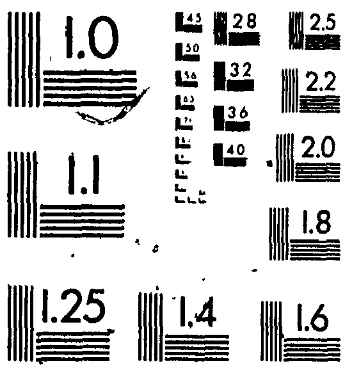
1



Production and Evaluation of a Self-Instructional Module for Teaching the Fundamentals of Computer Programming in the BASIC language

Jonathon Marsh

A Thesis Equivalent

in

The Department

of

Education

Presented in Partial Fulfillment of the Requirements for the Degree of Master of Arts (Educational Technology) at Concordia University

Montreal, Quebec, Canada

June 1988

ISBN 0-315-44821-0

ABSTRACT

Production and Evaluation of a Self-Instructional Module for Teaching the Fundamentals of Computer Programming in the BASIC Language.

Jonathon Marsh

This thesis equivalent presents the design and evaluation of a self-instructional module used in an undergraduate computer literacy course at Concordia University. The materials produced had to meet the following general goals:

1) provide an introduction to programming in general,
2) provide instruction on programming tasks in BASIC up to but excluding file management,
3) provide instruction on the use and design of flow charts.

Design and evaluation was conducted according to the method prescribed by Dick and Carey (1978) with some elaborations derived from the techniques outlined by Romizowski (1981). The package consisted of a compilation of existing text, video, and computer based materials reworked into a stand-alone self instructional format.

Because this was to be the last module in the course the evaluation was conducted in two stages. The first involved the piloting of the course wherein the materials were assessed for sequencing, coherence, timing, and completion rate. Subsequently the second stage was implemented, which involved two levels of evaluation. At the first data were drawn from embedded exercises, posttest scores, and on location observation of the entire enrollment. At the second, a more detailed evaluation of a representative subgroup was conducted based upon data drawn from demographic and course evaluation questionnaires, and pre/post/delayed test scores. Based on a 70/20 mastery criteria only 5 of the 10 subordinate objectives were successfully met. However, 100% mastery was achieved on both terminal objectives. In the conclusion the strengths and weaknesses of the materials are discussed and recommendations for revisions are presented.

I would like to thank the following people for their support, good humor, and faith in this work.

Richard Schmid

Andrew MacAusland

Johanne Roy

Mariela Tovar

Gary Boyd

Dennis Dicks

Annabel Marsh

TABLE OF CONTENTS

List of Tables

List of Figures

List of Appendices

CHAPTER ONE

Statement of the Problem

It was the purpose of this project to design and evaluate a self-instructional learning module on the topic of introductory programming in the BASIC computer language. The context in which the module was to be implemented was as the last unit of instruction in an introductory course in microcomputer literacy targeted at undergraduate students in the Arts and Science Faculty of Concordia University, Montreal, Quebec, Canada.

The course as a whole was experimental in that it was the only CAI based, self-instructional, in-house program of study available at the University. It covers a regular thirteen week semester and carries with it the standard credit value of the University for a single semester course. The content required for the course was determined by an organizing committee to include an introduction to:

1) the machine (the components and disk operating system);
2) word-processing;
3) electronic spreadsheeting;
4) database management;
5) problem solving; and
6) graphics and listhandling programming in BASIC and LOGO.

Sequencing of materials was left up to the course administrators judgment. Consequently, to enhance instructional effectiveness the course was organized into the following lesson structure.

LESSON 1 - Introduction to computer architecture (the machine),

LESSON 2 - Wordstar 2000 (word-processing),

LESSON 3 - SuperCalc 3 (spreadsheeting),

LESSON 4 - DBASE III (database management),Review module,

MIDTERM EXAM

LESSON 5 - Problem solving module, Programming in LOGO,

LESSON 6 - BASIC programming,

FINAL EXAM

Accordingly the course can be said to be divided into three general conceptual categories.

1) Computer architecture (lesson 1); which addresses the machine components, floppy disk design, binary logic, and the concepts needed to understand the function of a disk operating system.

2) Application software (lessons 2-4); which addresses common areas of non-expert computer usage.

3) Programming languages (lessons 5-6); which addresses the concepts inherent to the development and use of the logical sequences the computer invokes in order to solve problems.

Committee specification with regard to admissible content along with this topic structure imposed several constraints upon the design, implementation, and evaluation of the module on BASIC programming. These are as follows:

1) It must have an introduction to programming in general.
2) It must include instructional materials sufficient to enable the student to perform general programming tasks up to but not including file management.
3) It must provide a contextual focus for the concepts learned previously in the course.
4) It must require no more than four lab sessions, of four hours each, to complete it.
5) All the work relevant to completion of the requirements must be conducted in the computer lab.

Given these constraints, the choice of instructional materials was left up to the discretion of the instructional designer, subject to approval by the course administrator. The only requirement was that a multi-media format be used with a strong emphasis on CAI tutorials and hands-on computer based practice exercises.

CHAPTER TWO

Rationale

Basic Programming and Computer Literacy

It was not within the jurisdiction of the instructional designer for the BASIC module to decide upon the merit of including programming skills in the requirements for the computer literacy course within which it is situated. The arguments both for and against such inclusion are strong and the debate remains unresolved. There is however a prevailing attitude that the decision to include programming should be largely determined by the specific requirements of the given instructional context (Shively, 1984). Most of the literature reviewed, in particular the HumRRO (1982) and MECC (1982) projects emphasized the inclusion of some form of programming within the computer literacy curriculum. However such a curriculum is usually viewed as a multi year process with an increase in specialization at the higher grade levels. Little argument exists for or against its inclusion in a single course. Notably all the existing courses examined did include programming with the large majority of them utilizing BASIC. This sort of precedence in and of itself would help to legitimate the use of BASIC here, however it affords us no insight into how best to design the module. As such some examination of the arguments in favour of its inclusion was here necessary in order to help identify an optimal instructional approach.

The question was not whether or not programming skills represent an inherently valuable aspect of computer literacy. It was rather, given that programming was required, how to ensure that the appropriate concepts are identified and emphasized in such a manner as to maximize both the students' overall learning gain and the coherence of the course at large.

To accomplish this end it is necessary to organize these concepts so as to facilitate meaningful learning as much as possible. In order for meaningful learning to occur it is important that the learner be able to connect new information to related or domain specific knowledge already existing in memory (Bransford, 1979). This body of knowledge can then form what Ausubel (1968) terms "appropriate anchoring ideas" upon which to moor incoming information and according to which it can be manipulated. Novices by definition tend to lack domain specific knowledge (Spilich, Vesonder, Chiesi, & Voss, 1980; Greeno, 1980; Simon, 1980). As a result it is necessary, as Mayer (1980) points out, to build first the conceptual framework needed for the assimilation of new information. It is not difficult to imagine a design for the proposed module which does not address this need.

To illustrate, it is possible to envision this module designed so as to emphasize the proper syntactic form of a language while ignoring the very important aspect of its semantic structure; which involves an understanding of good problem solving and logical sequencing techniques. A functional understanding of programming syntax is dependent upon a previously established good understanding of the semantic structures used in programming languages (Schneiderman, 1980). The understanding of semantic structure can be seen to represent the set of anchoring ideas upon which the assimilation of the new syntactic forms can be established. Any instructional design which seeks to legitimately place emphasis on syntactic form must assume a good understanding of semantic structure as a prerequisite characteristic of the target population. Such an approach which may have merit for the somewhat more advanced computer science student would be discontinuous with the objectives of the course within which this module is situated. Furthermore, because of the increase in specificity, which is a necessary consequence of such a design, it would not serve the overall aim of general computer literacy as it is conceived of here.

Concerning the notion of computer literacy, it should be noted that, as Seidel (1982) points out, the concept of computer literacy is time bound in a rather more severe sense than other forms of literacy are. That is to say the concept itself is evolving as rapidly as the technology to which it refers. Hence, in order to maximize the usefulness of this module, not only in terms of quality and quantity of information presented but also over time in the face of rapid changes in the technology, it is important to establish a clear definition of what is meant here by computer literacy.

To date any attempt to provide a general definition of the term has generated much debate. Anderson (1982) argues that much of this conflict may stem from variance concerning the definition of the term "literacy" itself. He indicates that the term "literate" is generally used to indicate some arbitrary threshold on the literacy continuum. Just where this threshold lies seems to vary with the context of its use. When we speak of a literate individual we may mean anything from a general capacity to read and write to an individual who is well read and well spoken. The literacy threshold thus seems to fluctuate between minimal and maximal levels of competency depending upon the context in which it is being evaluated. On this note Anderson and Klassen (1980) have tentatively defined literacy as the ability to function effectively within a given role. The author does not entirely agree with this definition as it would appear to be a better definition of competence than of literacy. It is quite possible to conceive of an individual who is fully cognizant of all the required information particular to a given role but is unable to function within that role due to reasons of physical or emotional inadequacy. However this definition does allow for a varying literacy threshold and does afford us a flexible starting point, while endless dissension will afford us no workable solution whatsoever. As Deringer and Molnar (1981) point out:

"Diversity of opinion and even fervent advocacy is a characteristic of any rapidly advancing field and should be viewed as an opportunity. It should not be used as an excuse for lack of action. Ideas should be advanced, developed, and disseminated for users to judge their worth and value."(p. 3)

Given a recognition of the above stated reservation, it is in this spirit of advancement and development that the author here adopts Anderson and Klassen's definition of literacy and attempts to identify a threshold appropriate to the given mandate of this course.

In its simplest form the primary directive of the course is to provide the individual with a comprehension of the various work areas in which computer technology, and in particular micro-computers, may be of everyday use. This directive parallels closely the generally held belief (Deringer and Molnar 1981) that the demand for computer literacy derives from a rapidly increasing need to deal effectively with the information handling technology which is permeating business, industry, government, and even our homes and schools. In order to meet the above stated directive this course must take into account not only the individual's need for an understanding of information handling technology but also of the impact it's permeation into society has on the individual (which is furthered as a consequence of just such an understanding). It is with respect to considerations such as these that Seidel (1982) proposed a breakdown of the information handling curriculum into three general areas in order to enable the design of effective domain specific instructional materials.

1) Awareness of the social impact of information handling technology.

2) General information concerning the functions and processes of the technology itself.

3) Specific knowledge regarding the use of the technology.

Item three is further reduced to three subcategories.

a) knowledge of application software (i.e. what the computer can do);

b) specific knowledge and/or skill in the use of application software; and

c) skill development with regard to system procedures and programming.

The decision about which of these to include in a given course depends upon the course objectives but all of them are important with regard to an overall literacy curriculum. As the course, with which we are here concerned, is the only one of its kind available within the institution, it is important that it address each of these aspects. Interestingly enough, as it stands, Seidel's categorization is strongly reflected by the course structure recommended by the organizing committee. Seidel also argues that regardless of the ambiguities surrounding the concept of computer literacy there are three areas of general agreement concerning it's requirements. All three combine to indicate a qualitative change for the individual who seeks to become computer literate. They are as follows:

1) our society collectively and individually, must handle increasing amounts of information;
2) individuals need to become better problem solvers in order to deal with the issues raised by an increase in information; and
3) computers are a major component of the work environment with regard to helping solve problems and handling information; therefore all people should become computer literate.

The qualitative change implied for us as individuals has to do with the role we play within the work environment. Information handling technology affords us more information upon which to base our decision making processes. However, as in the case of an ungoverned positive feedback loop this information influx may easily become overwhelming causing the processing of it to breakdown altogether. In order to work effectively with this information, we must be able to manipulate the very technology which is bringing it to us in such a fashion as to maintain order and facilitate decision making. This implies that, while at one time our problem solving activity was reserved soley for dealing with issues directly related to the "stuff" of our working environment, now to an ever increasing degree it is becoming once removed. Now we must be able to apply our problem solving abilities to the manipulation of the new technology itself in order to maximize our efficiency. Thus it would seem that an increase in problem solving ability lies at the crux of the computer literacy issue.

As was previously stated the purpose of this examination of some of the issues involved in the concept of computer literacy was to help identify an optimal instructional approach given the requirements and constraints of the course. If we accept that problem solving is a crucial aspect of computer literacy then with respect to the question as to what role programming in BASIC is to play we must subsequently examine the following three hierarchically related issues: 1) the relationship between problem solving skills and programming skills in general (this in order to establish that there is some merit in including programming in the curriculum); 2) the relationship between problem solving and BASIC programming in particular (this in order to demonstrate that there is legitimate reason for favouring BASIC over other programming languages); and 3) the relationship between the development of both problem solving and programming skills and the process of learning itself (this to assist us in deriving a viable organization of concepts upon which to establish a method of instruction).

With regard to the first of these issues there is a general intuition prevalent among computer scientists, particularly those involved in educational applications, that hands on programming does encourage the development of problem solving skills. Papert (1979), Bork (1981), Luehrmann (1972), Howe & Dubouláy (1979), and Bell (1978) are certainly proponents of this viewpoint. All of them adhere to the notion that the activity of programming, due to the combination of required logical rigour and the interactive qualities of the machine, afford students a dynamic context in which to first experience the techniques of good problem solving and then to exercise and elaborate upon their understanding of them. The underlying support for their position is that educational researchers in general recognize the importance of dynamic interaction between the learner and the subject matter (the subject matter in this instance being problem solving itself). In the case of computing, the problem solving power of the machine is inherently reflected in the structure of any programming language.

These languages would not serve their purpose if they did not adhere to the principles of good problem solving. Indeed the various commands and functions available in any high level language are usually established on the basis of the frequency with which problems are encountered which require their particular characteristics to provide a solution. Given these considerations, while not conclusive, it is not unreasonable to posit a strong structural relationship between good problem solving and the effective manipulation of a programming language.

The question remains as to whether or not people with programming skills are better problem solvers in 1) the domain of computer usage, and 2) in other domains related to their work environment. Mayer's (1982) research, while far from comprehensive, would seem to provide at least some preliminary indication that there is indeed a relationship between competency as a programmer and the ability to maximally utilize the problem solving power of the machine; particularly when working with various kinds of application software. Similarly positive results in the research conducted by Howe, O'Shea, & Plane (1979) and Soloway, Lochead, & Clement (1982) on the relationship between programming skills and the ability to solve algebraic problems, along with the work of Skulicz (1984) on the relationship between programming and writing well structured English prose, would seem to provide us with some preliminary indication of the transferability of computer related problem solving skills to applications in other domains. While none of these researchers have produced results substantial enough to entirely legitimate either claim, the results they have produced, when combined with the intuitive considerations mentioned earlier argue strongly for the inclusion of programming of some sort in any computer literacy course; particularly one which has the development of problem solving skills as a critical objective.

The question as to the merit of teaching BASIC over other languages is a subject of much debate among individuals concerned with computers in education. The arguments both for and against its use are equally substantive and it is not the intention here to join in the dialogue; rather it is simply to point out some of BASIC's more valuable assets. Apart from the very pragmatic virtue that BASIC is almost universally included in the purchase price of any microcomputer, and that instructional manuals and software tutorials are abundant and readily available, there is some argument that it may be structurally well suited to the needs of the novice learner.

It is the authors opinion that one of the primary benefits of teaching programming is that the student, particularly the novice, learns to identify and manipulate effectively all of the logical sequences relevant to the solution of a given problem. In order to accomplish this end, it is most important that the student learns the same refinement techniques as have been shown to be inherent to good problem solving (Lochead, 1981; Lochead & Clement, 1979; Whimbey, 1980; Whimbey & Lochead, 1981). While such practice is inherent to the semantic and syntactic structure of most languages, BASIC requires more explicit iterations than most. Indeed it is often the case that more sophisticated languages are considered to be better precisely because they have avoided the need for such explicit detail when writing coding.

While arguments favouring other languages exist they are often based on the assumption that the person learning to program is a student of computing in the formal sense. They do not seem to be directed at the student who is learning to program for more general reasons such as developing analytic skills or simply to get a feel for the machine. For example Lewis (1982) argues that students should learn PASCAL because it forces them from the outset to learn the proper techniques and concepts pertinent to good structured programming. Doyle (1982) too recommends PASCAL because students are more likely to think a problem through initially, due to the relative difficulty involved in incorporating changes in a program. These arguments seem to refer to the issue of efficiency in machine utilization and may well have relevance for the student specializing in computer science. They do not seem to be at all concerned with the broader educational concerns of the individual, such as the suitability of the learning environment regarding learning style, or the establishment of a viable cognitive framework within which to situate new concepts. Soloway et al. (1982) have indicated that novice students learning PASCAL were easily confused about even the simple iteration constructs of the language. Students either made explicit those functions which are implicitly performed (e.g. loop-end testing, index variable incrementing) or they assumed that more actions would be performed than actually were (e.g. incrementing the index variable in the "WHILE" construct). These authors argue that such students do not have an adequate conceptual model of the primitive logical constructs which compose these more sophisticated operations and are as a result reduced to rote learning with little dynamic comprehension. The implication is that explicit iteration should be taught first using all the primitives (such as are available in BASIC), in order to develop a suitable conceptual framework upon which to base further learning. Then, after appropriate evaluation of the student, a slow graduation into more abstract and complex syntactical constructions may be implemented which would include the introduction of other languages.

It may be noted here that the semantic and syntactical structure of a language such as FORTRAN would fulfill all the above mentioned requirements and perhaps be of greater use in the work force. In the author's opinion such is not the case, if simply for the reason of BASIC'S ubiquity in the world of micro- computing. This combined with the availability of tutorial texts and software, as well as with the increased computing capability found in newer versions of BASIC, more than compensates for any utilitarian advantages learning FORTRAN may offer. Furthermore it is commonly held to be true that learning one language greatly facilitates the learning of a second, particularly when they are structurally similar (Schneiderman, 1980). Having taken advantage of all that BASIC has to offer it should be a relatively simple matter, should the need arise, for an individual to acquire a knowledge of FORTRAN at a later date.

There is one last consideration concerning the importance of including programming in a computer literacy course which derives from the socio-economic realm. Watt (1982) points out that there is an increasing danger of reinforcing social segregation on the basis of computer competency. Higher income learners tend to have greater access, both in terms of time and facilities, to computer technology. Thus they are more likely to learn more comprehensively about computers and are more likely to develop more highly refined utilization habits, including the ability to program, than are the less fortunate. This is because, given limited user access due to economic restrictions and the lack of available equipment, instruction for lower income students tends to focus around the ability to work with application software. This in order to maximize the utility of their shorter and more sporadic hands on computer experience. Learning to program is time consuming and the turn around time with regard to being able to apply newly acquired knowledge to a problem situation is long when compared to that of ready to use application software (e.g., word-processors, spreadsheets, and database managers). Thus the lower income learner (due to these restrictions and because it is in the nature of application software to be to a greater or lesser degree user friendly) is more likely to perceive the computer as a machine which can be of assistance only in so far as he or she is capable of following the instructions output by the software he or she wishes to use. It is possible that the learner will learn to perceive himself or herself as doing what the machine says to do and not as telling the machine what to do. Such a perception could severely limit the scope of machine utilization and subsequently restrict the amount of information which can be manipulated or to which access may be had. It is not difficult to see how this form of restriction could result in a means by which class discrimination may be further established within a technological society (Watt, 1982; Turkle, 1980).

Much work has been conducted on the subject of the social and psychological impact of information technology on modern society. Notable among researchers in the field are Papert (1981), Weizenbaum (1976), Wessel (1974), and particularly Turkle (1980), who in her work raises a number of questions specifically addressed toward computer literacy educators. However these concerns are largely beyond the scope of this document except in so far as they relate to the issues of including programming in the computer literacy curriculum and the design of computer literacy course material. With regard to the programming issue even the possibility of a computer related form of social discrimination, in as much as it is counter productive to the aims of education in general, represents a strong argument for, not only it's inclusion, but for it's primacy of importance over other areas of computer use. With regard to the design of courseware, the socio-psychological impact of information technology must be examined in the light of various observable response patterns, how they relate to learning styles, and how best to accommodate a maximum variety of styles within the design. This is an issue particularly pertinent to this course given the wide variety of student types represented by the target population. While specific information is not available concerning these patterns it is possible to accommodate a wide variety student types through the use of a spiral curriculum. This subject will be discussed at greater length in due course.

To recapitulate, it is interesting to note that after surveying numerous computer literacy courses, similar in intent to this one, all of them included programming of some sort, with a large majority using BASIC. Rationale for it's inclusion has varied somewhat with the specific directives of each course but it has been possible to identify five commonly cited items. These five account for most of the previously discussed issues and together compose the legitimation for including BASIC in this course as well as the basis upon which the instructional design was established. Briefly, they are as follows:

1) Programming skills are an integral part of computer use without which the user's under standing of how the machine solves problems will be severely impeded.

2) Understanding the machine's problem solving

processes is important to the development of a clear understanding of the machines capabilities even for high level usage.

3) In order to obtain optimum value from computer usage the user must be able to analyse the internal structure of a problem in such a manner as to allow him or her to enter it in a computer compatible format. The development of such a capability is facilitated by exposure to the rigours of logic inherent to any programming language.

4) An understanding of the general concepts inherent to any form of application software is facilitated by an exposure to the way in which such software is created.

5) The BASIC language was favoured because of its ubiquity and the utility of its detailed iterations in facilitating the development of a viable conceptual framework.

Educational Considerations

It may be noted from the aforementioned five considerations that the BASIC module may act as a powerful focusing agent for the concepts encountered previously in the course. As such it was scheduled as the last module. A spiral curriculum, in both the Deweyan and the Brunerian sense of the term, was consequently implied for the design of both the course in general and the BASIC module itself.

The Deweyan influence derives from Dewey's (1938) argument that growth in learning is dependent upon the student's use of intellect to overcome difficulties arising from experience. As the learner utilizes intelligence to overcome such difficulties new ideas are generated causing an increase in intellectual power. This increase then provides the basis for overcoming new problems which arise from ongoing experience. By engaging in this process the student becomes aware of the interrelationships between areas of knowledge as well as the wider social applications of any particular knowledge gained. Dewey (1938) saw this process as a continual spiral and believed that it should begin with learner experience and not from organized subject matter. This belief is reflected in the structure of this course in so far as the students' initial exposure to concepts such as DOS, RAM, ROM, editing, spreadsheeting, and database manipulation is very much situated within an exploratory, pragmatic, hands on experience of the machine. Having had this experience to bolster confidence, the student will be better able to utilize the concepts learned from this experience when dealing with the more complicated, and abstract world of programming.

The Brunerian influence, on the other hand, stems from the fact that while he, like Dewey, believes that the understanding of concepts emerges from the learners experience, he also believes that the manner in which the internal structure of a topic is expressed or encountered will have a large effect on the quality of the learning experience. In other words, unlike Dewey, he places equal weight on content presentation and the process of learning with regard to the efficiency of a learning experience. From an instructional design standpoint this allows for greater specificity with regard to the content sequence without sacrificing flexibility and the ability to generalize. The Brunerian version of the spiral curriculum is constructed such that the ideas which lie at the heart of any topic area, and which form its skeletal structure, should be presented developmentally in a way that enables the learner to assimilate them through a process of multiple exposures at ever deepening and more complicated levels of understanding and use. As Bruner (1960) states "a curriculum as it develops should revisit these basic ideas repeatedly, building upon them until the student has grasped the full formal apparatus that goes with them." This process of revisiting ideas is not simply one of reiteration. Rather Bruner identifies three levels of experience; the enactive, the iconic, and the symbolic. At the enactive level the learner experiences a concept in as tangible a form as possible, at the iconic level the experience is pictorial or directly representational, finally at the symbolic level it is abstract or indirectly representational. Winer and Schmid (1986) indicate that it is not necessarily the case that in any given sequence of instruction all three stages of development be addressed. Rather it is of the utmost importance, in order for the Brunerian model to be effective, that the learner characteristics be carefully identified with regard to prior experience in the topic area. "In a case where a learner's prior knowledge is sufficiently high that the referents are self evident, the instruction does not have to lead the learner explicitly through the stages. If, however, the learner does not have the capability of generating enactive or iconic representations of the concept, the instruction must either provide them or, preferably, guide the student to discover them via an instructional sequence from concrete to abstract" (Winer & Schmid, 1986, p. 155).

With regard to the design of the materials for the BASIC module it can be argued that the previous modules in the course serve to provide the learner with both enactive and some iconic experiences of the concepts inherent to general computer literacy as it has been here defined, particularly of the sub-concepts necessary to efficient machine utilization and good problem solving. For example the text and computer based tutorials in everything from DOS functions through to database management can be said to provide the learner with an enactive level experience of the machine and it's utilitarian problem solving capabilities without requiring a great deal of unassisted creative activity on the part of the learner. The use of strategically interspersed video based instruction combined with the instruction on graphics programming in the LOGO language may serve to generate iconic level experience of these same concepts but with less emphasis on application and more on the structure and technique of good problem solving itself. The Brunerian model was further utilized within the module itself to provide the learner with some more iconic level representations, such as are to be found in the instruction on flowcharting, as well as symbolic representations, such as are to be found in the form of instruction and exercises pertinent to the writing of listhandling programs. Winer and Schmid (1986) further identify three educational requirements which should be met by any self instructional package utilizing Brunerian theory as a support for it's design:

1) Bypass of remedial content for weaker students must be easily accessible to stronger students.
2) Primary content presentation must be standardized consistent with the supporting learning theory.
3) The materials must be amenable to widespread dissemination (i.e. usable by a large variety of learner types).

The latter point was of particular significance for the design of this module as homogeneity of learner characteristics was not expected within the target population.

In light of these design considerations it must be noted that the content parameters were clearly predefined for this module. Originally they were specified to include;

1) an introduction to general concepts in programming,
2) an introduction to programming tools, such as flowcharts and psuedocodes, and their relationship to structured problem solving techniques,
3) instruction on, and practice in, writing listhandling programs in BASIC.
4) instruction on, and practice in, writing graphics programs in BASIC.

The module requirements were subsequently revised to eliminate graphics programming as it was seen as unnecessarily redundant to the concepts learned in the previous module on LOGO graphics, and because the overall materials for the course were too lengthy to be covered in a single semester.

In addition to these delimiters strong emphasis was placed, by the course administrator on the use of a self instructional format which would minimize the need for classtime input by an instructor and/or lab assistant. The implication was that a strongly integrated multimodal instructional format be used following a linear progression through the required content. How best to organize the content so as to optimize the educational value of this progression was seen as the task of the instructional designer. This task was further restricted by the fact that there was little room for lateral exploration and enrichment due to time requirements (although such material could be referenced for students who wish to do out of lab work).

Lastly, there were demographic considerations with regard to the target population which were seen as possible sources of bias within the evaluation process. These were as follows:

1) There were no prerequisites for the course as entry skills were zero based. However students may have had a variety of exposures to computing at the secondary five and collegial levels creating difficulty in ensuring evaluative accuracy.

2) Similarly some students may have entered the course with advanced typing skills allowing them to move more rapidly through the materials thus reducing the likelihood of them experiencing excessive frustration of boredom and increasing the likelihood of them scoring higher than others not so endowed.

3) Because this course was open to all arts and science faculty students a large variation in academic background and linguistic ability was to be expected (for example, as Concordia University is located in a largely French speaking province many of its students are francophones who are being educated in their second language). As such, because of the semi technical nature of the subject matter and the unfamiliar terminology involved, comprehension may be easier for some students for strictly linguistic reasons.

CHAPTER THREE

Instructional Design

The design process used for this module was primarily based on the model prescribed by Dick and Carey (1978), with some modifications drawn from Romizowski (1981), particularly with regard to the process of selecting domain specific performance objectives and the use of a less linear approach to the identification of relationships among concepts and groups of concepts within the instructional analysis.

Instructional Goals

The materials produced for this module were used as the second stage of instruction in computer programming (the first stage being an introduction to problem solving and LOGO graphics programming provided in the module prior to this one). The broad goals specified by the course administrator were discussed in detail in Chapters 1 and 2 of this document. Stated here as terminal objectives, they are as follows:

1) Given a list of simple problems the student will be able to demonstrate their ability to utilize good structured problem solving techniques by generating a logically valid flowchart using the ANSI standard notation.

2) From the flowchart described in the previous objective the student will be able to demonstrate their understanding of rudimentary BASIC syntactic and semantic structure by successfully writing and debugging a simple listhandling program to solve the same problem.

An instructional analysis was subsequently performed to delimit the concept field, identify the relationships between specific subskills, and determine entry behaviors. Behavioral objectives were written for each of the identified subskills and criterion referenced test items were constructed for each behavioral objective.

Target Population

As mentioned earlier the module was designed for undergraduate students in the Faculty of Arts and Science at Concordia University. Because of the variety of expected student backgrounds regarding computer usage, math, and/or critical thinking skills, entry skills with regard to these areas were restricted to concepts learned in the previous modules of the course. Entry skill with specific regard to previous exposure to the BASIC language were zero-based. However, because of the strong possibility that many of the students, especially the younger ones, may have had previous exposure during high school or CEGEP, the use of a pretest was necessitated (this will be further discussed in the chapter on evaluation).

Because this was to be the last module in the course and because of the relatively high level of education achieved by the targeted population, a better than average ability to correlate and process information from a variety of sources was an assumed characteristic. Better than average ability to concentrate over the four hour required lab time was also assumed. Finally because of the pragmatic value of the material covered as well as the novelty of both the course materials and the mode of delivery high levels of motivation were expected.

Media

Romiszowski (1981) notes that from a simplistic viewpoint "any media of communication can teach any lesson content within broad limits and the only important factors in media selection are the economical and practical constraints that exist in any given situation" (p.304). However he further notes as do others (Weston and Cranton, 1986; Dick and Carey, 1979; Gagne and Dick, 1983) that not all types of media are equally effective and efficient in teaching any given topic and some consideration must be given to matching the use of a particular medium to the complexity of the subject matter and the cognitive levels addressed by the course objectives. With this in mind the earlier mentioned restrictions imposed upon the design process were re-examined.

The module had to be self instructional with a strong emphasis placed on a multi-media approach (particularly the use of computer based instruction). This implied that not only must all instruction, guidance, remediation, and enrichment be contained in the package, but all information pertinent to the access and operation of various media must also be integrated into it. Furthermore this had to be done in such a fashion as to guarantee a smooth transition from one delivery mode to the next without any loss of conceptual continuity within the course content. To minimize the possibility of confusion due to an inappropriate media mix the decision was made to utilize particular media forms only within specific learning contexts. Hence standard linear video, because of it's characteristic of non-interactive unidirectional transmission, was used only as a means to reinforce and/or demonstratively facilitate the iconic representation of concepts. The only content relevant computer based tutorials available were strong on drill and practice with limited tutorial characteristics, as such they were used only for instruction on lower order concepts (such as knowledge and comprehension type concepts identified according to the taxonomy defined by Bloom et al, 1956) regarding the syntactic and semantic structure of the language. Text and embedded exercises were used as a means to teach higher order concepts (those concepts which involve rule retention and manipulation in order to be understood) because of their flexibility in terms of revision and speed of access. Media were selected on the basis of Romizowski's (1981) procedure for selection by rejection whereby items were selected on the basis of suitability, availability, distribution potential, and cost.

Two computer based tutorials ("Edubas I" and "BasicTeach") and one video (one episode of the TVO series "Bits and Bytes") were identified as being capable of meeting the behavioral objectives and were targeted for use. Preliminary evaluation of the computer based tutorials was conducted according to the procedures outlined by Bork (1980) and they were found to be acceptable. The video was chosen because it met the objectives and in order to maintain continuity with other episodes from the same series used in other modules in the course.

Due to time and cost constraints most of the instructional media included was researched and selected from existing materials. As suggested by Romizowski (1981) elaborations were made to selected readings in order to streamline the overall flow of the module and to help ensure conceptual continuity. In all cases where unoriginal text materials were used, original materials consisting of embedded tests, exercises, and connecting texts were added in order to guarantee relevance to the course objectives.

Instructional Strategy

In light of the already discussed value of a spiral curriculum approach, and the value of advance organizers as a means of supplying effective anchoring ideas upon which to build a strong cognitive framework, further strategic considerations were as follows.

An attempt to increase motivation and facilitate learning was made by maintaining a close relationship between the work performed here and the concepts learned in the preceding course modules. This means that the same terminology and conversational style were preserved wherever possible (as the various modules of the course were designed independently by different designers this proved somewhat more difficult than expected). There was an underlying assumption for this module which asserted that what had gone before had been learned. Therefore less of a spoonfeeding approach was required and as a result the module was less detailed in it's step by step explanations. For example it was assumed that by this point students would not need to be reminded as to the nature of random access memory when it was referred to in the instruction concerning how to load a BASIC file. It was further assumed that a more technical approach could be utilized with a minimum of student discomfort.

As is desirable for any self-instructional materials, student participation was not only encouraged, it was required. Here it took the form of embedded exercises and computer based instruction. Personal initiative in problem solving was also encouraged through the lack of feedback in the form of supplied answers to questions (instead the individual was referred back to the specific place in the materials wherefrom the answer may be derived) and through the use of enrichment exercises. These features were further seen as a means by which the student may monitor their own progress and identify for themselves the measure to which remedial work should be engaged in. A five unit design was used to facilitate remediation to specific areas and to avoid initial discouragement due to the apparent quantity of material, as well as to allow for maximum flexibility during the revision process. Key concepts and terminology were placed at the end of each unit to provide study guidance and to minimize review time.

Content Outline

For Bruner (1966) any student only truly grasps the structure of a topic when they understand it in such a way that they are able to meaningfully relate many other things to it. For one to understand structure is to understand how things are related. With this perspective in mind, and in light of the previously outlined considerations, the following module design was implemented. It was divided up into five consecutive units. The first provided a brief overview of the module as a whole including a general goal statement, a rationale, and a description of the content covered in each of the ensuing units. It also contained a quick reference guide through the module as a whole (detailed instructions were provided with each unit). To facilitate the development of appropriate anchoring ideas, as discussed in the previous section, this quick reference guide was designed in such a manner as to reflect the structure and syntax of a BASIC program. It also served as a focusing agent by identifying and situating the new concepts to be encountered through references to concepts previously encountered in the course. Estimated completion time parameters for each unit were also included here.

Unit two consisted of a general look at programming languages and how they relate to the machine. Included was a short reading reiterating some of the concepts encountered in lesson one of the course (such as the machine components, DOS, and binary code) and expanding them by studying their relationship to the differences between high and low level languages, compilers and interpreters, and machine and assembly language. Subsequently there was a video covering similar material to provide reinforcement and a visual (iconic) context for these concepts. Learning objectives were stated as a means for the student to focus on important concepts in the material (this was true of all the units).

Unit three began with a reading covering the basic concepts in flowcharting. This was meant to compliment and focus the concepts encountered in the previous module on approaches to problem solving by teaching the student to map out the decisions and procedures necessary to the solution of a problem in an algorithmic format. The idea was to help the student, through the use of graphic organizers, to be able to identify the logical structure inherent to the solution of any problem, and to familiarize the student with the kinds of logical procedures inherent to programming the computer in any language. An exercise followed in which the student was required to generate two flowcharts for solving preset problems. The second and more difficult of these became the subject of the final programming assignment for the module.

Unit four represented the majority of the work for the module and was divided into two sections. The first section concentrated on teaching the elementary system and program commands necessary for accessing and manipulating programs written in BASIC, as well as the use of the BASIC editor for writing and modifying programs. It consisted of a reading taken from the BASIC manual, which was included with the IBM PC at the time of purchase, and the prepackaged software tutorial Basic Teach. Hands on exercises were provided in the software tutorial. Section two focused on the elementary commands necessary for writing listhandling programs up to and including the use of the DIM(ension) command for array handling (instruction on file management was not included). It consisted of a reading on the use of subscripted variables for array handling and some selected lessons from the Edubas 1 software tutorial.

The final unit consisted of instructions for the final programming assignment and a memory jogger containing most of the critical commands and concepts contained in the module. This acted both as a means of review as well as a quick command reference resource while programming. It was the intention of the author to design the entire module in such a fashion that each of the six units build one upon the other. This was be accomplished through a hierarchical structuring of the content, the visual layout format of text materials, and the strategic placement of both audio/visual materials and hands on exercises. Similarly review and enrichment of concepts previously encountered was provided for by the insertion of focusing agents throughout the module. Identification of the criteria for legitimately establishing both advance organizers and focusing agents was drawn from a level two and three type analysis as prescribed by Romizowski (1981).

CHAPTER FOUR

Method

Evaluation Questions

Romizowski (1981) states:

"Evaluating the outcomes of a course of instruction has three functions:

1) To quantify them, for the purposes of student certification or grading.
2) To measure and improve the effectiveness of the course.
3) To test out some hypothesis about the course structure and the processes which may give us some insights into the general problems of course design." (p. 369)

Formative evaluation of the module was conducted so as to fulfill these three functions. The very pragmatic issues of increasing the student's understanding of programming concepts and their hands on competence at programming were the primary aims of the module. As such there was no evaluative measure specifically directed toward learning influences emerging from the affective domain. Instead measures were directed largely toward the achievement of the learning objectives which were formulated according to the manner scribed by Dick and Carey (1978). These measures did however reflect some concern for noncontent based influences on instructional efficiency; such as those emerging from the demographic considerations mentioned earlier. The specific evaluation questions addressed were as follows:

1) Were the overall learning gains statistically and substantively significant?
2) Were there statistically significant learning gains on each behavioral objective? On a criterion referenced basis which objectives were not met and why?
3) Were there influences on the successful achievement of the terminal objectives due to variations in population demographics?
4) Was the assumption of zero level entry skills regarding the target population legitimate?
5) Were the estimated time requirements needed to complete the material accurate?
6) Was the use of a spiral curriculum effective in enabling the students to master the terminal objectives?
7) Was the media-mix effective in supplying a means for the student to achieve the specified goals?
8) Was the the decision to provide explicitly stated learning objectives perceived by the students to be advantageous to them.

Instrumentation

Preliminary evaluation with regard to the sequencing, the time per module requirements, timing and coherence of the course was conducted as part of a global course evaluation immediately following the pilot session. Because revisions made at this stage were pertinent to the overall course design detailed discussion about them is not relevant here. Instead a simple report is given itemizing the changes that have been made as a result of the preliminary evaluation.

Dick and Carey (1978) suggest that when dealing with self instructional units using selected materials a full scale implementation of their evaluation model need not be used. Instead elements of the first two stages (one to one and small group) were modified and incorporated into the field evaluation. In order to do this the sample population was divided into two levels the second one being a subset of the first. The larger group consisted of the majority of students enrolled in the course at the time of the evaluation; the smaller group, numbering forty subjects, was selected at random from the larger group and tested for representativeness on achievement, through a simple comparison of scores on the final exam. The smaller group was administered a variety of demographic and evaluative questionnaires as well as content related test instruments; the data from which were subsequently used for more detailed statistical evaluation. Evaluation of the larger group was conducted according to the following measures. All evaluation measures are to be seen as strictly formative and oriented toward isolating instructional content in need of revision, re-design, or even eradication. Suggestions to this effect will be forthcoming in the discussion.

1) Embedded test exercises consisting of short answer and multiple choice criterion referenced questions were distributed throughout the module. These were back referenced to the appropriate sections of the materials and were meant to be a means by which to provide feedback to the student as well as to identify areas of difficulty or contradiction in the content.

2) An ongoing, on location, date specific observation commentary was provided by the course monitors through the use of observation sheets (see Appendix E). These were used primarily as a means of identifying common areas of difficulty, as well as inconsistencies in and between the materials potentially leading to confusion and demotivation in the student population. It also provided an unstructured means by which to record student reactions to the material.

3) An evaluation of overall performance was conducted through the use of scores generated from the final exam (see Appendix I) for the course , one third of which was dedicated to testing for

knowledge of the materials found in the BASIC module.

4) One to one interviews with selected members of the target population other than those in the smaller group. This was done in order to broaden the scope of evaluative information without introducing any source of bias on the posttest results. As Dick and Carey (1978) point out, posttest scores obtained from individuals who have participated in a one to one evaluation interview may well be the result of both the instructional effectiveness of the materials and the information gleaned from the personal intervention involved in the interview. Such scores should be avoided when applying an evaluation measure based on statistical methods.

More detailed evaluation of the smaller group was conducted using the following measures:

1) A frequency analysis of demographic characteristics (gathered with the questionnaire found in Appendix F) to identify prominent subgroups within the population (e.g. students with strong mathematics backgrounds, students learning in a second language, etc.).

2) Score data from criterion referenced pretest, posttest (quiz found in Appendix C), and delayed test (final exam found in Appendix I) were collected and correlations and/or chi square

analyses were performed to test immediately evident areas of possible difference between the identified subgroups.

3) A non-parametric evaluation of student response to the materials through the use of a questionnaire given individually upon completion of the materials.

4) Objectives mastery analysis for all identified subskills (see Table 1 for objective/test item matchup with mastery criteria, see Appendix B for list of objectives).

5) Test item analysis was conducted on all test items representing objectives for which mastery was not achieved.

Table 1

Test Items and Mastery Criterion for Subordinate Objectives

Objective	Pretest/Posttest	Mastery Score
1	1a-b, 3	5/5
2	1c	1/1
3	1d	1/1
4	2	3/3
5	4, 5	4/4
6	6j	1/1
7	6a-i	8/8
8	9	2/2
9	8a-h	5/5
10	7a-e	10/10

(continued on next page)

Objectives evaluated by Assignments		
11	editing assignment	100% of requirements
12	editing assignment	100% of requirements
13	flowcharting assignment	100% of requirements
14	programming assignment	100% of requirements

Note. Scoring criteria vary for each test item. Scoring codes are detailed in Appendix J. Appendix B lists the objectives. Appendix C contains the pretest/posttest.

CHAPTER FIVE

Results

This chapter opens with a short description of the considerations and minor revisions which were noted and/or implemented as a result of observations made by all the members of the design team prior to and during the initial piloting of the course. Subsequently, the results of the data, gathered with each of the evaluation instruments during a full scale field trial, are presented in detail. A discussion concerning the implications of these results is presented in Chapter 6.

Preliminary Developments

Revisions made as a result of an examination by the course administrator and the other instructional designers on the course development team were all either typographical, syntactic, or organizational with regard to instructional consistency. As such they had little impact on the overall design and will not be discussed in detail here. However some concern was expressed by members of the group as to the clarity and complexity of the unit on flowcharting. As a result this unit was flagged for particularly close scrutiny during the subsequent evaluation process. Some concern was also expressed by the instructional designers that the language used in the selected readings was too technical. It was suggested that either they be rephrased or new readings selected. Again this became a subject of particularly close scrutiny.

The course was piloted and a global evaluation was conducted regarding it's overall design. As a result it was noted that the general timing of the course had been seriously misjudged. The BASIC module being last, suffered the most from this with a completion rate of only 15% of the total enrollment. Less than 45% managed to complete even the first unit. While some changes were implemented in the internal structure of some of the earlier modules, as well as to the overall sequence of the course, the only major modification made to the BASIC module was the removal of the initially proposed unit on graphics programming. As mentioned earlier, this was done because graphics programming had been introduced in the LOGO module and was seen as being largely redundant here.

The course was run for a second time the following semester with a resulting increase in the completion rate. This time 35% of the total enrollment completed the materials with 62% finishing all the materials for the BASIC module up to but not including the final assignment. At that time an attempt was made at a formal evaluation of the module, however the small group selected at random for more detailed evaluation contained only six people who completed the course. This proved problematic as the proposed evaluation measures relied to a significant degree on materials completion. Consequently this was considered to be too small a sample upon which to base comprehensive data collection. Furthermore any attempt to utilize the entire group of students who completed the course as the sample would represent a danger of biased results derived from data collected only from high achievers. Hence the decision was made to abandon the evaluation at that point. Instead more minor revisions were made to the course as a whole in an attempt to better meet the time requirements. Minor changes were made to the BASIC module on the basis of information gathered from the observation sheets in an attempt to streamline it. In addition some strategically placed motivational comments on the importance of finishing the materials were inserted. Also implemented was an ongoing weekly posting in the classroom as to what the students should be working on during each lab time in order to remain on schedule. Students falling seriously behind were actively encouraged to seek extra assistance. The course was run again with a resultant 85% completion rate consequently enabling a detailed evaluation of the BASIC module.

Formal Evaluation

The following procedures were carried out on the entire student population enrolled in the course at the time of the evaluation.

Observation Sheets

Daily information from the lab assistants was gleaned through the use of observation sheets (see Appendix E) which were dated and available in the lab at all times. They yielded interesting information on specific recurrent difficulties being encountered within each unit of the module. Similar comments for each unit were sorted into groups and noted for their frequency. Table 2 (see next page) summarizes the information gathered.Comments which occurred more than three times were considered to represent a definite flaw in the materials. Comments occurring three or less times were further sorted into three categories. These were as follows:

1) Indicators of obvious typographical, syntactic, and organizational type errors (items 1a,1b,1c,3b),
2) Items with a high probability that they were due to learner error (items 3a,4a,4c,4g,4h,5c,5g),
3) Items with a high probability that they were due to vague or confusing instructional materials (items 3c,4i,5b,5e,5h).

Categorization was made on the basis of type of error and the complexity of the concepts involved in the field of error. Hence because items related to the first category, regardless of their low frequency, were easily verifiable, they were subject to immediate revision. The decision to place items in the second and third categories was made on the basis of three criteria: 1) their relative frequency, 2) the strength of their relationship to identifiable errors

Table 2
Summary of Observation Sheet Comments

Unit	Comment	Frequency
1	a) I don't know if the reading is mandatory.	1
	b) There are inconsistencies between the naming and numbering of readings and the way they are listed in the guide.	1
	c) Onscreen instructions for the software are different from the ones found in the booklet	1
2	No comments	
3	a) I don't understand the instructions for the flowcharting exercise	2
	b) There seems to be some contradictions between the text and the "Building a Snowman" example, with regard to the use of ANSI standard symbols.	2
	c) I don't understand the value of working from a maximum to a minimum value when flowcharting	1
4	a) I don't know how to erase a file	1
	b) I don't know how to use the CONTROL/ BREAK key combination	15
	c) I don't know how to save a file	2
	d) I don't understand how to dimension an array	7
	e) I don't know how many lessons to complete in the BASICTEACH software	4
	f) The brackets surrounding the conversion formula in the second problem of the editing exercise were included in the program	5
	g) I don't understand how the RENUM command works	1
	h) I don't understand when to use the NEW command	1

(continued on following page)

	i) I don't know how to return to the EDUBAS software tutorial from BASICA	2
5	a) I have problems using the READ/ DATA statement pair	5
	b) I have problems using the FOR/ NEXT statement pair	2
	c) I have problems using the IF/ THEN statement pair	1
	d) I have problems using the DIM	5
	e) I have problems using the INPUT statement	2
	f) I have problems using the RESTORE statement	8
	g) I don't know how to reload a file	1
	h) I have problems with the GOSUB/ RETURN statement pair	2

or fuzzy descriptions in the material (the weaker the relationship the more likely the item would be included in the second category), and 3) the complexity of the concepts involved in the error (derived from the level of abstraction, as per Bloom's (1956) taxonomy, required to work with the concept and its location within the instructional analysis). Hence, items in the second category were seen as weak indicators of errors in the instructional design and in need of strong confirmation from other aspects of the evaluation before triggering revision. While items in the third category were seen as strong indicators of design errors and, upon investigation of the pertinent areas, likely to precipitate some form of change in the materials.

Interview Results

One to one interviews were conducted with ten students in which the following questions were addressed.

1) What are your overall comments on the value of this module?
2) Which areas or concepts did you find difficult? Why?
3) Which areas did you find too easy? Why?
4) Which areas did you enjoy the most? Why?
5) Did you find the transition from the previous modules to this one difficult? Why?
6) How did the lesson sequence help or hinder your learning?
7) How did the video segments help or hinder you in understanding the overall content of module?
8) How effective were the reading materials affect you? Did you find them interesting?
9) How did the course guide in UNIT 1 help you or hinder you in your progress through the materials?
10) Were the exercises and embedded tests relevant and useful to you in assessing your understanding of the materials? How so?
11) Were the key concepts and terminology supplied at the end of each unit useful as a means of reviewing the material? Were they comprehensive enough?

Overall the module was perceived favourably although three of the ten interviewees complained that the material was too difficult for a self instructional format. They argued that while the materials themselves were not difficult to work through the solutions to the subtler problems which arose when working alone on a program or exercise could have been more easily understood if an instructor were present. Indeed, as was evident from the evaluation questionnaire and the observation sheet data, most of the people in the course resorted to querying the lab monitor as a means of relating the information presented in the materials to the problems encountered while working on the final programming assignment.

None of those interviewed objected to the sequencing of the materials although five of the ten thought that flowcharting would be better placed in the preceding module on problem solving.

The video segments used were perceived as a good way to break the tedium of otherwise uninterrupted reading of dry text material. Although eight people commented upon the video as being at times childish they all agreed upon the overall validity and utility of the content covered therein.

The reading material was viewed almost universally (one person found them enjoyable) as a necessary evil; the content being viewed as valid but the text too dry and technical to be enjoyable. Three people commented that, while they found the content for each reading important and internally coherent, they also found the transition from one reading to the next rough and discontinuous generating a certain amount of confusion.

The module outline in UNIT 1 was perceived as useful insofar as it provided some perspective on why learning to program was a valuable endeavor. However the attempt to design the course guide as reflective of BASIC syntax went largely unnoticed (two people acknowledged it) and all ten people said they had found it confusing after using the page by page instructional format utilized by the other modules in the course.

Overall the exercises and embedded test questions were perceived as helpful insofar as they provided a means for self assessment. Four people commented upon the usefulness of being able to refer to the appropriate places in the materials where the answers could be found rather than being simply provided with the correct answer. Two people argued that this was a deterrent as they simply didn't bother to look up the answer due to feeling pressured for time.

The flowcharting exercise and the final programming assignment were seen as being at once both the most difficult and the most enjoyable aspects of the module. Debugging was seen as the most time consuming and difficult aspect of the final assignment. All ten people believed that their difficulty stemmed from the sudden lack of further support from the materials. Comments such as "feeling suddenly alone" or "cast adrift" were typical despite the provision of a memory jogger and some quick reference materials which they felt were confusing and not comprehensive enough. People said that they felt themselves unprepared psychologically for these less "cut and dried", and less guided problem solving activities. However, they all felt that they enjoyed them most, precisely because they afforded them the freedom to be somewhat creative within very clear parameters.

The transition from the previous module (LOGO programming) into BASIC was perceived as difficult with eight people commenting to the effect that they felt more understanding was assumed of the student than in previous modules. Six people reported difficulty with the transition from the very detailed step by step approach found in previous modules to this module's less directive presentation of information. Five of the six said that this was largely due to the fact that they had become used to the other style of presentation and it took some time to adjust.

The only changes suggested were to resituate the material on flowcharting into the problem solving module, and to improve the readability of the text materials. Four people expressed a need for more time in order to be more relaxed about finishing the assignments. Two people suggested that the module should be extended and made into an independent course.

Sub-Group Evaluation

Detailed evaluation of a representative sample selected from the total student enrollment was conducted and is described below.

Sampling Procedures

Because various statistical tests were used to evaluate the data collected from this group (and because this was not a small group evaluation in the sense that Dick and Carey (1978) use the term) a slightly larger sample size of forty students was selected rather than the ten to twenty suggested for small group evaluation. The group was selected at random from the larger group through the use of a random number generator. The numbers thus generated were matched with the corresponding record numbers of students in the class database, maintained for record keeping purposes (repeated numbers were ignored). The subgroup was tested for representativeness at the end of the course by a comparison between their delayed test (final exam) scores and those of the larger group which showed no significant difference (unpaired t- test, $t = .129$) at $p \geq .05$.

Pre, Post, Delayed Test Results

It was expected that the pretest would produce a distribution of scores at the low end of the scale, and that scores of the posttest (quiz), delayed test (final exam), and the sub-section of the delayed test pertinent to the would emerge at the top end of the scale. Fulfillment of this expectation is indicated in Table 3 by the results produced from an analysis of descriptive statistics calculated on the scores collected on all three tests and converted to percentage scores. For additional information the scores were compared for statistically significant differences. The pre and postest score gain (paired t-test, t = 27.51) and the post-delayed (BASIC only) test score gain (paired t-test, t = 9.738) were both found to be significant at $\underline{p} < .01$.

Table 3

Percentage Scores on Pre, Post, and Delayed Tests

	Mean	Range	SD
BASIC module			
PRE	7.4	0-16	5.61
POST	67.85	50-96	11.49
Delayed Test			
Basic content	80.21	62.86-97.14	9.04
Overall content	82.45	66-95	6.7

Demographic Evaluation

A background questionnaire (see appendix E) was distributed to the students in the subgroup. Information was collected with regard to age, sex, previous experience, typing skill, student status (regular or otherwise), mother tongue, number of courses taken simultaneously, math background, faculty, and department.

Frequency analysis on the data collected yielded the following results:

1) Out of the total population 90% was between 18 and 30 years of age.

2) Typing skills varied from very poor to very good with 47.5% of the sample population claiming moderate skill.

3) Most of the students (97.5%) were Montreal residents living within 30 minutes travel time to the university.

4) The majority of the students (75%) were enrolled as full time regular day students, 25% were enrolled as independant students taking only one or two courses for the purpose of upgrading specific professional skills.

5) English was the predominant language (72.5%), 17.5% were French, and the rest were from a variety of other nationalities.

6) Only 5% were taking 5 other courses simultaneously, 27.5% were taking 4 other courses simultaneously, 35% were taking 3, 25% were taking 2, and 7.5% only 1.

7) Most of the students (75%) were in the faculty of Arts and Science, 7.5% were Fine Arts students, 10% were independent, and 7.5% were offcampus.

8) Students came from a broad range of thirteen different departments yielding no specific

subgroupings large enough to be meaningfully compared. They were subsequently grouped into four categories; 1) hard science departments (cell size = 10), 2) soft science departments (cell size = 10), 3) humanities and fine arts (cell size = 12), 4) Independant (cell size = 8).

Chi-square and/or t-test analyses were performed on all combinations of interest in an attempt to identify significant differences either in performance on the terminal objectives or in overall learning as represented by scores achieved on either the post or the delayed tests. No such differences were discovered (see appendix H for details) and as a result no further multivariate statistical analyses were pursued.

Evaluation Questionnaire

Evaluation of item response frequency was performed on an attitudinal evaluation questionnaire (see appendix D) in an attempt to identify an overall picture of the student response to the materials. Observations derived from the results were as follows. Percentages where listed indicate the number of students who fall within the range of agreement to strong agreement.

1) The mean time to completion of the materials was 15.15 hours.

2) Overall the product was ranked as good (60.9%) to fair (29.5%), with the main criticisms being the excessive length and lack of entertainment value,

particularly with respect to the readings.

3) Overall the objectives were perceived to be adequate (60.97%) to clearly defined (29.26%).

4) Based on the responses on the Likert scale used in items 4-A through 4-O, stating the objectives for each unit was perceived as a positive means by which to:

- identify what was needed to be known in order to complete the materials (items A - 82.91%, J - 70.72%).
- identify how well each task must be done in order to meet the requirements of each exercise (items B - 82.91, M - 80.49%, K - 85.36%).
- self evaluate progress through the module (items C - 80.47%, I - 73.17%).
- identify exactly how each aspect of the lesson would be evaluated in order to facilitate study (items F - 85.37%, O - 78.05%, L - 92.68%).
- identify areas in the materials which conflicted with the achievement of the objectives (none were noted) (items G - 83.95%, D - 82.91%, N - 87.80%).

5) The content outline was identified as detailed enough detail and logically presented with 85.36% of the sample population identifying

specifying one, or the other, or both of these characteristics in item 5.

6) The activities identified as being the most enjoyable were, in order of preference, A) the final assignment (63.41%), B) the flowcharting exercise (17.07%), C) the editing exercises (9.76%), D) the computer based tutorials (4.88%), E) the videos (4.88%), F) the readings (0%).

7) Program debugging was considered the most difficult activity, with 55% of the subjects commenting to this effect.

8) Practice and embedded exercise items were perceived to be adequate (73.17%), with 14.63% thinking there were too few, 4.88% thinking there were too many, and 7.32% arguing that they were unrelated to the instructional sequence. Based on an analysis of the free form responses to item fourteen, it seemed evident that some confusion was experienced concerning the flowcharting exercise particularly with regard to the specific requirements.

9) Of the test population 92.68% thought placement of this module at the end of the course was appropriate.

10) Of the forty people evaluating, 55% indicated that overall they found the activities to be relatively difficult, 55% found them definitely interesting,

22.5% felt there was too much to do; 30% felt there was too little, 13 said there was too much theory, 10% thought the module was too easy, and 10% said they were bored. Assistance was needed at one time or another by 70% mostly due to the difficulty of the material (31.14%) or the specific activity required (26.83%), 24.4% did not require assistance, and 4.9% did not answer the question. Some help was required due to vague or confusing directions (34.15%), but little was needed due to confusing directions or problems locating required materials.

11) Overall only 17.07% had problems following instructions, only 12.19% had difficulties understanding the diagrams, and nobody had problems accessing and using the audio-visual materials.

Objective by Objective Analysis

Subordinate Objectives 1 to 11. Performance of each student on each objective was classified as either mastery or non-mastery. In order to maximize the information potential of test item analysis for those objectives not mastered, test items on different objectives were evaluated according to a variety of specifications depending on the complexity of the response required. Table 1 details the test items and their mastery score for each objective. Appendix J details the evaluation code for each item. Figure 1 summarizes the results of the overall student performance on each objective.

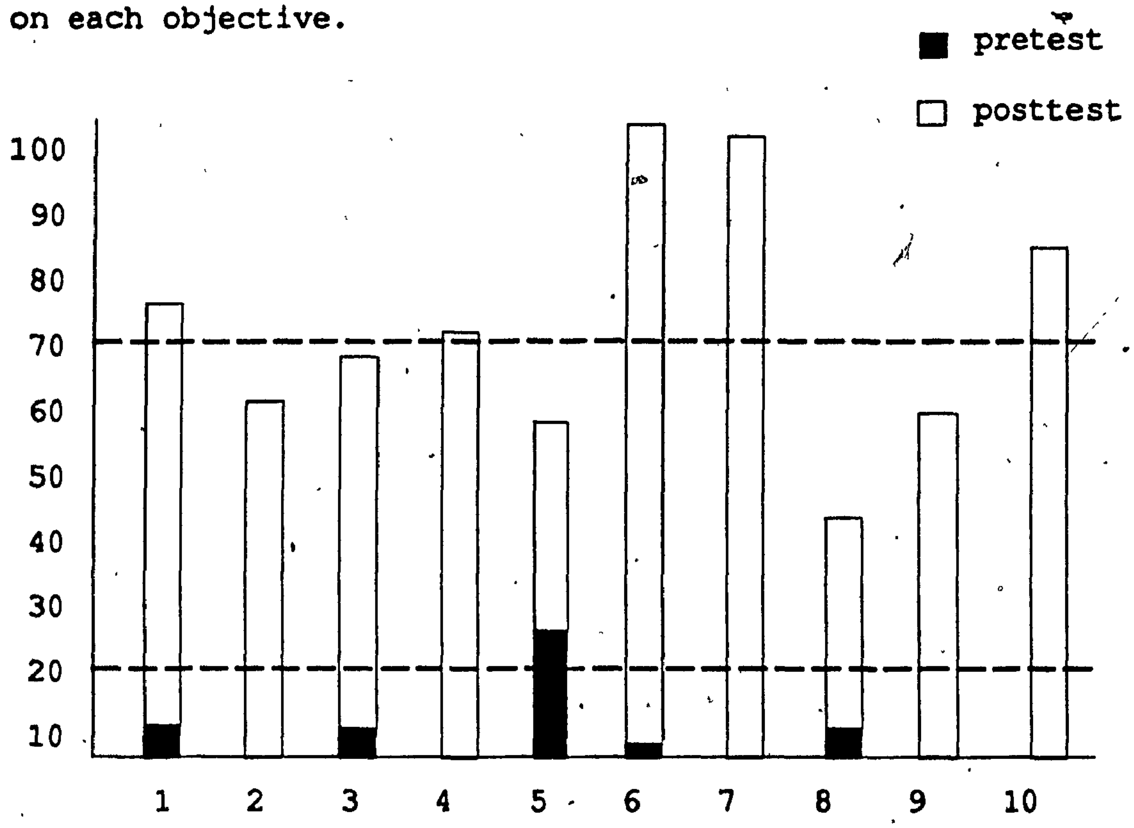


Figure 1.

Percentage of students mastering subordinate objectives 1 through 10.

Note: See Appendix B for list of objectives, Appendix C for sample pretest/posttest, and Appendix J for test item evaluation codes.

A non-parametric statistical analysis was subsequently performed using a Sign test for objectives measured by test items yielding a binomial distribution (correct or incorrect) and the Wilcoxon ranked-sign test for objectives measured by test items which yielded integer scores greater than one. Gains on all objectives were significant at *p*<.01. Appendix K details the test results for each objective.

Test item analysis was conducted according to the procedures prescribed by Berk (1980). Because of the assumption of zero based entry skills and the extreme variety of learner types and characteristics inherent to the target population, a somewhat liberal comparative standard of 20% - 70% was adopted as a means to identifying objectives in need of analysis. Thus the expectation was that less than 20% of the subjects would demonstrate mastery on each objective in the pretest while 70% or better would do so on the posttest. As is evident from Figure 1, objectives 3, 4, 5, 8, and 9 did not meet these standards. Detailed results of the test item analysis for these items can be found in appendix G.

Subordinate Objective 11. Objective 11 was tested for through the use of an embedded editing exercise and was evaluated on the basis of a completed or not completed criteria. Mastery was 100% on this objective.

Terminal Objectives 12 and 13. Both terminal objectives (see appendix B) were tested for through the use of in-class projects. Evaluation of these assignments yielded the results listed in Table 4. Mastery on both objectives was 100% with some secondary observations.

Table 4

Analysis of the Terminal Objectives

Objective	*Results of Analysis*
12	100% mastery BUT 22.5% produced a logically valid flowchart with the correct use of the ANSI standard symbols, however there were logical errors in the solution to the problem itself.
13	100% mastery BUT 47.5% produced a debugged, logically and syntactically valid program with logical errors in the solution to the problem itself.

CHAPTER SIX

Discussion

Information, derived from an analysis of the various data gathered, indicated that on the whole the module was successful in meeting the objectives set for it. However some areas of weakness were identified and the pertinent materials were marked for revision. This chapter details the implications of the previously itemized results with respect to each of the evaluation questions listed on pages 34-35, and identifies specific weaknesses. Suggestions for revisions intended to address and correct these weaknesses will be discussed in chapter seven.

Evaluation Question 1: Were the overall learning gains statistically and substantively significant?

Significant increases in the performance results between the pretest and both the posttest and delayed test for the entire sample population would indicate that overall learning gains were in fact achieved. This conclusion is further supported by the unexpected occurrance of an overall significant increase in score between the posttest and the delayed test which was presumably a result of the students' opportunity to study between the BASIC quiz (posttest) and the final exam (delayed test).

As the design of the module was heavily dependent upon mastery of the objectives for its evaluative measures, significant differences in results on test performance were expected and received (see Table 3). Furthermore, objective by objective analysis, of performance by the small group on criterion referenced items in both the pretest and posttest, indicated a significant increase on all the performance objectives tested for (see Appendix K). However the assumption was made that performance objectives which were being effectively addressed by the instructional materials would show mastery performance by less than 20% of the group before instruction and more than 70% after instruction. While all objectives met the less than 20% pretest standard, indicating some accuracy in the predicted level of entry skills, the 70% posttest standard was not met by objectives 2, 3, 5, 8, and 9. Subsequent item analysis revealed a number of possible reasons for this failure which will be further discussed in the objective by objective analysis.

Substantively significant increases in learning were evidenced by performance on the three assignments (objectives 11 through 14). All those evaluated showed no ability to meet the requirements before instruction and all of them showed 100% mastery after instruction.

Evaluation Question 2: Was the learning gain statistically significant for each objective? If not, on a criterion referenced basis, which ones were not met and why?

Objectives 1 - 10. As mentioned earlier, statistically significant increases were achieved in learning gains on each performance ojective, however the 20/70 comparative mastery standard for objectives 2, 3, 5, 8 and 9 was not met. Points of possible explanation for this outcome are as follows:

Objectives 2 and 3: Item analysis revealed that much of the difficulty here may be due to a lack of clarity in the phrasing of the test item. Typically the question evoked responses which were relevant only to the difference between high and low level languages (objective 1) and not to the advantages and disadvantages of each. Good overall achievement on objective 1, which involved a similar level of comprehension on a closely related topic, would indicate that poor performance on objective 2 was unlikely to be caused by an inability to understand the material as presented. It is more likely to be due to insufficient specificity in the test item. Dick and Carey (1979) warn that criterion referenced test items must be specific, unambiguous, and addressed to only a single objective; while this is generally a contentious issue, they do so precisely in order to avoid this type of difficulty. The author originally felt that because objectives 1, 2, and 3 were so closely related, combining their evaluation into a single test item was a reasonable way to test, not only comprehension, but also the student's ability to abstract and synthesize by explaining one newly acquired concept in terms of another. The results suggest that the test item failed to accomplish this end and should be rejected in favour of separate items for each objective.

Another consideration regarding the poor performance apparent on this objective is that, while some reference to the concepts inherent to these objectives was made in the introductory video, they were dealt with most substantively in the reading material found in Unit 2. Responses on both the Interview and Evaluation Questionnaires identified the reading material as the least enjoyable and most difficult aspect of the module to deal with. Preliminary observations from the other instructional designers on the course team also identified these materials as being problematic due to overly-technical language. Furthermore, the concepts particular to these objectives were neither referenced nor in any other way reinforced in any of the practice exercises other than in a direct format through the use of embedded test questions (for which the students were in no way held accountable). The design assumptions for these objectives were that the students would already have a solid understanding of related sub-concepts (e.g.: binary code, programming languages, and disk operating systems) as a result of instruction in previous modules. It was felt that the acquisition of the new concepts concerning the various attributes and distinctions between High and Low level langagues would be relatively easy and in need of minimal explanation. The poor results yielded by these objectives indicate that either this assumption was in error and hence some adjustments to the instructional analysis was required, or else the analysis was sound and the error lay either in the instructional strategy, particularly with regard to the mode of delivery, or in the means of evaluation. The positive commentary yielded by the Evaluation and Interview Questionnaires on the overall congruency and consistency of the material as well as the lack of overt difficulty expressed in the observation sheet would indicate that the latter was more likely to be the case.

Objective 5: The only expressed difficulty identified by the various measurement instruments, which could be related to poor performance on this objective, was again the tendency for students to lose their attentiveness while working through the readings due to overly technical and tedious language. On the whole Unit 3 generated comments concerning confusing requirements and seemingly contradictory instructions; as evidenced by the results on Items 8 and 10 of the Evaluation Questionnaire, responses to Questions 1 - 4, and the changes suggested during the interview. Test item analysis showed that students had little difficulty identifying the flow chart from its definition. However the term pseudocode remained unfamiliar. It is assumed here that it is the lack of retention on the concept of the pseudocode which accounts for poor performance on this objective. As this concept is dealt with in far less detail than is the concept of a flow chart, and again is not reinforced in any of the practice exercices, it is the opinion of the author that the concept is lost. Moreover the loss is most likely due to the strong emphasis placed on learning to flowchart and the combination of difficulty generated by having to adapt to a new less-directive style of instructional presentation, confusion in the instructions, and the dryness of the language used in the readings.

Objectives 8 & 9. The problems on these objectives would seem to be the result of the same textual problems outlined for Objectives 2, 3 and 5, with the added possibility of error in the instructional analysis. The instructional analysis indicates a direct hierarchal relationship between an understanding of simple Basic commands and the use of arrays, without any intermediary sub-concepts (see Appendix A). The excessive number of queries on this subject from students reported on the observation sheet, combined with the commentaries found in both the responses to the Interview and Evaluation Questionnaire concerning the difficulty of the program and exercises, indicate an overly large increment in learning is required to master the objectives. It would seem that the instruction provided is not suitable to meet the needs specified, insofar as it lacks both continuity and contiguity with previous instruction. Test item analysis revealed that while only 42.5% of the population achieved mastery on objective 9, 77.5% scored at least 1 mark indicating a good general ability to identify array handling as being related to the representation of data in a matrix. These results combined with a positive correlation between performance on Objective 8 and students with a good math background indicate that instruction on the concepts inherent to matrix behaviour is probably the missing element in the instructional analysis.

Analysis of item 8 on the pretest/posttest indicated a good general ability to comprehend the basic syntax and characteristics of an array. However more complicated representations showed poorer performance. This may be an indication of the effectiveness of the materials in imparting simple concepts, such as rudimentary syntactic form (which involve only lower level cognitive processes such as memorization). However they would seem to be less effective in evoking the higher order processing required to apply the principles of array handling to the process of problem solving.

Objectives 11 - 14. Each of these objectives, which include the terminal objective, involve the production of a pre-specified work assignment. All students reached mastery on all of the objectives indicating that while the errors found regarding previous objectives were serious enough to merit correction, they were not so serious as to cripple the students' ability to perform on the terminal objectives. Mastery was 100%, but as was noted in Table 3 of the previous chapter, there was some observation which indicated that some students suffered difficulties. As the difficulties encountered were relative to lower order concepts, as described by Bloom (1956), it is possible these would be eradicated by an effective response to the problems cited in reference to Objectives 1 - 10.

Evaluation Question 3: Were there influences on the successful achievement of the terminal objectives due to variations in the population demographics?

The lack of any significant results being indicated on the battery of chi square and t-tests performed comparing various demographically defined sub-groups, indicates that demographic characteristics are not an issue with regards to the performance of the module (See Appendix H). This generated some surprise for the author as at least some difference was expected to be evident between students with a relatively strong math and/or computer background and those without. In retrospect, given a near zero drop out rate for the course, it is possible that the lack of evidence forthcoming was due to the effective orientation to computing and problem solving acquired from the materials worked on in previous modules. Another area of surprise was the apparent lack of difference between students operating in a second language and those not. One possible explanation for the lack of difference may be that such students, particularly those whose mother tongue was French (these students have other options available to them), by virtue of the fact that they have chosen to undertake the difficult task of studying in a second language, are more likely to be high achievers and highly motivated. Lastly, contrary to expectations levels of typing skill showed no difference. In retrospect this was assumed to be the case most probably because speed of input is of little advantage when an individual is impeded with unfamiliarity with the structure and syntax of the input.

Evaluation Question 4: Was the assumption of zero level entry skills regarding the target population legitimate?

Extremely poor performance on the pretest, administered to the sub-group, would indicate that in fact the assumption of zero level entry skills for the population was legitimate (an entry skills test was not administered as it was assumed that if the student had completed the previous modules in the course then they must have met the necessary entry requirements). The assumption concerning the sophistication the students should possess with regard to their ability to integrate new concepts into their existing cognitive framework would seem to be supported by the high level of achievement on all terminal objectives. As predicted, motivation levels remained high as indicated by regular attendance and uninterrupted activity during any given lab session.

Evaluation Question 5: Were the estimated time requirements needed to complete the module accurate?

Apart from the difficulties encountered during the pilot run of the program, with regard to the overall timing of the course, the timing of the basic module itself was extremely accurate. Estimated time for completion was four lab sessions totalling 16 hours and the overall average time it took actual students to complete it was 15.5 hours, as indicated by the evaluation questionnaire (See Appendix D). This reinforces the belief of the author that most of the difficulties encountered during the running of the pilot were due to timing errors in the early stages of the course and not during the basic module itself.

Evaluation Question 6: Was the use of a spiral curriculum effective in enabling the students to master the terminal objectives?

While it is impossible to state that the successful performance results on the module's terminal objectives indicate in any way that the use of a spiral curriculum outperforms any other style of approach (no comparisons of this nature were made), it is safe to say that insofar as it has been implemented, it has proven to be an effective mode of delivery. Furthermore there was a tendency for students to perform better on objectives which involved an evident return to previous concepts, than they did on objectives for which this return was less evident. This phenomenon goes a long way to support the notion that the use of a spiral curriculum was a contributing factor to the level of success achieved on those objectives.

Evaluation Question 7: Was the media mix effective in supplying the students with a means to accomplish the specified goals?

Weston and Cranton (1986) identify three components of instructional materials as 1) the delivery system, or that which is to convey, 2) the message, or that which is to be conveyed, and 3) the form or condition of abstractedness, the degree to which the message is represented in reality at the time of the conveyance (e.g. the use of a concrete model as opposed to a verbal description). It is with regard to these three components that the success of the media mix used in the module is herein examined.

Given the overall success of the module in meeting the terminal objectives, it would seem that the general effectiveness of the media mix was acceptable. However in light of Romiszowki's comment, noted earlier, that any media can be made to teach any subject, the appropriateness of the mix bears some examination. Data collected from the observation sheet, evaluation questionnaire, and the personal interviews showed very little evidence of overt difficulty with the relationship between the subject matter (message) and the delivery system. Typically learner responses, as indicated by the opinions expressed during the interviews, were congruent with those anticipated at the time of the media selection.

The only commonly expressed difficulty was with the overly technical nature of the reading material. Romiszowski

(1981) identifies two types of communication which occur during the execution of an educational module; 1) informational communication, which involves a unidirectional transmission from the sender (teacher or package) to the receiver (learner), and 2) instructional communication, which involves a bi-directional link in which both messages and feedback are transmitted back and forth between both parties. As Weston and Cranton (1986) note both types have their place in education; the former being more potent in the initial dissemination of information, particularly with regard to lower order concepts and seminal expressions of higher order concepts; and the latter being more effective as a means to facilitating growth in the learner's ability to synthesize and abstract from concepts gained from informational presentations. In the case of this module it was hoped that the concepts presented in the negatively perceived informational format of the readings (typically deemed a necessary evil by the students) would be balanced and clarified by the interspersion of more entertaining delivery forms of informational communication, such as the videos, and the more interactive instructional format of the computer assisted instruction segments. It can be assumed by the overall success rate and the positive results on the evaluation questionnaire for items 12, 13, and 14 that this attempt was at least partially successful. However confusion generated by the difficult language, and the sudden switch to a different textual layout from that used in other modules, was noted in the results from all three of the above mentioned evaluation instruments. This may indicate that, while the delivery system was appropriate, the form of the delivery was somewhat lacking in power. Poor performance on pretest/postest items soley addressed by difficult textual passages (definition of psuedocode and the use of arrays) would further support this position.

Generally the positive response on all three pertinent evaluation instruments surrounding the ease of access and the use of the various media would suggest that the overall organization of the delivery system was efficient. However some confusion expressed on the observation sheet concerning instructions as to what to do next may imply again that some changes are required in the form of the delivery; this time with respect to the self administrative characteristics of the module. As mentioned earlier an attempt was made to provide the students with an advance organizer in the form of a course guide written in the format of a BASIC psuedocode and contained in a separate booklet. In the interest of clarity many of the course instructions detailed in this guide were reiterated at the beginning of each unit; again each unit was contained in a separate booklet. The lab monitors frequently noted informally that students were getting bogged down in the process of having to be continually cross referencing between many physically distinct sources of information (add to this the fact that several of the readings were also separate from the unit booklets). Furthermore students were encouraged to communicate with each other and to work together to solve problems encountered when attempting the assignments. This led to an undocumented, but often observed tendency for individuals to ask their neighbors what to do next rather than read the instructions carefully. It is the opinion of the author that this combination of characteristics in the form of the administrative delivery system, while each valid in its own right, served in reality to create unnecessary (insofar as it had to do with the administration of the module and not the subject matter itself) dissonance for the learner. The importance of disruptions of this nature cannot be underestimated, for until they are corrected it will be impossible to assess how much of any individual's inability to meet a given objective is due to weakness in the presentation of the subject matter and how much is due to the influence of these disruptions.

Evaluation Question 8: Was the decision to provide the learner with explicitly stated learning objectives perceived by the students to be advantageous to them?

Responses on the Likert scale, used in items 4a through 4o on the evaluation questionnaire, supplied the only clear evaluation results pertinent to this question. The items were deliberately designed and sequenced in such a fashion that, given a favourable attitude on the part of the student, responses with a high value on the Likert scale would be generated for the first eight questions and responses with a low value would be elicited on the remaining seven. Duplication of item content was to be found in both formats (positive or negative phrasing) for each of the five categories listed below: thus creating a built in check for irresponsible answers. No instances of such behaviour were noted and the data gathered indicated an overwhelming appreciation of the stated objectives as a means to:

1) identify exactly what they needed to learn;
2) identify exactly how well they needed to perform each task in order to complete each assignment acceptably;
3) evaluate their own progress;
4) identify weaknesses or confusing aspects of the materials; and
5) identify exactly how they were going to be evaluated.

It is to be noted however that the measure used was a simple evaluation questionnaire and was in no way powerful enough to indicate that stating the objectives was in actual fact a contributing factor in the students success. This issue remains a contentious one in the field of Educational Technology (Gerlach et al, 1976) and represents a broad scope of research questions; argument about which is well beyond the scope of this work. (Note: Stated objectives can be found at the beginning of each unit in the pilot version of the module found in Appendix L).

CHAPTER SEVEN

Revisions and Recommendations

The overall effectiveness of the materials produced by the author has been shown to be acceptable given the specifications originally outlined by the organizing committee for the course (see Chapter 1). The design has proven to be educationally sound with regard to enabling individuals within the target population to, using a self-instructional format, develop the skills detailed in the terminal objectives according to the criteria specified. The materials used would seem to be appropriate for the specified population in spite of them being identified as somewhat dry and overly technical in places. Furthermore given the heterogeneity of the target population and the typical transportability of a modular design it is highly likely that the materials could be used in any of a variety of settings meeting the basic population requirements (ie. office and industrial training scenarios).

In spite of the overall success of the module several weakness have been identified (see Chapter 6) and are indicative of areas in need of revision. These are as listed below:

1) Dry and overly technical language used in the readings generating reduced interest and/or confusion causing a tendency to skip over the reading materials too quickly.

2) A possibly inappropriate placement of Unit 3 on flowcharting.
3) The ineffective use of a multi-booklet format.
4) The ineffective use of a psuedo-code format for instructions in the introduction and course overview.
5) Clarification and/or stronger emphasis placed on the value and use of arrays.
6) Clearer description of the relationship between good structured problem solving, programming, and the use of subroutines.
7) Weak test items for objectives 2,3,4,8, and 9.
8) Discordance between the style of presentation used in this module and that used in previous ones.

In order to facilitate specificity with regard to the revisions intended to address these weaknesses and the materials they relate to, recommendations will be made first with regard to those changes which effect the overall design, and then with reference to each of the five units in the module.

Changes in the Overall Design

Three of the above listed revision items are pertinent to the overall design of the module; items 1,3, and 8. It is the author's view that revisions aimed at solving the problem referred to in item 8 (discordance between presentation styles) should be no more than epiphenomenal to the revisions targeted at the problems addressed by items 1 (overly dry and technical language), and 3 (use of a multi booklet format). This discordance was largely due to the combined effect of a very short allotted initial startup time (1.5 months) and little or no contact between the various instructional designers during that phase. It is probable that items 1 and 3, in as much as they represent deviations from the previous styles of delivery, would go a long way to correct this discordance should they be adequately corrected for. Details as to how this could be done will be considered in the following Unit by Unit breakdown.

Unit 1: Introduction and Overview

Essentially items 3 and 4 in the above stated list of weaknesses were pertinent to this unit. It had been hoped that the use of a multi booklet format would allow the student a great deal of flexibility in cross referencing materials. It was also intended to reduce the anxiety naturally generated in any student when confronted with an apparently enormous amount of information for which they are to be held accountable. However this format proved more trouble than it was worth both in terms of administering the materials and in terms of value to the student. Typically students became unnecessarily confused and annoyed when forced to deal with anywhere from two to four booklets simultaneously within the confined space of a study carrel;

already largely occupied by the computer itself. This problem was further exacerbated whenever a student attempted to reference materials from previous modules. It is the author's opinion that the materials for the entire course should be combined into a single workbook. This in order to minimize the space requirements of an individual and to maximize the availability of cross referenced materials (relevant material from both within the module and from previous modules could be easily page referenced within the text and/or located through the use of tables of contents supplied for each module). The workbook could be sold to the student allowing them to preview and/or review materials at their leisure helping them to concentrate better on the computer based activities during the allotted lab times. This format may go a long way to alleviating some of the dissonance created by the dryness of the reading materials (item 1) by affording the student more time to read and reread. Furthermore slower students would be afforded the means by which to accelerate their progress by working on assignments outside of lab times thus promoting course completion for a greater number of students (materials completion being a critical requirement in a skills based course). Lastly note taking, while not particularly encouraged in this course, may be made more relevant to specific content by allowing the student to make notes directly in the margins alongside the text. This could prove to be a powerful means by which to enable the student to clearly relate information presented in the texts to information encountered in the computer based tutorials.

Developing a single workbook format would render the instructional outline written in the form of a psuedo-code (item 4) entirely redundant. As students typically found this form of presentation more confusing than helpful it should be eradicated in favour of ongoing instructions interspersed throughout the module. In the author's opinion the psuedo-code layout should be maintained wherever activity related instructions are inserted. However the intended purpose should shift from that of an advance organizer to that of a focusing or reinforcement agent. Utilizing this somewhat more directive step by step approach to the design of the module would render it a good deal more like the previous modules, and perhaps alleviate some of the difficulties encountered due to discordance between the presentation styles. However, originally it was the author's deliberate intention not to use a strongly directive approach in the hope that students would be forced to use some of the problem solving skills and independent thinking capabilities built up during the early stages of the course. As this intention remains unchanged it is suggested that the proposed revisions to the presentational format not be elaborated to such an extent as to threaten this characteristic of the module.

Unit 2: Programming Languages a General Look

It was the authors original intention to use this unit as a means to reiterate some of the concepts previously encountered in the course. It was further hoped that by reviewing concepts such as RAM, ROM, and DOS a context would be created in which to introduce several novel but related concepts such as, compilers, interpreters, device drivers, and low and high level languages. Although only limited success on objectives 1 through 4 was achieved, test item analysis revealed significant weaknesses in the construction of the test items (see Appendix G). Furthermore students typically expressed enjoyment of the video based materials and little difficulty with the text material for this unit. Given this and the marginal increase in student achievement scores needed to indicate mastery on these objectives the author suggests that the test items be reconstructed to better reflect the conceptual requirements of the objectives. This change should be implemented and evaluated before any attempt is made to revise the instructional materials.

Unit 3: Flowcharting

Despite the high overall success rate on the flowcharting assignment far too much and too great a variety of difficulties were encountered in this unit. The author believes that this was due in part to incomplete text materials and in part to the poor placement of the unit. Incompleteness of the text is evidenced by poor performance on objective 5 with particular regard to the concept of psuedo-codes. Examination of the text by members of the instructional design team marked this reading as being plagued by unclear script and poor examples as well as being somewhat incongruous with previously encountered materials. This position was born out both by comments made on the evaluation questionnaire administered to the small group, and by comments made by the individuals subjected to one to one interviews. The strength and frequency of these observations indicate that a complete rewriting of this section is in order. It should include substantial elaboration on the concept of psuedo-codes and their relationship to both flowcharting and the programming process. New examples should be generated to replace the "Building a Snowman" example currently being used as much confusion was expressed concerning the logical validity of the solution presented therein. Emphasis on the "no best representation" aspect of flowcharting should be modified to clarify the point that the ANSI rules for flowcharting are best adhered to in the interests of consistency, but are by no means unique or absolutely necessary. Perhaps the idea of mathematical elegance (i.e., the simpler the algorithm the better) could be used here to clarify what is meant by effective flowcharting. Finally because of the conceptual contiguity inherent between the process of flowcharting and the process of problem solving, as discussed in the module on LOGO programming it is suggested that the unit on flowcharting either be removed altogether and placed in the LOGO module, or else the rewrite should use terminology which strongly reiterates the terminology used therein. It is possible that the exercises for this module should be rewritten to better accommodate a variety of learning styles. Perhaps the first section of the assignment could be rewritten as a step by step example of how to build a flowchart. Students could be asked to attempt the solution first on their own and then supplied with a step by step breakdown on how to do it. This would enable more tactile or slower learners to work through the problem without threatening their confidence, as well as serve as a means to reinforce the importance of top/down and bottom/up analysis, identifying decision points, and working from maximum to minimum values (these concepts should also be emphasized in greater detail in the foregoing text materials). It is possible that the test items on all the test instruments pertinent to this objective should be reassessed for their accuracy in indicating student comprehension of the concepts. However as the need for another field test is obviated by the extensive changes suggested such a reworking of the test items could be left until after the results of the new field test have been gathered and analysed.

Unit 4a: Fundamentals of Basic Programming: Writing and Editing

Of the five units of instruction contained in the module this unit appears to be the most problem free. Mastery on the editing assignment was 100% and generally the students seemed to find the instructional material both effective and enjoyable. The only consistent area of difficulty had to do again with the dryness of the reading found in the unit (in this case an excerpt from the IBM Basic manual covering editing features). Because this reading in particular had been singled out during the preliminary evaluation by the other members of the instrctional design team as being overly difficult the following flag was worked into the design in order to verify the accuracy of their observation. During the editing asssignment the students were asked to load, modify, run, and save a series of three short Basic programs. The students had not yet acquired sufficient knowledge of the Basic language to be able to understand what exactly they were doing; this was strictly a "follow the instructions" lesson on how the Basic editor works. Two of the programs contained infinite loop errors, in order to escape from which, the student needed to know about the CTRL - BREAK key stroke sequence. Information to this effect was only supplied to them in the reading in question. It was anticipated that if the reading was as difficult and uninteresting as suggested then most students would skim it or skip over it and consequently not know how to break out of the loops when they encountered them. (Note: Students were even warned that the loops existed and that they needed to have read the reading in order to be able to complete the editing exercise). As was noted from the observation sheet fifteen known instances of students having problems with this item were recorded. This combined with many other minor observations concerning weak use of the editor, acts strongly to support the design teams comments concerning the weakness of the reading. Again, the author suggests a complete rewrite of the reading to encourage a more hands on approach to the material. While a less directive approach is favoured for the module in general, it is possible that because it is strongly prerequisite to programming, each aspect of the editor be explored more efficiently with a step by step "do as you read" type of layout. As well, more comprehensive connecting text should be written, between the CAI tutorial "BASICTEACH" and the reading, detailing points of commonality and places to pay particular attention,

Item 6 on the revision list (relationship between good structured problem solving, programming, and the use of subroutines) is also relevant to this module. The only instruction which the students received regarding the use of subroutines was provided by the "BASICTEACH" CAI tutorial (Disk 2, lesson 5). Unfortunately this tutorial emphasizes the use of subroutines as a means to process many similar operations with a minimum of code. It does not place much emphasis on the concept of structured programming and makes no reference to the relationship between structured programming and good problem solving techinques. Given that the unit on flowcharting is revised to more closely parallel the techniques inherent to good problem solving, as discussed in the module on problem solving, it would make sense to extend this connection to include the notion of structured programming (already a concept tacitly encountered in the Logo module) as it applies to BASIC through the use of subroutines. It is the author's opinion that an additional step by step text based exercise, similar to the one recommended for UNIT 3 on flowcharting, would be appropriate for this purpose. This would encourage the learner to practice previously encountered concepts within a novel context. It would also serve to cement concepts newly acquired from the "BASICTEACH" tutorial by providing an opportunity to first attempt a solution in its entirety, and then a step by step comparison between their solution and a prewritten program (Note: again it must be made clear that there are no single "best" solutions).

Unit 4b: Fundamentals of BASIC Programming; Listhandling

Test item analysis (appendix 6) of items pertinent to objectives 8 and 9 indicated a strong likelihood that there was a general lack of student comprehension on the topic of dimensioning subscripted variables and array handling. In the author's opinion this was due to a lack of sufficient detail and practice exercises in the UNIT. Because adding such exercises would likely increase the length of an already long module, it is suggested that instead an attempt is made to more clearly relate the information presented in the text material to that found in the computer based tutorial Edubas I (lessons 8,12,13). Furthermore, although this unit's text material was not singled out as being particularly difficult, the students interest may be better sustained if it were rewritten to include examples which more closedly reflect typical student concerns (e.g. creating a bibliography).

It is possible that the level of cognitive processing required to fully comprehend the use of arrays is somewhat more complicated than can possibly be addressed by a single unit. It may be that it requires the equivalent of an entire module of instruction, in which case due to time constraints it should be eradicated. However, before such a decision is made it is suggested that the above mentioned revisions are implemented and field tested to see if the difficulties cannot be overcome within the existing framework.

Unit 5: Final Assignment

No major revisions are required for this unit. However in the event the aforementioned instruction on array handling is removed from the module, then the assignment would need to be modified to ensure that each of the options could be fulfilled without the use of subscripted variables (indeed two of them can be done without them as is).

The presence of the memory jogger seemed to be generally appreciated however it may be made more effective if it were redesigned to include page referencing to the IBM

BASIC manual for those in need of more detailed explanations (remediation), or who wish to try more sophisticated applications (enrichment).

Overall Conclusion

To the best of the author's knowledge consideration has been given to the ideas of removing the module and creating a seperate course entirely dedicated to BASIC programming. While such a move may have merit with regard to offering a more detailed BASIC course to students particularly interested in the language, the author feels it would be injurious to the overall value of the computer literacy course. Teaching BASIC as a language is something altogether different from using BASIC as an instructional vehicle with which to introduce students to the fundamental principles of programming. The latter should be viewed as an integral part of minimal computer literacy while the former remains a specialized skill of real interest to only a very small proportion of the computer using populous. It was the intended purpose of this module, as specified by the project coordinator and the organizing commitee, to meet the latter need. In the author's opinion the materials produced were successful in achieving their end and have proved their value to the course as a whole, and should be kept in place.

References
Text

Anderson, R. E. (1982). National computer literacy. In R. J. Seidel, R. E. Anderson, & B. Hunter (Eds.), *Computer literacy.* (pp. 9-18). New York: Academic Press.

Anderson, R. E., & Klassen, D. L. (1982). A conceptual framework for developing computer literacy instruction. Cited in R. J. Seidel, R. E. Anderson, & B. Hunter (Eds.), *Computer literacy.*(p. 12). New York: Academic Press.

Anderson, R. E., Klassen, D. L., Krohn, K., Smith-Cunnien, P.(1982). *Computer awareness and literacy: An empirical assessment.* Final Report Special Projects Division, Minnesota Educational Computing Consortium, Saint Paul, Minnesota.

Ausubel, D. P. (1960). The use of advance organizers in the learning and retention of meaningful verbal material. *Journal of Educational Psychology, 51,* 267-272.

Ausubel, D. P. (1968). *Educational psychology: A cognitive view.*New York: Holt, Rinehart, and Winston publications.

Bell, F. H. (1978). Can computers really improve school mathematics? *Mathematics Teacher, 71,* 428-433.

Berk, R. A. (1980). Conducting the Item Analysis. Cited in Berk, R.A. (Ed.), *A guide to criterion-referenced test construction.*Baltimore, MA: John Hopkins University press.

Bork, M. A. (1981). *Learning with computers*. Bedford, MA: Digital Press.

Bloom, B. S. (1956). *Taxonomy of educational objectives, handbook 1: Cognitive domain*. New York: David Mckay Inc.

Bransford, J. D. (1979). *Human cognition*. Monterey, CA: Wadworth Publications.

Bruner, J. S. (1960). *The process of education*. Cambridge, Mass: Harvard University Press.

Bruner, J. S. (1966). *Toward a theory of instruction*. Cambridge,Mass: Harvard University Press.

Dewey, J. (1938). *Experience and education*. NY: Macmillan.

Deringer, D. K., & Molnar, A. R. (1982). Key components for a national computer literacy program. In R. J. Seidel, R. E.Anderson, & B. Hunter (Eds.), *Computer literacy*. (pp. 3-7). NewYork: Academic Press.

Dick, W., & Carey, L. (1978). *The Systematic Design of Instruction*. Glenview, Ill: Scott, Freeman, and Company.

Doyle, T. E. (1982, February). PASCAL NOW: Let Pascal balance your NOW account. *Byte*, p. 290.

Dwyer, T. A. (1974). Heuristic strategies for using computers to enrich education. *International Journal of Man-Machine Studies*, *6*, 1-16.

Gagne, R. M., & Dick, W. (1983). Instructional Psychology. *Annual Review of Psychology, 34*, 261-295.

Gerlach, V. S., Haygood, R. C., Filan, G. L., Schmid, R.F., Wigand, D. L., Hagin, W. ~~V.~~ (19??). Performance Objectives. *Studies in Systematic Instruction and Training*, U.S.A.F. Office of Scientific Research, Technical Report # 81230 (unpublished), Arizona State University.

Greeno, J. G. (1980). Trends in theory of knowledge for problemsolving. In D. T. Tuma, F. Reif (Eds.), *Problem solving and education: Issues in teaching and research.* Hillsdale, NJ: Erlbaum.

Howe, J. A. M., & Doublay, B. (1979). Microcomputer assisted instruction: Turning the clock back? *Programmed Learning and Educational Technology, 16*, 240-246.

Howe, J. A. M., O'Shea, T., & Plane, J. (1979). *Teaching Mathematics Through Logo Programming* (D. A. I. Research paper 115) . Department of Artificial Intelligence, University of Edinburgh.

Human Resources Research Organization. (1980) *The HumRRO Computer Literacy Project*. Educational and Training Systems Division, Alexandria, Virginia.

Luehrmann, A. (1972). Should the computer teach the student or vice-versa? In *Proceedings of the Spring Joint Computer Conference*. Association of Information Processing Systems, Washington, D.C.

Lewis, T. G. (1982, February) Book review of Pascal programs for scientists and engineers. *The Computing Teacher*, p. 55.

Lochead, J. (1981). Research synthesis on problem solving. *Educational Leadership, 39(1)*, 68-70.

Lochead, J., & Clement, J. (Eds.) (1979). *Cognitive process construction: Research on teaching thinking skills*. Philadelphia,Penn: The Franklin Institute Press.

Mayer, R. E. (1978). Different problem solving competencies established in learning computer programming with and without meaningful models. *Journal of Educational Psychology, 67*, 725-734.

Mayer, R. E. (1976). Some conditions of meaningful learning for computer programming: Advance organizers and subject control of frame order. *Journal of Educational Psychology, 4*, 143-150.

Mayer, R. E. (1978). Advance organizers that compensate for the organization of text. *Journal of Educational Psychology, 70*, 880-886.

Mayer, R. E. (1980). Contributions of cognitive science and related research in learning to the design of computer literacy curricula. In R. J. Seidel, R. E. Anderson, & B. Hunter (Eds.), *Computer literacy*. (pp. 129-159). New York: Academic Press.

Papert, S. (1979). Computers and learning. In M. Dertouzos & J. Moses (Eds.), *The computer Age: A twenty tear view.*

Cambridge,MA: MIT Press.

Papert, S. (1981). *Mindstorms: Children, computing and powerful ideas*. New York: Basic Books.

Romiszowski, A. J. (1981). *Designing instructional systems.* NY:Kogan Page.

Schneiderman, B. (1980). *Software psychology: Human factors in computer and informations systems*. Cambridge, MA: Winthrop Publishers.

Seidel, R. J. (1982). On the development of an information handling curriculum: Computer literacy, a dynamic concept. In R. J. Seidel, R. E. Anderson, & B. Hunter (Eds.), *Computer literacy* (pp. 19-32). New York: Academic Press.

Seidel, R. J.,& Hunter, H. G. (1970). The application of theoretical factors in teaching problem-solving by programmed instruction. *International Review of Applied Psychology, 19*(1),41-81.

Skuliez, M. (1984). *Some analogies between computer programming and the composing process*. Paper presented at the New YorkState English Council. Amherst, NY.

Shively, J. E. (1984). Computer utilization in education: Problems and prerequisites. *AEDS Journal, 17*(3), 24-34.

Simon, H. A. (1980). Problem solving and education. In D. T. Tuma, & F. Reif (Eds.), *Problem solving and education:*

issues in teaching and research. Hillsdale, NJ: Erlbaum Publishers.

Soloway, E., Lochead, J., & Clement, J. (1982). Does computer programming enhance problem solving ability? Some positiveevidence on algebra word problems. In R. J. Seidel, R. E.Anderson, & B. Hunter (Eds.), *Computer literacy*. (pp. 171-185). New York: Academic Press.

Spilich, G. J., Vesonder, G. T., Chiesi, H. L., & Voss, J. F. (1980). Text processing of domain related information for individuals with high and low domain knowledge. *Journal of Verbal Learning and Verbal Behaviour, 18*, 275-290.

Turkle, S. (1980). Computer as rorschach. *Society, 17*(2).

Watt, D. H. (1982) Education for citizenship in a computer based society. In R. J. Seidel, R. E. Anderson, & B. Hunter (Eds.), *Computer literacy*. (pp. 53-68). New York: Academic Press.

Weizenbaum, J. (1976). *Computer Power and Human Reason*, San Francisco: W. H. Freeman.

Wessel, M. (1974). *Freedoms edge: The computer threat to society*. Reading, MA: Addison Wesley Publications.

Weston, C., & Cranton, P.A. (1986). Selecting Instructional Strategies. *Journal of Higher Education*, 57 (3), 259-288.

Whimbey, A. (1980). Students can learn to be better problem solvers. *Educational Leadership, 37,* 560-562.

Whimbey, A.,& Lochead, J. (1981). *Problem solving and comprehension: A short course in analytical reasoning.* Philadelphia, Penn: The Franklin Institute Press.

Winer, L. R., & Schmid, R. A. (1986). Using Brunerian learning theory with educational simulations to teach concepts. *Canadian Journal of Educational Communication, 15*(3), 153-165

Media

Micro Learning Systems (1984). *BASICTEACH.* Reston publishing company; Reston VG.

Europro Inc. (1983). *Edubas 1.* San Diego, Ca.

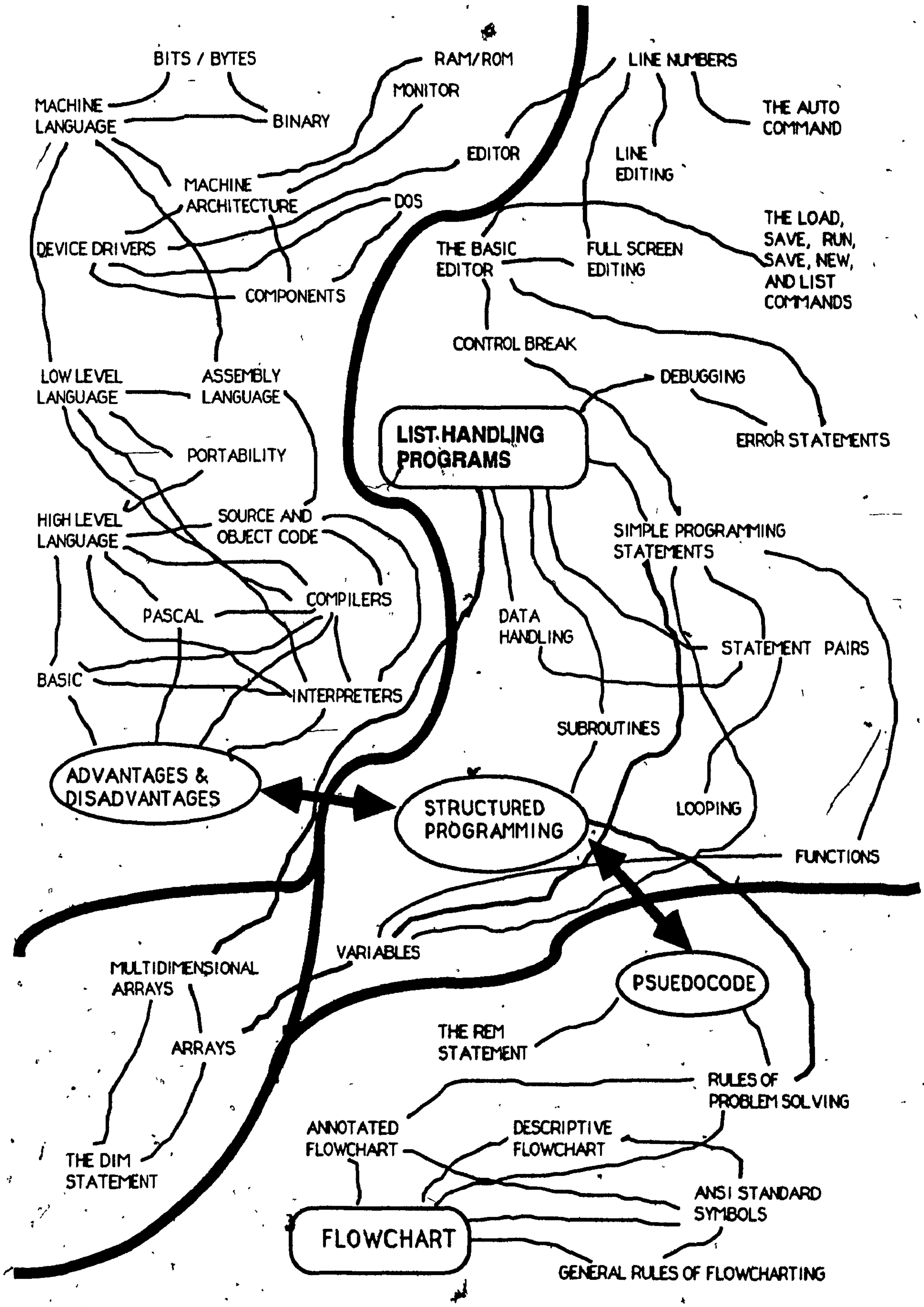
T V Ontario Ltd. (1980) *Bits and Bytes.* (episode 6), Toronto, Ontario.

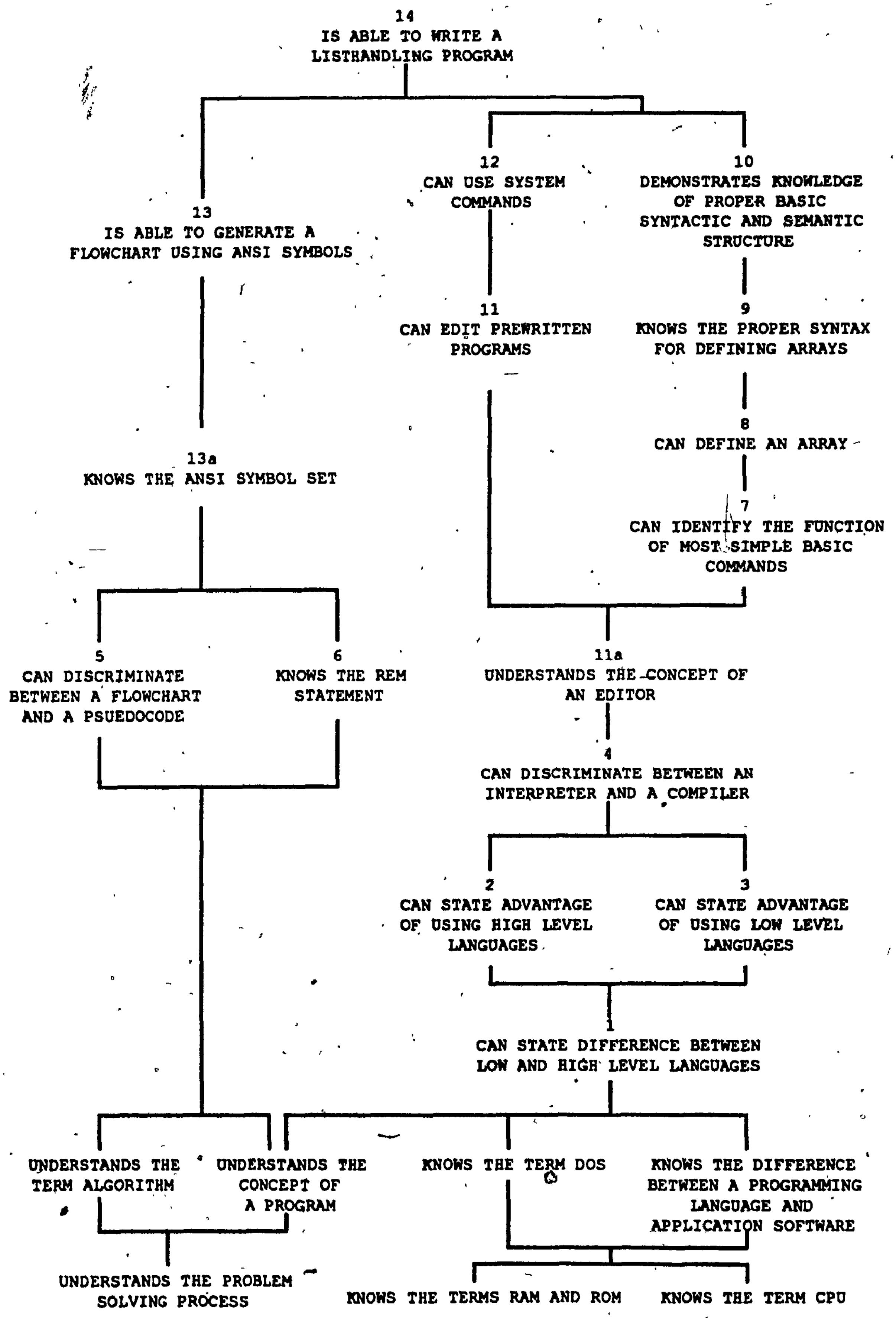
Appendix A

Instructional Analysis

Instructional analysis was conducted in two stages. The first recommended by Romiszowski (1981), involved a process of brainstorming and free association which was initially translated into a series of concept maps as typified by the representation found on page 91. Concepts were grouped into relationships on the basis of similarity, and contiguity with respect to the topics they addressed as well as the the level of conceptual difficulty involved (rated according to Bloom's Taxonomy as outlined in Romiszowski, 1981). Meta concepts, being those concepts singular in their identity but encompassing most of the other concepts in a category, were isolated and used to identify possible links between categories. Finally the relative importance of each concept was determined by the observed density of relational lines connected to it.

In the second stage information gleaned from the previous process was reworked into the more linear hierarchical model, found on page 92, such as is promoted by Dick and Carey (1978). This facilitated the sequencing of the materials and the specification of objectives without losing sight of the more associational relationships noted in stage one.


BITS / BYTES
RAM/ROM
LINE NUMBERS
MACHINE LANGUAGE
BINARY
MONITOR
THE AUTO COMMAND
EDITOR
LINE EDITING
MACHINE ARCHITECTURE
DOS
THE LOAD, SAVE, RUN, SAVE, NEW, AND LIST COMMANDS
DEVICE DRIVERS
THE BASIC EDITOR
FULL SCREEN EDITING
COMPONENTS
CONTROL BREAK
LOW LEVEL LANGUAGE
ASSEMBLY LANGUAGE
DEBUGGING
LIST HANDLING PROGRAMS
ERROR STATEMENTS
PORTABILITY
HIGH LEVEL LANGUAGE
SOURCE AND OBJECT CODE
SIMPLE PROGRAMMING STATEMENTS
COMPILERS
PASCAL
DATA HANDLING
STATEMENT PAIRS
BASIC
INTERPRETERS
SUBROUTINES
ADVANTAGES & DISADVANTAGES
STRUCTURED PROGRAMMING
LOOPING
FUNCTIONS
MULTIDIMENSIONAL ARRAYS
VARIABLES
PSUEDOCODE
ARRAYS
THE REM STATEMENT
RULES OF PROBLEM SOLVING
ANNOTATED FLOWCHART
DESCRIPTIVE FLOWCHART
THE DIM STATEMENT
ANSI STANDARD SYMBOLS
FLOWCHART
GENERAL RULES OF FLOWCHARTING


14
IS ABLE TO WRITE A LISTHANDLING PROGRAM
13
IS ABLE TO GENERATE A FLOWCHART USING ANSI SYMBOLS
12
CAN USE SYSTEM COMMANDS
10
DEMONSTRATES KNOWLEDGE OF PROPER BASIC SYNTACTIC AND SEMANTIC STRUCTURE
11
CAN EDIT PREWRITTEN PROGRAMS
9
KNOWS THE PROPER SYNTAX FOR DEFINING ARRAYS
8
CAN DEFINE AN ARRAY
13a
KNOWS THE ANSI SYMBOL SET
7
CAN IDENTIFY THE FUNCTION OF MOST SIMPLE BASIC COMMANDS
5
CAN DISCRIMINATE BETWEEN A FLOWCHART AND A PSUEDOCODE
6
KNOWS THE REM STATEMENT
11a
UNDERSTANDS THE CONCEPT OF AN EDITOR
4
CAN DISCRIMINATE BETWEEN AN INTERPRETER AND A COMPILER
2
CAN STATE ADVANTAGE OF USING HIGH LEVEL LANGUAGES
3
CAN STATE ADVANTAGE OF USING LOW LEVEL LANGUAGES
1
CAN STATE DIFFERENCE BETWEEN LOW AND HIGH LEVEL LANGUAGES
UNDERSTANDS THE TERM ALGORITHM
UNDERSTANDS THE CONCEPT OF A PROGRAM
KNOWS THE TERM DOS
KNOWS THE DIFFERENCE BETWEEN A PROGRAMMING LANGUAGE AND APPLICATION SOFTWARE
UNDERSTANDS THE PROBLEM SOLVING PROCESS
KNOWS THE TERMS RAM AND ROM
KNOWS THE TERM CPU

Appendix B

List of Objectives

GENERAL GOALS

The overall aim of the module was to provide the student with a thorough familiarity of the following concepts:

1) A programming language,
2) A hierarchy of software categories (i.e. the differences between varying kinds of languages).
3) The fundamental syntax of the BASIC language.

LIST OF LEARNING OBJECTIVES

UNIT 2: (objectives 1 through 4)

Upon completion of this section the student will be able to correctly define the following concepts in a short answer format and in accordance with the definitions found in READING # 1.

1) the difference between a high and low level language.
2) the advantages of using a high level language.
3) the advantages of using a low level language.
4) the difference between an interpreter and compiler.

UNIT 3: (objectives 5, 6, and 13)

After having completed the materials found in UNIT 3 the learner will be able to,

5) given both definitions, correctly discriminate between the definition of a flowchart and the definition of a psuedocode as described in READING # 2.

6) correctly identify, from a list of descrip tions, the function of the REM statement in the BASIC language as described in READING # 2. 13) generate a logically valid graphic representa tion of an algorithm to solve one of a set of prespecified problems in the form of a flow chart using the ANSI format.

UNIT 4a: (objectives 7, 11, and 12)

Upon completion of the activities prescribed in UNIT 4a the learner will be able to,

7) given a list of 10 simple system and language commands, identify the function of all 10 as defined in the BASIC TEACH program.

11) correctly enter and edit a short, prewritten BASIC program such that it may be run without error,

12) demonstrate a knowledge of BASIC system commands by correctly loading (from diskette), listing, running and saving (to diskette) a given program.

UNIT 4b: (objectives 8 through 10)

Upon completing the materials for UNIT 4b the learner will be able to,

8) in short answer format, correctly define what is meant by array handling in accordance with the definition provided in READING # 4,

9) given a list of BASIC statements related to the dimensioning of subscripted variables, correctly identify improper syntax for at least 70% of the items.

10) demonstrate a knowledge of proper semantic and syntactic structure in BASIC by indicating in writing
A) the correct output for a series of short logically valid and syntactically correct prewritten programs.
B) the errors in a series of logically and/or syntactically incorrect programs.

UNIT 5: (objective 14 - terminal objective)

14) Upon completion of all the materials found in

the previous UNITS (1-4), the learner, with the aid of a memory jogger related to proper syntax, will be able to write a simple, logically valid, bug free, listhandling program in BASIC to execute the same algorithm represented by the flowchart generated in fulfillment of objective 13.

Appendix C

Sample Pretest/Postest

LESSON V : QUIZ

NAME :____________________ DATE :____________

STUDENT # :________________

FACULTY :__________________

DEPARTMENT :_______________

ANSWER AS MANY OF THE FOLLOWING QUESTIONS AS YOU CAN. PLEASE DO NOT REFER TO YOUR NOTES OR ANY OF THE COURSE MATERIALS DURING THE QUIZ.

1) Briefly describe the difference between a HIGH and a LOW level language.____

____

____

____

2) Briefly describe the difference between a compiler and an interpreter.____

____

____

____

3) The BASIC language can be said to be

A) a high level language

B) a low level language

C) application software

D) object code

4) A graphic representation of a problem solving algorithm is called a ____________________ .

5) A preliminary program outline written in brief natural language (eg. English, French, German, etc.) phrases is called a ____________________ .

6) What is the BASIC language command required to fulfill each of the following functions?

A) To look at the program procedures ___________

B) To execute a program ___________

C) To take a program from a disk and place it in RAM ___________

D) To write a program to disk permanently ___________

E) TO clear RAM to start work on a new program ___________

F) To take information from a data list previously established in the current program. ___________

G) To ask for information from the user. ___________

H) To change a variable from one value to another. ___________

I) To output a result. ___________

J) To place a descriptive comment in a program which is not considered when the program is run. ___________

7) What output would occur as a result of running the following programs.

A)
```
5 X = 1
10 IF X >= 4 GOTO 60
15 Y = 3*X^2
20 PRINT Y,
25 X = X+2
30 GOTO 10
60 END
```
____

B)
```
10 X = 1
20 IF X = 10 GOTO 50
30 X = X+2
40 GOTO 20
50 PRINT X
60 END
```
____

C)
```
10 READ B,C
20 A = B+C
30 READ C,D
40 E = C+D
50 PRINT A,B,C,D,E
60 DATA 7,11,3,5
70 END
```
____

D)
```
10 DIM A(S), B(S)
20 FOR X = 1 TO 5
30 A(X) = 3+2*X
40 B(X) = X+2
50 NEXT X
60 FOR X = 1 TO 5 STEP 2
70 S = S+A(X)+B(X)
80 NEXT X
90 PRINT S
99 END
```
____

E)
```
5 PRINT "GIVEN APRICOT,CANTALOUPE,FIG,PEACH,PLUM,WATERMELON"
10 INPUT "WHAT FRUIT DO YOU LIKE?";F$
20 GOSUB 500
30 PRINT "EACH ";F$;" HAS ";X;" CALORIES."
40 GOTO 550
500 READ X$,X
510 IF X$ = F$ THEN 530
520 GOTO 500
530 RETURN
540 DATA APRICOT,18, CANTALOUPE, 120, FIG, 60
550 DATA PEACH, 35, PLUM, 25, WATERMELON, 1840
```
____

8) Identify whether each of the following BASIC statements is correct or incorrect. In the event that one is incorrect then state what the error is.

A) 11 DIM A(40,10), B$(50) ____________________

B) 45 PRINT W(1,Y) ____________________

C) 10 INPUT A$(X) ____________________

D) 65 READ (5),(7),(9) ____________________

E) 40 IF A$(y) <> A$(Y+1) THEN 100 ____________________

F) 50 PRINT X$(X+Z) ____________________

G) 20 A(3,4) = 50 ____________________

H) 35 DIM S(N) ____________________

9) What is meant by the term array handling? ____________________

____

____

Appendix D

Sample Evaluation Questionnaire

EVALUATION QUESTIONNAIRE FOR LESSON 5

NOW THAT YOU HAVE COMPLETED THE MATERIALS FOR LESSON 5, PLEASE COMPLETE THIS FORM AND RETURN IT TO THE RESERVE BOOTH.

CHECK THE APPROPRIATE STATEMENTS AND COMMENT AS REQUESTED.

IF YOU NEED MORE ROOM PLEASE USE THE BACK OF THE PAPER.

NAME (OPTIONAL) ....

DATE ....

DATE LESSON 5 WAS STARTED ....

1) AMOUNT OF ESTIMATED TIME USED TO COMPLETE LESSON 5

(NEAREST 1/2 HR) ....**(15.15 mean time)**......

2) OVERALL I WOULD RANK LESSON FIVE AS

(A) EXCELLENT	(B) GOOD	(C) FAIR	(D) POOR
7.31%	**60.97%**	**29.26%**	**2.43%**

W H Y ?

....

....

3) THE OBJECTIVES FOR LESSON FIVE IN GENERAL WERE

(A) CLEARLY DEFINED,	(B) ADEQUATE,	(C) NOT CLEAR,	(D) TOO BRIEF,
29.26%	**60.97%**	**4.87%**	**2.43%**

(E) TOO BROAD.
2.43%

4) PLEASE CIRCLE THE APPROPRIATE RESPONSE FOR THE FOLLOWING QUESTIONS (please use the following code):

strongly agree 1 2 3 4 5 strongly disagree

A) The list of objectives tells me exactly what I need to know to complete lesson five.

1	2	3	4	5
29.26%	31.7%	21.95%	17.07%	0%

B) The objectives indicate how well I must accomplish a task.

1	2	3	4	5
29.26	12.19%	41.46%	17.07%	0%

C) The objectives provide a means for me to periodically evaluate my own progress.

1	2	3	4	5
19.51%	29.26%	31.7%	17.07%	2.43%

D) Provisions are made for my accomplishments of the objectives.

1	2	3	4	5
14.63%	39.02%	29.26%	14.63%	2.43%

E) By reading the objectives I know what I am expected to do in this lesson.

1	2	3	4	5
41.46%	24.39%	21.95%	9.75%	2.43

F) The objectives show me how I will be evaluated.

1	2	3	4	5
26.83%	26.83%	31.71%	14.63%	0%

G) I see no restrictions as to how I am to meet the lesson objectives.

1	2	3	4	5
24.4%	34.15%	24.4%	9.76%	7.32%

H) The objectives describe precisely what I must be able to do to complete this lesson.

1	2	3	4	5
34.15%	24.4%	21.95%	19.51%	0%

I) I do not feel confident that I am meeting the instructional requirements even though I can do what is stated in the objectives.

1	2	3	4	5
12.19%	**14.63%**	**17.07%**	**26.83%**	**29.27%**

J) The statement of objectives does not make clear what I need to know for the final exam.

1	2	3	4	5
12.19%	**17.07%**	**17.07%**	**17.07%**	**36.58%**

K) As I read the stated objectives, I do not understand what I am expected to accomplish.

1	2	3	4	5
4.88%	**9.76%**	**19.51%**	**39.02%**	**26.83%**

L) I do not see how these objectives can be requirements of the lesson.

1	2	3	4	5
4.88%	**2.44%**	**21.95%**	**43.9%**	**26.83%**

M) The objectives do not inform me of the kinds of things that are accepted as evidence that I have done well.

1	2	3	4	5
7.31%	**12.19%**	**19.51%**	**29.27%**	**31.71%**

N) How I am to achieve each objective is not clear.

1	2	3	4	5
4.88%	**7.32%**	**21.95%**	**36.58%**	**29.27%**

O) The objectives do not help me to make a high grade on the final exam.

1	2	3	4	5
4.88%	**17.07%**	**14.63%**	**29.27%**	**34.15%**

5) THE CONTENT OUTLINE (feel free to circle more than one answer)

73.17%	(a) WAS LOGICAL.
43.90%	(b) CONTAINED ADEQUATE DETAIL.
00.00%	(c) WAS NOT LOGICAL.
12.19%	(d) DID NOT CONTAIN ENOUGH INFORMATION.
12.19%	(a and b)

6) WHICH ACTIVITY DID YOU ENJOY THE MOST? WHY? ....

..(Final assignment=63.41%, Flowcharting=17.07%, Editing=9.76%, Tutorials = 4.87%, Videos = 4.88%, Readings = 00.00%)..........

7) WHICH ACTIVITY DID YOU ENJOY THE LEAST? WHY? ....

..(Final assignment=2.44%, Flowcharting=4.88%, Editing=19.51%, Tutorials = 19.51%, Videos = 21.95%, Readings = 31.71%)........

8) THE PRACTICE AND EXERCISE ITEMS WERE (circle one or more)

73.17%	(a) ADEQUATE.
14.63%	(b) TOO FEW IN NUMBER.
4.88%	(c) TOO MANY IN NUMBER.
7.32%	(c) WERE UNRELATED TO THE INSTRUCTIONAL SEQUENCE.
0.00%	(d) INADEQUATE.

9) THIS LESSON FITS WITHIN THE SEQUENCE OF OTHER LESSONS IN THE COURSE.

Strongly agree	1	2	3	4	5	Strongly disagree
	68.29%	19.51%	4.88%	2.44%	4.88%	

10) WHICH TERM(S) BEST DESCRIBES THE ACTIVITIES IN THIS LESSON?

9.76%	EASY	21.95%	TOO MUCH READING .....

53.66%	HARD	29.27%	NOT ENOUGH ACTIVITIES .....
10.00%	BORING	9.76%	TOO MANY ACTIVITIES .....
53.66%	INTERESTING	31.71%	TOO MUCH THEORY .....

11) **DID YOU RECEIVE ANY HELP ON THIS LESSON?**

70.73%	YES (GOTO NUMBER 12) .....
24.39%	NO (GOTO NUMBER 13) .....
4.88%	unanswered

12) **WHY DID YOU NEED HELP?**

34.15%	THE MATERIAL WAS TO DIFFICULT .....
4.88%	EXPLANATION WAS POOR AS TO WHAT WAS TO BE LEARNED
34.15%	THE DIRECTIONS WERE CONFUSING AND VAGUE .....
26.83%	THE LEARNING ACTIVITY WERE TO DIFFICULT .....
14.63%	HELP WAS NEEDED TO LOCATE MATERIALS, AIDS, ETC. .....

OTHER...

....

13) **DID YOU HAVE ANY PROBLEMS RELATED TO**

17.07%	FOLLOWING INSTRUCTIONS? .....
12.19%	UNDERSTANDING DIAGRAMS? .....
	THE USE OF AUDIO - VISUALS? .....
	OBTAINING MATERIALS OR EQUIPMENT?.....

14) PLEASE COMMENT ON HOW IMPROVEMENTS MAY BE MADE

....

....

....

Appendix E

Sample Observation Sheet

Unit #	Comment	Page #

Appendix F

Sample Demographics Questionnaire

BACKGROUND INFORMATION

1) NAME :______________________

ID #______________

IF YOU ARE A UNIVERSITY STUDENT PLEASE COMPLETE THE FOLLOWING QUESTIONS, IF NOT THEN SKIP TO QUESTION 4.

2) FACULTY :______________________
(eg. arts & science, engineering, fine arts, etc.)

3) DEPARTMENT :______________________
(eg. history, biology, mathematics)

4) What ages are you between?

A) 18 - 25
B) 25 - 30
C) 30 - 35
D) 35 - 40
E) over forty.

5) What is your present occupation? ______________________
____

6) What is the highest level of formal education that you have completed?______________________

7) Have you ever worked with a computer before?

A) often
B) a moderate amount
C) a few times
D) once or twice
E) never

8) If you have please describe the nature of your computer experience (eg. type of machine, languages used, software used).

____

____

____

9) How would you rate your typing skills?

A) very good
B) good
C) fair
D) poor
E) non-existent

10) How many words a minute would you say you can type?__________

11) Are you a Montreal resident?______________

12) What is your approximate travel time to school?______________

13) Are you a full time student?______________

14) What are your preferred days for working in the computer lab (list three in order of preference).

____

____

____

15) What is your native language? A) English, B) French, C) other

If other please specify____________________

16) How many courses are you taking this term (please list by name,time,and location)?

____

____

____

17) What is your background in math and/or logic.

A) excellent
B) very good
C) good
D) fair
E) poor

18) Is there an address and/or phone number where you can be reached(optional)?

____

____

Appendix G

Test Item Analysis

Objectives 2 and 3 were only tested for as a subset of question 1 on the Pretest/posttest. Typically, due to the phrasing of the question which requested the student to supply only an explanation of the difference between high and low level languages (objective 1), students did not refer to the advantages or disadvantages (objective 2) of high or low level languages as a means to explaining the difference. As a result the two marks relevent to these objectives were deducted from the overall score for question 1. Good overall achievement on responses relevant to objective 1 was evident increasing the possibility that the attempt to combine the evaluation of several objectives into one poorly phrased test item was the cause of difficulty rather than a lack of learner comprehension.

Objective 5 was evaluated on the basis of correct or incorrect responses to questions 4 and 5. Overall mastery was set at 100% (or a total of 4 marks, 2 for each question). Only 27.5% of those tested achieved this level. However more detailed analysis indicated that 38 out of 40 people (95%) answered question 4 correctly while only 11 out of 40 (27.5%) answered question 5 correctly. None of those tested answered incorrectly to question 4 and correctly to question 5. As the subject matter tested in both questions is closely related, and the level of understanding required is similar, it is assumed that the results obtained on this objective are not reflective of comprehension inability on the part of the members of the sample population. The indication is that the problem lies in either the phrasing and pertinence of question 5, or else in the weakness of the relevent instructional materials. Objective 8 was tested for by item 9, mastery was set at 2 of a possible three marks. Only 17 people (42.5%) achieved this level, however another 14 people (35%) scored one mark indicating a strong general understanding of the relationship between array handling and the representation of data in a matrix. The weakness evidenced here may have to do with a discrepancy between the level of comprehension addressed by the instructional materials and that addressed by the required response to the question. Analysis of the question's phrasing indicated extreme generality, hence the level of detail required in the response may not have been registered by the test poulation.

Objective 9 was tested for by questions 8a through 8h the table below indicates the general response pattern.

Item	# correct	% correct	aspect tested
a	26	65	two and multi-deminsional arrays
b	31	77.5	using Alpha subscripts
c	23	57.5	combined use of string variables and alpha subscripts
d	29	72.5	error - no variable defined
e	24	60	use of arithmetically defined subscripts
f	19	47.5	combined use of arithmetically defined subscripts and string variables
g	28	70	assigning value to an array variable
h	36	90	use of a variable dimension array

While four individual show mastery of 70 % or better each of them reflect content dealing with the simple syntactic form and rudimentary characteristics of array usage in BASIC. The four items which showed less than 70% reflect more complex and unusual syntactic structure, with the poorest performance occuring on item 8e which represents the most difficult concepts. The lack of comprehension evidenced by examining these test items would seem to indicate that weakness in the materials is more likely to be the causal factor here, rather than inappropriate test items.

Appendix H

Comparisons made between various subgroups yielding non-significant results.

Simple comparisons were made using the following list of independent variables and performance on the flowcharting assignment, final assignment, and the pre, post, and delayed test.

- Department
- Faculty
- Age
- Math background
- language
- typing skills
- previous exposure to computers

Subgroupings were identified for each independent variable through frequency analysis. Because of the limited size of the test population some subgroupings were formed by collapsing across similar categories in order to guarantee cell sizes of greater than 5. The final subgroups tested were as follows.

DEPARTMENT

Group 1 = Students in either English, Theatre, Modern Languages, Journalism, History, or Animation. (total cell size = 12)

Group 2 = Students in either Math, Economics, Management, or geography. (total cell size = 10)

Group 3 = Psychology, Womens Studies, Sociology, or Biology. (total cell size = 10)

Group 4 = Independent students (total cell size = 8)

FACULTY

Arts and Science = 30 students Other = 10 students

AGE

18 to 25 = 28 students Over 25 = 12 students

MATH BACKGROUND

Very Good to Excellent = 13 students
Medium to Good = 14 students Poor to Medium = 13 students

LANGUAGE

English = 29 Other = 11

TYPING SKILLS

More than 50 words per minute = 9 students Between 20 and 50 words per minute = 19 students Less than 20 words per minute = 12 students

PREVIOUS EXPOSURE TO COMPUTERS

Some Programming = 8 students
Some application software = 14 students
None = 18 students

Appendix I

Sample Final Exam (delayed test)

INTE 300/4 SECT 01 ALLOTTED TIME: 90 MINUTES

FINAL EXAM

(Note: Items 66-100 are pertinent to the Basic module)

1) HARDWARE COMPONENTS COMMON TO MOST MICROCOMPUTER SYSTEMS ARE

A) the keyboard, diskdrive, dos, and rom.
B) the dos, diskdrive, CPU, and monitor.
C) ROM, RAM, DOS, CPU.
D) keyboard, diskdrive, CPU, monitor.

2) THE PRIMARY MEANS FOR ENTERING DATA DIRECTLY INTO THE MICROCOMPUTER IS THROUGH THE

A) a diskette.
B) a diskdrive.
C) a keyboard.
D) a program.

3) WHEN YOU FORMAT A DISKETTE

A) you copy the DOS on it.
B) you make it double sided.
C) you change its physical shape.
D) you put reference signals on it.

4) WHICH OF THE FOLLOWING CAN BE SAID TO BE THE THINKING ELEMENT OF THE COMPUTER?

A) The microchip.
B) The RAM
C) The CPU
D) The ROM

5) THERE ARE TWO DEVICES THROUGH WHICH A COMPUTER MAY OUTPUT INFORMATION DIRECTLY THEY ARE

A) the diskdrive and the diskette.
B) the monitor and the diskdrive.
C) the monitor and the cassette.
D) the monitor and the printer.

6) THE BRAIN OF THE COMPUTER IS CALLED THE

A) RAM
B) DOS
C) CPU
D) Diskette

7) TERMS RAM AND ROM STAND FOR

A) random active memory and read onto memory.
B) random actual memory and read only memory.
C) random access memory and read only memory.
D) read access memory and random only memory.

8) RAM AND ROM ARE DIFFERENT BECAUSE

A) ROM and RAM are both permanent but ROM can be changed.
B) ROM is preprogrammed and permanent RAM is not.
C) They are both transient but ROM can be saved to diskette.
D) RAM is a programming mode of ROM.

9) IT IS IMPORTANT TO SAVE INFORMATION ON A DISKETTE BECAUSE

A) there is usually to large an amount of information to be held in memory.
B) it is an essential step before obtaining a hardcopy.
C) the information will be lost when you turn the machine off.
D) ROM is volatile.

10) WHAT IS THE DIFFERENCE BETWEEN HARDWARE AND SOFTWARE?

A) Hardware is often machinery, software never is.
B) Hardware is tangible, software is not.
C) Hardware processes information, software is information.
D) All of the above.

11) WHAT IS MEANT BY THE TERM BIT?

A) Binary digit
B) The smallest piece of information
C) Is represented by a 0 or a 1 (on or off).
D) All of the above

12) WHAT IS MEANT BY THE TERM BYTE?

A) A series of bits defining a computer character.
B) A measure of computer memory.
C) Usually a series of eight bits.
D) All of the above

13) WHAT IS MEANT BY THE TERM BINARY CODE?

A) Machine language
B) A mathematical system based on two elements
C) A code which works on 0 and 1
D) All of the above

WRITE **TRUE** OR **FALSE** FOR EACH OF THE FOLLOWING EQUATIONS.

14) DOS = software ________

15) ROM = software ________

16) diskdrive = hardware ________

17) printer = software ________

18) SuperCalc 3 = software ________

19) keyboard = software ________

20) WORDSTAR = software ________

21) RAM = hardware ________

22) CPU = software ________

23) Exploring the IBM PC = hardware ________

24) WHAT DOES THE TERM DOS STAND FOR?

A) Disk organization standard
B) Diskette operating standard
C) Diskette operating system
D) Any of the above

25) INFORMATION CAN BE STORED ON

A) a diskette, cassette, or hardcopy.
B) a diskette or monitor.
C) a diskette or in ram.
D) all of the above.

26) WHICH OF THE FOLLOWING ARE DOS FUNCTIONS?

A) Formatting a diskette.
B) Outputting a directory of files.
C) Renaming a file.
D) Copying a file.
E) A, B, and D.
F) All of the above

27) WHEN A MICROCOMPUTER PROCESSES INFORMATION IT

A) performs a series of logical operations.
B) rearranges the ROM to accommodate the input.
C) performs a series of arithmetic computations very quickly.
D) performs a random series of mathematical operations.

28) WITH WORD PROCESSING SOFTWARE YOU CAN

A) format a file diskette.
B) make budget projections.
C) write and edit a text.
D) all of the above.

29) WITH AN ELECTRONIC SPREADSHEET YOU CAN

A) make budget projections.
B) output a directory of spreadsheet files.
C) manipulate mathematical formulae.
D) all of the above.

30) **WORDSTAR** WORD PROCESSING SOFTWARE IS DIVIDED UP INTO

A) sectors.
B) cells.
C) menus.
D) graph modes.

31) WITH THE **SUPERCALC 3** ELECTRONIC SPREADSHEET YOU **CANNOT**

A) calculate class averages.
B) plot data on a graph.
C) calculate monthly payments on a new car.
D) keep the annual Christmas letter in a file for next year.

32) BY THE TERM PROBLEM WE MEAN

A) any procedure which needs to be executed.
B) any complex task which needs to be completed.
C) an inability to reach a specific and clearly defined goal.
D) all of the above.

33) WHAT IS MEANT BY A **SYSTEMATIC PROBLEM SOLVING STRATEGY?**

A) Applying your solution procedure to a computer system.
B) A system based diagnostic program.
C) A step by step plan used to achieve a specific goal.
D) An approach to computer programming.

34) WHAT ARE THE STEPS COMMON TO MOST PROBLEM SOLVING S TRATEGIES?

A) 1: Find a solution,
2: write a program,
3: run the program,
4: output the result.

B) 1: Run the program,
2: output the result,
3: apply the solution,
4: check for errors.

C) 1: Assess and understand the problem,
2: relate the data,
3: find a solution,
4: check the results.

D) none of the above

35) TO SOLVE A PROBLEM MEANS

A) to have a solution.
B) to find a way to reach a specific and clearly defined goal.
C) to complete work on a task.
D) to execute a complex algorithm.

36) WHAT IS MEANT BY THE TERM STEPWISE REFINEMENT.

A) Following a planned procedure.
B) Developing an algorithm.
C) Breaking a problem down into smaller problems.
D) All of the above.

37) WHAT IS MEANT BY THE TERMS BUG AND DEBUGGING?

[✓ A)] Finding errors in a program and removing them.
B) Working out an algorithm and refining it.
C) Finding a new solution and replacing the old one with it.
D) All of the above.

WRITE **TRUE** OR **FALSE** BESIDE EACH OF THE FOLLOWING STATEMENTS

38) The LOGO language can be considered application software. ________

39) An algorithm is a problem solving device. ________

40) An algorithm is a step by step procedure used to solve a problem. ________

41) In the LOGO language a variable is used to vary the input to a procedure. ________

42) In the LOGO language a procedure is a built in command. ________

43) In the LOGO language a PRIMITIVE is a simple geometric shape. ________

44) In the LOGO language a subprocedure is a procedure defined in machine language. ________

45) In the logo language you can define new procedures using primitives. ________

46) In the LOGO language procedures are automatically saved in a file. ________

47) In the LOGO language naming a procedure is equivalent to naming a file. ________

48) APPLICATION SOFTWARE REFERS TO

A) all software.
B) programming languages.
C) pre-packaged programs designed to do specific things.
D) both B and C.
E) all of the above.

SPECIFY WHETHER OR NOT EACH OF THE FOLLOWING IS CONSIDERED APPLICATION SOFTWARE BY WRITING **YES** OR **NO** IN THE SPACE PROVIDED.

49) DOS ________

50) LOGO ________

51) SUPERCALC 3 ________

52) WORDSTAR ________

53) EXPLORING THE IBM PC ________

54) DISKETTES ________

55) CASSETTES ________

56) RAM ________

57) ROM ________

58) CPU ________

59) IN A DATABASE MANAGEMENT SYSTEM YOU WILL NOT FIND WHICH OF THE FOLLOWING FEATURES.

A) Create a database where the information is organized according to the users' needs.

B) Create graphs using fields and record in the database.

C) reorganizing the information of the database

according to new specifications.
D) add, delete, correct and update existing information in the database.

60) A FIELD WIDTH IN DBASE III CORRESPONDS TO

A) The length of the longest record.
B) The length of the field name.
C) The record number.
D) Eight characters.

WRITE **TRUE** OR **FALSE** BESIDE EACH OF THE FOLLOWING.

61) You absolutely need the Assist menu to create a database. ______

62) Any field with one or more numbers in it is necessarily a numeric field. ______

63) The display functions cannot be used with the Assist Menu. ______

64) With DBASE III index files are created based on your field names. ______

65) WHICH OF THE FOLLOWING IS NOT A FIELD TYPE FOUND IN DBASE III.
a) Date
b) Character
c) Time
d) Memo

WRITE **TRUE** OR **FALSE** BESIDE EACH OF THE FOLLOWING.

66) Designing a program simply means planning the math which will lead to the solution of the problem. ___

67) Designing a program means planning the input and output which are required in the solution of the problem. ___

68) Designing a program means planning the logic and detailed steps which will lead to the solution of the problem. ___

69) A flowchart can be used to graphically illustrate the steps required in the program. ___

70) A program is a set of instructions that is written by the programmer which will direct the computer system to perform the operations necessary to solve the problem. ___

71) Violation of the rules regarding a programming language will cause a syntax error in the program. ___

72) Incorrect output from a program could be the result of a single coding error. ___

73) To enter a program into auxiliary storage, a programmer keys the program on the keyboard and the program is recorded directly on the auxiliary storage device. ___

74) A program is normally saved on auxiliary storage. ___

75) After a program is entered it is permanently saved in the CPU. ___

76) A BASIC program entered by the programmer must be translated from BASIC to machine language after being executed. ___

77) BASIC is sometimes called a machine language. ___

78) An interpreter translates a program one statement at a time into machine language ____

79) An object program resluts from an interpreter that translates a computer program. ____

80) Interactive processing involves entering from the keyboard and processing it immediately. ____

CIRCLE THE CORRECT ANSWER

81) WHEN FLOWCHARTING A PARALLELOGRAM IS USED TO REPRESENT

A) processing.
B) input/output.
C) decision points.
D) terminals.

82) THE SYMBOL USED ON A FLOWCHART TO REPRESENT PROCESSING IS A

A) parallelogram.
B) circle.
C) diamond.
D) rectangle.

83) A TERMINAL SYMBOL ON A FLOWCHART REPRESENTS

A) The beginning or ending of a program.
B) An entry from or an exit to another part of the program.
C) A comparing operation that is to be made.
D) That all data has been made available for processing.

84) THE BASIC INTERPRETER IS OFTEN STORED

A) In the CPU.
B) In ROM.
C) On the printer.
D) Using a keyboard.

85) IN **BASIC**, LOOPING

A) is the execution of a series of instructions one time.
B) is seldom required when programming.
C) may be used to allow one set of instructions to process many records.
D) is performed by the CPU every time an instruction is executed.

86) LINE NUMBERS

A) can begin with 1 and be incremented by 1.
B) must begin with the number 100 and be incremented by 10.
C) must begin with the value 10 and be incremented by 10.
D) must begin with 100 and be incremented by 100.

87) A **REM** STATEMENT

A) identifies the entire statement as a remark.
B) causes no operation to be performed by the computer.
C) allows comments to be written in a basic program.
D) all of the above.

88) WHICH OF THE FOLLOWING IS A VALID DATA STATEMENT TO REFERENCE THREE FIELDS WHICH CAN BE REFERENCED BY THREE VARIABLE NAMES?

A) 200 714, 555-1232, SAM HORN
B) 200 DATA 714, "555-1212", SAM HORN
C) 200 DATA 714, 555-1212, SAM HORN
D) 200 DATA "714" 555-1212 SAM HORN

89) WHICH STATEMENT BELOW CONTAINS VALID NUMERIC VARIABLE NAMES?
A) 100 READ 1A,2B,3C
B) 100 READ A1,B2,C3
C) 100 READ A$,B$,C$
D) 100 READ 1A,B1,C$

90) A STRING CONSTANT IS

A) A single numeric digit.
B) A series of numeric digits.
C) Any constant containing a non-numeric character.
D) Any character in a data statement.

91) WHICH OF THE FOLLOWING IS A VALID **READ** STATEMENT?

A) 200 READ "A, T$, N$"
B) 200 READ "A T$ N$"
C) 200 READ A T$ N$
D) 200 READ A, T$, N$

92) WHICH OF THE FOLLOWING IS A VALID **IF** STATEMENT?

A) 100 IF N$ = "END OF FILE" THEN 370
B) 100 IF N$ = END OF FILE THEN 370
C) 100 IF END OF FILE THEN 370
D) 100 IF N = "END OF FILE" THEN 370

93) WHICH OF THE FOLLOWING STATEMENTS WILL PRINT A BLANK LINE?
A) 100 PRINT " "
B) 100 PRINT "BLANK"
C) 100 PRINT BLANK
D) 100 PRINT "BLANK LINE"

94) THE **GOTO** STATEMENT IS

A) The first statement found in a loop.
B) The last statement in a program.
C) Used to transfer control to the statement whoseline number appears after the word GOTO.
D) Used to transfer control to the beginning of the program.

95) VALID MATHEMATICAL OPERATORS ARE

A) +,-,*,/,^
B) +,-,*,-:-,^
C) +,-,*,/,
D) +,-,x,-:-,**

96) WHICH OF THE FOLLOWING **INPUT** STATEMENTS IS NOT VALID?

A) 340 INPUT A
B) 340 INPUT B
C) 340 INPUT "A", "B"
D) 340 INPUT A, B

97) IN THE FOLLOWING EXAMPLE:

410 FOR Y = 1 TO 100

A) Execution of the program will terminate because the STEP entry has been omitted.
B) An error message will be displayed when the statement is entered because there is no STEP entry.
C) The value in the counter-variable field Y will remain at 1 each time the for statement is executed.
D) The value 1 will automatically be added to the counter-variable field Y.

98) THE **DIM**ENSION STATEMENT IS USED TO

A) Create storage space on auxiliary storage for an array.
B) Create storage space in main computer storage for an array.
C) Define the size of main computer memory used for a program.
D) Define the data to be contained in an array.

99) THE **DIM**ENSION STATEMENT CAN BE USED TO DEFINE

A) Arrays containing numeric data.
B) Arrays containing string data.
C) Multi-dimension arrays.
D) All of the above.

100) THE SEQUENCE FOR EXECUTING A SUBROUTINE CONSISTS OF THE FOLLOWING STEPS

A) The subroutine is called; instructions within the subroutine are executed; program execution is terminated.
B) The subroutine is called; instructions within the subroutine are executed; control returns to the first statement in the program.
C) The subroutine is called; instructions within the subroutine are executed; control returns to the statement following the gosub that called the subroutine.
D) The subroutine is called; instructions within the subroutine are executed; control returns to the end of the program

Appendix J

Evaluation codes for test items and assignments.

Evaluation of Pretest/Postest test items.

The entire test was evaluated on a total score of 50 marks.

- Questions 1 and 2 were worth 4 marks each and a three point criteria was used to evaluate them. Students were awarded two marks for including any one of the three points in their answer and one subsequent mark for each of the others. These items were criterion referenced to objectives one through four. Mastery was set at achieving 3 out of the four possible marks.

- Questions 3, 4, 5, and 6 were evaluated on a strictly correct or incorrect basis with items 3 and 4 being worth 2 marks each, item 5 worth 1 mark, and 1 mark for each answer given in item 6. These items were referenced to objectives 5, 6, and 7. Mastery was set at 100%.

- Each of the five items in Question 7 were worth 3 marks and were again evaluated on a three point criteria. Students were awarded three marks for stating the output of the program correctly, 2 marks for stating an output which was incorrect obviously as a result of a calculation error, and 1 mark for stating an incorrect output stemming form a procedural error in their assessment of the workings of the program. No marks were awarded for unanswered or partially answered questions or if the student stated an output which involved both procedural and calculation errors. Mastery was set at 10 out of a possible 15 marks.

- Each of the eight items in Question 8 were worth 1 mark and were evaluated on a correct or incorrect basis. However some notice was taken of correct specification of proper syntax as opposed to simple identification of erroneous syntax. Mastery was set at 5 out of 8 correct answers.

- Question 9 was worth 3 marks. One mark was awarded for responses mentioning the use of a matrix to table information, two marks were awarded for mentioning the use of subscripted variables to designate addresses within such a matrix, and three marks were awarded for mentioning the DIM statement as a means to defining subscripted variables. Mastery was set at 2 marks out of the possible 3.

- Assignments 1 (flowcharting), 2 (editing), and 3 (listhandling program) were evaluated on the basis of the following codes,

ASSIGNMENTS 1 and 3 0 = not acceptable

1 = acceptable because the flowchart/listhandling program is logically valid but there are logical errors in the solution to the problem itself.

2 = both flowchart/listhandling program and problem solution are logically valid.

ASSIGNMENT 2

0 = not acceptable

1 = acceptable

Appendix K

Non-parametric test results.

TEST RESULTS ON PERFORMANCE GAINS FOR OBJECTIVES 1 THROUGH 10

WILCOXON RANK SIGN TEST

OBJECTIVE		NUMBER	RANK	MEAN RANK	Z	Z CORRECTED FOR TIES	NUMBER OF TIED GROUPS
1	-	32	528	16.5	-4.937	-5.055	4
	+	0	0	--			
2	-	21	231	11	-4.015	-4.583	1
	+	0	0	--			
3	-	21	241.5	11.5	-3.734	-4.264	1
	+	1	11.5	11.5			
4	-	34	658	19.353	-5.106	-5.17	4
	+	2	8	4			
5	-	32	548	17.125	-4.78	-4.998	2
	+	1	13	13			
6	-	37	703	19	-5.303	-6.083	1
	+	0	0	--			
7	-	40	820	20.5	-5.511	-5.53	7
	+	0	0	--			
8	-	31	520	16.774	-4.787	-4.875	3
	+	1	8	8			
9	-	38	812	21.368	-5.403	-5.424	6
	+	2	8	4			
10	-	39	780	20	-5.442	-5.448	11
	+	0	0	--			