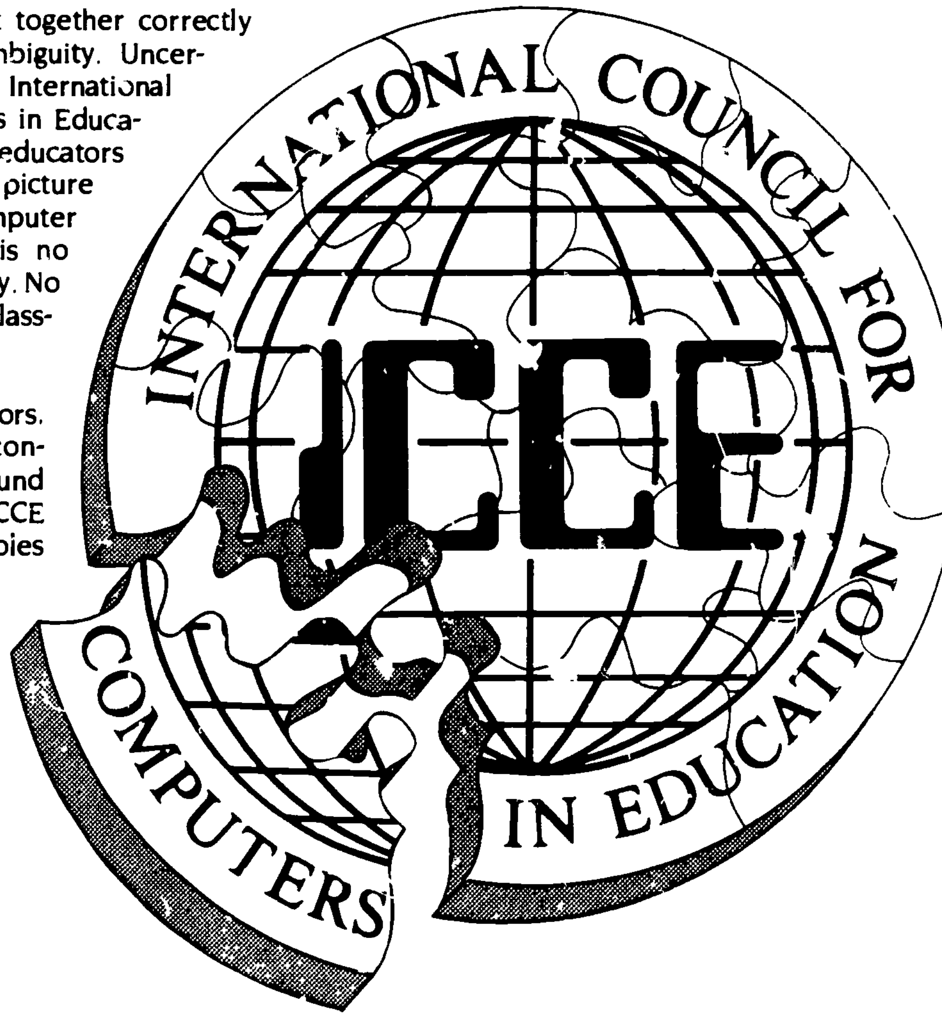
A Perfect Fit

ICCE's the one for you.

When pieces don't fit together correctly there's confusion. Ambiguity. Uncertainty. Since 1979 the International Council for Computers in Education has worked with educators to help shape a clear picture for the future of computer education. So there is no confusion. No ambiguity. No uncertainty in your classroom.

Educators, administrators, coordinators and concerned individuals around the world belong to ICCE —taking leadership roles in their classrooms, schools, districts and countries. ICCE is the organization they count on.

It's the one for you.



You Fit In

ICCE sponsors many diverse forums where your area, training or special interest fits right in

The Computing Teacher—novice or expert, you'll find up-to-date, practical information for computers in the classroom.

Special Interest Groups—share information to help your special interest area grow. SIGs include computer coordinators, teacher educators, administrators and special educators, and are planned for advanced placement computer science, community colleges and videodisc users. The quarterly **SIG Bulletin** serves as a forum for SIG information.

Booklets and Monographs are additional resources on specific topics. **ICCE Packets** provide you with teacher training materials. Members receive a 10% discount on all three.

ICCE Committees address a variety of ethical and practical issues important to you as a computer-using educator

Add to the picture of computer education —write for free information and a catalog today.

ICCE
University of Oregon
1787 Agate Street
Eugene, OR 97403
503/686-4414

the One for You

ISBN 0-924667-31-1