DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

U. S. ARMY RESEARCH INSTITUTE
FOR THE BEHAVIORAL AND SOCIAL SCIENCES

A Field Operating Agency under the Jurisdiction of the

Deputy Chief of Staff for Personnel

JOSEPH ZEIDNER
Technical Director

L. NEALE COSBY
Colonel, IN
Commander

NOTICES

DISTRIBUTION: Primary distribution of this report has been made by ARI. Please address correspondence concerning distribution of reports to: U.S. Army Research Institute for the Behavioral and Social Sciences, ATTN: PERI-TST, 5001 Eisenhower Avenue, Alexandria, Virginia 22333.

FINAL DISPOSITION: This report may be destroyed when it is no longer needed. Please do not return it to the U.S. Army Research Institute for the Behavioral and Social Sciences.

NOTE: The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: Technical Report 451	2. GOVT ACCESSION NO.: AD-A139 278
3. RECIPIENT'S CATALOG NUMBER:	
4. TITLE (and Subtitle): A CAI COURSE ON CONSTRUCTING PLANIT LESSONS: DEVELOPMENT, CONTENT AND EVALUATION	5. TYPE OF REPORT & PERIOD COVERED: Final Report February - July 1974
	6. PERFORMING ORG. REPORT NUMBER: TM-5364/000/00
7. AUTHOR(s): F. D. Bennik, W. G. Hoyt, & A. K. Butler (SDC); J. D. Baker (ARI)	8. CONTRACT OR GRANT NUMBER(s): DAHC19-73-C-0029
9. PERFORMING ORGANIZATION NAME AND ADDRESS: System Development Corporation Santa Monica, CA	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 2Q062106A721
11. CONTROLLING OFFICE NAME AND ADDRESS: U.S. Army Research Institute for the Behavioral and Social Sciences, 5001 Eisenhower Avenue, Alexandria, VA 22333	12. REPORT DATE: June 1980
	13. NUMBER OF PAGES: 145
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office): U.S. Army Research Institute for the Behavioral and Social Sciences 5001 Eisenhower Avenue Alexandria, VA 22333	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for open release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

—

18. SUPPLEMENTARY NOTES

—

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Automated Instruction	Army Tactical Data Systems
Computer-assisted Instruction	Embedded Training
PLANIT Author Language	Military Training

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Army's current emphasis on providing ET-capability (Embedded Training) in its tactical data systems has produced a requirement for the development of an Army in-house capability to produce its own courseware; to update and modify contractor developed automated instruction materials and to provide a quality assurance capability for user assessment of contractor delivered TACFIRE ET modules.

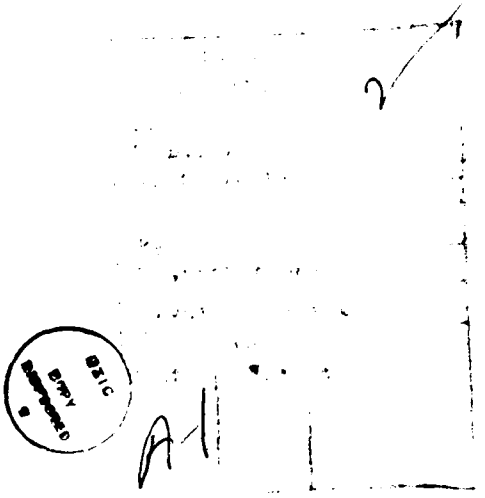
(continued on back)

Item 20 (Continued)

The research approach taken here was to develop a courseware package on how to program lessons for an automated system. Since PLANIT (Programming Language for Interactive Teaching) is the student/author language for providing ET Modules for Army tactical data systems, such as TACFIRE, PLANIT was chosen as both the course and the vehicle of instruction. The research effort had two phases: (1) to develop a courseware package to teach subject matter experts, who are novices with respect to computer-assisted instruction (CAI), how to program PLANIT lessons, and (2) to evaluate the effectiveness of the package for select Army users.

The resultant courseware package was found to be successful in teaching people who are subject matter experts, but who are CAI novices, to program lessons over a diverse set of topics and to execute and edit their lessons on-line. The majority (93%) of the participants strongly indorsed this method for learning how to develop instructional materials.

As a result of this study the Army now has an in-house capability to train its personnel to prepare, execute, edit and update PLANIT courseware. This courseware package now is also included at the front-end of the PLANIT operating system tapes distributed to other military groups, academia and industry.



Technical Report 451

A CAI COURSE ON CONSTRUCTING PLANIT LESSONS: DEVELOPMENT, CONTENT, AND EVALUATION

Fred D. Bennik, William G. Hoyt, and Alfred K. Butler
System Development Corporation
and
James D. Baker
Army Research Institute

Submitted by:
James D. Baker, Chief
MANPOWER AND EDUCATIONAL SYSTEMS TECHNICAL AREA

Approved by:

Edgar M. Johnson, Director
ORGANIZATIONS AND SYSTEMS
RESEARCH LABORATORY

U.S. ARMY RESEARCH INSTITUTE FOR THE BEHAVIORAL AND SOCIAL SCIENCES
5001 Eisenhower Avenue, Alexandria, Virginia 22333

Office, Deputy Chief of Staff for Personnel
Department of the Army

June 1980

Army Project Number
2Q062106A721

· Embedded Training

ARI Research Reports and Technical Reports are intended for sponsors of R&D tasks and for other research and military agencies. Any findings ready for implementation at the time of publication are presented in the last part of the Brief. Upon completion of a major phase of the task, formal recommendations for official action normally are conveyed to appropriate military agencies by briefing or Disposition Form.

FOREWORD

The Manpower and Educational Technology Technical Area of the Army Research Institute for the Behavioral and Social Sciences (ARI) conducts on-going research on computer-based educational systems (Army Project 2Q162722A791, FY 80) and training simulation (Army Project 2Q163744A795 FY 80). The forerunner of the current research is described in this report.

An effort in the Command Systems Work Unit of ARI had been designed to optimize commanders' use of Army tactical data systems for command and staff information processing and decision making, by developing software packages that would use the actual system as the instructional vehicle for training users and maintaining their proficiency. The research reported here stemmed from that effort.

The initial research which examined the potential for tactical data systems to be used in a computer-assisted instruction (CAI) mode to support MOS 11B40 infantry training at the combat unit level was conducted as MASSTER Test FM 122 at Fort Hood, Texas. The particular problem area was selected when the training 11B40 soldiers came up as a critical item in the report of the Board for Dynamic Training in 1971 and in the Continental Army Command (CONARC) Task Group on Computer-Assisted Instruction in 1972. As spinoff from MASSTER Test 122, a requirement surfaced for the Army to acquire its own in-house capability to provide CAI courseware. This requirement led to the research reported here.

ARI's programs are conducted as in-house research augmented by contracts with organizations having unique capabilities in the area. Much of this experiment was conducted by personnel of the System Development Corporation (SDC) under contract DAHC19-73-C-0029. The entire effort responded to requirements of Project 2Q062106A721, Human Performance in Military Systems, FY 1973 Work Program, and to special requirements levied by what were then Assistant Chief of Staff for Force Development and the Director of Army Research, Office of the Chief of Research and Development. Current programs are responsive to requirements of Army Training and Doctrine Command (TRADOC), the successor to CONARC.

Acknowledgement should also be made of the contributions by the civilian scientists at the U.S. Army Research Institute for the Behavioral and Social Sciences and the military personnel at the U.S. Army Infantry School. Sincere thanks go to Mr. John Larson, Dr. Michael Strub and Mr. Cecil Johnson, of ARI, and Colonel B. E. Rutherford, Lt. Colonel Russ Baker, Mr. G. J. Schnabl, and Major Daniel Boone of The Infantry School. Appreciation is also expressed to Mr. Harold Holoter and Ms. Jeanette Condon, SDC, for their contributions to this project.

JOSEPH ZEIDNER
Technical Director

A CAI COURSE ON CONSTRUCTING PLANIT LESSONS: DEVELOPMENT, CONTENT AND EVALUATION

BRIEF

Requirement:

The U.S. Army has been developing highly complex tactical data processing systems which will produce considerable computer capability in the field. As tactical data systems become part of the Army's inventory, there are immediate and sustained requirements for skilled personnel to operate these systems. To meet this primary requirement, ARI has developed the ET-concept (embedded training). Embedded training, in the context of training tactical data system users, means using the tactical computer itself to train the user on how to use the system. But as doctrine changes, tactics are updated and new weapon systems enter the inventory, a secondary requirement surfaces: How to develop an Army in-house capability to update and expand embedded courseware, as well as how to develop new course materials as required.

Procedure:

The approach taken was to develop courseware on how to program lessons for an automated instructional system. The target system chosen both as the course and the vehicle of instruction was PLANIT (programming language for interactive teaching). The goal was a PLANIT course on PLANIT, or PLANIT^2. The research effort had two phases: (1) to develop a courseware package to teach people who are subject matter experts but who are novices with respect to computer-assisted instruction (CAI) how to program PLANIT lessons; and (2) to evaluate the effectiveness of the package for two groups of subject matter experts selected from ARI and the Infantry School at Fort Benning, GA.

Findings:

The PLANIT^2 course was found to be successful in teaching students to program CAI lessons and to execute and edit the lessons on line. The majority (93%) of the military and civilian students who participated in the project approved of this method of learning how to develop instructional materials using PLANIT. Some students taking the PLANIT^2 course reported problems in organizing their lessons, preparing answer strategies, and controlling the student performance track--problems commonly inherent in individualized instruction where the lesson materials, rather than the instructor, carry the instructional burden. In other words, PLANIT^2 provides a means for developing instructional programs which will execute

flawlessly all of the tools available within PLANIT but it does not guarantee that the materials will teach. Additional work is necessary to provide subject matter experts with the skills necessary to produce appropriate strategies for presentation, answer processing, testing, feedback and control of the student learning process.

Utilization of Findings:

As a result of this study, the Army now has the capability to train its military and civilian personnel on how to prepare, execute and edit (update) PLANIT courseware. This capability subsequently has been successfully used by personnel at the Field Artillery School, Fort Sill, OK to provide quality assurance checks on TACFIRE Embedded Training modules. Additionally, PLANIT^2 now is also included on the front-end of the PLANIT operating system tapes distributed within the military, industry and academia.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

A CAI COURSE ON CONSTRUCTING PLANIT LESSONS: DEVELOPMENT, CONTENT, AND EVALUATION

Section 1. INTRODUCTION AND SUMMARY

A. PROJECT BACKGROUND AND PURPOSE

Army tactical data systems such as TACFIRE and TOS[2] are prime contenders for incorporation of a computer-assisted instruction (CAI) capability to augment the decentralized training requirements and to provide training resources on the computer at the tactical unit level. Along with the thrust toward integrating automated instruction into Army tactical systems, a continuing trend exists at a variety of Army schools, including the U.S. Army Infantry School (USAIS) at Fort Benning, Ga., the Signal School at Fort Gordon, Ga., and the Command and General Staff College at Fort Leavenworth, Kans., to integrate computers in support of individual instruction and instructional management functions.

MASSTER Test 122 conducted at Fort Hood, Tex., showed that computer-assisted instruction (CAI) could be conducted on tactical computers to train combat personnel in weapons and tactics. The USAIS subsequently used these weapons and tactics courses in its audiovisual library at the Individual Learning Center. From this experience, certain observations peculiar to Army CAI needs have resulted. For example, as Army tactical doctrine is changed, tactics and weapons employment is updated. The course materials need to be updated and expanded as these changes are made, and new course materials need to be developed to take advantage of the current capability for computer-assisted instruction.

Recognizing this need for recurring modification and the Army's desire to develop capability to produce CAI lesson materials, the present effort was undertaken. The objective was to develop courseware on how to program lessons for an automated instructional system. The target system chosen both as the course and the vehicle of instruction was PLANIT (Programming Language for Interactive Teaching): More specifically, the Army Research Institute (ARI) version 1.1 of PLANIT, currently operating on Army UNIVAC and Control Data* computers, was used. Because it is specially designed for machine transferability, PLANIT also operates on a variety of computers and operating systems in university, government, and military instructional settings in the United States and overseas. The Army and the National Science Foundation have supported recent improvements and extensions of PLANIT software capability, and individual CAI groups continue to install and improve PLANIT for their needs.

---

*Commercial equipment and materials are identified by trade names to provide a precise description. Their use does not constitute endorsement or approval by ARI or the Department of the Army.

The research effort had two phases: (1) to develop a courseware package to teach subject matter experts, who are novices with respect to CAI, how to program PLANIT lessons; and (2) to evaluate the effectiveness of the package for two groups of subject matter experts selected from ARI and the USAIS staff at Fort Benning. Specific attention was directed toward identifying courseware capabilities for teaching the basic PLANIT programming language needed to develop instructional programming material, the time required to negotiate the course of instruction, and the user acceptance of this mode of instruction.

B. VALUE AND IMPORTANCE OF THIS STUDY

ARI has successfully installed the CAI system PLANIT on the DEVTOS tactical computer system at Fort Hood, on the UNIVAC 1108 computer system at Edgewood Arsenal, Md., and on the CDC 3300 computer system at ARI. PLANIT is a versatile and powerful CAI system which is machine transferable (computer independent) and has been installed at a number of universities in the United States and foreign countries.

MASSTER Test 122, conducted on the DEVTOS tactical computer system, has shown that CAI is an effective and highly acceptable method of training combat personnel in weapons and tactics.

A logical next step is to provide the Army with an in-house capability to develop their own courseware programmed in PLANIT, and modify it as tactical doctrine changes. The present weapons and tactics courses and future courses operate on any computer which has the PLANIT CAI system. In some cases minor adjustments would be needed to fit the parameters of the particular computer system's CRT display presentation, e.g., line length or number of lines presented. PLANIT has the capability to easily and quickly make these adjustments.

As a result of this study, the Army now has an in-house capability to train military personnel to prepare, execute, and edit (update) PLANIT courseware.

C. SCOPE OF THE REPORT

This document describes the courseware development and the courseware package (Section 2), the procedures for the two field evaluations at ARI and USAIS (Section 3), the results from the two field evaluations (Section 4), and the conclusions and recommendations of the study (Section 5).

Section 2: DEVELOPMENT OF PLANIT COURSEWARE

Sufficient hours of PLANIT CAI lessons and adjunct materials were to be developed consistent with the following goals:

- Teaching subject matter experts, naive with respect to PLANIT, both at ARI and Fort Benning, the PLANIT language capabilities to permit them to construct rudimentary lessons in subject areas of their own choice.

- Providing a resource to acquaint users with features available in PLANIT.

- Providing a practical exercise to assess the extent to which the first goal could be met with researchers at ARI and with USAIS instructors and staff at Fort Benning.

The first and third goals required selection of a basic subset of the PLANIT language to teach at a sufficient level of detail so students could demonstrate competency through a practical exercise in instructional programming. The second goal implied the ability to learn about a wider variety of PLANIT capabilities, albeit at a level of competence greater than that required for the practical exercise. All of these goals were aimed at student populations at ARI and at Fort Benning, which are diverse with respect to educational level, years of experience, type of work experience, and other background variables.

From March to May 1974, development of the PLANIT courseware proceeded through the course design steps: (1) preliminary analysis; (2) development of course outline, objectives, and a practical exercise for evaluation of instructional programming skills; (3) ARI review and revisions; (4) preparation of CAI lessons; (5) preparation of adjunct materials; (6) delivery of courseware materials; and (7) online tryout and editing of the courseware package at ARI. These design and development steps are described below, followed by a description of the PLANIT lessons resulting from these steps.

A. COURSE DESIGN

1. Preliminary Analysis

A preliminary analysis determined the scope, content, and sequencing of PLANIT subject areas and topics. PLANIT is a highly versatile instructional programming language with English-like statements and commands that permit users to construct, enter, try out, check, and edit lessons. PLANIT contains a computational language equivalent in power to several popular programming languages. This computational language can be used directly by any PLANIT user, by instructional program authors in PLANIT frames, or by students during interaction with a lesson. There are more than 100 separate components in the PLANIT user language. This required differentiation of those components essential for a basic ability to write instructional programs from those that offer an extended capability.

First, basic PLANIT skills were defined to enable the students to construct, enter, try out, check, and edit instructional programs in subject areas of their choice using PLANIT. Several ground rules were derived to delimit this analysis:

- The PLANIT course would teach and evaluate practical uses of the PLANIT language.

- The practical exercise in instructional programming would require students to construct up to 20 frames for displaying information, eliciting responses, providing feedback, and specifying instructional decisions.

- The student's choice of subject matter for the practical exercise would not be constrained, but requirements would be specified that the student's instructional program should meet with respect to the PLANIT capabilities incorporated. In this way, the practical exercise would relate directly back to the material taught by the lessons as well as provide the same evaluation baseline for the ARI and USAIS student groups.

Within these basic ground rules PLANIT was further analyzed to determine:

- The general sequence in which PLANIT components are normally used by lesson authors.

- The order in which the PLANIT language elements logically build upon one another.

As a result of this initial analysis, the basic subject areas and topics as shown in Table 2-1, headings A, B, C, and D, were defined. These are the functional elements and skills necessary to construct rudimentary instructional programs.

Two other requirements demanded further analysis of topics that would build upon the basic skills by covering some of PLANIT's more specialized capabilities:

- A requirement to familiarize PLANIT users with some of the versatile answer-matching processors, action control options, decision control options, and PLANIT commands.

- A requirement to familiarize users with capabilities of the (P)rogramming frame, which requires a familiarity with PLANIT decision statements and portions of the PLANIT computational language (CALC).

In particular, the second requirement required considerable analysis to determine which elements of CALC should be selected to give novice PLANIT users an understanding of their use in general lesson construction and in the (P)rogramming frame. Apart from PLANIT decision capabilities and those CALC elements selected, the only new items required for the (P)rogramming frame were (1) how to branch line-to-line within the frame and (2) how to return from a (P)rogramming frame to the main flow of the lesson.

TABLE 2-1. OUTLINE OF SUBJECT AREAS AND TOPICS

A. BASICS OF PLANIT LESSON CONSTRUCTION

1. Introduction to PLANIT and PLANIT frame types
2. Constructing (Q)uestion frames
3. Constructing (M)ultiple choice frames
4. Constructing (D)ecision frames

B. ENTERING LESSONS FROM A TERMINAL

1. Signing-on as a lesson author
2. How to make frame entries
3. How to display frame entries
4. How to name a lesson and sign-off

C. RUNNING AND CHECKING A LESSON

1. Types of checking during tryout
2. How to execute the lesson
3. Handling execution error messages

D. BASICS OF LESSON EDITING

1. Reasons for frame editing
2. Switching to editing from other activities
3. How to replace material
4. How to insert material
5. How to add-on to material
6. How to delete material

TABLE 2-1. OUTLINE OF SUBJECT AREAS AND TOPICS (Continued)

E. SPECIAL CAPABILITIES AND COMMANDS

1. Overview of special capabilities
2. Detecting two or more key words
3. Procedure for a series of student answers
4. Group 4 action options and controls
5. (D)ecision frame options and controls
6. How to Break lesson execution
7. How to Search for and Modify frame entries

F. BASICS OF (CALC)ULATION AND THE (P)ROGRAMMING FRAME

1. CALC overview
2. Methods of entering and exiting CALC
3. CALC arithmetic
4. CALC assignment statements
5. Evaluating CALC expressions
6. Storing CALC data in tables
7. Displaying CALC data
8. Repetitive operations and subscripting
9. Purpose of the (P)rogramming frame
10. Structure of a (P)rogramming frame
11. Elements unique to (P)rogramming frames
12. Operation of a (P)rogramming frame

As a result of the second analysis, the expansion of PLANIT language coverage determined most useful for the lessons is shown in Table 2-1, headings E and F.

2. Course Objectives and Practical Exercises

In accordance with the broad subject area and topic coverage shown above, a detailed course outline was developed that showed task and subtask learning objectives associated with each step in the outline. The PLANIT Course Outline showing the task objectives and order in which the PLANIT language components are introduced is contained in Appendix A. Concurrently, requirements for the practical exercise in instructional programming were established pending review and recommendations from ARI.

3. ARI Technical Review and Revisions

ARI personnel reviewed two products of the course design steps--the PLANIT Course Outline covering the basic PLANIT topics and a draft of student instructions for the practical exercise in instructional programming--in March 1974. At this meeting, working agreements were reached on the scope and content of lesson materials, requirements for the practical exercise, types of student adjunct materials for the practical exercise, and dates for the field evaluations at ARI and the USAIS. Plans were also discussed for evaluation criteria, student scheduling, equipment arrangements, and logistics.

Subsequently, further ARI review of the expanded course outline covering PLANIT special capabilities and commands, CALC, and the Programming frame resulted in the following changes:

- The commands BREAK, Search, and Modify were substituted for commands DISPLAY, LIST, and PUNCH in Part V.6 of the outline. ARI already has a procedures handout for use of the DISPLAY command, and the commands LIST and PUNCH are not available for use at USAIS.
- The control word RELATED was added to Part V.4.c of the outline.
- Various changes were made to control symbols and key names peculiar to the UNIVAC installation and ADDS Consul 880 terminal.

In addition, as courseware development proceeded after incorporating the above changes, it was decided to cover the main steps in planning a lesson (outline Part I.5) in a student handout rather than the online lessons. Minor changes were also made to requirements of the practical exercise to remove material requiring the use of KEYWORD ALL and KEYWORD(number).

B. COURSE DEVELOPMENT

ARI's review resulted in a final selection of PLANIT topics and language elements. Subsequently, the work focused on developing CAI lessons and adjunct materials for the topics shown in the Course Outline. The work was

originally estimated to comprise 400 to 600 frames and to provide 3 to 6 hours of online instruction and practice. The expansion of topic coverage and changes resulted in more than 800 frames, providing an estimated 8 to 16 hours of online instruction.

1. Preparation of CAI Lesson Materials

Each topic and task objective specified on the Course Outline became a basic instructional production unit. For each task objective, the associated course topics and subtopics determined a basic instructional sequence. The sequence proceeded from one task objective to the next, with enabling objectives appropriately interspersed, as determined by prerequisite order of elements in the PLANIT language (reflected by the ordering of topics in the course outline).

A series of frames were prepared in conjunction with each task or subtask objective in the course sequence. Enabling test items were interspersed to diagnose individual student difficulties and provide immediate remedial help. Criterion performance items ended each task objective sequence so as to either gate students forward or to skip them back for review of earlier material. Each frame in the PLANIT course sequence was designed to perform one or more of the following functions:

- To present PLANIT content information, examples of PLANIT frame code, examples of student interaction with PLANIT frames, practice problems, test items, instructions, or lesson control choices to the students.
- To evaluate student responses as correct, incorrect, neutral, or unanticipated.
- To provide feedback messages appropriate to the category of response and, in many cases, to the correct or incorrect response given.
- To decide on the next lesson control action to be taken, i.e., prompt another response, proceed ahead in sequence, skip elsewhere in the lesson, or skip to another lesson.

Lesson authors exercised these basic frame capabilities, using the character presentation, answer matching, and lesson control statements of PLANIT (ARI Version 1.1). In addition to the lesson authors' own experiences with PLANIT, the following reference materials were used as resources for PLANIT information, content examples, and instructional style:

PLANIT Author's Guide, System Development Corporation (SDC) TM-4422/001/01, 1 October 1970.

PLANIT Language Reference Manual, SDC TM-4422/002/01, 1 October 1970.

Application of Tactical Data Systems for Training, Final Report, Volume II - AI/DEVTOS Automation Studies (Section 4 and Table 4-1), SDC TM-5261/001/00, 2 January 1974.

PLANIT CAI lesson listings for Light Antitank Weapon instruction.

PLANIT briefing charts for PLANIT course taught by SDC.

Michigan State University partial lesson on PLANIT, provided by ARI.

Listing of PLANIT lesson INTRO, provided by ARI.

PLANIT messages listing for PLANIT Version 1.1 cold start deck, provided by ARI.

In creating and sequencing the frames providing content information, several ground rules of instructional technique were established:

- To maintain a functional context training approach by introducing PLANIT frame types and commands in the order they are normally used, and by introducing PLANIT language elements in the order that they build upon one another.

- To provide to the student a preview of the instructional topics sequence.

- To inform the student how well he has done on sets of subgoals and on each frame.

- To provide clear instructions, avoiding ambiguity.

- To keep information and feedback as concrete and specific as possible.

- To avoid the use of programming or PLANIT jargon that does not aid in lesson writing performance with PLANIT.

- To use content examples of PLANIT frame entries and execution interaction that would be relevant for both ARI and Fort Benning students.

- To maintain simplicity and continuity of example content by using basic examples as building blocks, so that the focus remains on PLANIT rather than on the content of a particular example.

Frame sequences were developed for teaching students how to construct PLANIT frame entries or to use PLANIT commands. These sequences incorporated some or all of the following steps:

1. Brief statement of purpose or function of the PLANIT language element.

2. Presentation of an example showing use of the frame entry or command and its effect on lesson execution or on PLANIT's next response.

3. Tutorial questions related to the example, to bring out key points.

4. Identification of the general entry form for the language element.

5. Practice entries using the language element for situational problem requirements, or for responding in a particular way to a given PLANIT message.

6. Constructing one or a series of entries to meet the requirements of a situational problem designed to assess learning for the task objective.

Where the requirement for Steps 5 and 6 was for the student to provide a series of constructed frame entries, the verification and feedback technique most often used was to have the student first work out the construction on a PLANIT frame worksheet. A spot-check of this frame construction followed, having the student enter certain lines. This was followed by a full verification, allowing the student to compare his or her worksheet with a completed online example and to modify the worksheet as required. When the requirement for Step 5 and 6 was for the student to use a PLANIT command, the entire construction was usually obtained and evaluated in a simulated interaction with PLANIT online messages.

The CAI applications software used for encoding the online instructional materials was PLANIT Version 1.1, as installed by ARI on the Edgewood Arsenal UNIVAC 1108 under the Exec 8 operating system. This UNIVAC PLANIT installation incorporated nearly the same functional capabilities as the PLANIT Version 1.1 which had been installed earlier on CDC 3300's at ARI and at Fort Hood. The only major difference encountered in functional capability between the CDC and UNIVAC Version 1.1 installations was in the relative permanence of storage with respect to PLANIT user files (lessons and records). Because this difference did not directly affect lesson development and construction, it is not dealt with here.

The completed instructional materials, in the form of card decks and listings, were delivered to ARI, using the IBM 029 character set. The syntax of the instructional frame entries adhered to rules and conventions described in the PLANIT Author's Guide and PLANIT Language Reference Manual. ARI converted the IBM 029 card decks to UNIVAC 1108 punched cards. ARI then used the PLANIT card-to-disk lesson building capability via remote card reader to generate the PLANIT lessons onto the Edgewood 1108 for online use at ARI and Fort Benning consoles. Figure 2-1 shows the 029 punch codes SDC used for PLANIT special characters having different punch codes on the UNIVAC, and the character substitutions for those PLANIT symbols not available on the 029 punch.

The course checkout and the ARI and USAIS evaluations were completed using the ADDS Consul 880 display terminal and keyboard, which permitted a usable display area of 21 rows and 78 columns. However, at ARI's request, course authors limited presentations to 17 rows by 48 columns to accommodate the Control Data 211 display console available at ARI and at other PLANIT installations with CDC equipment. Therefore, all lesson construction was done on 48-column PLANIT frame coding sheets to insure that lesson presentations would not exceed the smaller of the two display screen sizes. Two of the lesson construction techniques accommodating the display limitation were (1) to insert frames where the student had to respond with GO to advance to the next display and (2) to provide internal lesson loops that would repeatedly rewrite an example on the screen while fetching a new question or information frame about the example.

PLANIT SPECIAL CODE	SDC CHARACTER[1]	IBM Ø29 PUNCH CODE
+	+	12-6-8
-	-	11
*	*	11-4-8
/	/	Ø-1
(	(	12-5-8
)	)	11-5-8
"	"	12-3-8
%	%	Ø-4-8
=	=	6-8
,	,	Ø-3-8
:	:	2-8
;	;	11-6-8
'	'	5-8
#	#	3-8
?	?	Ø-7-8
$	$	11-3-8
\[2]	¬	11-7-8
↑	|	12-7-8
←	<	12-4-8
	![3]	11-2-8
	&	12

[1] Alphanumeric characters and blank are not depicted since their keypunch code is common among IBM, UNIVAC, and CDC systems.

[2] Since the characters \, ↑, and ← are not available on the IBM 029 punch, SDC made the indicated substitutions.

[3] The characters ! and & required special handling by the ARI character conversion software.

Figure 2-1. IBM 029 Character Set for PLANIT Lessons

When a set of frames was completed on PLANIT coding sheets, it was submitted to keypunch operators for conversion to 029 punched cards, and a listing was then generated from each set of cards for in-house checking. The lesson author reviewed each listing for errors and logical inconsistencies. Figure 2-2 shows the listing format designed and programmed for lesson checking and debugging. The frame count, frame numbers, blocked entries, line counts, and card sequence numbers made it easier than a tightly packed listing format would to spot the types of errors that would prevent successful card-to-disk lesson building, and to insert correction cards into decks. Corrections made to the listings were keypunched and the card decks updated accordingly.

This production and quality control cycle was repeated until frames were prepared for all the task objectives within a course unit. Lesson authors grouped card decks of frames for each unit to form named PLANIT lessons from which another listing was produced and checked. PLANIT limits each named lesson to a maximum number of frames, depending upon the storage parameters set for a given PLANIT installation. However, there is no PLANIT limit to the number of named lessons which can be linked to form a logical lesson, unit, or course. Therefore, sometimes a logical lesson was named as two PLANIT lessons in the interest of staying within frame limitations. For example, the logically adjacent material dealing with Groups 3 and 4 of a PLANIT (Q)uestion frame is artificially divided between two named PLANIT lessons (PLANIT12 and PLANIT13).

For control purposes, frames between connecting PLANIT lessons were numbered from 1.00 through 91.00 in PLANIT1, upward to 101.00 through 186.00 in PLANIT12, to 710.00 through 838.00 in PLANIT6 and PLANIT61.

Frames representing embedded test items were labeled corresponding to task objectives on the course outline with a mnemonic formed from the identifier that appears on the course outline. For example, in lesson PLANIT5 the first frame of the topic on how to handle a series of student responses to a question is labeled STEPS. Then, the criterion test item corresponding to task 3.0 in the course outline is labeled STEPS30, while the four enabling questions leading up to this objective are labeled STEPS31, STEPS32. . . . In this manner, the frame labeling logic for the courseware serves two primary purposes:

- To mark the criterion and enabling test items for each topic in the lesson listings.

- To mark the start points of topics corresponding to topics in the course outline.

Frame labels marking the start point of topics are also punctuated with a period control character such that, should the PLANIT REVIEW function be made operable, when the student enters ←REVIEW, the labels would be presented according to the logic and conventions documented in TM-4422/001/01 (PLANIT Author's Guide, IV-58, 59).

Frame labels also served secondary purposes in lesson development as reference points, useful when the lesson listings are used to observe student progress, and as control points for branching internal to the lessons.

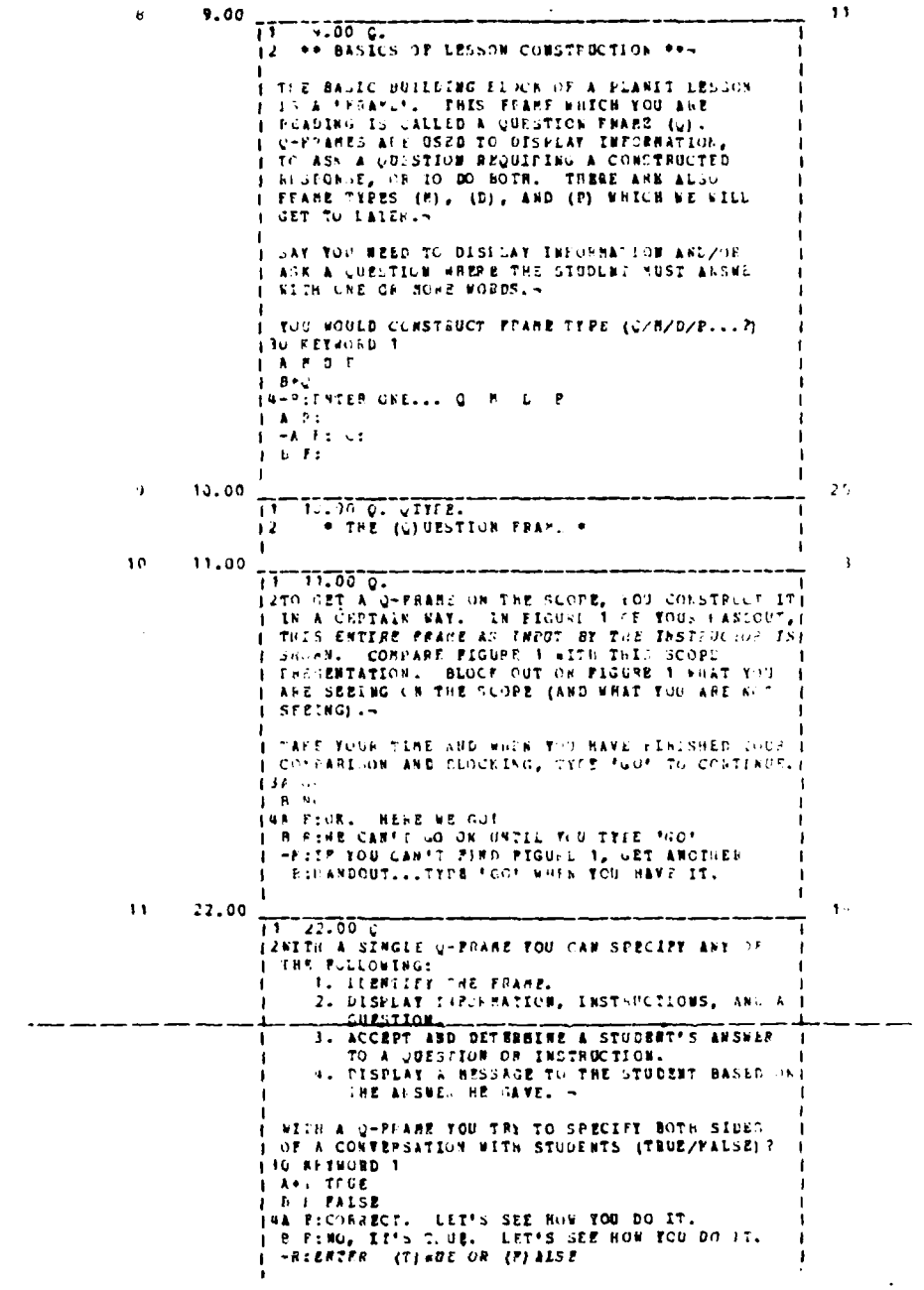


Figure 2-2. SDC Listing Format for Courseware Checkout

2. Preparation of Adjunct Materials

Adjunct materials were developed for students to use in conjunction with preparation of the CAI lesson materials. Five types of adjunct materials support both the online learning and offline/online instructional programming tasks and are described below.

a. Examples of PLANIT Frame Types

The CAI lessons introduce each of three PLANIT frame types--Q, M, and D--in sequence. The first example of each frame type a student encounters is contained in Figures 1, 2, and 3 and is referenced by the CAI lessons as follows:

	Adjunct Materials Figure	Lesson
1	Q-Frame Input and Interaction	PLANIT1
2	M-Frame Input and Interaction	PLANIT12
3	D-Frame Input and General Form	PLANIT13

The CAI lessons use these figures to aid the student to (1) identify the frame groups and group functions; (2) differentiate those frame entries which are displayed on the scope from those entries which are essential for lesson control but are not displayed; and (3) trace the order of execution among groups, lines, and individual statements on lines. Outside of these examples, all further examples of frame entries and execution are contained in the CAI lessons themselves. Thus, these initial hardcopy examples remind the students as they study the CAI lesson examples and their offline programming deskwork.

The example used for the M-frame is shown in Figure 2-3.

b. PLANIT Frame Worksheets

Students use worksheets to draft frame entries for frame types Q, M, and D according to practice and test item instructions provided by CAI lessons PLANIT1, PLANIT12, and PLANIT13. Each student has one set of 10 worksheets for this purpose. Thirty more worksheets are provided each student for the frame writing deskwork portion of the practical exercise. A PLANIT Frame Worksheet is shown in Figure 2-4.

c. Instructions for the Practical Exercise

At the end of CAI lesson PLANIT13, the student's attention is drawn to Figures 4 through 6 of the adjunct materials. These give the student practical exercise requirements and instructions. Figure 4 gives the PLANIT frame requirements for the student's lesson-writing exercise. Figure 5 is a diagram making explicit the decision logic requirement of the exercise, as an aid to sequencing and checking frames. Figure 6 is a reminder of the lesson writing, checking, online entry, execution, and editing steps the student follows in the practical exercise of instructional programming.

```
                                COLUMNS

                00000000001111111111222222222233333333334444444444
M-FRAME         12345678901234567890123456789012345678901234567890
AS INPUT
                1  2.00 M  TEST

                2WHAT TYPE OF FRAME IS THIS?
                 (CHOOSE A LETTER)

                3  A.   QUESTION
                   B.+  MULTIPLE CHOICE
                   C.   DECISION
                   D.   PROGRAMMING

                4ACD F:WRONG.  HINT- SEVERAL ALTERNATIVES
                 R:ARE DISPLAYED.  TRY AGAIN.
                 B F:YES, A (M)ULTIPLE CHOICE FRAME.




 M-FRAME        WHAT TYPE OF FRAME IS THIS?
DISPLAY &       (CHOOSE A LETTER)
INTERACTION
                  A.   QUESTION
                  B.   MULTIPLE CHOICE
                  C.   DECISION
                  D.   PROGRAMMING

                *QUESTION
                CHOOSE ONE OF THE ABOVE LETTERS
                *A
                WRONG.  HINT- SEVERAL ALTERNATIVES
                ARE DISPLAYED.  TRY AGAIN.
                *D
                WRONG.  HINT- SEVERAL ALTERNATIVES
                ARE DISPLAYED.  TRY AGAIN.
                *B
                YES, A (M)ULTIPLE CHOICE FRAME.
```

Figure 2-3. A PLANIT Frame Type Example

PLANIT FRAME WORKSHEET

Name _____ Lesson _____ Page _____ of _____

SS Number _____ Phone _____ Date _____

Organization _____ Date/Time: Entered: _____ Checked_____

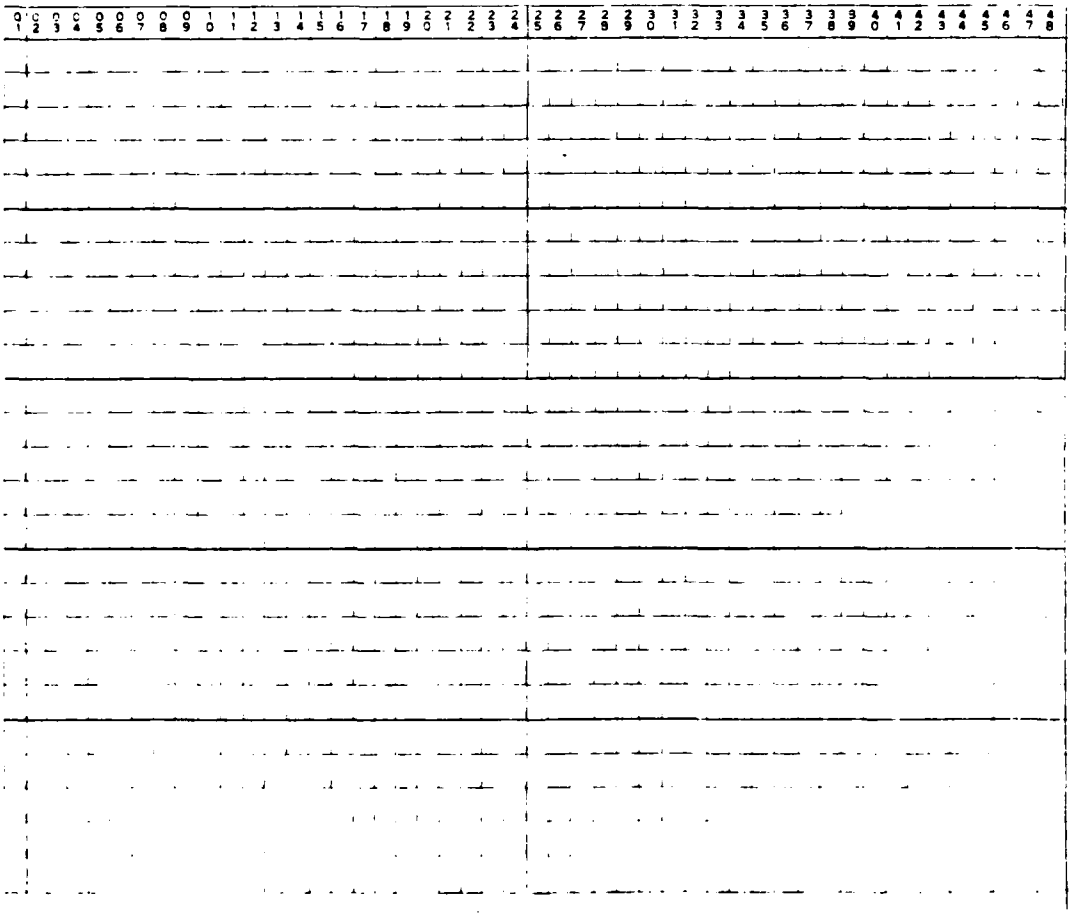


Figure 2-4. A PLANIT Frame Worksheet

Figure 2-5 shows the lesson logic diagram for the practical exercise.

d. Summary of PLANIT Frames

Exhibits 1 through 6 are a set of single-page job aids to remind the novice PLANIT programmer of the legal frame entries and entry formats for frame types Q, M, D, and P. When students sign off PLANIT at the end of lesson PLANIT13, the lesson reminds them that Exhibits 1 through 6 are available as aids. Figure 6 (the lesson exercise steps) also reminds them of this.

Exhibit 1 is an overview of the functional groups in each of the four PLANIT frame types. Exhibits 2 through 5 show the PLANIT language conventions that are available for each of the four frame types. Entry formats are shown in general syntax form, with terse explanations. Exhibit 6 summarizes how PHONETIC works, for the instructor who is inclined to figure out the misspellings that PHONETIC can and cannot handle.

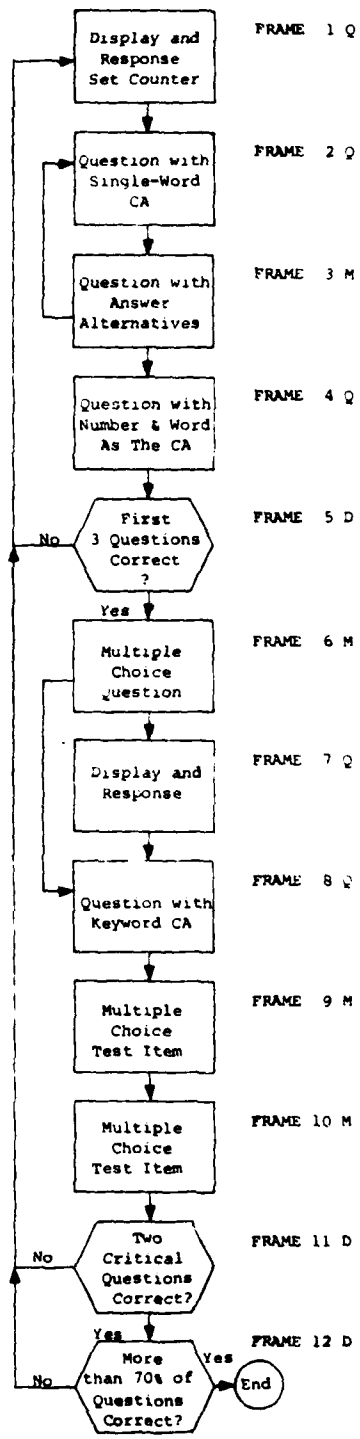
The frame entry legalities shown in Exhibits 1 through 6 summarize and go beyond the information covered by lessons leading up to the practical exercise (PLANIT1, PLANIT12, PLANIT13), including information contained in the more advanced lessons (PLANIT5, PLANIT51, PLANIT6, PLANIT61). For example, the Q-frame reminder sheet (Exhibit 2) shows KEYWORD ALL and KEYWORD (number) as legal entries in Group 3 which is information covered in CAI lesson PLANIT5. Similarly, the D-frame reminder (Exhibit 4) shows connectives AND and OR as optional entries in the general form of a decision statement, even though compound decision statements are not covered until lesson PLANIT51. The P-frame reminder (Exhibit 5) incorporates information presented in lessons PLANIT12, PLANIT13, PLANIT51, PLANIT6, and PLANIT61.

The decision to incorporate language conventions beyond those introduced by lessons leading up to the practical exercise was a tradeoff intended to maximize the flexibility of Exhibits 1 through 6 as a self-contained reference aid for those students who might later have occasion to use the more advanced PLANIT lessons, or for those using PLANIT without the lessons. The tradeoff recognized that students might be confused during the practical exercise evaluation upon seeing legal entries they could not remember from earlier lessons.

Figure 2-6 shows the page summarizing D-frame PLANIT language conventions (Exhibit 4).

e. Error Messages Reference List

The PLANIT Error Messages Reference List is an alphabetical list of PLANIT execution error messages; each message is cross-referenced to one or more frame entry conditions that could have caused the error. Students are expected to use this aid to find frame entries causing execution error messages (if any) during the online portion of their practical exercise. Lesson PLANIT3, on executing and checking a lesson, gives students practice in using this aid to locate and trace the cause of simulated execution errors. The reference list does not include all PLANIT error messages, nor



- THIS FIGURE SHOWS ONE ORDER OF Q & M FRAMES THAT WILL SATISFY THE REQUIREMENTS OF FIGURE 4. YOU MAY USE ANY ORDER OF Q & M FRAMES FOR YOUR TOPIC AS LONG AS IT SATISFIES THE REQUIREMENT.
- NOTE THE POSITION OF THE THREE REQUIRED D-FRAMES IN TERMS OF THE FIGURE 4 REQUIREMENTS. THE DECISION CONDITIONS AND BRANCHING LINES ARE SHOWN.
- THIS EXAMPLE ASSUMES A 12-FRAME LESSON WITH THREE EXTRA Q-FRAMES.
- THE "CA" STANDS FOR THE CORRECT ANSWER.

Figure 2-5. Lesson Logic Diagram for the Practical Exercise

GROUP 1 IDENTIFIER (Enter 1 in column 1)

Enter a frame number: N.ØØ or NN.ØØ
Enter frame type: D
Enter frame label (optional) 7 or less letters and/or numbers, all letters before any numbers.

GROUP 2 CRITERIA (Enter 2 in column 1)

TO SPECIFY CONDITIONAL ACTIONS

- General form- IF condition1 (AND)/(OR) (condition2)
  (AND)/(OR) (condition3)...
  action1 (action2)...
  (ELSE)
  action3 (action4)...

  Where- ALL entries except IF condition1 action1 are optional.
  "AND" and "OR" can serve as logical connectives where more than one condition is specified.
  ELSE (if used) means "otherwise". If ELSE is not used and no condition is found, control goes to next frame.

CONDITIONS MAY BE ANY OF THREE FORMS

- Computational- IF (name) (relational) (number) (expression)

  where- "name" is user-defined and given a value with C:actions.
  "relationals" are GR GQ EQ NQ LQ LS
  "expression" is any legal CALC statement giving a number.

- Summary- IF (FROM frame,) (relational) (number) (ALL)/(NONE) RIGHT/WRONG/SEEN frame/frame-frame

  where- "relationals" are GR GQ EQ NQ LQ LS
  "frame" is a no. or label and frames are given frame-frame.

- Pattern- IF (FROM frame,) frame,tags (frame,tags)...
  where- "tags" are any letter A-Z, number 1-9, + or -
  "frame" is a no. or label.

  The Computational Form addresses records assigned with C: in frames.
  The Summary Form and Pattern Form address PLANIT student records.
  FROM (if used) begins a search for conditions at the most recent recorded entry of the frame specified.

- Actions may be:

  F:XXX -display message XXX.
  C:Calcstatement -do an assignment, calculation, or control statement.
  B:Frame B:Lesson -branch to a frame no. (label) or to named lesson.

TO SPECIFY UNCONDITIONAL ACTIONS

Actions F: C: B: on lines above conditions (or no conditions in frame).
Actions F: C: B: following control word END on lines below all conditional actions in the frame.

Figure 2-6. A PLANIT Language Conventions Example

does it include all the error conditions that might lead to a given error message. Instead, it is a selection of those error messages and conditions considered most likely to follow from the requirements of the practical exercise.[1] It goes beyond the practical exercise requirements to include some errors that might be encountered in using the capabilities covered by lessons PLANIT5, PLANIT51, PLANIT6, and PLANIT61.

Figure 2-7 shows a page from the PLANIT Error Messages Reference List.

The above adjunct materials were arranged into sets to be given in folders to each student at the start of the PLANIT course. Each student received the following:

- PLANIT Lessons Figures 1 through 6 (1 copy).
- PLANIT Frame Worksheet (4 sets of 10 worksheets).
- Summary of PLANIT Frames (1 copy).
- PLANIT Error Messages Reference List (1 copy).

A complete set of adjunct materials for the PLANIT course is contained in Appendix B.

3. Delivery of Materials and Review by ARI

The complete PLANIT courseware card decks and listings and adjunct materials were delivered to ARI in May 1974. This included the following 10 PLANIT lessons and 2 unit tests with associated frame counts as shown:

	Course Unit	PLANIT Lessons	Frame Count
I	Basics of Lesson Construction	PLANIT1	84
		PLANIT12	93
		PLANIT13	99
II	Online PLANIT Use	PLANIT2	72
		PLANIT3	33
		PLANIT4	79
III	Special Capabilities and Commands	PLANIT5	99
		PLANIT51	100
		PLNTST5 (Unit test)	22
IV	CALC and the P-frame	PLANIT6	70
		PLANIT61	74
		PLNTST6 (Unit test)	17
		Total	842

[1] Appendix C, PLANIT Language Reference Manual, SDC TM-4422/002/01. Santa Monica: 1 October 1970. PLANIT Version 1.1 Cold Start messages listing, ARI, received 30 March 1974.

Message	Condition
ILLEGAL USE OF F:--NO CORRECT ANSWER SPECIFIED	Occurs during execution of a Group 4 of a Q or M frame when the Action Command F: is encountered and no correct answer has been specified in Group 3.
ILLEGAL USE OF R:	Occurs during execution of a Group 4 in a Q or M frame when the Action Command R: is encountered and no answer tags exist in Group 3.
IMPROPER FROM SPECIFICATION	Occurs during execution when the optional component FROM is used in a conditional clause and nothing follows, or the subsequent component is not a frame number or label, or the label does not exist, e.g., IF FROM or IF FROM -X.
IMPROPER KEYWORD SPECIFICATION	a. Occurs when the component following KEYWORD is not ALL, OFF, ON or a number. b. Occurs when KEYWORD number is greater than the number of key words specified in an answer.
INTEGERS ONLY FOR GROUPS/LINES/COLUMNS	Occurs when the group, line, column entries in an edit command statement are not integers, e.g., 2,3.4,M or 3,2,1.5,P.
INVALID	Occurs when a user enters a log-in value which does not match one of the specified identifications for the installation.
LEFT PARENTHESIS MISSING	Occurs whenever parentheses (required or optional) are entered in a CALC statement which are not balanced, e.g.,) or ( )).
LINE CONTINUATION NOT PERMITTED	Occurs during execution when an author has attempted to extend a CALC statement more than a single line.
LINE DOESN'T EXIST	Occurs when an author attempts to Insert a line into a group above a line which doesn't exist or Edit, Modify or Print a nonexistent line or Append after a non-existent line.

Figure 2-7. Portion of the PLANIT Error Message Reference List

Also delivered to ARI were the PLANIT Data Sheet, PLANIT Data Questionnaire, and PLANIT Debriefing Questions contained in Appendix C.

4. Online Tryout and Editing of Lessons

Online tryout of the delivered course materials started in May 1974 at ARI. The six lessons teaching PLANIT basics and directly supporting the practical exercise work were checked first. Each lesson was run on the computer, and as problems were discovered, editing changes were marked onto listings. ARI updated card decks and the master UNIVAC lesson files to incorporate the changes, and produced a new lesson listing. The online tryout was particularly sensitive to execution and logic errors by testing decision frames and branching, and by providing both correct and incorrect answers to open-ended questions to find problems in answer-matching or in feedback. Virtually no problems were found in display size, and only a few changes were needed to correct execution, logic, and typographic problems. The majority of editing changes made to the first six lessons were for one of the following reasons:

- Improving correct and incorrect answer-matching for some constructed response questions.
- Editing to reduce ambiguity of student instructions and feedback.
- Inserting frames to provide additional or transitional content.

After the first six lessons had been tested, ARI students used the lessons. Student suggestions resulted in more changes of the above kinds. This process was repeated for the last four lessons and the two online tests in the package. The majority of problems encountered in this tryout were in the second of two lessons about CALC and were problems of content and execution failure due to exceeded PLANIT capacities. These problems were recorded, and solutions were tested and fixed as discussed below.

On-computer checkout of the lesson materials continued in parallel with USAIS student evaluations at Fort Benning in June. The types of changes incorporated during this period were as follows:

- A few changes were made to the basic PLANIT lessons where USAIS students detected conflict in some details of content.
- Several changes were also made to shorten the character count in certain frames when ARI reported that these frames would not load successfully on the CDC 3300 due to storage restrictions.
- The lesson covering portions of CALC and the Programming frame was extensively edited to improve content compatibility with other lessons and to fix certain content and execution problems known to exist from online tryout at ARI.

When changes were few, the Fort Benning editing changes were transmitted by telephone to ARI. For extensive editing, PLANIT editing changes were listed on the printer at ARI. In both cases, ARI followed up by updating

the card decks at ARI and updating the UNIVAC master lesson files using system utility software.

C. COURSE DESCRIPTION

The following figures, tables, and text describe the PLANIT courseware content and logic.

1. Course Activities Flow

The overall flow of students through the PLANIT course CAI lessons, adjunct materials, and practical exercise is shown in Figure 2-8. Each block in Figure 2-8 shows the PLANIT lesson names and the main subject areas covered. As shown, students log in to PLANIT and first take lesson INTRO,[1] which covers basic communication conventions with PLANIT and the ADDS Consul 880 terminal keyboard. Each student then takes the Unit I lessons (PLANIT1, PLANIT12, and PLANIT13) in sequence, using the adjunct materials shown as required by the lessons. Lessons within each unit call upon each other automatically--there is no need for students to take a separate action in moving from one lesson to another. At the end of PLANIT13, the students look at Figures 4 through 6 of the student handout, which gives them the requirements for the practical exercise in instructional programming, a sample lesson logic diagram, and the steps to follow for the offline and online portions of the exercise.

Students then begin Part 1 of the Practical Exercise. This requires the planning, writing, sequencing, and encoding of 12 to 20 PLANIT frames in a subject matter area of his choice. Students can perform this deskwork part of the practical exercise either in the CAI work area or elsewhere. In either case, reference materials available include sample frame types Q, M, and D (Figures 1-3); the students' completed worksheets from the first three lessons; and the "Summary of PLANIT Frames" (Exhibits 1-6).

When students have encoded their frames onto PLANIT Frame Worksheets and have checked them against the exercise requirements (Figures 4 & 5), they begin Unit II of the online instruction, which is designed to prepare them for the online portion of the practical exercise. They take lessons PLANIT2, PLANIT3, and PLANIT4 in sequence. These lessons cover PLANIT conventions to input frames, display frame entries, name a lesson, call-up and continue working on a lesson, try out a lesson, check for errors and trace error causes, and perform selective editing of material. When students finish these lessons, they begin Part 2 of the practical exercise. This requires using the terminal keyboard to enter the frames from the

---

[1] ARI wrote lesson INTRO for the UNIVAC Version 1.1 PLANIT and ADDS Consul 880 terminal, based upon a similar lesson for the CDC PLANIT Version 1.1 and the Control Data 211 terminals. A listing of this lesson is contained in Appendix D. It constitutes 12 frames and occupies students for about 10 to 15 minutes of online time. Because each version of INTRO will be unique to the particular installation, it is not covered in the subsequent discussion.

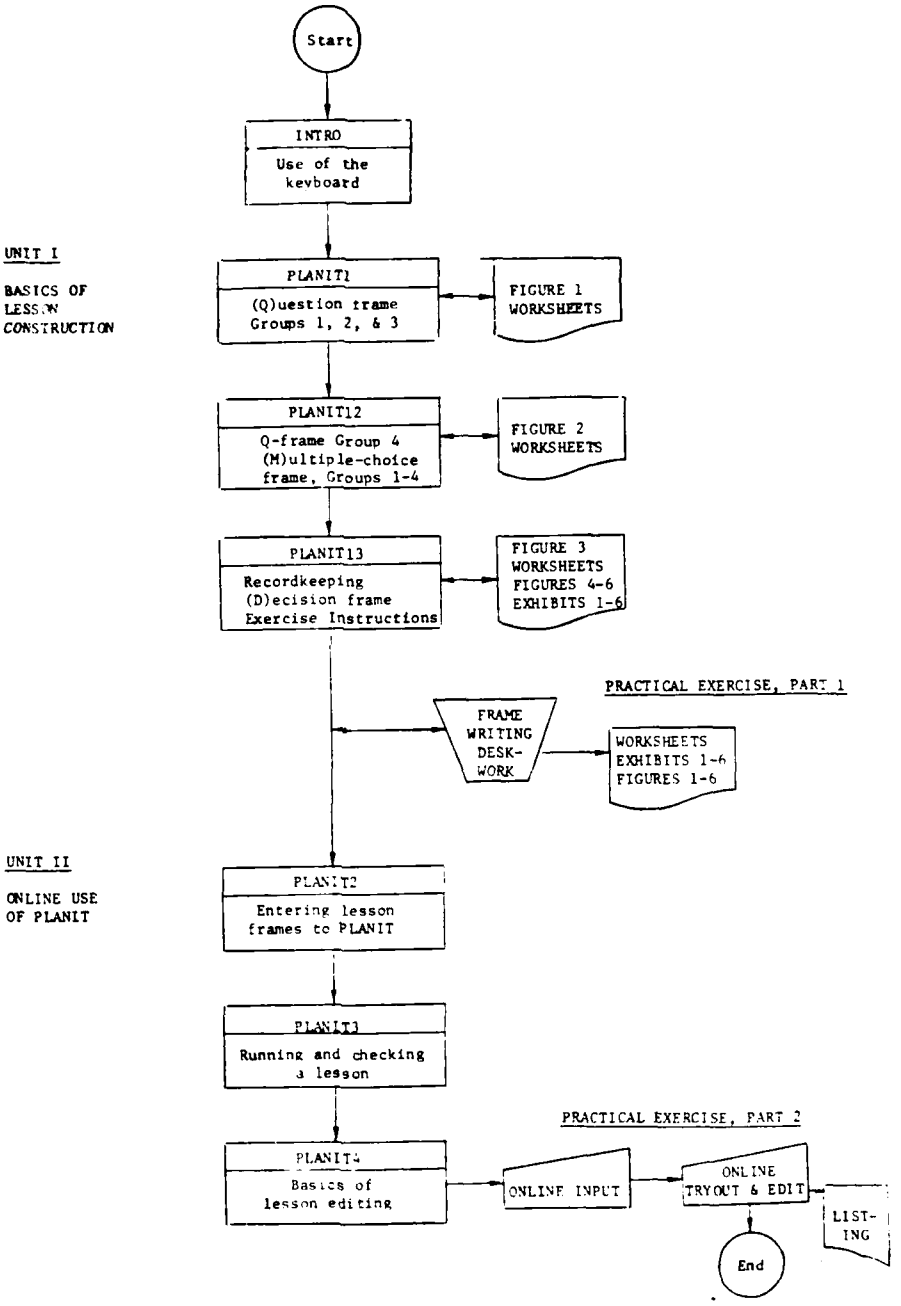


Figure 2-8. PLANIT Course Flow

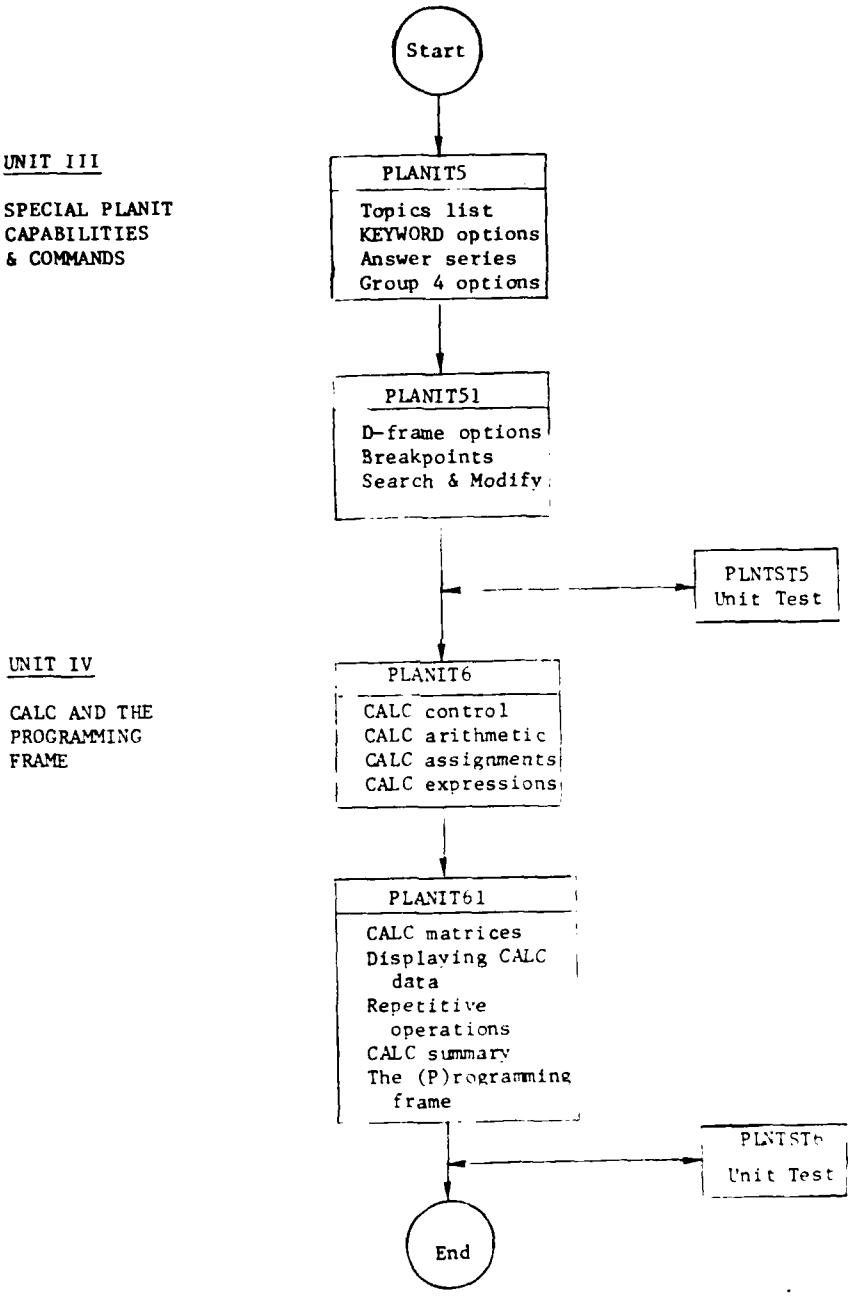


Figure 2-8. PLANIT Course Flow (cont)

students' worksheets directly to PLANIT.[1] When entry is completed, students use PLANIT to execute their frames, check for execution error messages or improper frame constructions, and use PLANIT editing commands to modify the material until it runs successfully. Should they encounter execution errors while running their lessons, they can refer to the "PLANIT Error Messages Reference List" to find the cause of the error. They continue the tryout, check, and edit process until they are satisfied that their lessons work as intended. At this point, a listing of the lesson frame entries is made--either at a printer connected to the ADDS console and/or at a remote printer by using the PLANIT Print or LIST commands, respectively.

For purposes of the evaluations at ARI and the USAIS, Units I and II comprising six lessons (PLANIT1 through PLANIT4) and the practical exercise were the PLANIT courseware package for which evaluation results are reported. The practical exercise in instructional programming served as the criterion test for the evaluation.

The second half of Figure 2-8 shows the remainder of the PLANIT course material, Units III and IV. These units provide instruction for additional PLANIT language features to include special capabilities and commands for lesson authors, as well as features of PLANIT's CALCulation language and the (P)rogramming frame. The lessons require the student to identify correct PLANIT entries, to answer questions on how frame entries would influence lesson execution, and to respond with PLANIT commands to simulated sequences of PLANIT messages. Two end-of-unit tests are provided to assess student learning on Unit III and IV topics. The materials contained in Units I and II are considered prerequisite to Units III and IV.

Questions at the start of PLANIT5 determine student prerequisite status. If students have completed all lessons in Units I and II, they can proceed. If not, they are instructed to take these lessons first and are signed off PLANIT5 automatically. For continuing students, if they have not taken Units III and IV, they begin an overview of topics and continue in sequence through the four lessons and two unit tests. If this is not the first time, students begin with a topic menu and repeatedly select from the topics in the four lessons. Each time they finish a topic, they return to the menu in PLANIT5 with instructions to select another topic or sign-off. The linking to topic start points within the four lessons and back to the menu is automatic; students merely select their topics.

While Units III and IV are intended to run in sequence, beginning with the status questions at the start of PLANIT5, the two units can be run independently. If this is done, the first lesson of each unit should be considered prerequisite to the second, with both lessons of each unit prerequisite to the end of unit tests.

---

[1] During the evaluations at ARI and USAIS, the monitors performed this step in an attempt to stabilize the evaluation baseline among students. The lesson was entered to the computer without correction, just as the students had provided it on their worksheets, prior to the students' online checkout. Also, an uneven level of typing ability among students would have made this step very time-consuming and error-prone.

2. Structure of Course Materials

Table 2-2 on the next four pages shows the topic and frame number progression for each unit and lesson in the PLANIT courseware. The topics and frames are also shown where students are directed to use the various kinds of adjunct materials.

3. Course Logic Diagrams

The logic diagrams comprising Figures 2-9 through 2-18 supplement the course and topic progressions above by showing the major decision points and branching patterns among the topics and lessons of each unit. They indicate the major performance check points in each lesson and how they tie in with course content. They are useful for those who attempt to use lesson listings to monitor student progress, or for further editing of the lessons.

To aid the user who attempts to use logic diagrams with lesson listings, PLANIT frame labels are above selected diagram entries, showing all of the frame label markers in each lesson. In most cases, the labeled frame begins the sequence of frames for each boxed diagram entry. Frame labels followed by a period are those which are the logical start points for topic-by-topic review within each lesson. They are so punctuated to enable the PLANIT REVIEW function to operate. Other frame labels refer to figures or examples.

A number of the frame labels (without periods) mark the embedded test items. These usually signal the start of a sequence of frames that begins with the student constructing entries on a PLANIT Frame Worksheet, typing in one or more entries from the worksheet to be checked by PLANIT, and ending with feedback of results and a chance to compare and correct entries as required. This logic is used repeatedly in lessons PLANIT1, PLANIT12, and PLANIT13.

The logic diagrams do not show the many instances of branching among the frames in order to keep examples and their associated questions from rolling up and off the display, nor the CALC items that keep track of these repetitive loops. Nor is branching internal to frames shown in the logic diagrams. This means that none of the many attempts are shown to prompt students into giving a correct frame entry using general entry forms or other more direct forms of hints and cues. The reader goes to the lesson listings for this level of detail.

The logic diagrams are ordered by course unit. Figures 2-9 through 2-11 are those for Unit 1, Figures 2-12 through 2-14 for Unit II, Figures 2-15 and 2-16 Unit III, and Figures 2-17 and 2-18 for Unit IV lessons.

Table 2-2. PLANIT Courseware Structure

COURSE UNIT	PLANIT LESSON NAME	COURSE OUTLINE TOPIC	FRAME NUMBERS BY TOPIC	NO. OF FRAMES BY TOPIC	TOTAL FRAMES BY LESSON & UNIT	ADJUNCT MATERIALS
I BASICS OF LESSON CONSTRUCTION	PLANIT1	Introduction	1-8	8		
		The Q-Frame				
		Q-frame Groups	9-29	13		Figure 1
		Constructing Group 1	31-42	12		Worksheet
		Constructing Group 2	43-58.5	16		Worksheet
		Constructing Group 3	59-73	16		Worksheet
		Use of Number Tags	74,75	2		
		Basic use of KEYWORD ON	76-86	11		Worksheet
		Basic use of PHONETIC ON	87-91	6	84	Worksheet
	PLANIT12	Constructing Group 4 actions				
		Overview	101-110	11		
		Command F:	111-113	3		
		Command R:	114-117	5		
		Command C: for answer feedback	118-120	3		
		Unanticipated Answers	121-123	3		
		Command B:	124-130	7		
		Spacing and order of action commands	131,132	2		
		Practice	133-137.5	6		Worksheet
		Construction Exercise	138-147	12		Worksheet
		The M-Frame				
		M-frame Groups	147.5-156	11		Figure 2
		Group 1 Frame Labels	157-162	6		Worksheet
		Constructing Groups 2 & 3	163-172	10		Worksheet
		Constructing Group 4	173-186	14	93	Worksheet

Table 2-2. PLANIT Courseware Structure (cont.)

COURSE UNIT	PLANIT LESSON NAME	COURSE OUTLINE TOPIC	FRAME NUMBERS BY TOPIC	NO. OF FRAMES BY TOPIC	TOTAL FRAMES BY LESSON & UNIT	ADJUNCT MATERIALS
I BASICS OF LESSON CONSTRUCTION (cont)	PLANIT13	Command C: to record and update scores	201-230.75	32		Worksheet
		The D-Frame				
		D-frame groups	231-243	15		Figure 3
		Constructing a computational decision	244-250	7		Worksheet
		Student records	251,252	2		
		Constructing a summary decision	253-275	25		Worksheet
		Constructing a pattern decision	276-289	13		Worksheet
		Instructions for practical exercise	290-293	5	99 276	Figures 4-6
II ONLINE USE OF PLANIT	PLANIT2	Entering Lessons From a Terminal				
		Overview	301-305	3		
		How to begin authoring	306-309.7	6		
		Entering frames	310-320	12		
		Frame entry exercise	321-336	19		Worksheet
		Displaying frame entries	337-343	16		
		Procedure for naming lessons	344-350	9		
		How to call-up a named lesson	351-356	6		
		Entry errors	357	1	72	
	PLANIT3	Running and Checking Lessons				
		Overview	358	1		
		How to execute a lesson and clear records	359-372	14		
		Execution errors and causes	373-387	18	33	Error Messages Ref. List

Table 2-2. PLANIT Courseware Structure (cont.)

COURSE UNIT	PLANIT LESSON NAME	COURSE OUTLINE TOPIC	FRAME NUMBERS BY TOPIC	NO. OF FRAMES BY TOPIC	TOTAL FRAMES BY LESSON & UNIT	ADJUNCT MATERIALS
II ONLINE USE OF PLANIT (cont)	PLANIT4	Basics of Lesson Editing				
		Overview	401-409	9		
		Replacing material	410-419	10		
		Inserting material	420-432	13		
		Adding material	433-439	7		
		Deleting material	440-445	6		
		Symbol to switch to & from editing	446-460	15		
		Exercise	461-478	18		
		Final instructions	479	1	79 184	Figure 5
III SPECIAL CAPABILITIES & COMMANDS	PLANIT5	Group 3 and 4 Options & Controls				None
		Topics & menu for Units III & IV	500.5-504.5	7		
		Optional uses of KEYWORD	505-532.5	29		
		Procedure for a series of answers	533-543.5	12		
		Action command options and controls	545-598	51	99	
	PLANIT51	D-Frame Options & PLANIT Commands				None
		D-Frame review	600.5-607	8		
		Unconditional actions	608-617	10		
		Compound conditions	618-628	11		
		Controlling search for conditions	629-644	16		
		Execution breakpoints	645-668	26		
		Finding and modifying entries	669-696	29	100	

Table 2-2. PLANIT Courseware Structure (Cont.)

COURSE UNIT	PLANIT LESSON NAME	COURSE OUTLINE TOPIC	FRAME NUMBERS BY TOPIC	NO. OF FRAMES BY TOPIC	TOTAL FRAMES BY LESSON & UNIT	ADJUNCT MATERIALS
III SPECIAL CAPABILITIES & COMMANDS (cont)	PLNTST5	Unit Test				
		Instructions	801	1		None
		PLANIT5 items	802-815	14		
		PLANIT51 items	816-820	5		
		Scores	821,822	2	22 221	
IV CALC AND THE P-FRAME	PLANIT6	CALC				None
		Overview	709.5-710.5	3		
		Methods for Using CALC	711-731	36		
		Arithmetic Operators	732,733	2		
		Assignments Using C:	734-739.5	8		
		Evaluating CALC Expressions	740-747.5	10		
		CALC Practice	748-754.1	11	70	
	PLANIT61	More CALC				None
		Storing Data in Tables	754.5-773	22		
		Displaying CALC Data	774-779	7		
		Repetitive Operations & Subscripting	780-780.5	6		
		CALC Review	781-784	4		
		The P-Frame				
		Structure and Operation	810-831	29		
		Performance Summary	834-838	6	74	
	PLNTST6	Unit Test				None
		Instructions	825	1		
		PLANIT6 items	826-836	13		
		PLANIT61 item	837	1		
		Scores	838,839	2	17 161	

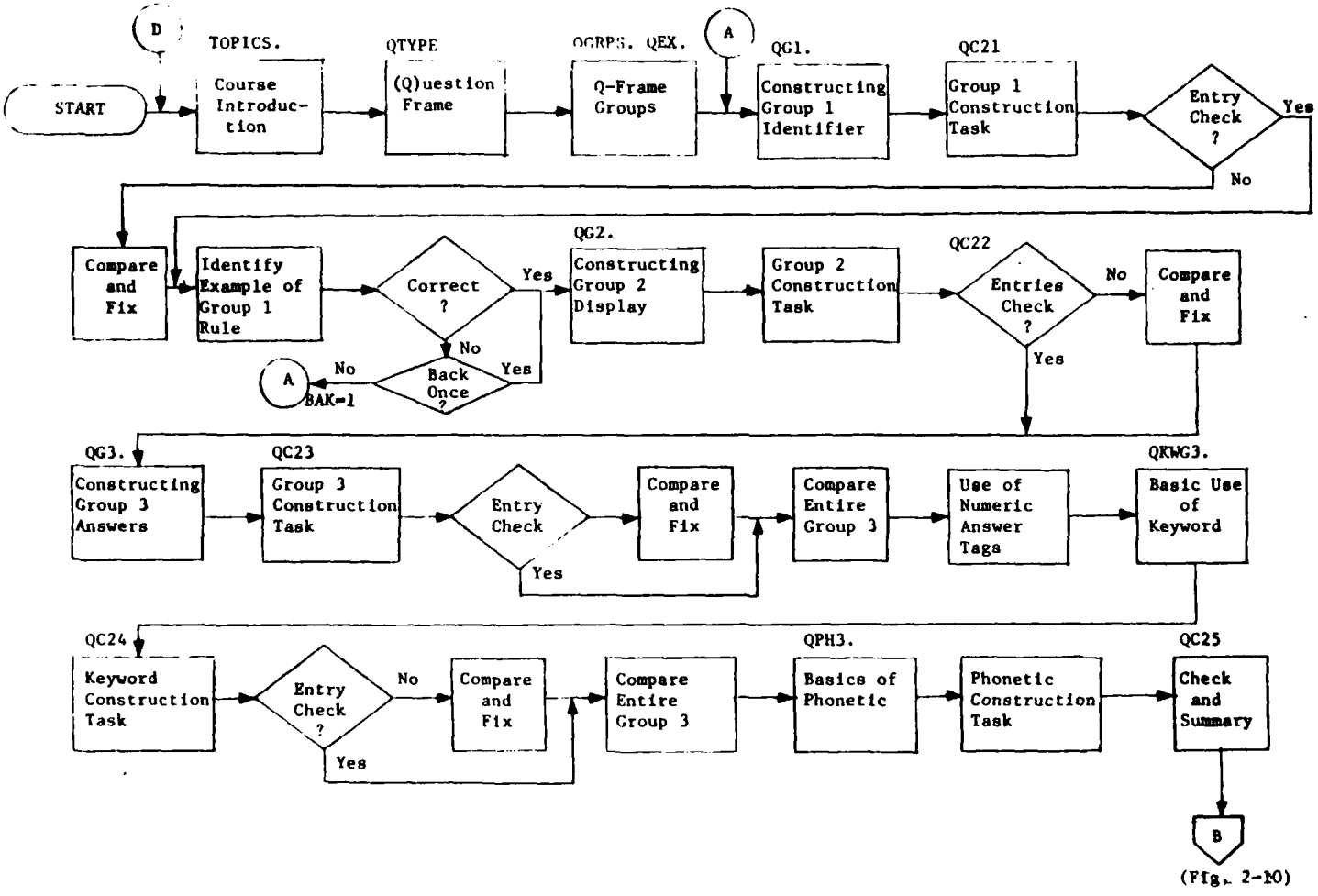


Figure 2-9. Unit I Logic Diagram: Lesson PLANIT1

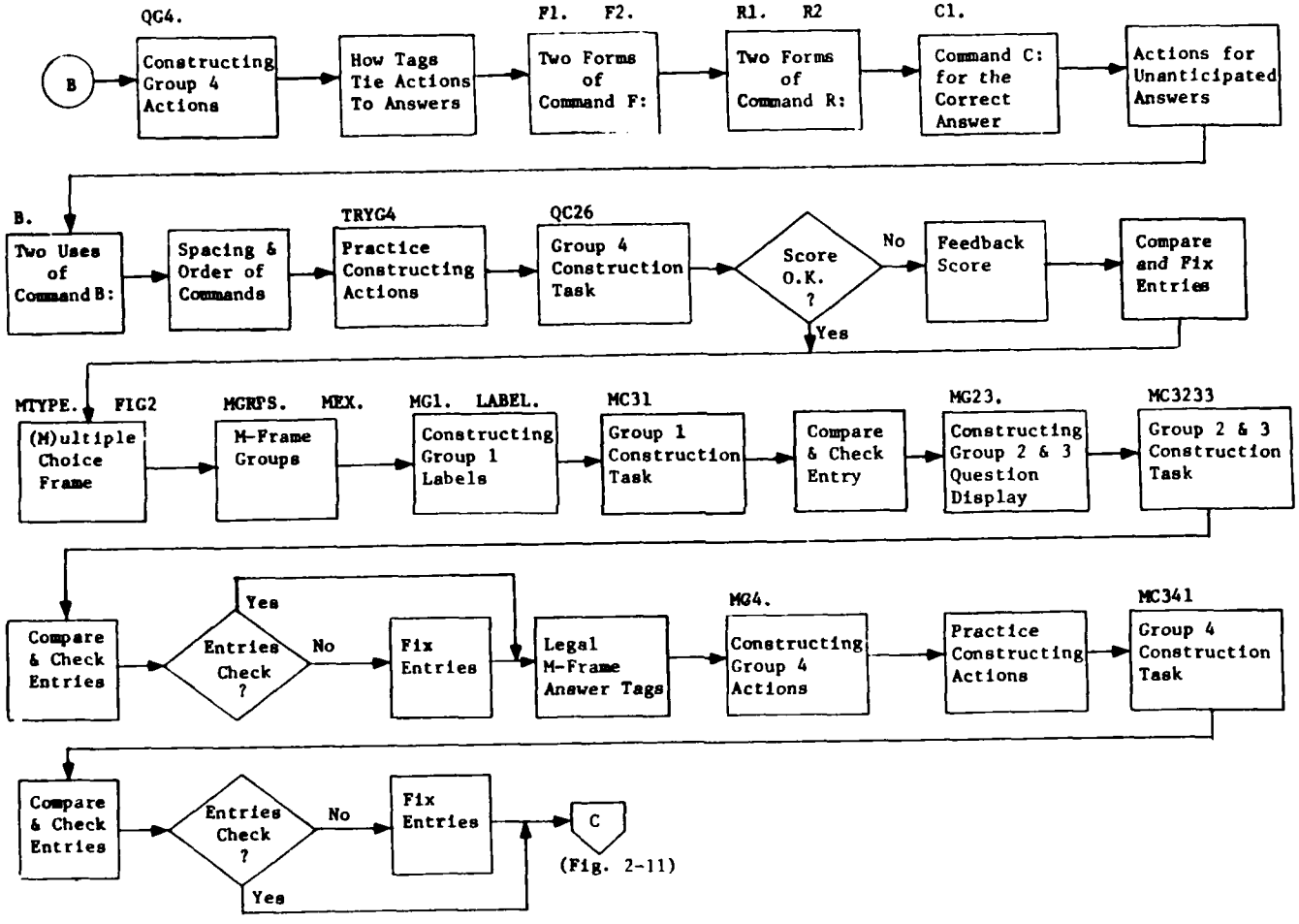


Figure 2-10. Unit I Logic Diagram: Lesson PLANIT12

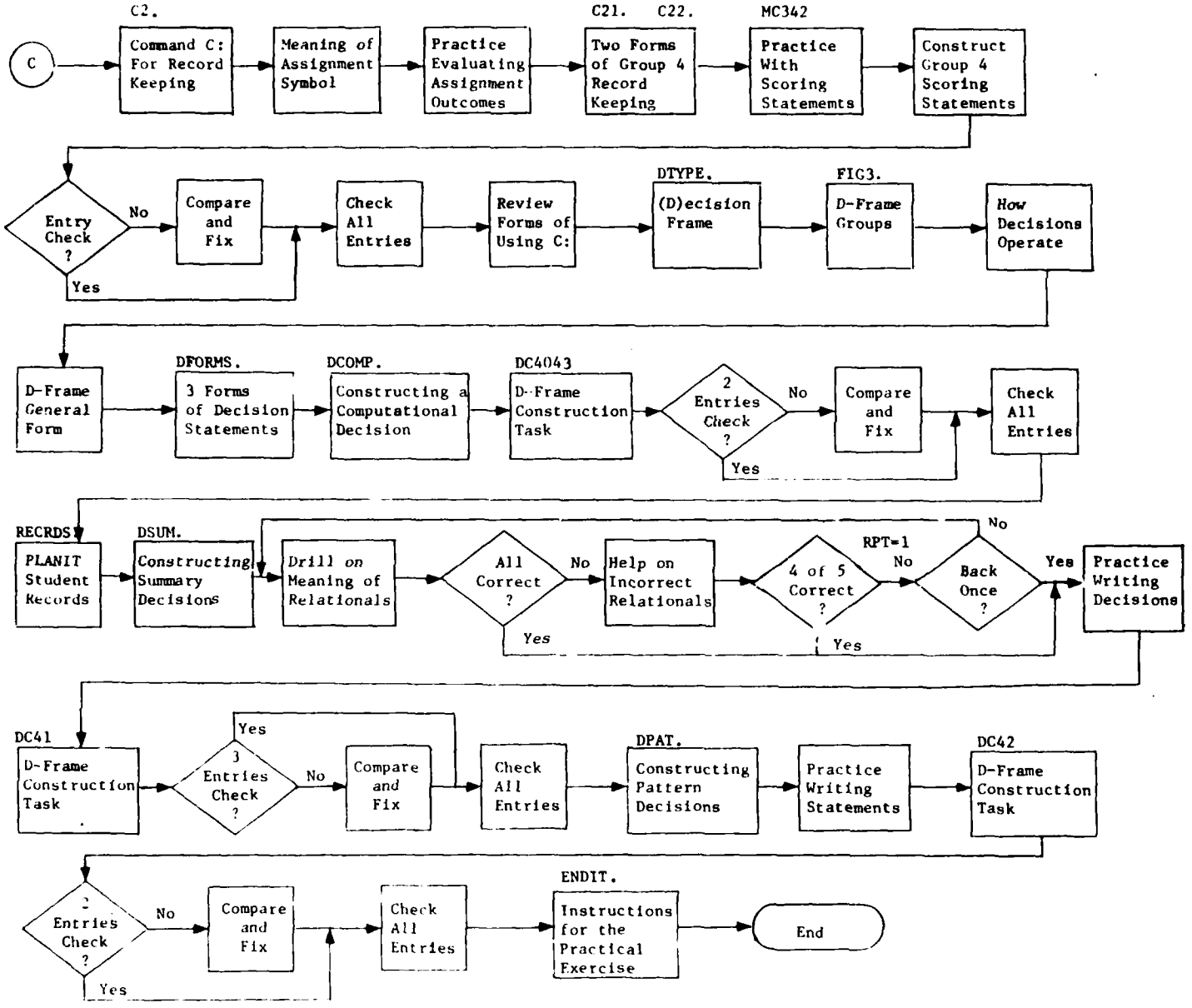


Figure 2-11. Unit I Logic Diagram: Lesson PLANIT13

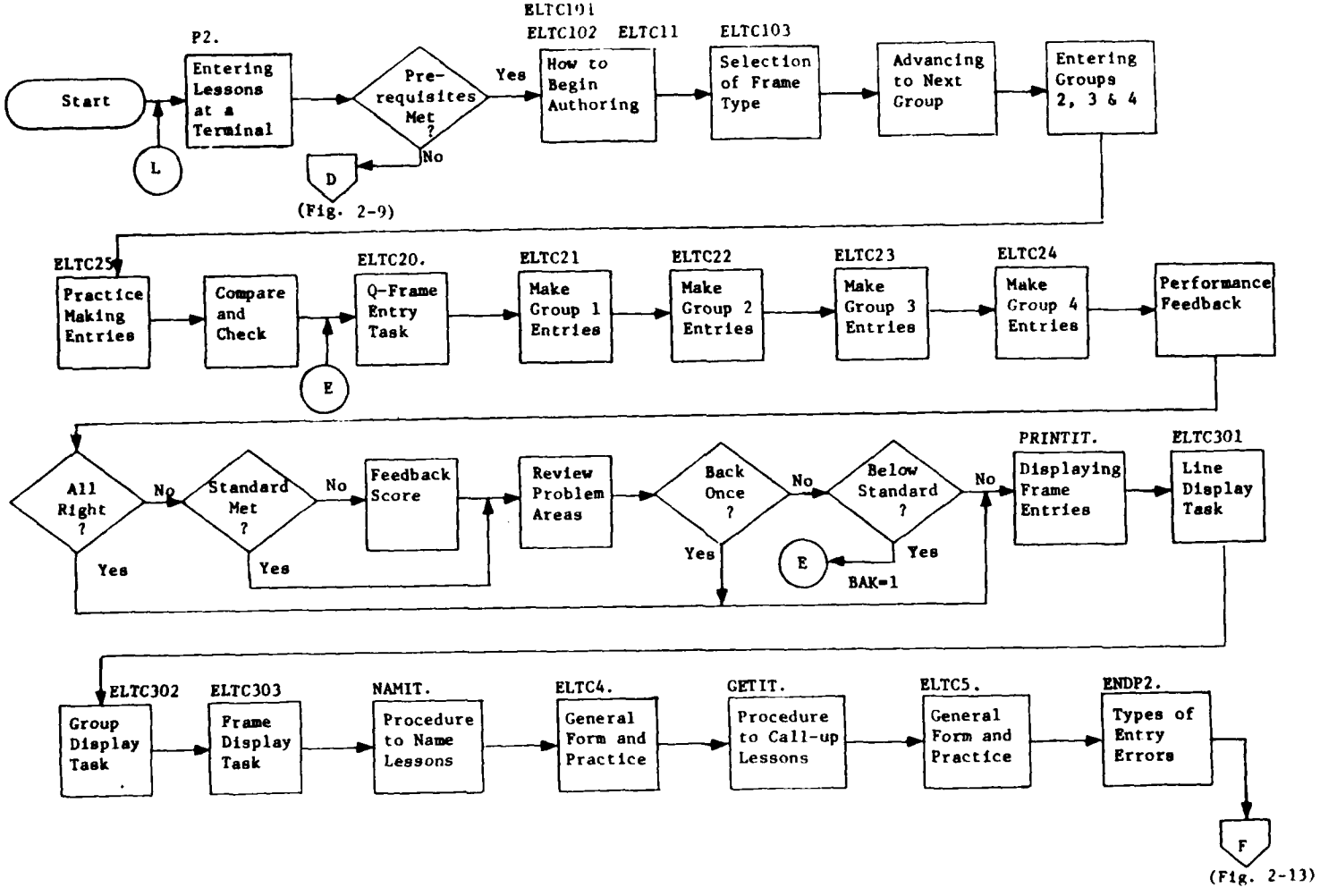


Figure 2-12. Unit II Logic Diagram: Lesson PLANIT2

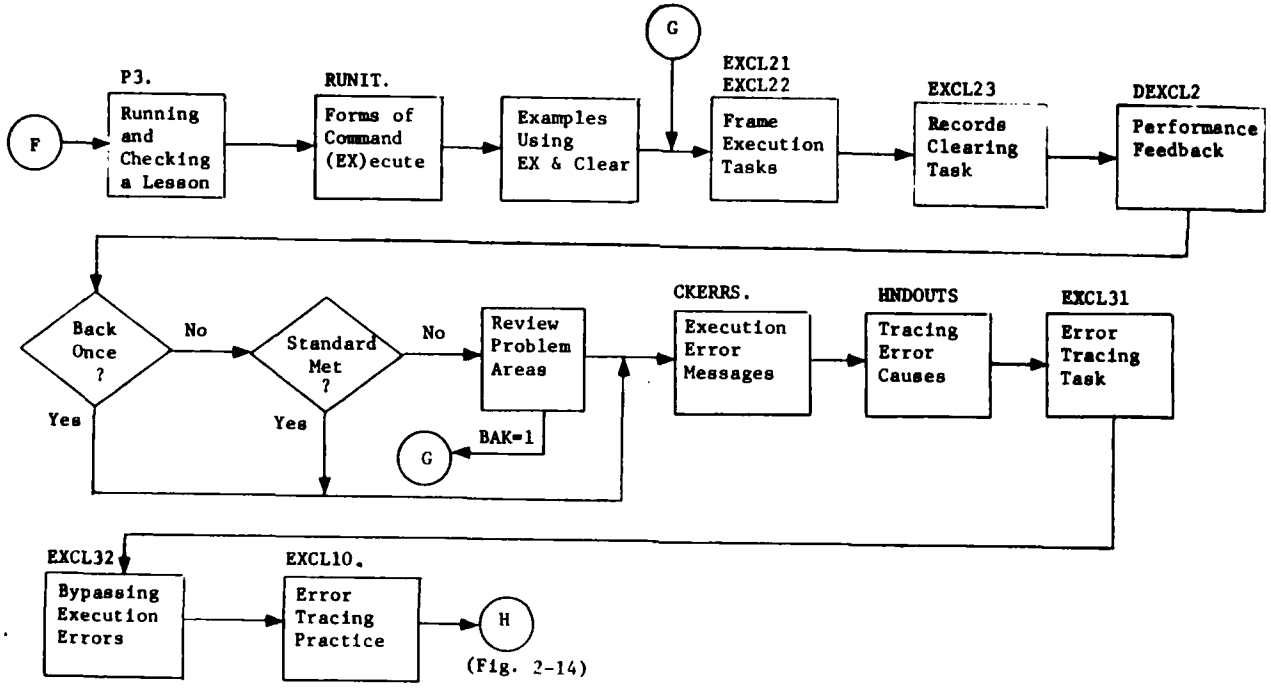


Figure 2-13. Unit II Logic Diagram: Lesson PLANIT3

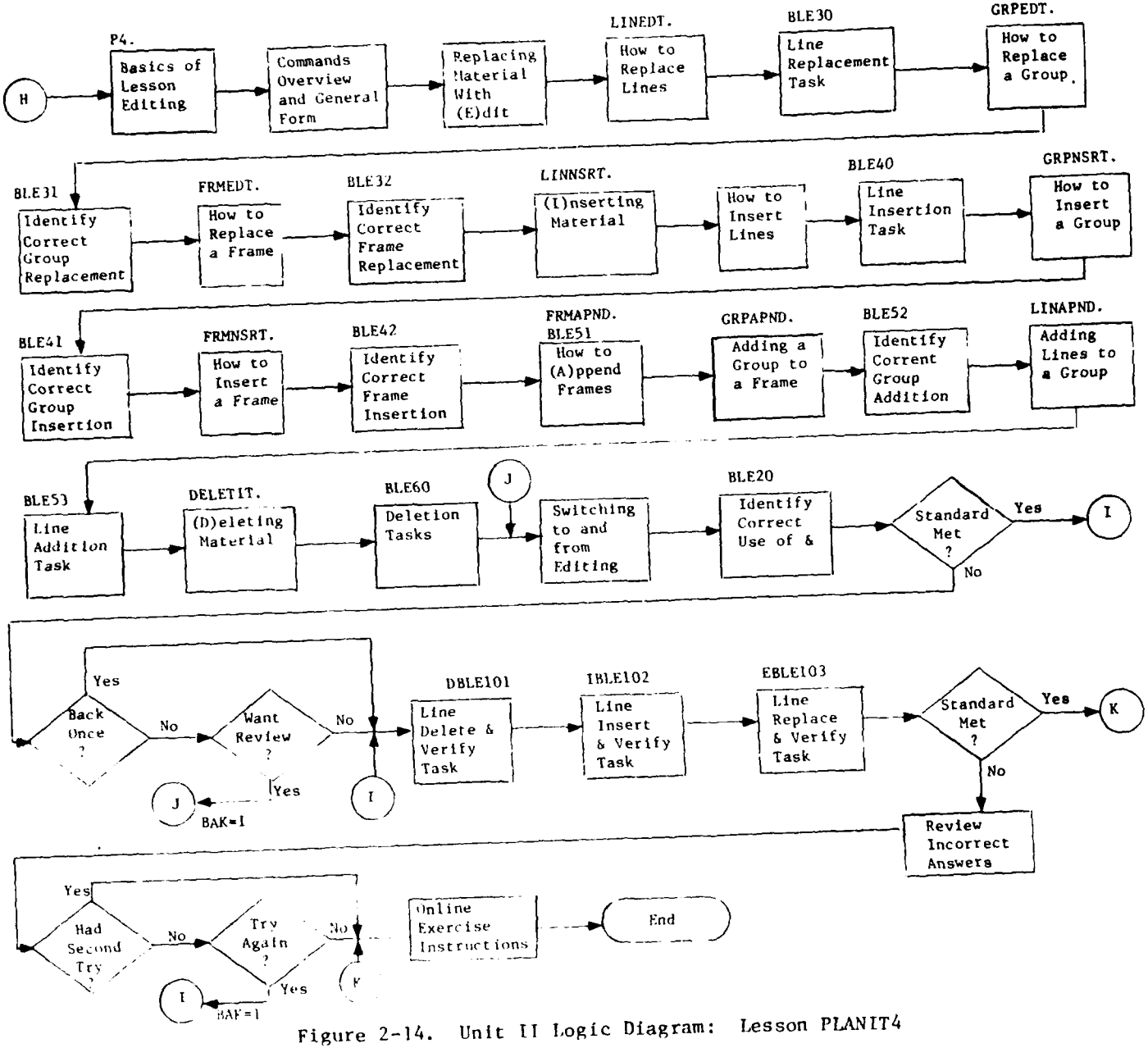


Figure 2-14. Unit II Logic Diagram: Lesson PLANIT4

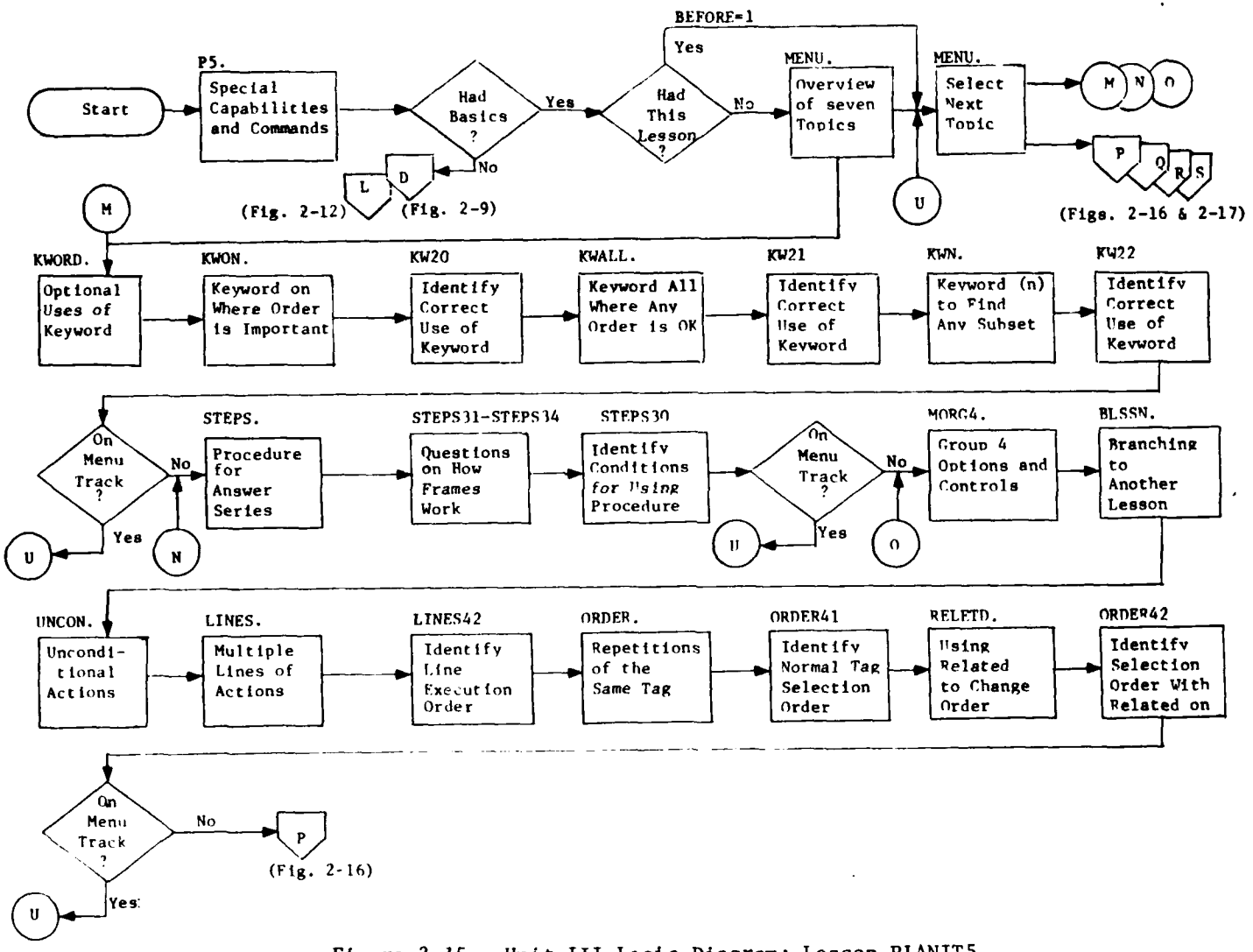


Figure 2-15. Unit III Logic Diagram: Lesson PLANIT5

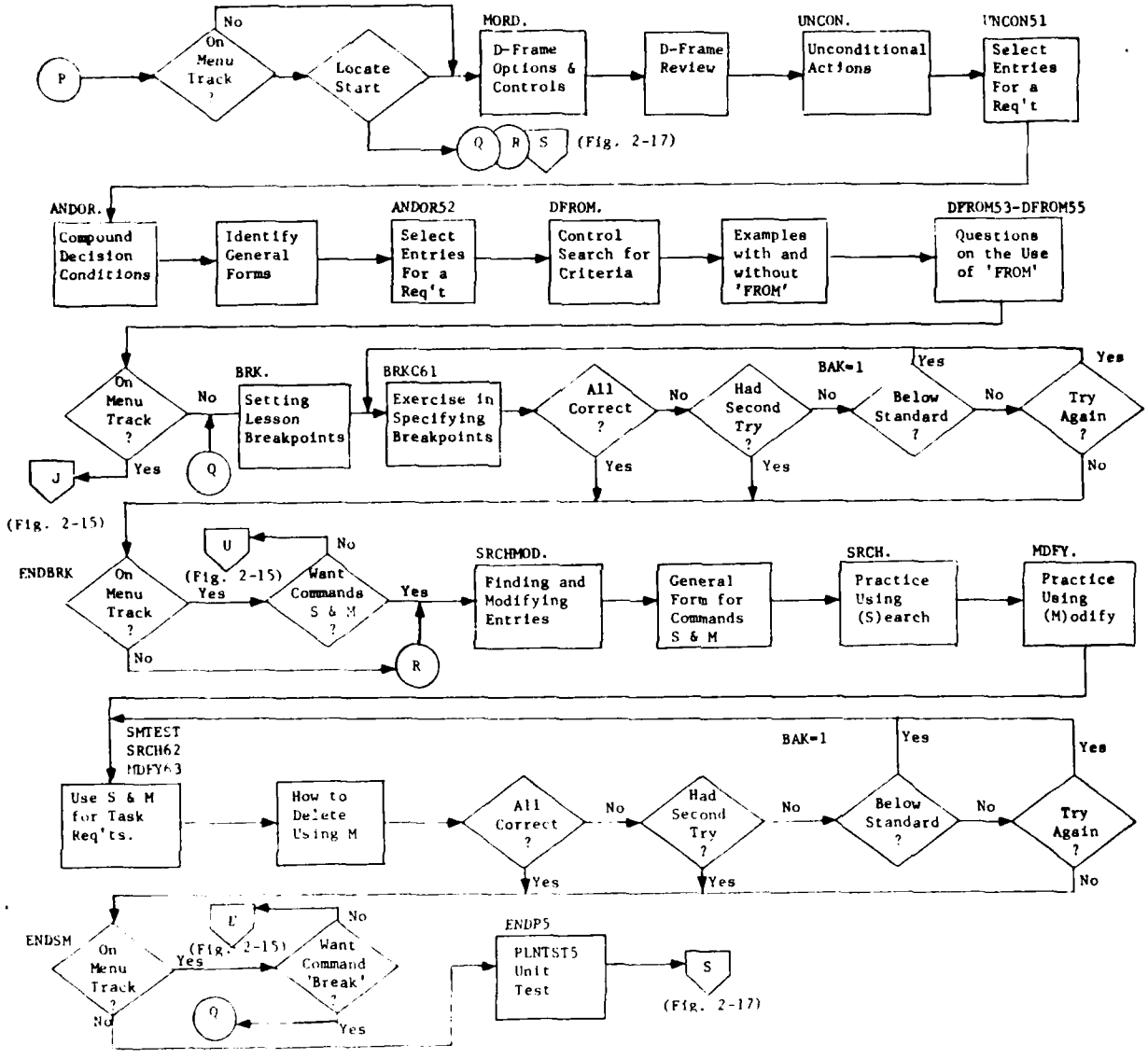


Figure 2-16. Unit III Logic Diagram: Lesson PLANIT51

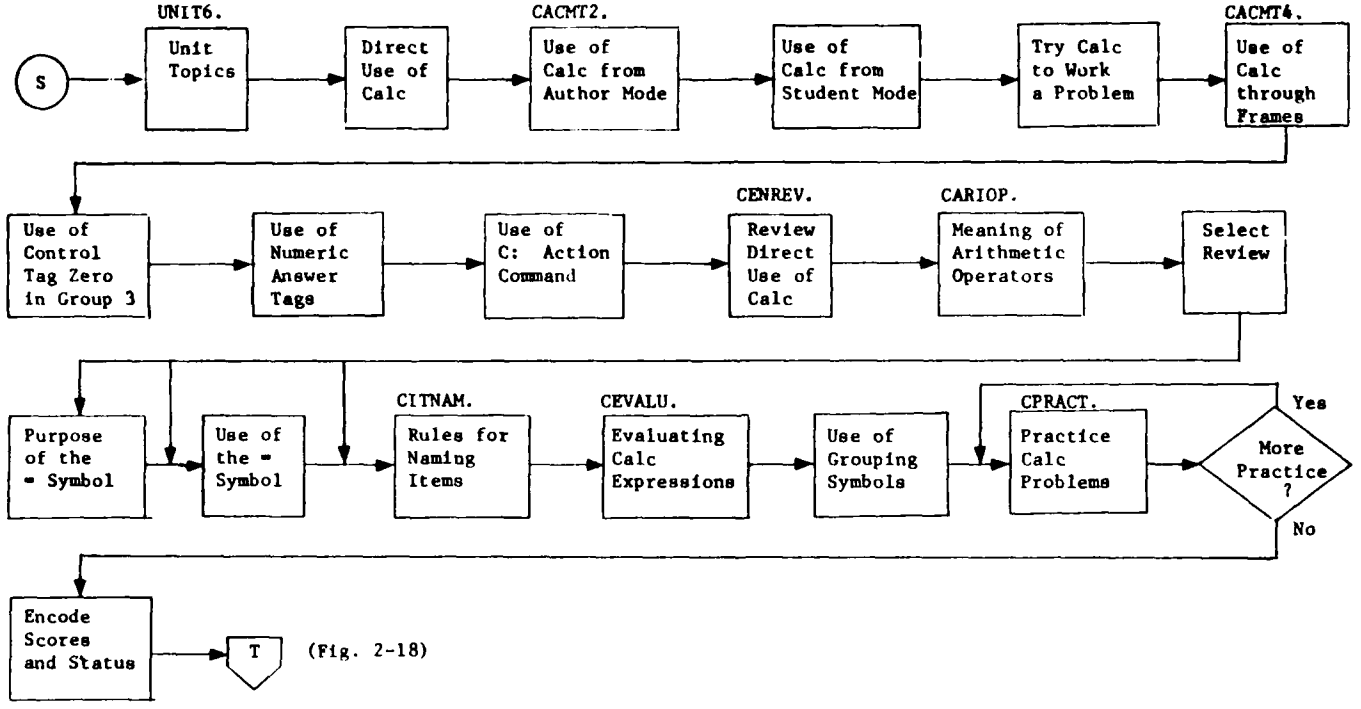
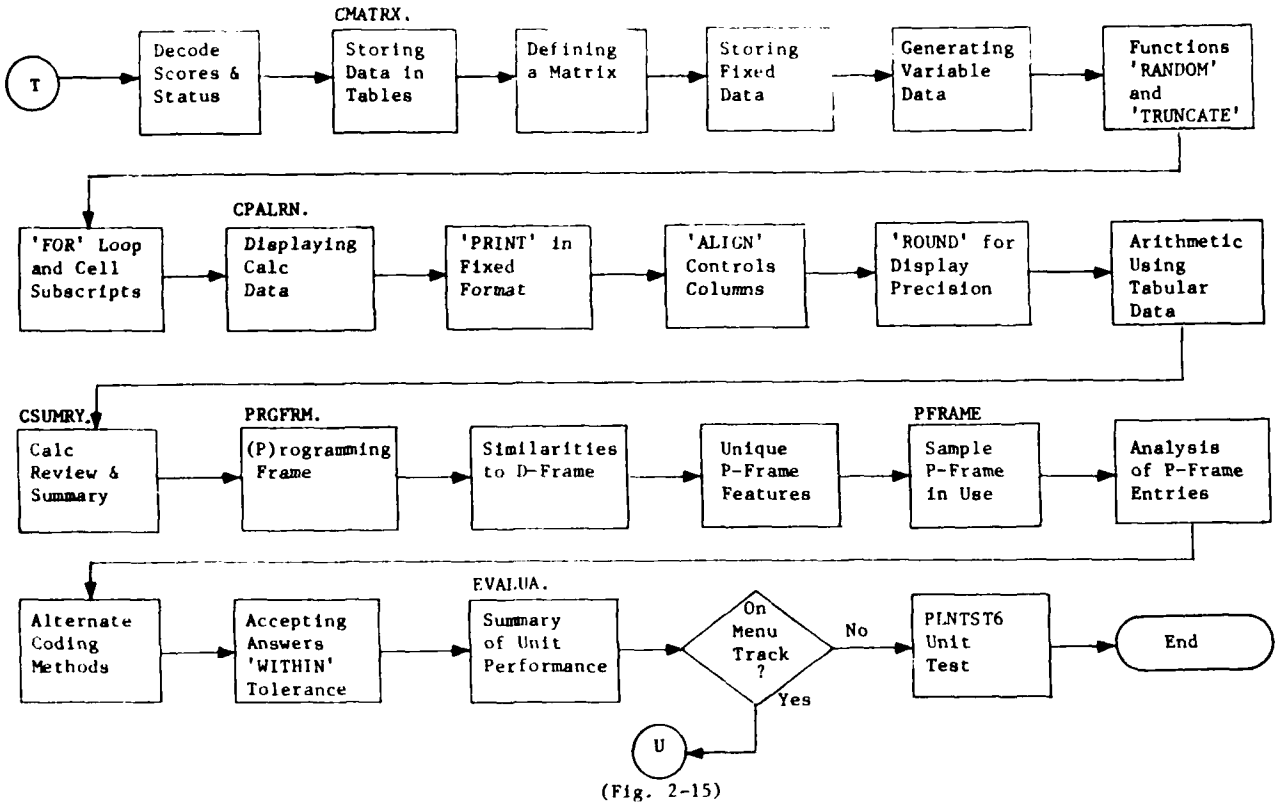


Figure 2-17. Unit IV Logic Diagram: Lesson PLANIT6



(Fig. 2-15)

Figure 2-18. Unit IV Logic Diagram: PLANIT61

Section 3. CONDUCT OF ARI AND USAIS EVALUATIONS

This section describes the selection of students, the flow of students through the evaluation, and the associated monitoring procedures and evaluation data collection at each step. ARI and USAIS differences are noted.

A. IDENTIFICATION AND SELECTION OF STUDENTS

The study was designed for professional ARI personnel and instructional USAIS personnel. ARI selected five research scientists to serve as students.

The ARI student group had relatively homogeneous job titles (research scientist), work interests, and educational levels (five Ph.D.'s, one M.S.). Ages ranged from late 20's to late 40's, with corresponding variations in years of professional experience.

ARI performed the coordination tasks with USAIS to obtain 10 students from various staff and instructional units within the school. As a result, the following working units furnished the USAIS students:

Unit	Students	Rank
IDAD Committee, Tactics Group, BBOD	1	Major
Leadership Instruction	2	Captain
Weapons Training	1	Captain
Infantry Instruction	2	1Lt. MSGT E-8
Combat Engineering	1	SGT E-7
Education Consultation Div. (D.E.T.)	3	1Lt. SGT E-5 Civilian

The USAIS student group was less homogeneous in composition. Ages ranged from 24 to 42, years of service ranged from 9 months to over 18 years, and education levels ranged from high school completion to considerable graduate work toward advanced degrees.

B. TIME ESTIMATES AND SCHEDULING OF STUDENTS

A preliminary estimate of time, before online tryout of the PLANIT lessons, facilitated scheduling students from their work commitments. This first estimate was that the PLANIT lessons and the practical exercise would require approximately 6 to 8 hours of student time, occurring as two 4-hour blocks of time on consecutive days. The initial plans for scheduling students at ARI and the USAIS were set up on that basis.

Initial runs with the first three of six students in the ARI evaluation showed the combined times for the basic PLANIT lessons (Unit I and II) and the practical exercise to be as follows:

Course Segment	Mean Time (Mins.)	Time Range (Mins.)
Unit I lessons: PLANIT1, PLANIT12, PLANIT13	230	178 - 274
Practical exercise deskwork	220	180 - 240
Unit II lessons: PLANIT2, PLANIT3, PLANIT4	135	108 - 161
Exercise online execution & editing	45	30 - 55
TOTAL TIME	630	496 - 730
HOURS	10.5	8.25 - 12.2

These data indicated that the initial time estimates were low. As a result, Units III and IV of the PLANIT lessons were not used in the remaining evaluation sessions. Evaluations at the USAIS incorporated those course segments and lessons as shown above, beginning with the short lesson, INTRO. Due to the slight overlap in finishing the ARI evaluations and starting the USAIS evaluations at Fort Benning, the above time results were used as parameters for the student scheduling plans at USAIS. Unfortunately, these initial empirical times were too low for the remaining students at ARI and the subsequent pool of students at the USAIS. This resulted in one case of missing performance exercise data at ARI; the missing data problem became more severe at USAIS as schedule conflicts required that a few students return to work assignments before completing all lessons or the practical exercise.

The daily time blocks scheduled for student evaluation activities were 1700 through 2400 hours at ARI, and 0800 through 1700 hours at the USAIS. The student accomplished deskwork portion of the practical exercise between scheduled online sessions at his convenience.

Each student had two 4-hour periods with the computer on consecutive days. When students scheduled into the evaluation had completed their first online session, subsequent sessions were scheduled by the SDC monitor as required, based upon the student's progress during the first session. The initial schedules were treated flexibly for the individuals as daily variations in lesson progress, progress on the practical exercise, computer availability, and conflicts in student commitments caused queues or gaps to develop in the computer usage schedule. The SDC monitor performed student rescheduling tasks as required.

C. COMPUTER OPERATION

The PLANIT software and user files for the evaluation--CAI lessons and student records--reside on the UNIVAC 1108 Exec 8 time sharing computer system in Aberdeen Arsenal, Edgewood, Md. Users tie remotely into the system via acoustic couplers and telephone lines. The ADDS Consul 880 terminals, two at ARI and two at the USAIS, were coupled to this facility for the evaluations. Each day's use of PLANIT at these locations was initiated when the evaluation monitor entered run and load statements to call up

PLANIT, in accordance with procedures which have been documented by ARI in an operator's guide.

Hardcopy output from the ADDS consoles was obtained via thermal printers which merely plug into the back of the console and activate with the PRINT LOCAL key. One printer was shared among the two consoles at ARI; the other two were allocated to each console at the USAIS. Hardcopy records included student interactions with the CAI lessons, PLANIT student performance records for individual students and lessons, student interaction during online execution and editing of his own frames, listings of the student's edited lesson frames, and hardcopy records of PLANIT lesson editing changes. A high-speed printer at ARI obtained listings of the student's practical exercise lessons and of PLANIT lesson changes through use of the PLANIT LIST command from terminals at ARI and USAIS.

PLANIT operated reliably and predictably throughout the period of checkout and evaluation. The UNIVAC system facility itself was characterized by weekly stretches of reliable operation interspersed with 1 or 2 days of unreliable operation and/or excessive response times. In one case, on two successive days at USAIS, response times became so excessive and reliability such a problem that students were excused and rescheduled.

Another problem, related to the interface installation of PLANIT with the time sharing system, is protection of PLANIT user files. Protection of newly authored lessons, editing changes, and student records on the UNIVAC PLANIT installation is achieved through a procedure requiring use of UNIVAC system file management software to update master copies of PLANIT files. During the entire period of online checkout and evaluations at ARI and USAIS, the ARI supporting programmer/analyst performed file update support functions on a daily basis, including update of master punched card decks. ARI performed the protection operation frequently to preclude potential loss of authoring and editing work done by students and course authors. Newly updated files are copied with system software to provide a primary and backup PLANIT system, each with its associated set of user files. ARI has documented the procedures for UNIVAC PLANIT 1.1 file maintenance.

D. STUDENT FLOW AND MONITORING PROCEDURES

The flow of students through the various course activities and the evaluation data obtained is shown in Figure 3-1.

Each student was briefed on the purpose of the study. The students filled out the PLANIT Data Questionnaire, received course handouts, logged in to PLANIT, and took the first lesson, INTRO.

Students were scheduled on the console until they completed the first three lessons which lead up to the practical exercise deskwork, the preparation of their own PLANIT lesson. The monitor surveyed and logged student activities and time, including computer downtime. Students were rescheduled as necessary. The monitor obtained PLANIT lesson records for each student after the student signed off from each online session.

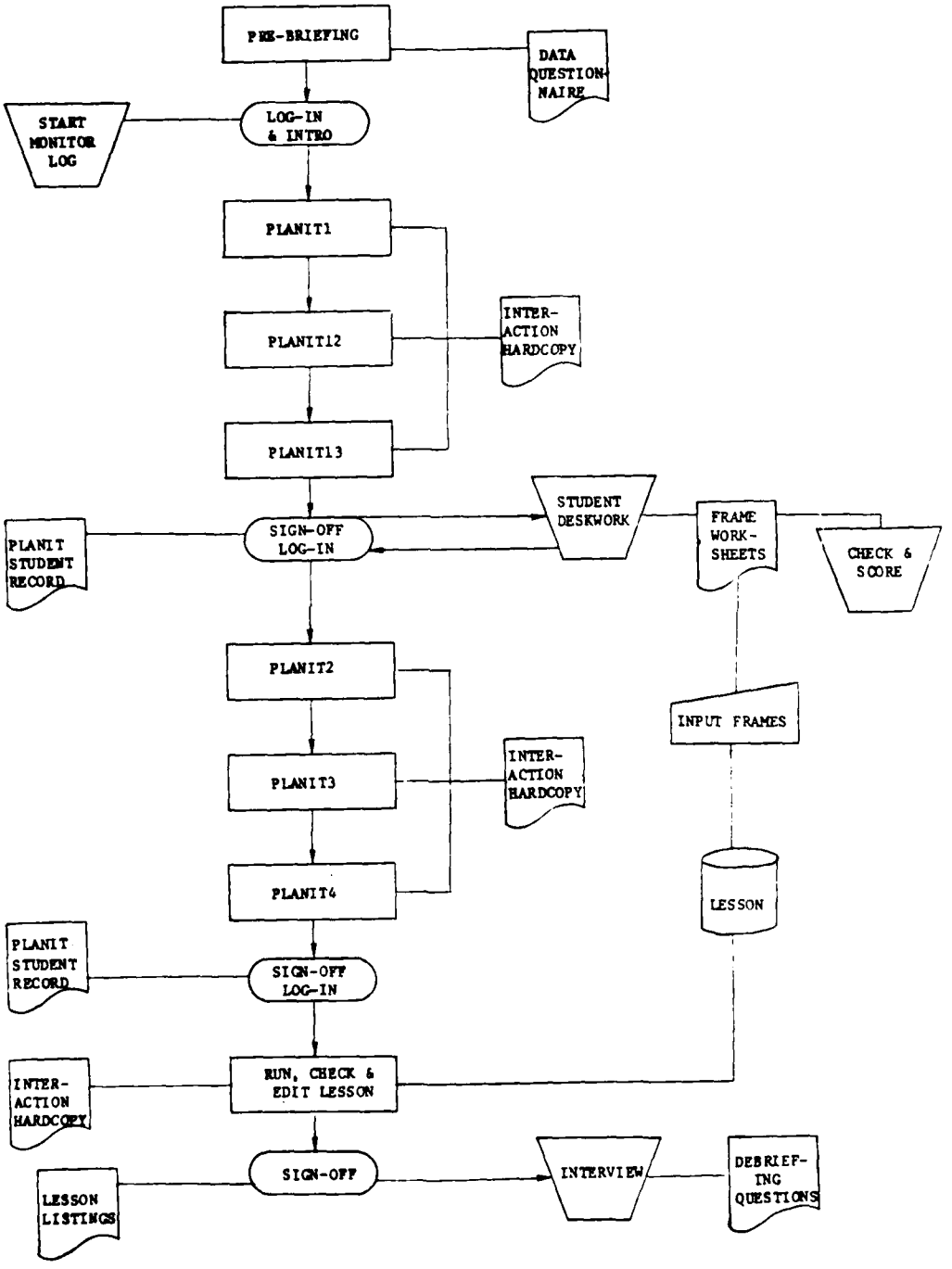


Figure 3-1. Student Flow

The vast majority of the students accomplished the offline portion of the practical exercise away from the CAI area (in the office or at home). When students left the CAI area to start the practical exercise, they were usually scheduled to return the next day.

After the students completed their practical exercise, they logged in and started the second series of three lessons on entering, running, checking, and editing lesson frames while monitors input the students' practical exercise lesson frames into the computer. This was done in parallel with the student taking the lessons, if a console was available. Otherwise, the students waited between completion of the last three lessons and their online use of PLANIT to run, check, and edit their own lessons. The monitor entered the student lessons exactly as prepared by the student on the PLANIT worksheets, except where so many missing entries were apparent that the monitor found it necessary to provide missing material prior to entry. The monitor noted these additions as required, and this material became part of the subsequent evaluation data.

When the student's lesson had been entered, the student logged in to PLANIT, called up the lesson, and executed it as a student. Students would usually try patterns of both right and wrong answers to test decision points in the lesson. Upon receiving a PLANIT error message or encountering frame constructions which did not operate as intended, the student displayed frame entries with the Print command and looked for causes of the error, using PLANIT Error Messages Reference List as an aid. Having found the cause of the error, the student formulated the correction needed, using the editing commands to make the correction. The student would resume lesson execution to verify that the change then operated as intended and was free of execution error messages. The student continued this process until the errors were corrected. When the student finished execution and editing, the SDC monitor obtained copies of the student's program from the printer--one for the student and one for evaluation records.

Following completion of the online portion of the practical exercise, SDC interviewed students in depth to determine their reactions to the CAI experience, the CAI lesson materials, and the adjunct materials. The PLANIT Debriefing Questions form was used as a basis for the interview. Some questions were open-ended, but many required selection of a multiple-choice response along a five-point scale. SDC gave students a copy of the form so they could read the question, but the interviewer filled out the form. All interviews were recorded on a cassette recorder. One interview form was completed, but the interview was not tape-recorded due to recorder malfunction. Another student could not be interviewed because of other commitments. The PLANIT Debriefing Questions are shown in Appendix C.

Section 4: ANALYSIS OF RESULTS

A. INTRODUCTION

The data were analyzed to determine the following:

- The degree to which students learned how to construct PLANIT frames using correct PLANIT syntax.
- The time required to complete the basic course CAI lessons and the practical exercise.
- The kinds of errors most frequently made on the practical exercise.
- Student attitude toward the CAI method of learning, the instructional package content and format, and associated procedures.

Each of these form one of the four results areas discussed below.

Data sources for these analyses, in the temporal order that the data were acquired for each student during the evaluations, were as follows:

- PLANIT Data Questionnaire.
- Hardcopy of lesson execution.
- Hardcopy of PLANIT student records.
- Student report of deskwork time.
- Student frame worksheets.
- Hardcopy of online student lesson execution.
- Final listing of the student's lesson.
- Monitor's log of online time for lessons and the practical exercise (with records of downtime).
- PLANIT Debriefing Questions.
- Interview audio tapes.
- PLANIT Data Sheets (compilation of data from several data sources).

B. PERFORMANCE ON THE PRACTICAL EXERCISE

1. General Performance

The fact that students in the available time successfully completed their own lesson encoded in the PLANIT language as a practical exercise shows that PLANIT skills learning did take place by means of CAI.

The practical exercise was completed by 12 of the 16 students who began the lessons. At ARI, all six of the students finished the Unit I and II lessons. Five submitted PLANIT frame worksheets to meet the offline requirements of the practical exercise, and finished the online portion of the exercise. Work commitments precluded the sixth student from completing the frame writing, despite having completed all of the lessons.

At the USAIS, of 10 students who started the PLANIT course, the disposition of students with respect to completing the exercise was as follows:

- Five students submitted frame worksheets to meet the initial exercise encoding requirements.
- Two students completed all lessons and submitted structured narratives for the presentation and feedback portions of their exercise, but with little attempt to meet the encoding requirements of PLANIT frame types Q, M, and D.
- One student completed all but one lesson and had just begun the practical exercise when she was called away on extended Army TDY.
- One student started the lessons, but became ill. Upon return to duty, work commitments and revisions to the CAI schedule made continuation impractical.
- One student completed half of the three lessons of Unit I in the time available to him away from other duties. While he expressed willingness to persevere, it became impossible to arrange mutual schedules for the requisite amounts of online time.

Assessing results on the practical exercise in broad terms, at ARI five of the six students met overall requirements and completed the course. At the USAIS, of the seven students who completed the course, five met the overall requirement and two did not.

2. Levels of Performance

The next question was how well students mastered basic PLANIT skills. Relative levels of student competence in using the PLANIT language were determined as reflected by numbers and types of errors on their frame worksheets.

A PLANIT skills index compared the level of success on the practical exercise and, by extension, showed the extent to which the lessons taught skills which could usefully be applied in the practical exercise. The PLANIT skill index was derived from the following ratio:

$$\frac{\text{Number of PLANIT Worksheet Errors}}{\text{Number of Frames Written}}$$

Using this as an index of applied learning, the 12 pooled ARI and USAIS students who completed the practical exercise were rank-ordered as shown in Table 4-1.

TABLE 4-1. RANK ORDER OF STUDENT SKILLS FOR POOLED GROUPS*

STUDENT	ORG.	RANK ORDER	SKILL INDEX	SKILL RATIO ERRORS/FRAMES	TYPE I ERRORS	TYPE II ERRORS	DESKWORK TIME (HOURS)	KINDS OF ERRORS
S1	ARI	1	0.31	4/13	2	2	3.0	3
S16	USAIS	2	0.42	5/12	3	2	2.5	3
S3	ARI	3	0.43	6/14	3	3	4.0	5
S4	ARI	4	0.89	8/9	2	6	4.0	6
S2	ARI	5	1.15	15/13	3	12	4.0	6
S10	USAIS	6	1.30	30/23	2	28	8.0	7
S13	USAIS	7	1.80	27/15	12	15	2.0	9
S5	ARI	8	2.33	28/12	10	18	3.0	5
S8	USAIS	9	2.50	30/12	19	11	1.5	7
S9	USAIS	10	3.09	34/11	14	20	2.0	12
S15	USAIS	11	3.66	46/12	N/A	46	4.0	6
S7	USAIS	12	4.08	49/12	N/A	49	3.0	6

*Students are identified as S1 through S16 in presenting results. This refers to the temporal order in which 16 students began the course. The analysis pool for practical exercise consisted of 12 students.

The top demarcation line (dotted) in Table 4-1 ends a band within which four students made less than one error per frame written, as denoted by the skill index. Also, the numerator of the skill ratio shows that error counts jumped appreciably for subsequent students. In the top band, three of the four students met the practical exercise requirement in the 12 to 20 frames stipulated. One student, S4, managed to cover nearly all of the practical exercise requirements in nine frames. It is concluded that these four students learned basic PLANIT skills to a highly acceptable level.

In conclusion, the performance of the six students who lie between the two dotted lines in Table 4-1 indicates that they learned PLANIT basics to an acceptable level. This conclusion was reached because the work submitted by these students was close to meeting the requirements of the practical

exercise and, without modification before input, was successfully executed and edited online by the student. The skills evidenced by the two students below the last line were "unacceptable." These students, S15 and S7, turned in partially completed PLANIT frames and did not meet either the practical exercise structure or the PLANIT coding syntax requirements, though the time they reported spending at the practical exercise task was as long or longer than most students.

Looking at the time data in Table 4-1, one might presume that students S13, S5, S8, and S9 might have higher skill indices had they spent as much time at the task as did students S2 and S10. The top two performers spent less time at the task than did those who ranked three through six. The interview comments showed that students S1 and S16 liked the discipline required by lessons and applied tasks more than some of the other students did.

Table 4-1 also shows the counts for two classes of errors found on student frame worksheets. These were:

- Type I errors. Coding syntax errors of the kind that halt lesson execution and cause PLANIT to display an error message (for example, a D-frame line which starts with any entry other than IF, ELSE, END, or an action command F: C: or B:).

- Type II errors. These are improper PLANIT constructions that would not half lesson execution, but which clearly would *not* operate in the way the PLANIT user intended. (For example, several lines of feedback for a given answer where only one of the lines is prefaced by an action command. This would still execute without an error message, but only the line prefaced by the action command would be displayed.)

The PLANIT course covered both syntax for use of language elements, as well as constructions of language elements to achieve desired ends basic to instructional programming. The higher number of Type II errors for any total error count shown in Table 4-1 shows that the lessons were more successful in teaching the elements and syntax for a given kind of PLANIT entry than in teaching how to structure elements into frame constructions which would operate as intended. Said another way, it is easier to teach the syntax than the semantics of using a programming language, and these results are generally supported by the experience of others.[1]

Finally, Table 4-1 shows that in terms of the performance skill index, the ARI group as a whole was better able to apply immediate learning from the lessons than the USAIS group. This may reflect that the ARI personnel proceeded more carefully on the encoding task, or that they more quickly comprehended the general forms of legal entries shown in lessons and on

---

[1] cf. Fenichel, R. R., Weizenbaum, J. and Yochelson, J. C. "A Program to Teach Programming." Communications of the ACM, Vol. 13, No. 3, March 1970, 141-146.

adjunct materials. All three of the top ARI performers had some experience with at least one other programming language, but the top USAIS performer did not.

3. Rationale for Skill Index Scoring

In deriving error counts for the skill index, SDC used the original student frame worksheets rather than their post-edited instructional program listings so as to provide comparable baseline data. The Type I and Type II coding errors discussed above were summed for each student's worksheets to derive the numerator of the skill index ratio. This equivalent weighting of the two types of errors reflects a tradeoff. Type I errors are damaging by virtue of halting lesson execution and yielding a PLANIT error message. Type II errors are serious in that PLANIT offers little assistance in identifying them--the burden is upon the PLANIT user to detect their occurrence and to determine the cause of the problem.

Both errors of omission and commission were counted. For example, a missing group or a missing action command within a group each counted as one error. Similarly, an incorrect group structure or an incorrect action command used within a group each counted as one error.

The ratio of errors to written frames assumes that those who had learned the most about the PLANIT language would make the fewest errors per number of frames written. The ratio tends to give more credit to those who carefully planned their content to stay within boundaries of the practical exercise requirements and to those who focused on the correct syntax for accomplishing the coding. The student who tried to push beyond the defined rules could magnify ratio errors if he tried to accomplish instructional programming techniques for which appropriate PLANIT language structures had not been presented to him in Unit I and II lessons. An example is the student who tried to write a compound decision statement where the AND/OR connectives were not taught in the basic lesson leading to the practical exercise, but were taught in other lessons (Unit III). Attempts such as these contributed to higher error counts among a few students (e.g., student S2 in Table 4-1).

4. Kinds of Errors on the Practical Exercise

Further analysis of the syntax and construction error counts (Type I and Type II) shown in Table 4-1 yielded 25 kinds of errors of omitted or incorrect entries. These kinds of errors with frequency and distribution of occurrence among students are shown in Appendix E.

The last column in Table 4-1 shows how many different kinds of errors were made by each student. The general trend toward upward progression in the number of error types committed among the first 10 students is in basic agreement with the rank ordering of proficiency based on the skill index ratio. That is, those students who made the fewest errors by and large also made the fewest kinds of errors. S15 and S7 are the two clear exceptions--their errors were fewer in kind because they were mostly errors

of omitted material and, therefore, could not be tallied as particular kinds of incorrect entries. The implications of omitting entire groups or attempts at PLANIT encoding within groups are decidedly more severe from the standpoint of demonstrated proficiency in using PLANIT than are finer-grained errors based upon incorrect entries.

The pattern of errors shown in Appendix E lends further support to the demarcation separating the four highly acceptable performances from the six acceptable ones. Errors among the first four students are quite sporadic, except where S1 and S16 tried to employ PLANIT techniques they had not been taught (Error #12). Most of the student errors were due to intermittent forgetting and/or inadequate check of their worksheet entries, rather than to a lack of learning.

The next six students (S2 through S9) show divergent distributions of the types of errors made; for example, S2 and S5 made relatively fewer kinds of errors than S10, S13, S8, and S9, but made them with greater regularity. Also, certain kinds of errors contributed heavily to the error counts among most of these students (errors 17-19, 21, 23, and 25). Suggestions for improving the course based upon performance errors which occurred with high regularity are contained in Part C of this section.

5. Background Data and Student Performance

The data in Table 4-1 showed that 10 of the 12 students (83%) who took the PLANIT lessons were able, on their first attempt at PLANIT programming, to program at an acceptable or highly acceptable error rate. This includes all five of those who completed the practical exercise at ARI, and five of the seven at The Infantry School.

Table 4-2 provides additional data on these students. Table 4-2 shows that the ability of apply PLANIT skills to an acceptable level of programming accuracy extended to those instructors with a high school level of education and no prior experience with computers or CAI instructional programming.

In general, students least experienced with computers and CAI took longer to take the lessons, when compared with students having higher levels of education and prior experience with computers. Six of these 10 students had previous computer experience and four had no experience. The six with computer experience averaged 6.83 hours to take the PLANIT lessons; the four without computer experience averaged 9.74 hours. Thus, prior computer experience seems to reduce the length of time required to learn how to program CAI course materials. However, attitude or a combination of attitude, skills, and knowledges brought to the task may also reduce learning time rather than skills and knowledge alone. Personnel working with CAI for the first time appear to be somewhat cautious in making their entries, giving the impression that they want to be absolutely certain that their response is correct. Those that have online computer experience know that if they are wrong, the feedback provided by the instructional material will so advise them, and that their progression through the material will continue as appropriate. Consequently, time required to take the lessons may

not be a function of computer skills and knowledges but simply knowing how programmed instruction, including CAI, operates. Lessons on how to plan programmed instruction sequences for CAI, to include basic strategies for presentation, answer processing, feedback, decisionmaking, and control of the student during the learning process would meet this need.

TABLE 4-2. PERFORMANCE DATA AND BACKGROUND VARIABLES

STUDENT	ORG.	SKILL INDEX RANKING	TIME TO WRITE FRAMES (HRS.)	NO. FRAMES WRITTEN	TIME FOR LESSONS (HRS.)	AGE	PRIOR COMPUTER EXPERIENCE	EDUCATION LEVEL
S1	ARI	1	3.0	13	4.78	29	YES	Ph.D.
S16	USAIS	2	2.5	12	6.75	36	NO	BMS & continuing
S3	ARI	3	4.0	14	7.25	43	FORTRAN	Ph.D.
S4	ARI	4	4.0	9	9.0	47	FORTRAN	Ph.D.
S2	ARI	5	4.0	13	6.18	34	YES	Ph.D.
S10	USAIS	6	8.0	23	5.25	26	YES	BS & continuing
S13	USAIS	7	2.0	15	8.5	30	Overview	BA
S5	ARI	8	3.0	12	9.0	38	NO	MS
S8	USAIS	9	1.5	12	11.95	35	NO	High School
S9	USAIS	10	2.0	11	11.25*	42	NO	High School
S15	USAIS	11	4.0	12	10.0	25	NO	BA
S7	USAIS	12	3.0	12	9.3	33	NO	BA

*Includes projected estimate of 70 minutes for lesson PLANIT4.

The time reported by students to write their own lessons in Table 4-2 averages 3.3 frames per hour for the first six students (S1 through S10) and 5.9 frames per hour for the next four (S13 through S9). The error rate for the last four students is considerably higher than for the first six. This indicates that these four students wrote their frames at a faster rate with less attention paid to checking for the somewhat stringent rules required to program the CAI material. Skill in PLANIT programming, as well as for any CAI system, requires careful attention to syntax and frame logic requirements. Student comments and monitor observation indicate that, as a group, the first six spent a considerable amount of time in checking their worksheets for potential execution errors before submitting their work. The last four apparently spent more time on developing their lesson content with less time applied to the coding aspects of

the lessons, as reflected by their error rate. This may indicate an instructor-centered orientation rather than an instructional programming centered orientation for some of the students. A course on developing CAI courseware, mentioned above, would probably help these students and certainly would help the two whose work was unacceptable (S15 and S7) because they confined their activities to course content, rather than tackling the coding and logic requirements of stand-alone, self-paced individual instruction.

The number of frames per hour written by these beginning students should not be used as a measure of what might be expected from an experienced CAI lesson author. Predictably, as students gained experience in writing frames and executing them on line, their productivity and skill in developing more complex lesson frames would increase.

C. RELATIONSHIP BETWEEN ERRORS AND THE COURSE

The paragraphs which follow examine each of the most prevalent kinds of errors with respect to recommendations for further improvements to the CAI lessons and adjunct materials. Errors #2, #7, and #8 bear some explanation. Their frequency and distribution show primarily that students S15 and S7 omitted almost all tries at entering PLANIT code or group structuring from their work. Conclusions about the lessons, therefore, are not based on error counts contributed by these students and errors.

1. Use of Action Commands (F: C: R: B:)

a. The high count for error #23 is because the procedure for continuing lines of feedback using commands F: or R: was inadequately covered in Unit I and II lessons leading to the practical exercise. Except for an example containing the technique, with no explanation or practice specific to line continuation, this topic is not treated until Unit III (PLANIT5) which the students did not take. The students who made no or few errors on line continuation of actions appear to have either limited their feedback statements to one line and/or relied upon the information contained in the handouts which shows examples and general form for line continuation.

The high incidence of this error warrants further action. Three alternatives are apparent. One would be to tell students taking basic lessons to avoid feedback with F: and R: which requires more than one line. Another way would be to modify lesson PLANIT12 to call upon lesson PLANIT5 and branch directly to the line continuation topic, switching back to PLANIT12 automatically at the end of the topic. This would require adding data linkage and decision points within both of these lessons. The third and most straight forward solution would be to move the topic out of PLANIT5 and into PLANIT12 and PLANIT13. Because of the high frame counts which now exist in PLANIT12 and PLANIT13, this would require the simple procedure of inserting a new PLANIT lesson name into the sequence rather than adding to either of the existing lessons.

b. The counts for errors #19, #20, and #21 show that students had sufficient trouble in choosing the correct action command for a situation and in writing the command statement in proper syntax to warrant some action. During interviews, the students often commented on the need for more discrimination drill and usage practice with each action command. Also, there are certain cases where order of the action commands on a single line is important to proper execution of the author's intent; for example, the C: must precede a B: or R: on a line when both are used, because the B: and R: are types of branches which take precedence over a subsequent C: command.

It is recommended that two drill sequences be added to lessons PLANIT12 or PLANIT13. The first would give practice in discriminating among the basic meanings of the four action commands and each of their general forms. The second would be a sequence where the student must state the appropriate action commands to use and select the correctly formatted action line for at least five action requirements. The lesson would cause the student to repeat each of these drill sequences until performance met criterion.

2. Use of Counters for Scoring

Testing the values of counters did not pose any special problems in the Decision frame. However, how to initialize and increment counters in Group 4 of Question and Multiple-Choice frames did pose a problem, despite a whole topic devoted to this in the first half of PLANIT13. Error counts for #5, #6, #17, and #18 indicate this problem. Where students entered a statement to initialize a correct answer counter as required by the practical exercise (e.g., C:SET RITE=0), they did use the correct syntax. But several students never entered the required statement, or put the statement in an improper position with respect to the occurrence of an answer (errors #5 and #18). Two of the four students who left out statements to initialize a counter also left out subsequent statements to increment the counter (error #6). The part of PLANIT13 covering counters does not emphasize either the need for or the technique of how to initialize counters before incrementing them, explaining the absence of statements to initialize counters. Initializing counters in the wrong Group 4 line position also can be explained; it was not emphasized by the lesson. Errors in incrementing a counter in the wrong position with respect to an answer tag shows that, unlike other uses of action commands, students did not always connect this action to the occurrence or the nonoccurrence of a response matching an answer tag.

It is recommended that the counters topic (first half of PLANIT13) be reworked, but not necessarily expanded. The rework would emphasize how to initialize a counter to zero on the first Group 4 line, unconditionally. It would then show how line 2 of Group 4 would increment the counter by one for a correct answer, followed by feedback, while an incorrect answer would not result in a statement to increment the correct answer counter. Also, another figure for the handouts should show an explicit example of counters usage to support the lesson activities and to serve later as a reminder to the student. Frames now existing in PLANIT13 would appear to

give enough practice in differentiating the meaning and uses of the two forms of C: command, and the general forms for each use.

3. Decision Statements

All students tried to write Decision frames to satisfy requirements except S15 and S7. Of these 10 students, only S5 and S9 had problems with their decision statements as shown by errors #24 and #25. Student S5 had only minor errors with spacing among the elements of each decision statement, while S9 had trouble employing the correct decision words and order of language elements among lines of decision frames. Based upon the low error count across students for decision frames, no immediate recommendations are offered to modify the second half of PLANIT13 nor the optional topics in PLANIT51. The exercise instructions, logic diagram, and Exhibit 4 of the handouts apparently reminded some students of the decision structures during their deskwork--e.g., S16 commented on using all of these, especially the logic diagram.

4. Response Processing

The frequency of error for improper Group 3 constructions (error #12), even for the top performers, shows what every CAI author knows--that the type of open-ended question constructed will dictate how difficult it will be to cover the variety of response contingencies for correct and incorrect answers to the question. Most of these errors occurred because students tried to write questions which needed some of the special techniques for KEYWORD and for accepting and evaluating a series of responses which were contained in Unit III (PLANIT5). No specific recommendations are made on how to counter the occurrence of this error, other than making PLANIT5 available as an option and continuing practice in writing question frames. Expansion of the series of handouts is suggested to include specific frame application examples using the three forms of KEYWORD for given response requirements. A less desirable solution would be more severe constraints placed upon the content requirements of each student's practical exercise; e.g., legal forms of questions pre-specified to avoid the student author's overcommitment.

D. TIME FOR COURSE SEGMENTS AND LESSONS

Table 4-3 presents the composite mean times and range of times for each CAI lesson unit, as well as for the two parts of the practical exercise. The average time to complete the course for the two groups combined was 13 hours. The ARI group overall was faster than the USAIS group. The ARI average was 12.4 hours; the USAIS average was 13.7 hours. Unit I lessons for the pooled group took an average of 5.5 hours, ranging from a minimum of 3 hours for a student at ARI to a maximum of 7.7 hours for a student at USAIS. Unit II lessons for the pooled group average 3 hours, ranging from a low of 1.8 hours at ARI up to 4.25 hours at USAIS. The ARI group spent about 0.3 hours more at exercise deskwork on the average than did the USAIS group, with the amount of time spent in this activity varying considerably more at USAIS than at ARI. The groups spent about the same time in the online activities.

TABLE 4-3. TIMES BY COURSE SEGMENT*

COURSE SEGMENT	USAIS		ARI		BOTH GROUPS	
	MEAN TIME (Mins.)	TIME RANGE (Mins.)	MEAN TIME (Mins.)	TIME RANGE (Mins.)	MEAN TIME (Mins.)	TIME RANGE (Mins.)
UNIT I (3 Lessons)	361	195-462	303	178-410	332	178-462
Exercise Deskwork	197	90-480	216	180-240	207	90-480
UNIT II (3 Lessons)	198	120-255	161	108-200	180	108-255
Online Tryout & Editing	66	25-100	62	30-90	64	25-100
TOTALS FOR COURSE	822		742		783	

*Lesson timing data was sufficiently complete across the three lessons of each unit for a combined groups N=14. The ARI N=6 and the USAIS N=8.

Tables 4-4 and 4-5 show the mean and range of times for each CAI lesson for students in the ARI and USAIS groups, respectively.

With reference to Table 4-4, at ARI the slowest students took 1.25 to 2.5 times as long as the fastest students, depending on the lesson. Table 4-5 shows that at USAIS the slowest students took from two to three times as long (depending on the lesson) as the fastest students. These differences probably reflect the relative homogeneity of the ARI group with respect to experience and education, as compared to the relative diversity within the USAIS group.

The lessons themselves, except for a few decision points with review loops (see Section 2, logic diagrams), generally require all students to traverse the same route of frames and checkpoints. The time differences among students, therefore, are accounted for primarily by (1) differences in time among students in responding to each question; (2) amount of time spent looking at student handouts and writing on worksheets during the lessons; (3) amount of time spent taking additional notes; and (4) amount of time contributed by various distractions to student focus (e.g., sporadic occurrences of system downtime and slow system response times for certain students).

TABLE 4-4. ARI STUDENT TIMES BY LESSON

Student	LESSON						Student Total
	PLANIT1	PLANIT12	PLANIT13	PLANIT2	PLANIT3	PLANIT4	
S1	50	68	60	40	20	48	286
S3	85	92	97	73	33	55	435
S4	95	130	135	80	45	55	540
S2	60	85	92	52	35	47	371
S5	95	130	135	85	45	50	540
S6*	75	150	185	105	35	60	610
Total	460	655	704	435	213	315	
Lesson Mean	77	109	117	73	36	53	
Lesson Range	50-95	68-150	60-185	40-105	20-45	47-60	

*This student completed all lessons, but not the practical exercise.

TABLE 4-5. USAIS STUDENT TIMES BY LESSON

Student	LESSON						Student Total
	PLANIT1	PLANIT12	PLANIT13	PLANIT2	PLANIT3	PLANIT4	
S16	60	110	95	55	35	50	405
S10	45	60	90	45	30	45	315
S13	70	115	130	90	35	70	510
S8	120	182	160	105	65	85	717
S9	110	155	190	105	45	70**	675
S15	90	145	160	95	35	75	600
S7	100	110	138	100	30	80	558
S14*	110	140	200	130	35	70**	685
S11*	110	-	-	-	-	-	-
S12*	185	-	-	-	-	-	-
Total	1000	1017	1163	725	310	545	
Lesson Mean	100	127	145	91	39	68	
Lesson Range	45-185	60-182	90-200	45-130	30-65	45-85	

*These students did not complete the practical exercise.

**Estimated times are due to noncompletion of PLANIT4.

E. ATTITUDES TOWARD THE PLANIT COURSE

1. Summary

Twelve students were interviewed to determine their overall reactions to the course, confidence in skills learned, reactions to specific course characteristics, and suggestions for improvements. Of the 12, one had completed all the lessons but not the practical exercise.

The debriefing questionnaire (Appendix C) was used to record interview responses, with the entire interview recorded onto cassette tapes.

The attitude survey generally supports the relative scalings of students with respect to level of learned skills and learning efficiency. The most accurate or efficient students and those with related experience or work interests tended to be the most positive to the experience, while those at the unacceptable level were the most critical of both the course and of their own personal approach toward it. ARI researchers as a group tended to offer a variety of suggestions regardless of their performance level.

In summary, the response to and acceptance of learning about PLANIT via the CAI course and practical exercise is generally positive, both among personnel at ARI and The Infantry School. The questionnaires and taped interviews show that the material was well received. In several cases highly enthusiastic responses were forthcoming at ARI and the USAIS. Only two USAIS interviewees revealed sufficient ambivalence with respect to learning about PLANIT to warrant a doubtful conclusion on how fruitful would be further learning in the automated instruction mode. Therefore, some students should be eased into online simulation and practice with PLANIT via more traditional individual training methods such as PI text, audiovisual presentations, and over-the-shoulder training. This is probably also true of one student at ARI who expressed a preference for learning about CAI by means other than CAI, though this student was the only one interviewed who had not done the practical exercise portion of the course.

The positive and negative interview comments and observation of students at work show that the CAI lessons are challenging. PLANIT becomes more meaningful to students as pieces fall into place during the Unit II simulations and subsequent hands-on PLANIT work. Any difficulty with course content, an unreliable system, or ambivalence of personal priorities with respect to learning PLANIT can disrupt some students sufficiently such that the necessary perseverance in the lessons and exercise is obtained at the cost of negativism during learning.

Student comments and data on the questionnaires elicit the following general reactions:

- Overall Reaction to the Course

  1. Good--but not perfect.

  2. Most instructors can do it, but college background is helpful.

  3. Simpler than anticipated at the start.

4. Became more eager after starting.

5. Became more eager during Unit II--tough at the start.

6. Adjunct handouts helpful.

7. PLANIT is simple, easy language to learn.

8. PLANIT is very logical and straightforward.

9. Working at own pace with one-on-one approach is good.

- Confidence in Learning PLANIT

  1. Understanding satisfactory or very satisfactory.

  2. Learned enough to develop lessons and continue with PLANIT practice.

  3. Very few problems in online execution and editing.

- Specific Course Characteristics

  1. Hardest part of deskwork is sequencing content and checking against exercise requirements.

  2. Editing is the simplest PLANIT topic.

  3. All pretty good.

  4. Unit II the best.

  5. Amount of time about right or fairly short.

  6. Amount of information about right.

  7. Technical detail very satisfactory.

  8. Satisfactory or very satisfactory organization of material.

  9. Fairly large quantity of adjunct materials.

- Suggestions for Improvement

  1. Keep system up and response times short to maintain continuity of thought and application.

  2. Provide more drill on action commands selection and syntax.

  3. Add review options under control of the learner.

  4. Actions for unanticipated answers is confusing as presented.

5. Use of counters is confusing; maybe eliminate counters and just use pattern and summary forms of decision statements.

6. Cover more goal orientation and big-picture of PLANIT capabilities at the start, before the details.

7. Provide more or different kinds of offline review materials (more applications examples, summary of topic review points).

8. Earlier PLANIT simulation and hands-on with abbreviated lessons leading up to this.

2. Overall Reaction to the Course

Five questions (questions 1, 2, 8, 24, and 25) obtained scalings and commentary related to the course, the CAI method of teaching, and overall value of the experience to the student.

Eleven of the 12 interviewees said they liked the course or liked it very much (one-half). One reported disliking it because of fatigue and conflicting priorities. Those who liked it very much said they found PLANIT logical, English-like, and easy.

Nine students reported that learning about a CAI language (PLANIT) in the CAI mode was effective, or very effective. Three students reported the method as borderline, citing personal learning style preferences of a multimedia approach (one efficient student noted that the lag in computer response tended to break his continuity of thought).

The ease with which students reported they learned PLANIT varied considerably. Eight reported it as easy or very easy (four each), while four reported it difficult or borderline difficult (two each). Those finding it easy also reported compatible interests and experience. The others cited negative transfer from FORTRAN, nonanalogy to other kinds of learning, and critical predispositions because of the difference from normal Army teaching methods as contributing to their difficulty.

Summarizing commentary about the experience and the course ranged from "very enjoyable and would like to go into advanced features of PLANIT" (USAIS), to "an invaluable way to learn to appreciate the skills of CAI authors" (ARI), to "trapped and fatigued with too many personal commitments" (USAIS).

3. Confidence in Learned Skills

Three questions (questions 5, 13, and 20) tapped the student's self-estimate of the strength of newly acquired skills and knowledges. Virtually all students said they knew enough about PLANIT to develop their own lessons, with one at USAIS and several at ARI expressing an eagerness to learn more about PLANIT. Four students qualified these expressions of confidence to indicate they would need more practice, a reference manual, or

personal guidance. One felt he could write PLANIT frames short of the encoding needed for entry to the computer.

Ten of the 12 students rated their understanding of PLANIT content as very satisfactory (6) or satisfactory (4). Nobody rated it as unsatisfactory, but two students in the lower half of the skill index distribution did rate their understanding as borderline. All of the students except one estimated that they understood from 80% to 100% of the instruction either at the time it was presented or after they had gone through the practical exercise. One student estimated 50% understanding. Several students qualified their estimates with respect to amount retained, noting that loss was fast until the lesson simulation practice and subsequent online work with PLANIT. This argues for quicker lesson transition out of the academics and into simulation practice of online PLANIT use.

4. Reactions to Course Specifics

Fifteen questions (questions 3, 4, 6, 7, 9-12, 14-19, and 21) obtained scalings and commentary on course characteristics, including specific suggestions for improvements to CAI lessons, adjunct materials, and procedures.

Eight students reported no problems in interacting with the ADDS console. Four students reported a series of annoyances including lack of keyboard familiarity, initial oddity in taking a separate key action to make an entry, feedback delay times, inability to use the quotation mark key during editing, and fatigue.

Some students reported the course as "all pretty good"--no standout strong or weak points. Others indicated that the best aspect either began from the Decision frame instruction (PLANIT13) onward, or from the Unit II lessons covering online entry, execution, and editing. The practical exercise itself frequently was singled out as the reinforcing experience which caused the prior work to fall into place.

The amount of time available during the course sessions for a satisfactory understanding of PLANIT was seen as about right by half the students, fairly short by four students, and fairly long or much too long by two students. Those seeing it as too short wanted more time for review, offline deskwork, and online practice. Interestingly, these also were the students who ranked 4 through 7 on the skill index indicating a recognition that performance improvement could come through more application time. On the other hand, those ranking the course too long saw it as quite dry and laborious for teaching simple concepts. They ranked in the bottom half on the skill index for performance on the practical exercise.

Nine students felt the amount of PLANIT content information in the course was about right, two rating it fairly large and one fairly small. Six rated the technical detail level as very satisfactory and three said it was satisfactory. One rated it as unsatisfactory because he wanted to learn more precisely how PLANIT operates upon the coded entries provided by a lesson author. All students rated organization of the course material satisfactory (seven) or very satisfactory (five).

Two students would have liked more adjunct materials, while nine thought the quantity was on the high side. Only 2 of the 12 pointed to adjunct materials they considered irrelevant or too unique (e.g., the specific Q-frame application shown). Specific suggestions for future revisions to adjunct materials included (1) combine the general forms and examples by following an outline of legal entries for each frame with several examples illustrating applications of the general forms; (2) provide a summary sheet of main review points from each lesson; and (3) provide a compilation of key PLANIT language terms with meanings and formats for quick reference.

Virtually no problems were reported by students when it came to executing and editing their own lessons except nervousness about mistakes until they saw how PLANIT helped them to correct their mechanical errors.

Three students reported the following problems in preparing worksheets for their lessons: spending too much time worrying about content considerations, difficulty in organizing a teaching strategy for the topic, and desiring personal question-answer review while writing. Other items mentioned as difficult aspects of this phase were selecting a topic and checking worksheets against the exercise requirements.

The suggestions for improving the course which emerge with the greatest clarity and regularity can be summarized as follows: (1) have more drill and practice in use of action commands; (2) have more online and off-line review options; (3) move earlier to simulated and hands-on use of PLANIT; (4) start with the big picture of lesson writing and PLANIT capabilities first, rather than so many details so fast; and (5) avoid fatigue and time-crunches by considering carefully the scheduling of each student.

Section 5. CONCLUSIONS AND RECOMMENDATIONS

The following are conclusions and recommendations as a result of the development and evaluation of a PLANIT course on constructing PLANIT lessons.

A. CONCLUSIONS

- The PLANIT CAI course has been run successfully on the UNIVAC 1108 computer at Edgewood Arsenal, is capable of being run on the CDC 3300 computer at ARI, and was evaluated from remote consoles at ARI and the U.S. Army Infantry School.

- The PLANIT course is successful in teaching students to program their own PLANIT lessons and to execute and edit the lessons online (on computer).

- The majority (93%) of military and civilian students who participated in this project approves of this method of learning how to use PLANIT to develop instructional material.

B. RECOMMENDATIONS

- As a result of MASSTER Test 122 and this study, the Army has an operational CAI system, PLANIT, and a course of instruction for training military personnel to develop their own PLANIT lessons in their subject matter areas. The Infantry School should continue to use the PLANIT course to train subject matter experts in programming PLANIT courseware. These trained personnel should then develop CAI courses to be used at the school for the following purposes: (1) to train personnel in military subjects; (2) to determine the problems, cost, user acceptance, and effectiveness of CAI as a training vehicle; and (3) to gain experience and expertise in the development and use of computer-assisted instruction.

- Other service schools, in addition to The Infantry School, have an ongoing and continued interest and requirement for developing computer assisted instruction. The capability operational at one service school should be made available to other service schools. The PLANIT course should be installed at other service schools including the Command and General Staff College.

- The PLANIT course developed in this study is effective in teaching how to use PLANIT. The evaluation and tryout of the course has pointed to areas in which either change to or reorganization of the course materials would make it an even more effective instructional vehicle. The changes to the PLANIT course discussed in this report should be implemented.

- Some students taking the PLANIT course reported problems in organizing their lesson, preparing answer strategies, and controlling the student--problems inherent in individualized instruction where the lesson materials, rather than the instructor, carry the instructional burden. These problems are associated with any individualized instruction whether it is programmed instruction (PI), audio-visual instruction (AVI), or multimedia instruction. A separate PLANIT course should be developed to cover the preparation of course materials for individualized instruction. This course would cover strategies for presentation, answer processing, testing, feedback, and control of the student learning process. This course would be a prerequisite for the course on PLANIT instructional programming developed in this study.

- Programming lesson materials for computer input requires a fairly strict adherence to the form and syntax required, especially in the areas of answer processing, decision statements, and branching. Successful completion of the PLANIT course indicates mastery of these kinds of skills which are required for successful instructional programming. If the lack of skills is indicated by failure on the course, then the PLANIT course itself becomes a potential selection instrument. The PLANIT course should be used to select personnel to develop CAI course materials.

- PLANIT is in use in many universities and other computer centers in the United States and a number of foreign countries. Those using PLANIT or intending to install PLANIT are members of the PLANIT Users Group, which is supported by the National Science Foundation. Each of these has the problem of training personnel to program PLANIT courseware. The PLANIT course should be made available to members of the PLANIT User's Group.

APPENDIX A

PLANIT COURSE OUTLINE AND OBJECTIVES

COURSE OUTLINE

USING PLANIT FOR AUTOMATED INSTRUCTION

I LESSON CONSTRUCTION, PART 1

II ENTERING LESSONS AT A TERMINAL

III EXECUTING AND CHECKING A LESSON

IV BASIC LESSON EDITING

V LESSON CONSTRUCTION, PART 2

VI BASICS OF CALC AND THE (P)ROGRAMMING FRAME

USING PLANIT FOR AUTOMATED INSTRUCTION

CONTENT DEVELOPMENT OUTLINE

Subject Breakdown	General Task/Objective
I. LESSON CONSTRUCTION, PART 1	
1. INTRODUCTION	1.0 Match the four PLANIT frame types with short statements of functional purpose.
a. Unit learning objectives	
b. PLANIT, AI lessons, & PLANIT frames	
c. PLANIT frame types: Q, M, D, P	
2. THE (Q)UESTION FRAME	2.0 Construct a Q frame to present information, a question requiring a single word answer, and which uses action commands for feedback and branching.
a. Purpose of Q frame	
b. Example and explanation of a Q frame and how it appears to students	
c. Group layout by function	
d. Constructing a Group 1 Frame Identifier	2.1 Construct a Q frame Group 1 identifier.
(1) Enter group number in Col 1.	
(2) Format for frame number and frame type	
(3) Frame label optional	
e. Constructing a Group 2 display	2.2 Construct a Q frame Group 2 to present information and a question.
(1) Displays information, instructions, questions	
(2) Example of Group 2 display	
(3) Plan as you want display to appear	

Subject Breakdown	General Task/Objective
(4) Plan Group 2 to fit screen size (5) Question formats which require single words or numbers as answers (6) Enter group number in Col. 1 (7) Begin display lines in Col. 2 (8) Use \ symbol to cause line skip	
f. Constructing a Group 3 answer list (1) Group 3 accepts an answer (2) Group 3 detects answers (3) List acceptable correct answers (4) List most likely incorrect answers (5) Assign + to correct answers (6) Assign letter tags for a straight match (7) Example of answers matched in Group 3 (8) Example of single numbers matched if number tags are used (9) Enter group number in Col. 1 (10) Line format for correct answers (11) Line format for incorrect or neutral answers	2.3 Construct a Q frame Group 3 for the Group 2 question above (2.2).
g. Basic use of KEYWORD in Group 3 (1) To find answers among other words (2) Example of answer matching with KEYWORD (3) Line format for Ø KEYWORD ON (4) Use KEYWORD one frame at a time	2.4 Change the Q frame Group 3 above (2.3) to detect answers as key words.

Subject Breakdown	General Task/Objective
h. Basic use of PHONETIC in Group 3 (1) To match misspelled answers (2) Example of answers matched with PHONETIC (3) Example of answers not matched (4) Line format for Ø PHONETIC ON (5) Use PHONETIC one frame at a time	2.5 Change the Q frame Group 3 above (2.4) to match misspelled correct answers.
i. Constructing Group 4 Actions (1) To tie actions to Group 3 tags (2) To handle unanticipated answers (3) Examples of actions and effects (4) Select a feedback message (F:, R:, C:) (5) Print instructor's messages (F:,R:) (6) Skip ahead or back in the lesson (B:) (7) Keep track of logic when using B: (8) Enter group number in Col. 1 (9) Line format for action entries (10) Use of -tag for unanticipated answers (11) Same action for several tags e.g. BCD F: LET'S REVIEW B:2 (12) Actions extending to a second line	2.6 Construct a Q frame Group 4 using F:, R:, C: feedback and B: for the Group 3 entries above (2.5).
3. THE (M)ULTIPLE CHOICE FRAME	3.0 Construct an M frame to present information, ask a question with four answer choices, give feedback for all answers, and advance only if a correct answer is given.
a. Purpose of M Frame	
b. Example and explanation of an M frame and how it appears to students	

Subject Breakdown	General Task/Objective
c. Group Layout by Function	
d. Group 1 frame number and type	3.1 Construct an M frame Group 1 identifier.
e. Constructing a Group 2	
(1) Same display capabilities as Q (2) It must ask a question (3) Only question stem is in Group 2 (4) Same format rules as Q (5) Example of Group 2 display	3.2 Construct an M frame Group 2 to present information and a question stem. 3.3 Construct an M frame Group 3 answer set for the Group 2 above (3.2).
f. Constructing Group 3 Answer Choices	
(1) Example of Group 2/3 display (2) Format answer choices for display (3) Assign letter tags to answers (4) Assign + to one answer only	
g. Constructing Group 4 Actions	3.4 Construct an M frame Group 4 using F:, R:, and C:name=value for the Group 3 answer set above (3.3).
(1) Same action commands as Q (2) Same line format rules as Q (3) Automatic message for unanticipated answers (4) Recording and updating scores using C:name=value (5) Example of actions and effects	
4. THE (D)ECISION FRAME	4.0 Construct a D frame to tailor feedback and branching according to number right and wrong over a sample set of frames.
a. Purpose of D frames	
(1) Check student performance over frames (2) Check student path through frames (3) Tailor student feedback to conditions (4) Skip back or ahead based on conditions	

Subject Breakdown	General Task/Objective
b. Constructing a D Frame Group 2 (1) Example and explanation of conditions and actions in a D frame (2) Conditions (the IF part) (3) Actions (F:, B:, C:) (4) Alternate actions (the ELSE part) (5) Enter group numbers in Col. 1 (6) Assign frame number and type (7) Put IF statement on first Group 2 line (8) Put IF statement actions on next line (9) Put alternate (ELSE) actions on last line	
c. Basic use of Summary Decision (1) Example of PLANIT student records (2) Summary form tests RIGHT, WRONG, SEEN (3) Format for testing ALL or NONE (4) Example of D frame and effects (5) Format for using relationals (GR, GQ, LQ, LS) (6) Example of D frame and effects	4.1 Construct a D frame of summary form to check if at least three of four frames were answered correctly.
d. Basic use of Pattern Decision (1) Example of PLANIT student records (2) Pattern form tests +,-, answer tags (3) Format for testing answer tags (4) Examples of D frames and effects	4.2 Construct a D frame of pattern form to check if three frames were each answered with a specific tag.

Subject Breakdown	General Task/Objective
(5) Format for testing + or - (6) Example of D frame and effects	
e. Basic Use of Computational Decision (1) Example of score values changing in a score counter (2) Scores are set and changed in Q or M frames (C:name=value) (3) Computational form checks on values in named counters (4) Format for testing counter values (5) Example of D frame and effects	4.3 Construct a D frame of computational form to check if a score counter has reached a specified value.
5. FIRST THINGS LAST: PLAN THE LESSON a. Write objectives for students b. Write teaching steps c. Write questions for each step d. Write lesson test questions e. List correct and wrong answers	5.0 Arrange a short list of information, questions, feedback messages, and decision checks into proper order for a teaching objective.
f. Tie feedback and branching to answers g. Decide on go/no-go check points h. Decide on go/no-go standards	5.1 Assign the material ordered above to PLANIT frame types.
i. Sequence the information, questions, and go/no-go decisions j. Assign frame numbers and frame types k. Name the lesson (naming rules)	5.2 Assign frame numbers and PLANIT lesson name.

Unit Entering Lessons at a Terminal

Subject Breakdown	General Task/Objective
II. ENTERING LESSONS FROM A TERMINAL	
1. HOW TO SIGN ON THE SYSTEM	1.0 Enters identifier, author mode, and choice of frame type in response to simulated PLANIT messages.
a. Obtain PLANIT author identifier	
b. Sign on to PLANIT	
(1) PLEASE LOGIN** appears	
(2) Type in your identifier and hit carriage return (CR)	1.1 Enters ? to obtain simulated help message after ENTER COMMAND.
(3) ENTER COMMAND appears	
(4) * appears next	
(5) PLANIT coaches and prompts	
(6) Use ? for more help	
c. Erasing lines and characters	
(1) Procedures to erase before a CR	
(2) Always use CR to enter material	
2. ENTERING A NEW LESSON	2.0 Enters a short Q frame, advances to the next frame, and selects frame type M in response to simulated PLANIT messages.
a. ENTER COMMAND appears on scope	
b. Always wait for the *	
c. Type and enter A	
d. Q/M/D/P appears next	
e. Select and enter frame type	
f. PLANIT numbers each frame	
g. Group 1 message appears on scope	2.1 Enters a Q frame Group 1
(1) FRAME 1·ØØ (example)	
(2) Bypass optional LABEL entry	
(3) Advance to Group 2 with CR	

Subject Breakdown	General Task/Objective
h. Group 2 message appears on scope (1) G2.TEXT for Q or M frame (2) G2.CRITERIA for D frame (3) G2.STATEMENTS for P frame (4) Type and enter each line after * (5) Use space bar to space across (6) Use \ for display line skips (7) Advance to next group, if any, or to next frame (*CR)	2.2 Enters a Q frame Group 2.
i. Group 3 message (if any) appears (1) G3.ANSWERS (Q & M only) (2) Use answer list (3) Type and enter each line after * (4) Advance to next group (*CR)	2.3 Enters a Q frame Group 3.
j. Group 4 message (if any) appears (1) G4.ACTIONS (Q & M only) (2) Type and enter tags and action lines (3) Rule for continuing actions on subsequent lines (example) (4) Advance to next frame (*CR)	2.4 Enters a Q frame Group 4.
k. Entering subsequent frames (1) Use CR or #A to start next frame (2) Q/M/D/P message appears (3) Select next frame type (4) Group 1 identifier appears (5) Check frame number (6) Advance to and enter groups as before	2.5 Advances to the next frame and selects type M.

Subject Breakdown	General Task/Objective
3. HOW TO DISPLAY FRAME ENTRIES a. Enter # after any * b. ENTER COMMAND appears c. Use (P)rint command (1) Display a frame (e.g., 1,P) (2) Display a group (e.g., 1, 3, P) (3) Display a line (e.g., 1, 3, 2, P)	3.0 Enters commands to display frame, groups, and lines built in 2.0 (simulated displays).
4. HOW TO SIGN-OFF THE SYSTEM a. Name the Lesson Frames Built (1) PLANIT saves frames automatically (2) Name is for finding lesson later (3) Assign a name (naming rules) (4) Enter SAVE and lesson name (5) Enter SAVE identifier when IDENTIFY YOURSELF appears (6) Enter UNLOCK when asked (7) Record the lesson name and identifier b. Enter QUIT to Sign Off	4.0 Performs a simulated lesson save and sign-off from PLANIT.
5. HOW TO CONTINUE A LESSON a. Sign on with author identifier b. Enter GET and lesson name c. Enter the save identifier after IDENTIFY YOURSELF appears	5.0 Performs a simulated login, lesson retrieval, and entry to author mode.

Subject Breakdown	General Task/Objective
d. Enter A after ENTER COMMAND	
e. Continue building after Q/M/D/P appears	

Unit Executing and Checking a Lesson

Subject Breakdown	General Task/Objective
III. EXECUTING AND CHECKING A LESSON	
1. REASONS TO EXECUTE YOUR LESSON	1.0 *Identifies instances and non-instances of display, answer-matching, and branching problems resulting from an execution.*
a. See lesson as a student would see it	
b. Check displays	
c. Check answer matching	
d. Check frame sequencing	
2. HOW TO EXECUTE YOUR LESSON	2.0 Executes the (simulated) set of frames built in Modules I & II from the start and from a specified frame.
a. Log in with your author identifier	
b. GET the lesson by lesson name	
c. Enter save identifier after IDENTIFY YOURSELF appears	
d. CLEAR any prior execution record	
(1) EX to execute from start (example)	
(2) EX from a specific frame number (example)	
e. Answer questions when * appears	
f. Use # before EX to re-execute	
g. Re-execution with and without CLEAR	

Subject Breakdown	General Task/Objective
3. ERROR MESSAGES DURING EXECUTION	3.0 Identifies probable causes of common error messages displayed during a simulated execution using an "Execution Errors Reference List".
a. Messages when PLANIT finds an error (1) Error condition message (2) Frame, group, line where error occurred (3) PLANIT halt message	
b. Finding cause of the error (1) Use P to display frame, group, and line (2) Find the error condition message in the "Execution Errors Reference List" (3) Examine the line in error against explanation in the Reference List (4) Decide how to fix the error	3.1 Enters P command to display a frame, a group, and a line after a simulated execution halt.
c. Actions available after finding the error (1) Record error and execute from another frame (2) Enter # to fix the error by editing	3.2 Enters EX command to execute from the next frame after the frame in error.

Unit Basic Lesson Editing

Subject Breakdown	General Task/Objective
IV. BASIC LESSON EDITING	
1. REASONS TO CHANGE LESSON FRAMES	1.0 Fixes errors found in a simulated execution (3.0 above) using E, I, and D commands in response to simulated PLANIT messages.
a. Fix errors causing execution halt	
b. Edit sentences and other displays	
c. Fix frame sequencing instructions	
d. Improve answer-matching lists	
2. WHEN EDIT COMMANDS ARE USED	2.0 Obtains simulated message ENTER COMMAND under three conditions.
a. Directly after GETing the lesson	
b. Switching from Lesson Building (#)	
c. Switching from Execution (#)	
d. You can edit after ENTER COMMAND appears	
e. Edited material is saved automatically	
3. HOW TO REPLACE MATERIAL (Edit)	3.0 Replaces a line in error with a new line in response to simulated PLANIT messages.
a. Replace means substitute new for old	
b. Replace a line	
(1) Example and explanation (2) Specify frame, group, line (e.g., #2, 3, 1, E) (3) PLANIT prints old line (4) Type and enter your new line	

Subject Breakdown	General Task/Objective
c. Replace a Group (1) Example and explanation (2) Specify frame and group (e.g., #2, 3, E) (3) Group message (e.g., G3. ANSWERS) appears (4) Type and enter all lines for group	3.1 Identifies the correct command-message sequence for replacing a group.
d. Replace a Frame (1) Example and explanation (2) Specify frame (e.g., #2, E) (3) Selection message Q/M/D/P appears (4) Select frame type (5) Check Group 1 identifier (6) Select and enter all groups needed	3.2 Identifies the correct command-message sequence for replacing a frame.
4. HOW TO INSERT MATERIAL (Insert)	4.0 Inserts a line between two existing lines in a group in response to simulated PLANIT messages.
a. Insert means put new material between existing material	
b. Insert a line between lines (1) Example and explanation (2) Specify frame, group, and position of new line (3) Line must exist at referenced position (4) Line referenced is moved down in sequence (5) Type and enter new line	
c. Insert a Group between groups (1) Example and explanation (2) Specify frame and new group (e.g., 11, 3, I) (3) Referenced group must not exist	4.1 Identifies the correct command-message sequence for inserting a group.

Subject Breakdown	General Task/Objective
(4) Group message (e.g., G3.ANSWERS appears (5) Type and enter all lines for group	
d. Insert a Frame between frames (1) Example and explanation (2) Specify new frame (e.g., #11.5, I) (3) Referenced frame must not exist (4) Selection message Q/M/D/P appears (5) Select frame type (6) Check Group 1 identifier message (7) Select and enter all groups needed	4.2 Identifies the correct command-message sequence for inserting a frame.
5. HOW TO ADD-ON TO MATERIAL (Append)	5.1 Identifies the correct command-message sequence for adding a frame to a lesson.
a. Add the next Frame to a Lesson (1) GET the lesson (2) Enter #A (3) Q/M/D/P appears (4) Select frame type (5) Check Group 1 identifier (6) Select and enter all groups needed	
b. Add the next Group to a Frame (1) Give command for frame (e.g., #5,A) (2) Group message (e.g., G4.ACTIONS) appears (3) Type and enter all lines for the group	5.2 Identifies the correct command-message sequence for adding a group to an existing frame.

Subject Breakdown	General Task/Objective
c. Add the next Line to a Group (1) Example and explanation (2) Specify the frame and group (e.g., #5, 3, A) (3) Wait for the * to appear (4) Add line or lines after existing lines	5.3 Adds a line to a group in response to simulated PLANIT messages.
6. HOW TO DELETE MATERIAL (Delete)	6.0 Deletes a line, a set of lines, and a frame in response to consecutive instructions and simulated PLANIT messages.
a. Delete means destroy	
b. Specify frame, group, or line (1) A frame (e.g., #2, D) (2) A group (e.g., #2, 3, D) (3) A line (e.g., #2, 3, 1, D) (4) A set of lines (e.g., #2, 3, 1-5, D) (5) * shows deletion is done (6) Use P command to verify deletion	6.1 Uses P command to verify each simulated deletion.

Subject Breakdown	General Task/Objective
V. LESSON CONSTRUCTION, PART 2	
1. INTRODUCTION	1.0 Student selects his start point in this unit (or in Unit VI) from a list:
a. Unit learning objectives	. Questions requiring more than one answer entry.
b. Unit adds more Q and D frame capabilities	. More on action commands.
c. Unit adds more on PLANIT commands	. More on use of the D-frame.
d. Unit gives document references for more information	. Basic capabilities of the CALCulation mode and the (P)rogramming frame.
	. How to get a lesson listing or a new punched card deck.
	. How to display a student's lesson record.
	. All of these in sequence.
2. DETECTING TWO OR MORE KEY WORDS (Q, GROUP 3)	2.0 Selects a correct Q frame Group 3 construction to detect two key words only if they are given in order.
a. Two or more key words: Order important	
(1) KEYWORD ON	
(2) Example and explanation	
b. Two or more key words: Order unimportant	2.1 Selects a correct Q frame Group 3 construction to detect two key words given in any order.
(1) KEYWORD ALL	
(2) Example and explanation	
c. Any number of a set of key words: Order not important	2.2 Selects a correct Q frame Group 3 construction to detect any two of three possible correct answers.
(1) KEYWORD number	
(2) Example and explanation	
d. Problems in using KEYWORD	
(1) Undesired answers may match	
(2) Example where desired and undesired answers would match	
(3) Line length limits word entries	
(4) Requires student typing skills	
(5) Reference to more information	

Subject Breakdown	General Task/Objective
3. PROCEDURE FOR QUESTIONS REQUIRING SEVERAL ANSWERS	3.0 Orders a list of frame requirements for accepting a list of steps one entry at a time.
a. Types of questions (1) Matching letters-numbers from 2 lists (2) Listing parts or steps in a given order	3.1 Answers questions about the purpose and operation of frame entries for the frames sequenced in 3.0.
b. Accepting answer words (or tags) an entry at a time (1) Requires instructions to students (2) Uses one Q frame per required entry (3) Uses one D frame after the set of Q frames (4) Example and explanation of a frame sequence (three Q frames, one D frame)	
4. MORE ON GROUP 4 ACTIONS (Q/M, GROUP 4)	4.0 Identifies the correct Group 4 entries format for certain unconditional action requirements.
a. Branch to another lesson (1) General form: B:lessoname (2) Example: B:LESSON3	4.1 Identifies branch to lesson entries which do/do not conform to the general input form and naming rules.
b. Unconditional actions (1) Actions F:, C:, or B: with no tags (2) Place at top of Group 4 before the tagged actions. (3) Example and explanation of a Group 4 with both unconditional and conditional (tagged) actions.	
c. Multiple action lines for the same tags (1) Group 3/Group 4 example of two A-tagged action lines and two unanticipated answer action lines. (2) Explanation of normal order of execution of like-tagged actions (without the use of RELATED)	4.2 Selects statements which describe the normal order of execution of like-tagged actions in a Group 4.

Subject Breakdown	General Task/Objective
5. MORE CAPABILITIES OF THE D-FRAME	
a. Unconditional actions taken before a decision statement (1) Legal actions: F:, C:, or B: (2) The entire D-frame could be used only for unconditional actions. (3) Example and explanation	5.1 Identifies the correct D-frame entries for specified actions to be taken unconditionally before a conditional decision.
b. Compound conditions (AND/OR) (1) Meaning of AND -- example and explanation (2) Meaning of OR -- example and explanation (3) General form: IF / AND / OR } conditions...actions (4) AND/OR can connect conditions of pattern, summary, or computational forms.	5.2 Chooses AND/OR as the appropriate connector to use for given decision requirements.
c. Using FROM to control search for conditions (1) Purpose of FROM (2) Example and explanation--PLANIT's search of student records for a decision condition specified with and without the use of FROM. (3) General form: IF FROM frame, ...etc. (4) FROM applies only to search for conditions of pattern or summary form; i.e., searches against PLANIT records.	5.3 Selects decision statements with and without FROM to control search of PLANIT student records as specified. 5.4 Completes a decision statement by providing the FROM entry so as to control search of records as specified.

Subject Breakdown	General Task/Objective
6. MORE ON PLANIT COMMANDS	
a. How to DISPLAY a student's lesson record	6.1 Enters the correct GET, ATTACH, and DISPLAY command sequence given a requirement to display summary and history data for a fictitious student.

(1) Reasons: To check a student's progress, to check how your lesson is operating, etc.
(2) General form: DISPLAY ... entire record
DISPLAY frame number(s)
(3) Sequence: GET lesson, ATTACH record, DISPLAY
(4) GET the lesson using the author's identification--if you are not already using it as the author.
(5) ENTER COMMAND appears
(6) Enter ATTACH studentID... to get a record
(e.g., ATTACH HOYTWG)
(7) DONE appears
(8) Enter DISPLAY command (e.g., DISPLAY)
(9) SUMMARY ONLY (Y/N)? appears
(10) Enter Y ...Y for summary data only
(11) PLANIT displays the record summary data
(12) DONE appears
(13) Enter next DISPLAY command (e.g., DISP 5-25)
(14) SUMMARY ONLY (Y/N)? appears
(15) Enter N ...for detail data frames 5-25
(16) PLANIT displays detail execution history
(17) Sign OUT or use # to another activity

b. How to LIST and PUNCH your lesson after editing	6.2 Enters the command sequence to obtain a listing and punch a card deck for a given lesson.

(1) To get a listing for review
(2) To make or update a backup card deck
(3) ENTER COMMAND appears
(4) Enter SYSTEM to use printer or punch
(5) ENTER SYSTEM COMMAND appears
(6) Enter LIST lessoname ...for a listing
(e.g. LIST SQUAD2)
(7) DONE appears
(8) Enter PUNCH lessoname ...for punched cards
(e.g. PUNCH SQUAD2)
(9) DONE appears
(10) Sign OUT or # to another activity

Subject Breakdown	General Task/Objective
VI. BASICS OF CALC AND THE (P)ROGRAMMING FRAME	
1. CALC OVERVIEW	1.0 Identifies the major functional subdivisions of the CALC language.
a. Unit learning objectives	
b. Functional components of CALC	
(1) For computational work	
(2) For instructional programming	
2. HOW TO ENTER AND EXIT CALC	2.1 Identifies CALC as the command word for any user to enter CALC after Login.
a. Any user after Login	
(1) ENTER COMMAND appears	
(2) Enter the word CALC	
(3) ENTER CALC STATEMENT appears	
(4) Use CALC as needed	
(5) Sign OUT or # to another activity when done with CALC	
b. Author from any PLANIT activity	2.2 Identifies #CALC as the command for an author to switch to CALC from any other PLANIT activity.
(1) * appears	
(2) Enter #CALC	
(3) ENTER CALC STATEMENT appears	
(4) Use CALC as needed	2.2.1 Identifies # as the control symbol for an author to exit CALC to another PLANIT activity.
(5) # to another activity when done with CALC	
c. Student from a lesson (or author trying out lesson as a student)	2.3 Identifies ↑ as the control symbol for a student (or an author using a lesson as a student) to enter CALC from the lesson.
(1) * appears for a question requiring the use of CALC	
(2) Type and enter ↑ alone -- or with a CALC statement	
(3) ENTER CALC STATEMENT appears. If CALC NOT AVAILABLE appears, you must do the computation without CALC's help.	2.3.1 Identifies ◄ or READY as how a student (or an author using a lesson as a student) switches from CALC back to the lesson.
(4) Use CALC to get an answer	
(5) Type and enter ◄ or READY when you have your answer (◄ can include the answer)	
(6) ENTER YOUR ANSWER appears	
(7) Type and enter numeric answer	
(8) Continue with the lesson	

Subject Breakdown	General Task/Objective
d. Author's use in lesson construction (1) C:(calc statement) -- in Q/M frames, Group 4 or in D/P frames, Group 2. (2) O (calc statement in Q Group 3 (3) tags 1-9 (answer) in Q Group 3	2.4 Identifies C:, O, and tags 1-9 as control symbols for using CALC through lesson frames to give lesson control instructions.
3. BASIC ARITHMETIC OPERATIONS	
a. General form: operand operator operand...etc. e.g. 5 + 2 ...where the operands are constants (non-varying numbers) or variables (names which give a number)	3.1 Identifies the meaning of CALC arithmetic operators.
b. Arithmetic operators + add - subtract * multiply / divide ** raise to a power	3.2 Identifies arithmetic expressions which do/do not conform to the general input form. 3.3 Writes simple arithmetic expressions with constants as operands. 3.4 Evaluates arithmetic expressions containing constants connected by two or three operators. e.g., 10**2/25-1 = ? 25-1/2**2 = ?
Examples (with constants): ENTER CALC STATEMENT *8+2 10.0 *8*2 ... 8 X 2 16.0 *8/2 ... $2\overline{)8}$ 4.0 *8**2 ... 8^2 64.0	

Subject Breakdown	General Task/Objective
c. Order of operations	3.5 Evaluates arithmetic expressions containing constants and two operators with one or two sets of grouping symbols; e.g., (8+2)/2 = ? 8+(2**(4-2)) = ?
(1) Moving left-to-right along a line (2) Operations enclosed by grouping symbols ( ) done first (3) ** done second (4) * or / done third (5) + or - done last (6) Exception: if - or + is taken as unary plus or minus, this is done first. (7) Nested grouping symbols (( )) are expanded from innermost to outermost (8) Examples (using constants)	3.6 Evaluates arithmetic expressions equivalent in all ways except the use or non-use of grouping symbols, e.g., 1*3**2 versus (1*3)**2 e.g., 25-5*2+1 versus (25-5)*2+1 e.g., 10/2+3 versus 10/(2+3)

```
ENTER CALC STATEMENT
*6*4/2
8.0              ...6+2 (division takes precedence)
*(6+4)/2         ...10/2 (addition takes precedence)
5.0
*(6+(4**2)/2)    ...4**2=16, 16/2=8, 6+8=14
14.0
```

4. SAVE A NUMBER BY NAME: ASSIGNMENT STATEMENT (=)		4.1 Identifies names which are legal/illegal.
(1) General form: name=constant name=expression ...yielding a number (2) Examples of legal and illegal names (3) Examples of assignment statements		4.2 Writes simple assignment statements to store, increment, and decrement named numbers.

```
                ENTER CALC STATEMENT
C:A=2           *A=2
                A=2.0    ...2.0  saved at location A
C:B=A+10        *B=A+10
                B=12.0   ...12.0 saved at location B
C:A=A+4         *A=A+4   ...increment current value
                A=6.0        of A by 4 (2+4)
```

4.3 States the value at a named location after a sequence of assignment statements.

Subject Breakdown	General Task/Objective
5. ARITHMETIC WITH NAMED NUMBERS	
a. Same order of operations and general input form as with constants; i.e., operand operator operand ... etc. X * Y	5.1 Writes simple expressions with constants and variables as operands.
b. One or more of the operands are named variables which will yield a variable number.	5.2 Evaluates the expressions in 5.1 given values for the variables.
c. All variables in the arithmetic expression must have values assigned before the expression is evaluated by CALC, otherwise PLANIT will give message UNDEFINED VARIABLE	
d. Interactive example:	

```
*A=2
A=2.0
*B=A+10          ...give values to
B=12.0              variables A and B
*A+B             ...add two variables
14.0
*X=A+B-5         ...add variables and
X=9.0               subtract a constant
*T=X*3**2        ...9x3^2=9x9=81
T=81.0
*SQRT(T)         ...use PLANIT's built-in
9.0                 function SQRT
```

6. SAVE A SET OF NUMBERS BY NAME (MATRIX)	
a. Reasons (1) To perform numeric computation on the set of numbers (e.g., find the average, the range, the sum, etc.) (2) To perform logical operations on the set of numbers (e.g., sort them, rank-order them, reformat them, etc.)	6.1 Identifies examples of interactive calculation and lesson data requirements which would or would not require the use of a matrix.

Subject Breakdown	General Task/Objective
b. Two steps to store a set of numbers--define a storage table by name and then enter data into the table.	6.2 Identifies sample matrix definitions as correct/incorrect with respect to the general input form.
(1) Define a storage table:	
General form: MATRIX (name, rows, column) e.g., C:MATRIX(DATA,3,2)	6.3 Matches storage table name and size requirements with MATRIX definition statements.
...defines a table named DATA with 3 rows and 2 columns (6 cells).	
e.g., C:MATRIX(NUM,10)	
...defines a table named NUM with 10 rows and one column (10 cells).	
(2) Enter data into the table	6.4 Identifies examples of calculation and instructional requirements which would require entry of numbers into a matrix.
• General form: Fixed entries	
matrixname(cell)=ARRAY(number, number..etc.)	
e.g., DATA(1,1)=ARRAY(2,4,6,8,10,12)	
...where(1,1) specifies that the first entry 2.0 goes in row 1 column 1 of the table DATA size 3 by 2 defined above.	6.5 Identifies samples of entering data into a defined matrix which are/are not correct with respect to the general input form.

	Col. 1	Col. 2
1	2.0	4.0
row 2	6.0	8.0
3	10.0	12.0

• General form: Generated entries

matrixname(cell)=expression

...where the expression yields a number

e.g., NUM(1)=RANDOM*100

NUM(2)=RANDOM*100

.

.

etc.

...where each entry randomly generates a number from 0.0 to 0.9999, multiplies it by 100, and puts the result in a cell of table NUM defined above until the 10 cells are filled.

Subject Breakdown	General Task/Objective

7. DISPLAYING CALC DATA (PRINT, ALIGN, ROUND)

a. Purpose of PRINT and ALIGN

(1) Used to display variable data stored by name; e.g., the results of a computation.
(2) Can be used to display a mixture of fixed and variable information.
(3) No need to use if all information is fixed (then F: would be sufficient).

7.1 Identifies display requirements which do/do not require the use of PRINT or ALIGN.

7.2 Identifies examples of PRINT and ALIGN statements which do/do not conform to the general input form.

b. General form for PRINT and ALIGN

```
PRINT  name                              name
ALIGN  expression  ,column;  expression...etc.
       'literal'             'literal'
```

The only difference is that with ALIGN you specify the column where the material is to be displayed.

7.3 Selects the display that would result from each of two PRINT and two ALIGN statements.

c. Use of ROUND

(1) General form: ROUND(number) ...where (number) is the number of decimal places to be retained for the display. (Only TRUNCATE will change precision of a stored number.)
(2) Review rules for rounding numbers.

7.4 Selects ROUND as the control word that changes the precision of a number during display and selects TRUNCATE as the control word that changes the precision of a number both for computation and display.

d. Example of PRINT, ALIGN, AND ROUND

```
ENTER CALC STATEMENT
*N=25.738           ...name a decimal number N
N=25.7380
*PRINT N ROUND(2)   ...display N rounded to 2 places
N=25.74
*TRUNCATE(N)        ...remove non-zero decimal digits
N=25.0                 from the number at location N
*ALIGN N,8
     25.0           ...display N with decimal point in column 8
*PRINT 'THE VALUE OF N IS ';N
THE VALUE OF N IS 25.0      ...mix literal and variable for display
*PRINT 'N = ';N ROUND(0)    ...ROUND(0) removes
N = 25                      decimal point and
                            digit for the display
```

Subject Breakdown	General Task/Objective

8: THE (P)ROGRAMMING FRAME

a. Purpose--to define a procedure for computation or for computation and display which requires more than one line of PLANIT entries.

8.1 Identifies examples of instructional requirements which would/would not require the use of a P-frame.

(1) Can be useful when the same procedure is to be used repeatedly within a lesson and PLANIT does not provide a built-in capability for the procedure.

- Examples of built-in procedures provided by PLANIT: *square root*(SQRT), randomly generate a number(RANDOM), truncate a number(TRUNCATE), absolute value function(ABS), automatic lesson review capability (REVIEW).

- Examples of procedures <u>not</u> provided for by PLANIT which could be specified by an author: automatic scoring, computing statistics from scores or grades, sorting numbers into various orders, generating problems or simulations in electronics or the geometry of tactical fire.

(2) A student is not aware of any direct interaction with a P-frame, as is also the case in a D-frame. However, the procedure in a P-frame may operate on data provided by the student in previous frames and may display the results of a computation or logical operation to the student.

b. Legal P-Frame Line Entries (G2. STATEMENTS)

8.2 Identifies exmples of PLANIT line entries as legal/illegal for a P-frame.

(1) Any CALC statement (with or without the C: prefix)
(2) Any form of decision statement (IF,ELSE)
(3) Action commands F:feedback
C:CALC statement
B:line,frame,lesson
(4) RETURN to lesson frame that branched to the P-frame

Subject Breakdown	General Task/Objective
c. Capabilities unique to P-Frame--which are different than the D-Frame. (1) Decision statements can control line-to-line branching within the P-frame. (2) RETURN can serve as an unconditional branch back to the next PLANIT frame (or next instruction in a frame) after the one that branched to the P-frame.	8.3 Identifies "line-to-line branching" and "automatic return to another frame" as the two capabilities that distinguish a P-frame from a D-Frame.
d. Basic Functions of a P-Frame Procedure (1) Store data--e.g., numbers prestored in the frame, numbers generated by the frame, numbers obtained from a user. (2) Perform operations on data--e.g., mathematics, statistics, scoring, or logical operations such as sorting and formatting. (3) Make decisions--e.g., select the next operation, select a display format, etc., and branch or update records based on outcomes of the decision. (4) Display the outcomes of the procedure--e.g., display data stored by the procedure, or display any keyboard character according to instructions given by the procedure (e.g., a graph of a sine wave).	8.4 Matches statements of P-frame functions with the 4 basic types of functions.
e. Example of Defining a Procedure (1) Requirement: You need a procedure to repeatedly generate a column of 3 decimal numbers which students will be instructed to add. (2) Analysis: You can use CALC function RANDOM to cause PLANIT to generate a number from 0.0 to 0.9999 each time it is used. Multiplying these numbers by 100 will give you decimal numbers ranging from 0.0 to 99.99. PRINT can be used to display the numbers and ROUND(2) to insure that the decimal numbers do not exceed two place.	8.5 Identifies in a sample P-frame the statements containing a line label, a branch to the line label, and a return to the main lesson flow.

Subject Breakdown

(3) Steps in the procedure and example of P-frame entries

Step 1: Give student instructions	F:ADD THE NEXT 3 NUMBERS
Step 2: Define storage area for three numbers.	C:MATRIX(NUM,3) C:CELL=1 C:PASS=1
Step 3: Set counters to keep track of where the generated number is to be stored (CELL) and how many numbers have been generated (PASS).	LN4:NUM(CELL)=RANDOM*100 C:CELL=CELL+1 IF PASS LQ 2 C:PASS=PASS+1 B:LN4 ELSE C:ROUND(2)
Step 4: Generate a number from 0.0 to 99.999 and store it. Label this line (LN4) so you can use it again.	
Step 5: If 3 numbers have been generated and stored, go to Step 6. If not, increment CELL and PASS and go back to Step 4 (LN4).	
Step 6. Display the numbers vertically.	PRINT NUM(1) PRINT NUM(2) PRINT NUM(3)
Step 7: Return to the next frame after the instruction that branched to this P-frame (should be a Q-frame to accept an answer for this example).	RETURN

APPENDIX B

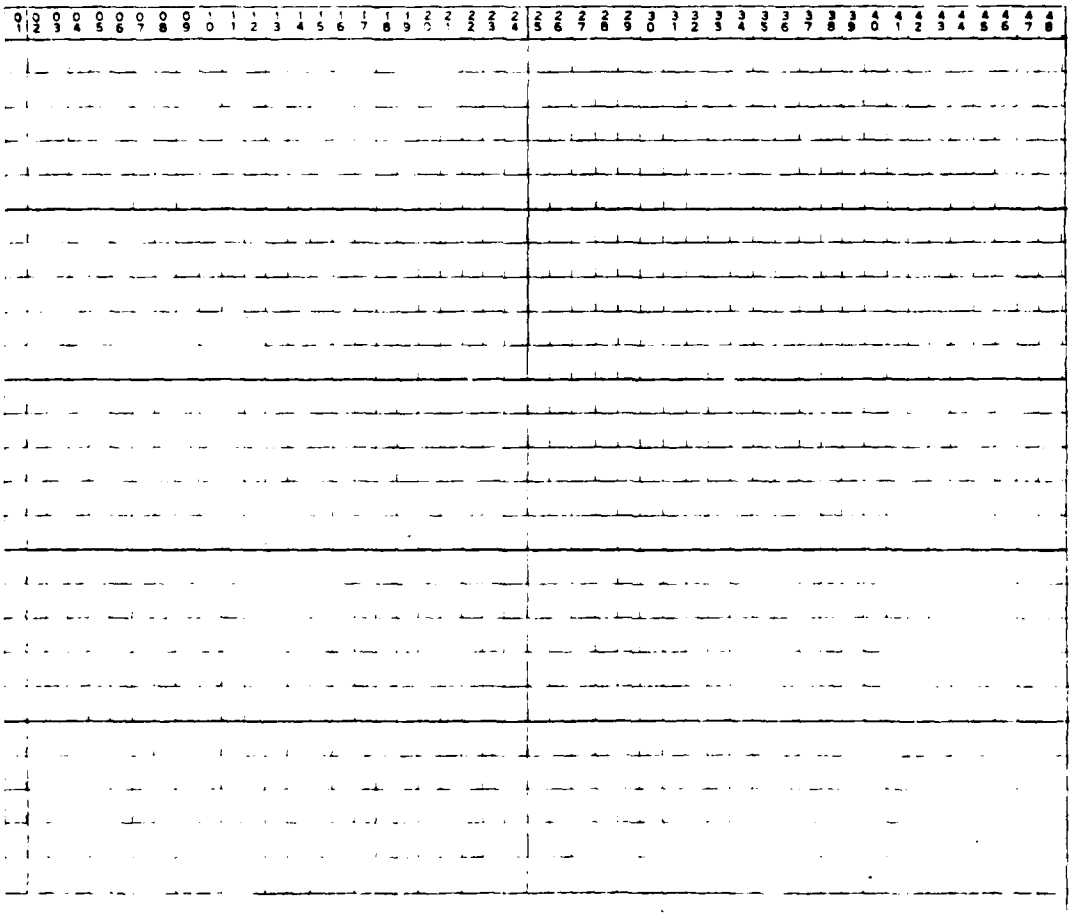
PLANIT COURSE ADJUNCT MATERIALS

PLANIT FRAME WORKSHEET

Name _____ Lesson _____ Page _____ of _____

SS Number _____ Phone _____ Date _____

Organization _____ Date/Time: Entered: _____ Checked_____



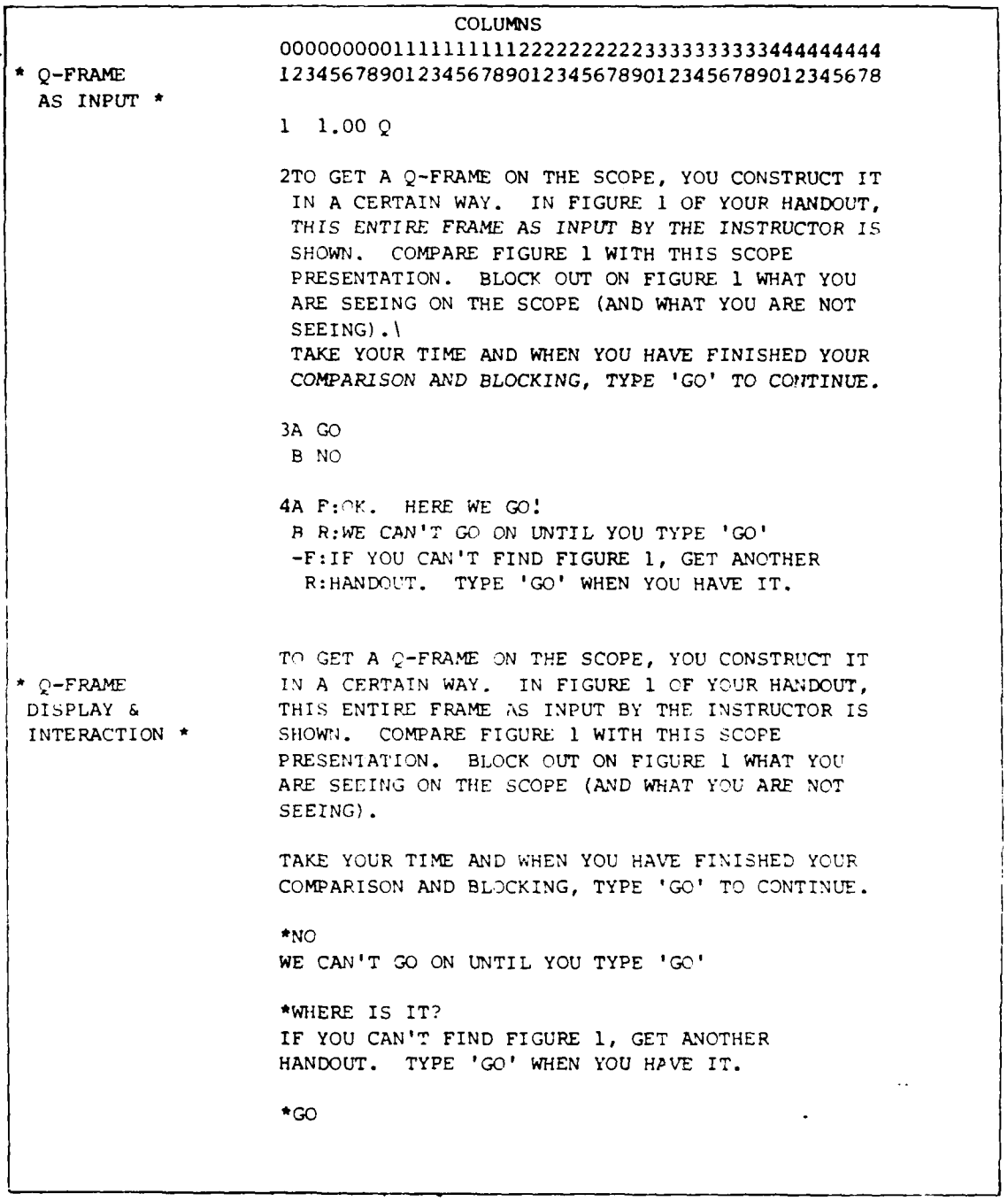


Figure 1. Example of Q-Frame Input and Interaction

```
                                    COLUMNS

                  000000000011111111112222222222333333333344444444
* M-FRAME         123456789012345678901234567890123456789012345678
  AS INPUT *
                  1  2.00 M  TEST

                  2WHAT TYPE OF FRAME IS THIS?
                   (CHOOSE A LETTER)

                  3  A.   QUESTION
                     B.+  MULTIPLE CHOICE
                     C.   DECISION
                     D.   PROGRAMMING

                  4ACD F:WRONG.  HINT- SEVERAL ALTERNATIVES
                   R:ARE DISPLAYED.  TRY AGAIN.
                   B F:YES, A (M)ULTIPLE CHOICE FRAME.




* M-FRAME         WHAT TYPE OF FRAME IS THIS?
 DISPLAY &       (CHOOSE A LETTER)
 INTERACTION
                    A.   QUESTION
                    B.   MULTIPLE CHOICE
                    C.   DECISION
                    D.   PROGRAMMING

                  *QUESTION
                  CHOOSE ONE OF THE ABOVE LETTERS
                  *A
                  WRONG.  HINT- SEVERAL ALTERNATIVES
                  ARE DISPLAYED.  TRY AGAIN.
                  *D
                  WRONG.  HINT- SEVERAL ALTERNATIVES
                  ARE DISPLAYED.  TRY AGAIN.
                  *B
                  YES, A (M)ULTIPLE CHOICE FRAME.
```

Figure 2. Example of M-Frame Input and Interaction

- Input Example -

```
                    COLUMNS


          000000000111111111122222222223333333333444444444
          123456789012345678901234567890123456789012345678

          1  4·ØØ D  BIGD

          21F ALL RIGHT 1-3
           F:GOT THEM ALL! C:SWIFT=1
           ELSE
           F:PRACTICE MAKES PERFECT. B:DRILL
```

- General D-Frame Form -

```
          IF CONDITION1              (IF 1ST CONDITION IS TRUE)
(THEN DO) ACTION1 ACTION2 ...ETC.
          IF CONDITION2              (IF 2ND CONDITION IS TRUE)
(THEN DO) ACTION3 ACTION4 ...ETC.
            .
            .
            .
          ELSE                       (OTHERWISE)
     (DO) ACTION5 ACTION6 ...ETC.
```

Figure 3. Example of D-Frame Input and General Form

1. WRITE YOUR OWN PLANIT LESSON OF 12 TO 20 FRAMES ON A TOPIC WHICH YOU KNOW (E.G., SOME PART OF YOUR JOB) WHICH WOULD BE USED TO TEACH THE TOPIC.
   ...PREPARING THE LAW FOR FIRING
   ...RANGE SAFETY PRECAUTIONS
   ...IMMEDIATE ACTION ON THE M60 MACHINE GUN
   ...CLASSES OF FIRE

2. YOUR PLANIT LESSON OF 12 TO 20 PLANIT FRAMES SHOULD CONTAIN AT LEAST THE FOLLOWING NUMBERS AND TYPES OF FRAMES (USE ANY TYPE ADDITIONAL FRAMES TO FILL OUT YOUR LESSON):
   3 Q-FRAMES 3 M-FRAMES 3 D-FRAMES

   A. INSTRUCTIONS FOR ALL Q-FRAMES:
      . THE PRESENTATION (INFORMATION & QUESTION) MUST FIT IN 17 LINES AND 48 COL'S.
      . FOR EACH CORRECT ANSWER, GIVE A CORRECT ANSWER MESSAGE (E.G., GOOD).
      . EACH FRAME SHOULD CONTAIN A GROUP 1, 2, 3, & 4.
      . USE A COUNTER TO RECORD THE TOTAL NUMBER OF RIGHT RESPONSES A STUDENT HAS MADE AFTER EACH Q-FRAME TRIED.

      FOR THE 3 Q-FRAMES--ONE SHOULD REQUIRE A SINGLE-WORD ANSWER TO A QUESTION.

      ONE SHOULD LOOK FOR A SINGLE KEY WORD AS THE CORRECT ANSWER AMONG SEVERAL STUDENT RESPONSE WORDS.

      ONE SHOULD ACCEPT AS THE CORRECT ANSWER EITHER A SINGLE WORD FOR A NUMBER OR A NUMBER. (E.G., HOW MANY..?)

      ALSO IN THE ABOVE-- WHEREVER POSSIBLE, ALLOW MISSPELLINGS TO BE COUNTED AS CORRECT.

      LOOK FOR AT LEAST ONE BUT NO MORE THAN 4 WRONG ANSWERS TO A QUESTION. FOR EACH WRONG ANSWER, GIVE A MESSAGE SAYING THE ANSWER WAS WRONG AND WHY, THEN ACCEPT ANOTHER ANSWER TO THE QUESTION.

      EACH Q-FRAME SHOULD PROVIDE A MESSAGE FOR 'UNANTICIPATED ANSWERS' WHICH GIVES THE CORRECT ANSWER & THEN GOES TO THE NEXT FRAME.

   B. INSTRUCTIONS FOR THE 3 M-FRAMES
      . EACH M-FRAME SHOULD HAVE 4 ANSWER CHOICES (ALTERNATIVES).
      . THE P[illegible]TATION (INFORMATION, QUESTION STEM, AND CHOICES) MUST FIT IN 17 [illegible] 48 COLUMNS.
      . EAC[illegible]OULD CONTAIN A GROUP 1, 2, 3, & 4.
      . FO[illegible]ECT ANSWER CHOICE, GIVE A HELPFUL MESSAGE AND ASK FOR ANOT[illegible] TO THE SAME QUESTION.
      . CAUS[illegible] TO RECORD THE TOTAL NUMBER OF RIGHT RESPONSES MADE.
      . USE AT LEA[illegible] ONE BRANCH AHEAD PAST A FRAME FOR A CORRECT ANSWER, AND ONE BRANCH BACK TO AN EARLIER FRAME FOR AN INCORRECT ANSWER.

Figure 4. Requirements for Your PLANIT Lesson (Page 1 of 2)

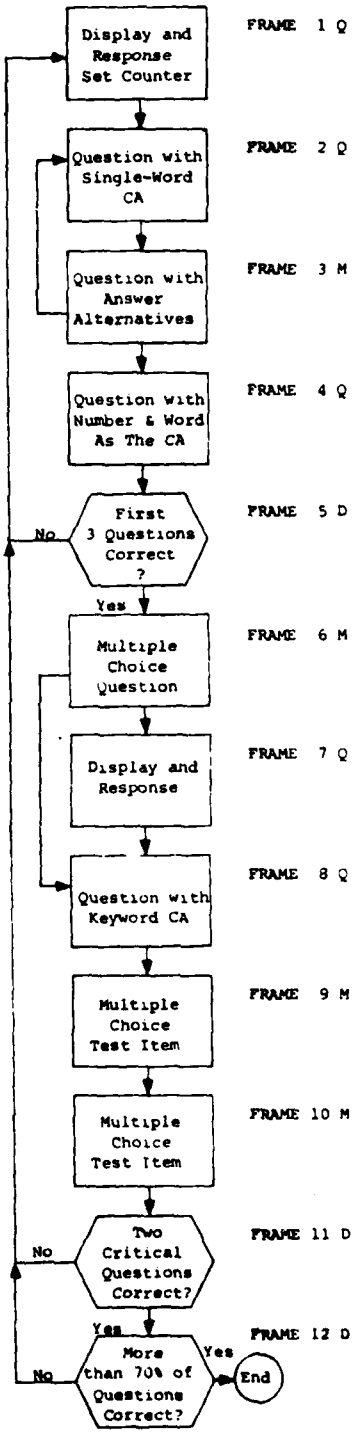
C. INSTRUCTIONS FOR D-FRAMES:

FOR THE 3 D-FRAMES--ONE SHOULD TEST IF THE FIRST 3 QUESTIONS IN THE LESSON WERE ALL ANSWERED CORRECTLY. IF SO, TELL YOUR STUDENT HE IS DOING WELL. IF NOT, TELL THE STUDENT REVIEW IS NEEDED AND GO BACK TO THE START.

THE ONE JUST BEFORE THE LAST FRAME, SHOULD TEST TO SEE IF THE 2 MOST CRITICAL QUESTIONS WERE ANSWERED CORRECTLY. IF SO, TELL THE STUDENT HE GOT THE MOST IMPORTANT ONES. IF NOT, TELL HIM HE MISSED THE MOST IMPORTANT QUESTIONS AND BRANCH TO LESSON START.

THE LAST FRAME SHOULD TEST IF THE STUDENT'S TOTAL RIGHT SCORE IS MORE THAN 70% OF THE QUESTIONS ASKED OF HIM...FOR EXAMPLE, MORE THAN 2 OUT OF 4 OR MORE THAN 4 OUT OF 7. IF IT IS, TELL THE STUDENT HE DID WELL, THANK HIM FOR HIS TIME, AND TELL HIM HE'S FINISHED. OTHERWISE, TELL HIM HE MISSED TOO MANY AND LOOP BACK TO THE START.

Figure 4. Requirements for Your PLANIT Lesson (Page 2 of 2)



- THIS FIGURE SHOWS ONE ORDER OF Q & M FRAMES THAT WILL SATISFY THE REQUIREMENTS OF FIGURE 4. YOU MAY USE ANY ORDER OF Q & M FRAMES FOR YOUR TOPIC AS LONG AS IT SATISFIES THE REQUIREMENT.
- NOTE THE POSITION OF THE THREE REQUIRED D-FRAMES IN TERMS OF THE FIGURE 4 REQUIREMENTS. THE DECISION CONDITIONS AND BRANCHING LINES ARE SHOWN.
- THIS EXAMPLE ASSUMES A 12-FRAME LESSON WITH THREE EXTRA Q-FRAMES.
- THE "CA" STANDS FOR THE CORRECT ANSWER.

Figure 5. Lesson Decision Logic Flow

THESE ARE THE STEPS TO FOLLOW AFTER YOUR LESSON UNITS ON PLANIT FRAME TYPES Q, M, AND D. (AS YOU PERFORM THESE STEPS YOU MAY WANT TO REFER TO THE "SUMMARY OF PLANIT FRAMES".

STEP 1: CHOOSE A LESSON TOPIC. DECIDE ON A LEARNING OBJECTIVE FOR THE TOPIC AND WRITE SEVERAL TEST ITEMS IN A SEQUENCE.

STEP 2: READ LESSON REQUIREMENTS & DECISION LOGIC FLOW (FIGURES 4 & 5).

STEP 3: WRITE 9-18 PLANIT Q & M FRAMES ON WORKSHEETS TO COVER YOUR TOPIC TEACHING POINTS AND TEST ITEMS. CHECK TO INSURE THAT Q-FRAMES AND M-FRAMES PERFORM THE FUNCTIONS STATED IN FIGURE 4. CHECK YOUR DISPLAYS FOR SCREEN SIZE.

STEP 4: STUDENT PERFORMANCE STANDARDS ARE ALREADY GIVEN IN THE FIGURE 4 D-FRAME REQUIREMENTS. WRITE 3 PLANIT D-FRAMES TO SATISFY THE DECISION CONDITIONS AND BRANCHING SHOWN IN FIGURES 4 & 5. SEQUENCE THESE FRAMES AS SPECIFIED.

STEP 5: CHECK WORKSHEET FRAME NUMBERS (NO DUPLICATES), FRAME TYPES, GROUP NUMBERS (COLUMN 1) AND THE FRAME NUMBERS BRANCHED TO WITH EACH B: ACTION.

STEP 6: WHEN YOU HAVE COMPLETED YOUR WORKSHEETS, COME TO THE COMPUTER TERMINAL, LOG-IN WITH YOUR I.D. AND, WHEN "ENTER COMMAND" APPEARS, TYPE:

GET PLANIT2

YOU WILL RECEIVE INSTRUCTION ON HOW TO ENTER YOUR LESSON FRAMES TO THE COMPUTER, HOW TO RUN THEM AS A STUDENT WOULD SEE THEM, AND HOW TO RECOGNIZE AND EDIT LESSON ERRORS.

TYPE < FINISHED AFTER THESE LESSONS.

STEP 7: LOG-IN WITH YOUR I.D. AND ENTER THE COMMAND TO BEGIN AUTHORING YOUR LESSON. ENTER YOUR 12-20 FRAMES IN SEQUENCE FROM YOUR WORK-SHEET. WATCH FOR TYPING ERRORS ON THE SCREEN BEFORE YOU ENTER EACH LINE.

STEP 8: TRY OUT YOUR LESSON AS A STUDENT WOULD SEE IT, GIVING THE CORRECT ANSWER TO EACH QUESTION. IF YOU GET A PLANIT ERROR MESSAGE, TRY TO DETERMINE THE CAUSE, EDIT THE LINE IN ERROR, AND THEN TRY THE FRAME AGAIN. DO THIS UNTIL YOUR LESSON RUNS WITHOUT ANY ERROR MESSAGES.

TELL A MONITOR WHEN YOU ARE DONE, OR WORK ON IT UNTIL THE DAY'S SESSION IS OVER (WHICHEVER COMES FIRST).

Figure 6. Steps for the Lesson Exercise

1. (Q)UESTION FRAME

ENTER GROUP NUMBER (FIRST LINE OF GROUP, COL. 1)

GROUP 1 IDENTIFIER (FRAME NUMBER, FRAME TYPE, OPTIONAL LABEL)

GROUP 2 TEXT (INFORMATION AND/OR QUESTION)

GROUP 3 ANSWERS (ANTICIPATED ANSWERS NOT DISPLAYED)

GROUP 4 ACTIONS (F: C: R: B:Frame B:Lesson)

2. (M)ULTIPLE CHOICE FRAME

ENTER GROUP NUMBER (FIRST LINE OF GROUP, COL. 1)

GROUP 1 IDENTIFIER (FRAME NUMBER, FRAME TYPE, OPTIONAL LABEL)

GROUP 2 TEXT (INFORMATION AND/OR STEM OF A QUESTION)

GROUP 3 ANSWERS (LISTED ANSWERS PART OF DISPLAY)

GROUP 4 ACTIONS (F: C: R: B:Frame B:Lesson)

3. (D)ECISION FRAME

ENTER GROUP NUMBER (FIRST LINE OF GROUP, COL. 1)

GROUP 1 IDENTIFIER (FRAME NUMBER, FRAME TYPE, OPTIONAL LABEL)

GROUP 2 CRITERIA (CONDITIONS OF COMPUTATIONAL, SUMMARY, OR DATUM
FORM WITH ACTIONS F:Message C:Calcstatement
B:Frame B:Lesson)

4. (P)ROGRAMMING FRAME

ENTER GROUP NUMBER (FIRST LINE OF GROUP, COL. 1)

GROUP 1 IDENTIFIER (FRAME NUMBER, FRAME TYPE, OPTIONAL LABEL)

GROUP 2 STATEMENTS (PROCEDURE WITH CONDITIONS OF ANY FORM AND
ACTIONS C:Calcstatement F:Message OR
B:Linelabel ONLY. LINE LABELS AND SUBROUTINE
'RETURN' ACTION ARE OPTIONAL)

Exhibit 1. PLANIT FRAMES AND FRAME GROUPS

GROUP 1 IDENTIFIER (Enter 1 in column 1)

Enter a frame number: N.ØØ or NN.ØØ
Enter frame type: Q
Enter frame label (optional): 7 or less letters and/or numbers, all letters before any numbers.

GROUP 2 TEXT (Enter 2 in column 1)

- To display information and/or a constructed response question.
- Legal line entries:

  Any printable keyboard characters.
  Use blank columns for spacing across.
  Use \ at end of line for each line-skip.
  Limit vertical display to 12-15 lines.

GROUP 3 ANSWERS (Enter 3 in column 1)

- To anticipate correct and most likely incorrect answers.
- Line entry form for answers:

  tag+answer1 -for correct answers
  tag answer2 -for incorrect or neutral answers
  where- "tag" is any letter A-Z -for character-by-character match.
  number 1-9 -for equivalent numbers matching.

- Line entry form for response processing aids:

  Ø processorname ON / ALL / number / OFF

  where- processor names are KEYWORD -to detect one or more key words.
  PHONETIC -to equate a misspelled word.

GROUP 4 ACTIONS (Enter 4 in column 1)

- To provide actions for answers.
- Line entry form for tags and actions:

  tag(s) action1 (action2) (action3)...
  where-"tag" is A-Z or 1-9 from Group 3
  - for unanticipated answers

- Legal actions are:

  F: or F:XXX -select right-wrong feedback or display XXX.
  R: or R:XXX -display WRONG, TRY AGAIN or XXX and repeat.
  C: -select and display the correct answer.
  C:Calcstatement -do an assignment, calculation, or control statement.
  B:Frame B:Lesson -branch to frame no.(label) or to named lesson.

Exhibit 2. Q-FRAME LANGUAGE CONVENTIONS

GROUP 1 IDENTIFIER (Enter 1 in column 1)

Enter a frame number: N.ØØ or NN.ØØ
Enter frame type M
Enter label (optional) 7 or less letters and/or numbers,
all letters before any numbers.

GROUP 2 TEXT (Enter 2 in column 1)

- To display information and/or the stem of a question.
- Legal line entries:

  Any printable keyboard characters.
  Use blank columns for spacing across.
  Use \ at end of line for each line-skip.

GROUP 3 ANSWERS (Enter 3 in column 1)

- To list letter tags and answer alternatives for display.
- Line entry form for answers:

  tag.+ ▢ answer1 -for correct answers
  tag. ▢ answer2
  tag ▢ answer3 -for incorrect or neutral answers

  where- "tag" is any letter A-Z, . is optional, and ▢ is one or more spaces, punctuation symbols, or both.

  All material is displayed except the + (displays a space)
  Limit vertical display for Groups 2 and 3 together to 12-15 lines.

GROUP 4 ACTIONS (Enter 4 in column 1)

- Provide actions for answer tags.
- Line entry form:

  tag(s) action1 (action2) (action3)...
  where- "tag" is a letter tag from Group 3

- Legal actions are:

  F: or F:XXX -select right-wrong feedback or display XXX.
  R: or R:XXX -display WRONG, TRY AGAIN or XXX and repeat.
  C: -select and display the correct answer.
  C:Calcstatement -do an assignment, calculation, or control statement.
  B:Frame B:Lesson -branch to frame no.(label) or to named lesson.

Exhibit 3. M-FRAME LANGUAGE CONVENTIONS

GROUP 1 IDENTIFIER (Enter 1 in column 1)

Enter a frame number: N.ØØ or NN.ØØ
Enter frame type: D
Enter frame label (optional) 7 or less letters and/or numbers, all letters before any numbers.

GROUP 2 CRITERIA (Enter 2 in column 1)

TO SPECIFY CONDITIONAL ACTIONS

- General form- IF condition1 (AND)/(OR) (condition2)
  (AND)/(OR) (condition3)...
  action1 (action2)...
  (ELSE)
  action3 (action4)...

  Where- ALL entries except IF condition1 action1 are optional.
  "AND" and "OR" can serve as logical connectives where more than one condition is specified.
  ELSE (if used) means "otherwise". If ELSE is not used and no condition is found, control goes to next frame.

CONDITIONS MAY BE ANY OF THREE FORMS

- Computational- IF (name) (relational) (number) (expression)

  where- "name" is user-defined and given a value with C:actions.
  "relationals" are GR GQ EQ NQ LQ LS
  "expression" is any legal CALC statement giving a number.

- Summary- IF (FROM frame,) (ALL)/(relational)/(NONE) (number) RIGHT/WRONG/SEEN frame/frame-frame

  where- "relationals" are GR GQ EQ NQ LQ LS
  "frame" is a no. or label and frames are given frame-frame.

- Pattern- IF (FROM frame,) frame,tags (frame,tags)...

  where- "tags" are any letter A-Z, number 1-9, + or -
  "frame" is a no. or label.

The Computational Form addresses records assigned with C: in frames.
The Summary Form and Pattern Form address PLANIT student records.
FROM (if used) begins a search for conditions at the most recent recorded entry of the frame specified.

- Actions may be:

  F:XXX -display message XXX.
  C:Calcstatement -do an assignment, calculation, or control statement.
  B:Frame B:Lesson -branch to a frame no. (label) or to named lesson.

TO SPECIFY UNCONDITIONAL ACTIONS

Actions F: C: B: on lines above conditions (or no conditions in frame).
Actions F: C: B: following control word END on lines below all conditional actions in the frame.

Exhibit 4. D-FRAME LANGUAGE CONVENTIONS

GROUP 1 IDENTIFIER (Enter 1 in column 1)

Enter a frame number: N.ØØ or NN.ØØ
Enter frame type: P
Enter frame label (optional): 7 or less letters and/or numbers, all letters before any numbers.

GROUP 2 STATEMENTS (Enter 2 in column 1)

- To specify a lesson procedure requiring several lines of statements that can be branched to as often as required.
- Legal line statements:

Decision Statements- of Computational, Summary, and/or Pattern forms (see Exhibit 4).

Action Commands- F:Message C:Calcstatement B:Linelabel
where- "Linelabel" is of the form... label colon e.g., LN4:
and can be followed on a line by decision statements or action commands.

B: actions in a P-frame must address a line label in the frame. No other usage of B: is legal.

Any line that is not a decision statement (IF,AND,OR ELSE,END) or action command (F: C: B:) will be taken as a calculation statement even if the C: is not used.

CALC statements (partial list)-
RETURN -this control word will cause a return to the next frame entry following the B: that branched to the P-frame.

Define data item: Form- SET name=value e.g., SET ITEM=Ø

Assign item value: Form- name=value / expression e.g., ITEM=1Ø / ITEM=ITEM+1 / ITEM=RANDOM*1Ø

Define data table: Form- SET MATRIX (name,rows,columns)
e.g., SET MATRIX(DATA,1Ø)
e.g., SET MATRIX(GRADES,4,2Ø)

Assign table values:
Fixed entries form- matrixname(cell)=ARRAY(number,number...)
e.g., DATA(1)=ARRAY(35.5,82.69...)
Generated entries form- matrixname(cell)=expression
matrixname(cell)=expression FOR(subscripts)
e.g., GRADES(1,J)=TRUNCATE(RANDOM*10) FOR(J=1,2Ø)

Arithmetic on stored data: Form- name=expression
e.g., MEAN=SUM GRADES(I,J) FOR(I=1,4 J=1,2Ø)/80

Display stored data:
Form- PRINT / ALIGN name / expression / 'literal' ,column; ROUND(number)

e.g., PRINT DATA(1)
35.5

e.g., PRINT 'AVERAGE = ';MEAN ROUND(2)
AVERAGE = 58.32

e.g., ALIGN DATA (2),6 ROUND (1)
82.7

Exhibit 5. P-FRAME LANGUAGE CONVENTIONS

IN PLANIT, THE PHONETIC ENCODING PROCESS IS ACCOMPLISHED IN FOUR STEPS:

STEP 1: LETTER EQUIVALENT: ALL LETTERS ARE TRANSFORMED INTO THEIR LETTER EQUIVALENTS. ANY REMAINING CHARACTERS INCLUDING BLANKS ARE UNCHANGED. THE LETTER IN ROW 1 IS TRANSFORMED INTO THE LETTER IMMEDIATELY BELOW IN ROW 2. PLANIT IGNORES ALL OTHER CHARACTERS.

ROW 1 ABCDEFGHIJKLMNOPQRSTUVWXYZ (ORIGINAL LETTER)
ROW 2 ABCDABCHACCLMMABCRCDABHCAC (LETTER EQUIVALENT)

STEP 2: THE H REPLACEMENT: EACH H IN A WORD IS TRANSFORMED TO THE PRECEDING LETTER PROVIDED THE CHARACTER IS A LETTER. IF NOT A LETTER (e.g., a blank), H IS UNCHANGED.

STEP 3: ELIMINATION OF SUCCESSIVE IDENTICAL CONSONANTS: ALL BUT THE FIRST ELEMENT OF AN UNINTERRUPTED SEQUENCE OF A SINGLE CONSONANT IS ELIMINATED, (e.g., CC=C, TT=T).

STEP 4: ELIMINATION OF AS: ALL VOWELS, TRANSFORMED INTO A'S, ARE ELIMINATED EXCEPT IF A IS THE FIRST CHARACTER OF THE WORD TO BE ENCODED. THE FINAL WORD CONTAINS ONLY CONSONANTS AND A LEADING A IF THERE IS ONE.

EXAMPLES:

ORIGINAL WORD

	1	2	3	4
PHONETIC	BHAMADAC	BBAMADAC	BAMADAC	BMDC
HAZARD	HACARD	HACARD	HACARD	HCRD
ON-LINE	AM-LAMA	AM-LAMA	AM-LAMA	AM-LM
AWHILE	AHHALA	AAAALA	AAAALA	AL

Exhibit 6. HOW 'PHONETIC' WORKS

PLANIT ERROR MESSAGE REFERENCE LIST

Message	Condition
ALIGN STATEMENT MUST SPECIFY A COLUMN	Occurs when a user has not inserted a comma between the expression and column specification in an ALIGN statement or the column value is missing, e.g., ALIGN X;7 or ALIGN X, or ALIGN X.
R/O CANNOT MODIFY, DELETE OR SAVE PROGRAM	Occurs when a Read Only author attempts to manipulate an instructional program as follows: ...Use PLANIT commands other than BREAK, EX, Print or Search. ...SAVE a named instructional program without renaming it. ...LOCK an instructional program.
COMMA MISUSED	a. Occurs when a user has not inserted a semicolon (;) as a separator between expressions in a PRINT or ALIGN statement, (e.g., PRINT 'ABC', X or ALIGN X, 1ø, Y, 2ø). b. Occurs when a user has inserted a comma on the right side of the replacement operator (=) in an assignment statement which can't be evaluated, e.g., X=5,6 or M=Y+,4. c. Occurs when nothing follows a comma in a conditional clause, e.g., IF 5, d. Occurs whenever one or more commas cannot be evaluated within an expression.
DECISION STATEMENT MUST START WITH A CONNECTIVE OR F, C, B.	Occurs during execution of a D frame when a line does not start with the appropriate decision language or Action Command
DUPLICATE PROGRAM NAME	Occurs if an author attempts to SAVE an instructional program using an already existing name.
ELSE/END MUST FOLLOW IF	Occurs during execution of a decision statement when the control words ELSE or END are encountered which do not follow a preceding conditional clause commencing with IF.

PLANIT ERROR MESSAGE REFERENCE LIST

Message	Condition
ALIGN STATEMENT MUST SPECIFY A COLUMN	Occurs when a user has not inserted a comma between the expression and column specification in an ALIGN statement or the column value is missing, e.g., ALIGN X;7 or ALIGN X, or ALIGN X.
R/O CANNOT MODIFY, DELETE OR SAVE PROGRAM	Occurs when a Read Only author attempts to manipulate an instructional program as follows: ...Use PLANIT commands other than BREAK, EX, Print or Search. ...SAVE a named instructional program without renaming it. ...LOCK an instructional program.
COMMA MISUSED	a. Occurs when a user has not inserted a semicolon (;) as a separator between expressions in a PRINT or ALIGN statement, (e.g., PRINT 'ABC', X or ALIGN X, 1ø, Y, 2ø). b. Occurs when a user has inserted a comma on the right side of the replacement operator (=) in an assignment statement which can't be evaluated, e.g., X=5,6 or M=Y+,4. c. Occurs when nothing follows a comma in a conditional clause, e.g., IF 5, d. Occurs whenever one or more commas cannot be evaluated within an expression.
DECISION STATEMENT MUST START WITH A CONNECTIVE OR F, C, B.	Occurs during execution of a D frame when a line does not start with the appropriate decision language or Action Command
DUPLICATE PROGRAM NAME	Occurs if an author attempts to SAVE an instructional program using an already existing name.
ELSE/END MUST FOLLOW IF	Occurs during execution of a decision statement when the control words ELSE or END are encountered which do not follow a preceding conditional clause commencing with IF.

PLANIT ERROR MESSAGE REFERENCE LIST

Message	Condition
ERROR IN DECISION STATEMENT	Indicates the conditional clause in a decision statement when executed is not complete or violates the prescribed format. These errors may include: ...Tags missing in the Pattern Form, e.g., IF 5, 6, ...RIGHT, WRONG, SEEN, USED not encountered in a Summary Form, e.g., IF GR 6 ...A label specified which does not exist. ...A missing comma following a frame number or label which preceeds answer tags, e.g., IF 5A- ...When nothing follows the dash to specify a range or the component is not a frame number or label, e.g., IF 5- or IF 5-%
FRAME DOESN'T EXIST	Occurs when an author attempts to COPY Edit or Print a frame which is non-existent
FRAME EXISTS	a. Occurs when an author attempts to Insert a frame over an existing frame. b. Occurs when a duplicate frame number is encountered in a card file.
FRAME LABEL EXISTS. LABEL=*	Occurs when an author attempts to enter a duplicate frame label.
FRAME NUMBER TOO LARGE OR TOO SMALL	Occurs when a frame number exceeds 1ØØ.ØØ.
GROUP DOESN'T EXIST	Occurs when an author attempts one of the following actions within his instructional program: ...Insert a line into a nonexistent group. ...Append onto a nonexistent group. ...Edit a nonexistent group. ...Modify material in a nonexistent group.
GROUP EXISTS	Occurs when an author attempts to Insert a group over an existing group.
ILLEGAL ANSWER FORMAT	Occurs during execution when an Answer Tag/Period/correct answer indicator is encountered in a Q or M frame in the form .A or +.A.

PLANIT ERROR MESSAGE REFERENCE LIST

Message	Condition
ILLEGAL BRANCH	Occurs during execution when a branch is attempted following the Action Command B: and the branch references a nonexistent frame/label or instructional program, or line label in a P frame.
ILLEGAL C:--NO CORRECT ANSWER SPECIFIED	Occurs during execution of a Group 4 of a Q or M frame when the Action Command C: is encountered and no answer has been specified as correct in Group 3.
ILLEGAL CHARACTER ON LINE	a. Occurs when a non arithmetic operator is encountered in an arithmetic expression, e.g., X+Y#6. b. Occurs when a semicolon (;) is encountered in a CALC statement other than a PRINT or ALIGN statement, e.g., X=FACT(5);SIN(3Ø c. Occurs when characters are encountered in a CALC statement which cannot be evaluated or treated as character strings in a PRINT or ALIGN statement.
ILLEGAL COMMAND FORMAT. TRY AGAIN...	This message is output for a variety of conditions. Some of these are: ...Occurs when a command is illegal or omitted from a command statement. ...Occurs when the user attempts to specify a range of frames using the Append, Edit, EX, or Insert commands. ...Occurs when numbers are not specified for groups, lines, or columns. ...Occurs when Group 1 is specified when using the commands Append, Delete, Insert, Modify. ...Occurs when an edit command other than Append is entered without specifying a frame number, e.g., #D or #P. ...Occurs when using the SAVE command and the command statement has more than three components, (e.g., SAVE MATH ON 1115) or the author is not associated with an instructional program. ...Occurs when using the GET command and the command statement has more than three components.

PLANIT ERROR MESSAGE REFERENCE LIST

Message	Condition
ILLEGAL CONNECTIVE...AND/OR MUST FOLLOW IF	Occurs during the execution of a decision statement when the connectives AND or OR are encountered which do not follow a preceding conditional clause commencing with IF.
ILLEGAL DECIMAL POINT IN NUMBER	Occurs whenever a decimal point is inappropriately used in a number and therefore cannot be evaluated, e.g., 1.3.5 or .17.5.
ILLEGAL FRAME TYPE CHANGE	Occurs when an author attempts to change a Q or M frame into a D or P or vice versa using the Edit command.
ILLEGAL GROUP NUMBER	Occurs when an author attempts to Append or Insert information into a frame using an illegal group number for the frame type, e.g., 3,3,A for a D frame or 3,5,I for a Q frame.
ILLEGAL LABEL/FRAME. ENTER COMMAND	a. Occurs when the frame number following the Append command is larger than the frame number in the instructional program. b. Occurs when using the Insert command and an author attempts to: ...Enter a frame over an existing frame. ...Enter a label which is not alphanumeric c. Occurs when an author specifies a nonexister or illegal label in the BREAK command. d. Occurs when the frame or label is illegal or nonexistent when using the EX command.

PLANIT ERROR MESSAGE REFERENCE LIST

Message	Condition
ILLEGAL USE OF F:--NO CORRECT ANSWER SPECIFIED	Occurs during execution of a Group 4 of a Q or M frame when the Action Command F: is encountered and no correct answer has been specified in Group 3.
ILLEGAL USE OF R:	Occurs during execution of a Group 4 in a Q or M frame when the Action Command R: is encountered and no answer tags exist in Group 3.
IMPROPER FROM SPECIFICATION	Occurs during execution when the optional component FROM is used in a conditional clause and nothing follows, or the subsequent component is not a frame number or label, or the label does not exist, e.g., IF FROM or IF FROM -X.
IMPROPER KEYWORD SPECIFICATION	a. Occurs when the component following KEYWORD is not ALL, OFF, ON or a number. b. Occurs when KEYWORD number is greater than the number of key words specified in an answer.
INTEGERS ONLY FOR GROUPS/LINES/COLUMNS	Occurs when the group, line, column entries in an edit command statement are not integers. e.g., 2,3.4,M or 3,2,1.5,P.
INVALID	Occurs when a user enters a log-in value which does not match one of the specified identifications for the installation.
LEFT PARENTHESIS MISSING	Occurs whenever parentheses (required or optional) are entered in a CALC statement which are not balanced, e.g.,) or ( )).
LINE CONTINUATION NOT PERMITTED	Occurs during execution when an author has attempted to extend a CALC statement more than a single line.
LINE DOESN'T EXIST	Occurs when an author attempts to Insert a line into a group above a line which doesn't exist or Edit, Modify or Print a nonexistent line or Append after a non-existent line.
LINE EXCEEDED	Occurs when the characters specified for modification in the Modify command will cause the number of characters entered on a line to exceed the allocated line length.

Message	Condition
NAME MISUSED	This message is output for a variety of conditions. Some of these are: ...Occurs when PHONETIC, does not have the proper number of modifying names; either SET before and/or ON/OFF following. ...Occurs when the name entered in an assignment does not conform to the PLANIT naming rules. (This message is not output when CALC is being used interactively. The expression to the right of "=" equates to a single value and is printed but no assignment is made, e.g., 1X=2+4 prints as 6.Ø). ...Occurs when user attempts to use SET interactively in a redefinition of an item, matrix, or function or use SET with a matrix and no argument exists or a character other than a left parenthesis exists between the matrix and its argument, e.g., SET M or SET M/1,2. ...Occurs when a value does not follow ROUND or WITHIN, or the value following KEYWORD is less than zero. ...Occurs when the name in a conditional clause is not an item or is unrecognized as an item.
NO G3. ANSWERS. ERROR	Occurs during execution in a Q or M frame when answer tags occur in Group 4 and Group 3 contains only Ø tags or doesn't exist.
NO SUCH FRAME/LABEL	Occurs when an author specifies a label in an edit command which does not exist or attempts to Insert or Edit a group or line for a nonexistent frame or attempts to Append a group within a nonexistent frame.
NUMBER OR SYMBOL MISPLACED	a. Occurs when attempting to evaluate a CALC statement and ...a number or symbol is missing or misplaced, e.g., 5*, 5*-65. ...a value does not exist to the right of a comma, e.g., F(2,,3). ...an operand is missing to the left of the arithmetic operator **, e.g., **5.

Message	Condition
NUMBER OR SYMBOL MISPLACED (continued)	...an operand is missing to the right of an arithmetic operator, e.g., 5*. ...the component to the left of the ** is not a number or a name for a number, e.g., 5**/ or 5**3M. ...arithmetic operators exist for which no operands exist. b. Occurs when a relational is missing following IF in the Computational Form of a decision statement, e.g., IF X 5.
PAIRED PRIME MISSING	Occurs when either the left or right prime used to enclose a character string is missing in a *PRINT* or *ALIGN* statement.
RESTART OR *SAVE* PROGRAM FIRST	Occurs when an author attempts to GET an instructional program and he is already associated with a named or unnamed instructional program.
RIGHT PARENTHESIS MISSING	Occurs whenever parenthesis (required or optional) are entered in a CALC statement which are not balanced, e.g., (or (( ).
TAGS MISSING IN DECISION STATEMENT	Occurs during execution when answer tags do not follow a comma in the Pattern Form of a decision statement, e.g., IF 5,
'=' MISUSED	a. Occurs when a second replacement operator (=) is encountered in a function definition and does not follow a FOR operator, e.g., FUNCTION EQUAL(X)=X5. b. Occurs when the replacement operator is encountered in a *CALC* statement and it doesn't belong or is not logical or can't be evaluated.
% MISUSED	a. Occurs during execution if the % symbol is encountered following the Action Command F: e.g., F%. b. Occurs when the % is encountered in a CALC statement following a name or an intervening blank occurs between the number and the %, e.g., 25 % or MEAN%.

PLANIT ERROR MESSAGE REFERENCE LIST

Message	Condition
VARIABLE HAS NOT PREVIOUSLY BEEN DEFINED	Occurs during execution when an item is encountered in a CALC statement which has no value.

APPENDIX C

EVALUATION DATA FORMS

PLANIT DATA QUESTIONNAIRE

NAME_____ DATE _____

SSAN_____ GRADE _____ MOS_____

JOB TITLE_____ EDUCATION_____
(Grade completed or degree)

ORGANIZATION_____

PHONE NUMBER_____DATE OF BIRTH_____

GT SCORE_____ LENGTH OF SERVICE_____

PLANIT DATA SHEET

ID NUMBER: 1._____ 2._____3. _____ DATE_____

NAME_____GRADE_____SSAN_____

MOS_____ _____JOB TITLE_____

ORGANIZATION_____

PHONE NUMBER_____DATE OF BIRTH_____

GT SCORE_____EDUCATION_____LENGTH OF SERVICE_____
(Grade completed or degree)

PLANIT COURSE - PART 1A

DATE & TIME: START_____END_____

DOWN TIME: 1. START_____END_____2. START_____END_____3. START_____END_____

4. START_____END_____5. START_____END_____6. START_____END_____

TOTAL TIME ON LINE_____

TOTAL FL FRAMES_____TOTAL ENTRIES_____

FL FRAMES PER MINUTE_____ENTRIES PER MINUTE_____

ENTRIES PER FL FRAME_____

PLANIT COURSE - PART 1B

DATE & TIME: START_____END_____

DOWN TIME: 1. START_____END_____2. START_____END_____3. START_____END_____

4. START_____END_____5. START_____END_____6. START_____END_____

TOTAL TIME ON LINE_____

TOTAL FL FRAMES_____TOTAL ENTRIES_____

FL FRAMES PER MINUTE_____ENTRIES PER MINUTE_____

ENTRIES PER FL FRAME_____

PLANIT COURSE - PART II

DATE & TIME: START_____END_____

DOWN TIME: 1. START_____END_____2. START_____END_____3. START_____END_____

4. START_____END_____5. START_____END_____6. START_____END_____

TOTAL TIME ON LINE_____ TEST SCORE_____

TOTAL FL FRAMES_____TOTAL ENTRIES_____

FL FRAMES PER MINUTE_____ENTRIES PER MINUTE_____

ENTRIES PER FL FRAME_____

STUDENT LESSON

1. PREPARING LESSON - OFF-LINE

DATE & TIME: ASSIGNED_____COMPLETED_____

TIME REQUIRED_____

NUMBER OF FRAMES: TOTAL_____: Q_____M_____D_____

BRANCHING FRAMES_____

2. ENTERING LESSON

DATE & TIME ENTERED: START_____END_____

ERRORS: NUMBER_____TYPES: 1._____

2._____3._____4._____

3. EXECUTING LESSON

DATE & TIME EXECUTED: START_____END_____

EXECUTION ERRORS: NUMBER_____TYPES: 1._____

2._____3._____4._____

4. EDITING LESSON

EDITING: START_____END_____

NUMBER OF FRAMES/LINES EDITED_____

_____

EDITING ERRORS: NUMBER_____TYPES: 1._____

2._____3._____4._____

5._____6._____7._____

5. LISTING OF FINAL STUDENT LESSON

DATE & TIME_____

NUMBER OF ERRORS_____

TYPE OF ERRORS_____

PLANIT DEBRIEFING QUESTIONS

NAME_____ SSAN_____

ORGANIZATION_____ PHONE NUMBER_____

INTERVIEWER_____ DATE_____

1. What did you think of the PLANIT course that you just went through?

_____

_____

_____

_____

2a. My attitude toward the course was that I . . .

( ) disliked it very much

( ) *disliked it*

( ) neither liked nor disliked it

( ) liked it

( ) liked it very much

2b. Please comment on your answer.

_____

_____

_____

3. Did you have any problems or difficulties in using the console or interacting with the computer?

( ) *yes* ( ) *no*

4. (If "yes" to item 3) please describe your most se·ious problem or difficulty.

_____

_____

_____

5. I estimate that I understood_____% of the instructional material (course content) presented.

6. Describe any part of the total course content that was particularly good, and tell why.

_____

_____

7. Describe any part of the total course content that was particularly bad, and tell why.

_____

_____

8a. I think that this method of learning to construct PLANIT Lessons is . . .

( ) very effective

( ) effective

( ) borderline

( ) ineffective

( ) very ineffective

8b. Why? _____

_____

_____

_____

9. For satisfactory understanding of the subject being studied, the amount of time provided was:

( ) much too long

( ) fairly long

( ) about right

( ) fairly short

( ) much too short

10. For satisfactory understanding of the subject being studied, the amount of material (information) provided was:

( ) much too large

( ) fairly large

( ) about right

( ) fairly small

( ) much too small

11. The technical detail provided was:

( ) very satisfactory

( ) satisfactory

( ) borderline

( ) unsatisfactory

( ) very unsatisfactory

12. The organization of the material presented was

( ) very satisfactory

( ) satisfactory

( ) borderline

( ) unsatisfactory

( ) very unsatisfactory

13. My understanding of the material presented

( ) very satisfactory

( ) satisfactory

( ) borderline

( ) unsatisfactory

( ) very unsatisfactory

14. The quantity of figures, worksheets and other supplementary material provided was:

( ) very satisfactory

( ) sacisfactory

( ) borderline

( ) unsatisfactory

( ) very unsatisfactory

15a. Were any of the supplementary materials irrelevant or unnecessary?

( ) yes ( ) no

15b. If yes, which? _____

16a. Can you think of any other supplementary materials that should be added to the course?

( ) yes ( ) no

16b. If yes, please describe. _____

17a. Did you have problems preparing your own lesson on the worksheets?

( ) yes ( ) no

17b. If yes, what were they. _____

_____

_____

18a. Did you have problems in executing your own lesson as a student?

( ) yes ( ) no

18b. If yes, what were they? _____

_____

_____ _____

19a. Did you have problems in editing your lesson?

( ) yes ( ) no ( ) no editing required

19b. If yes, what were they? _____

_____

_____

20a. Have you learned enough about PLANIT to develop PLANIT lessons in your own subject matter area?

( ) yes ( ) no

20b. If "no", please explain.

_____

_____

_____

_____

_____

21. Based upon your experience in constructing and checking out your own PLANIT lesson, what suggestions do you have for improving the PLANIT course.

_____

_____

_____

_____

22. Have you ever had experience using a computer before?

( ) yes ( ) no

23. If yes, to 22, have you ever taken a course on computers before?

( ) yes ( ) no

24a. Did you find learning PLANIT was

( ) very easy

( ) easy

( ) borderline

( ) difficult

( ) very difficult

24b. Why? _____

_____

_____

_____

25. How would you summarize your experience?

APPENDIX D

INTRODUCTORY LESSON

```
[ 1.00 9  START].
?THIS IS AUTOMATED INSTRUCTION PROVIDED BY THE U S ARMY
?RESEARCH INSTITUTE.  IN USING THE 'ARIS COURSE ARU' THERE
?ARE A FEW RULES OR CONVENTIONS YOU MUST FOLLOW SO THAT YOU
?MAY TALK TO THE COMPUTER WHICH WILL MONITOR YOUR PROGRESS
?THROUGHOUT THE PRESENTATION OF INSTRUCTIONAL MATERIAL.\
?YOU'LL SEE (TYPE 'GO' TO CONTINUE) AT THE BOTTOM OF THE
?SCREEN.  SIMPLY TYPE THE LETTERS 'G' AND 'O' AND PRESS THE
?RECTANGULAR KEY ON THE RIGHTHAND SIDE OF THE KEYBOARD LABELED
?'NEW LINE'.  MAKE SURE AND PRESS IT ONLY ONCE -- THEN WAIT
?FOR THE COMPUTER TO GIVE YOU FEEDBACK.(NORMALLY ABOUT 5 SECONDS)\
?                (TYPE 'GO' TO CONTINUE)
$A GO
$  GO
4A F:VERY GOOD.  NOW LET US CONTINUE.
4-F:YOU TYPED THE LETTER 'O' BUT THE NUMBER '0' INSTEAD OF THE
4-F:LETTER 'O'.  TRY AGAIN.
4-F:PLEASE TYPE THE LETTERS 'GO' AND THEN PRESS THE KEY LABELED
4 F:'NEW LINE'.
[ 1.50 ]
?YOU ARE IN DIRECT COMMUNICATION WITH A UNIVAC 1108 COMPUTER
?AT EDGEWOOD ARSENAL, MARYLAND.  THEREFORE, EACH TIME YOU
?DEPRESS A KEY ON THE CONSOLE, THAT CHARACTER, LETTER, NUMBER
?OR COMMAND IS SENT IMMEDIATELY TO THE COMPUTER.  SO ONE VERY
?IMPORTANT RULE IS NEVER DEPRESS A KEY UNLESS YOU WISH TO
?COMMUNICATE WITH THE COMPUTER.
[ 2.00 ]
?YOU ARE PROBABLY WONDERING---
?HOW WILL I KNOW WHEN THE COMPUTER EXPECTS A RESPONSE FROM ME?
?WELL IT'S EASY --- ANYTIME THE LAST CHARACTER ON THE SCREEN IS AN
?ASTERISK ( * ) YOU WILL KNOW THE COMPUTER EXPECTS YOU TO INPUT
?A RESPONSE.  SIMPLE, ISN'T IT.
[ 3.00 ]
?THE RESPONSES WHICH YOU WILL BE REQUIRED TO GIVE WILL VARY
?ACCORDING TO THE INSTRUCTIONAL MATERIAL.  THEY MAY CONSIST
?OF A SINGLE LETTER, A SINGLE WORD, SEVERAL WORDS, OR EVEN
?A SENTENCE.  SOME OF THE QUESTIONS MAY ASK FOR A NUMERIC
?ANSWER.  WHEN YOU ARE READY TO RESPOND TO A QUESTION, YOU
?SHOULD KEYIN THE COMPLETE ANSWER.  THE COMPUTER WILL NOT
?PROCESS YOUR ANSWER UNTIL THE 'NEW LINE' KEY IS DEPRESSED.
?                (TYPE 'GO' TO CONTINUE)
$A GO
$  GO
4  F:, A, LET'S MOVE ON.
4  F:YOU TYPED THE NUMBER '0' INSTEAD OF THE LETTER 'O'
4 F:F MORE CAREFUL NEXT TIME.  NOW, LET'S MOVE ON.
4-F:PLEASE TYPE 'GO' AND DEPRESS THE 'NEW LINE' KEY
4-F:PLEASE TYPE 'GO' (BE SURE TYPE 'O' THEN PUSH THE KEY
4 F:LABELED 'NEW LINE'
```

```
1 4.00 0
2EACH TIME YOU INPUT A RESPONSE TO THE COMPUTER -- WHETHER
2IT BE A SINGLE CHARACTER OR A SENTENCE --- YOU SHOULD DEPRESS
     INTRO                          JTI

2THE 'NEW LINE' KEY ONCE AND ONLY ONCE.  AS STATED BEFORE, THE
2COMPUTER WILL NOT PROCESS YOUR RESPONSE UNTIL THE 'NEW LINE'
2KEY IS DEPRESSED.  IF YOU WERE TO PRESS THE 'NEW LINE' KEY
2TWICE, THEN THE COMPUTER WOULD THINK YOU WERE ANSWERING TWO
2QUESTIONS (THE LAST QUESTION AND NEXT QUESTION TO COME ALONG).
2YOU MAY CHANGE YOUR ANSWER OR RESPONSE ANYTIME PRIOR TO DEPRESS-
2ING THE 'NEW LINE' KEY, BUT ONCE YOU HAVE DEPRESSED IT, DON'T TRY
2TO MAKE ANY CHANGES.  WAIT UNTIL THE COMPUTER HAS PROCESSED YOUR
2RESPONSE.  IF YOUR RESPONSE HAS A MISTAKE IN IT, YOU WILL PROBABLY
2GET THE OPPORTUNITY TO TRY THE QUESTION AGAIN.
2                        (TYPE 'GO' TO CONTINUE)
3A GO
4A F:HERE'S THE WAY TO MAKE A CHANGE OR CORRECTION.
4-F:EVEN THOUGH YOU DIDN'T TYPE 'GO', HERE'S THE WAY TO MAKE
4F:A CHANGE OR CORRECTION TO YOUR RESPONSE.
1 5.00 0
2THERE IS A RECTANGULAR KEY ON THE LEFTHAND SIDE OF THE KEYBOARD
2 WICH HAS THE LABEL 'CONTROL'.  DO YOU SEE THE 'CONTROL' KEY?
2       (ENTER 'YES' OR 'NO'  THEN PUSH THE 'NEW LINE')
3A YES
3- NO
4A F:VERY GOOD, NOW.....
4- R:IT IS THE FIRST KEY OF THE SECOND ROW.  DO YOU SEE IT NOW?
4-F:ENTER 'YES' OR 'NO' THEN DEPRESS THE 'NEW LINE' KEY.
1 6.00 0
2IF YOU WERE TO HOLD DOWN THE CONTROL KEY AND DEPRESS THE
2CHARACTER 'X' , THE COMPUTER WILL IGNORE (OR CANCEL) ALL INPUT
2SINCE THE LAST TIME 'NEW LINE' WAS DEPRESSED.  YOU WILL KNOW THAT
2THIS ACTION HAS TAKEN PLACE WHEN THE CURSOR JUMPS TO NEXT LINE ON
2SCREEN AND RETURNS TO THE LEFT MARGIN.  LET'S ASSUME YOUR ENTRY IS:\
2                              RIGHT\
2BUT WHAT YOU WANTED WAS LEFT, THEN YOU WOULD HOLD DOWN THE CONTROL KEY
2AND PRESS THE X KEY AND THEN ENTER THE CORRECT ANSWER.  THE SCREEN
2WILL NOW LOOK LIKE THIS:\
2                              RIGHT
2                              LEFT\
2TRY THE CONTROL X FEATURE SEVERAL TIMES.  TYPE IN A RESPONSE AND
2THEN HOLD DOWN THE CONTROL KEY AND PRESS X.  WHEN YOU ARE READY
2TO PROCEED  TYPE 'GO'.
3A^GO
4A F:\
4- F:\
```

```
1 7.00 \
2'OW WE ARE READY TO TRY SOME QUESTIONS.....\
2YOU WILL BE ASKED VARIOUS KINDS OF QUESTIONS THROUGHOUT YOUR
2COURSE.  ONE OF THESE WILL BE MULTIPLE CHOICE QUESTIONS.  HERE
2YOU ARE EXPECTED TO TYPE IN THE LETTER WHICH HAS THE RIGHT
2ANSWER AND THEN DEPRESS THE 'NEW LINE' KEY.  FOR THE FOLLOWING
2QUESTION DO THIS.\
2              WHAT NUMBER COMES AFTER FOUR?
3  A. 3
3  B. 2
     INTRO                        JTI


3  C.+5
-------------------------------------------------------------------
3  D. 1
4A R:NO.  5 COMES BEFORE 4. WHAT COMES AFTER 4?
4BD R:NO. TRY AGAIN.
4C F:RIGHT. LET'S GO ON.
4AND F:NO. YOU SHOULD HAVE ENTERED THE LETTER 'C' TO
4F:INDICATE THE ANSWER '5'.
1 8.00 0
2ANOTHER TYPE OF QUESTION WILL GIVE YOU THE CHOICE OF ANSWERS
2IN PARENTHESES.  LET'S TAKE THE SAME QUESTION AND ASK IT LIKE
2THIS.  THE NUMBER THAT COMES AFTER FOUR IS (THREE) (FIVE). HERE
2YOU ARE EXPECTED TO TYPE THE CORRECT ANSWER, THREE OR FIVE, AND
2THEN PRESS THE 'NEW LINE' KEY.  PLEASE DO THIS NOW.
3A+FIVE
3B THREE
3C+5
3D 3
3E+5
3F 5
4A F:GREAT.  LET'S GO ON.
4BD2 F:YOU TYPED IN THE ANSWER O. K., BUT NOT THE RIGHT ONE.
4F:ENTER THE WORD THREE OR FIVE THIS TIME.
4-F:PLEASE TYPE IN 'THREE' OR 'FIVE' AS YOUR RESPONSE.  DO
4F:THIS NOW.
4C1 F:YOU HAVE THE 'RIGHT' NUMBER BUT WE WANTED YOU TO ENTER
4F:THE WORD 'FIVE'.
4-F:OOPS, WE WANTED YOU TO ENTER THE WORD 'FIVE'.
```

```
1 9.00 Q
2THIS ONE IS MORE DIFFICULT TO DO.  WE WILL GIVE YOU A SCRAMBLED
2LIST AND ASK YOU TO PUT THEM IN THE RIGHT ORDER.  THE QUESTION
2MIGHT BE: PLACE THE FOLLOWING IN NUMERICAL ORDER FROM THE
2LOWEST TO THE HIGHEST.
2                              A. 4
2                              B. 3
2                              C. 5
2YOU WOULD TYPE IN YOUR ANSWER AS FOLLOWS: B A C AND THEN
2DEPRESS THE 'NEW LINE' KEY.  PLEASE DO THIS NOW.
3A+B A C
3B+BAL
3C+B,A,C
3C+B, A, C
4A F:GREAT. LET'S GO ON.
4B F:YOU TYPED IT IN O. K., BUT DIDN'T LEAVE THE SPACE BETWEEN
4B:THE LETTERS.  TRY AGAIN.
4C B:NO COMMAS PLEASE.  TRY AGAIN
4-F:TYPE THREE LETTERS TO SHOW THE ORDER OF THREE NUMBERS,
4B:LOWEST TO HIGHEST.
4F:NO. YOU SHOULD HAVE ENTERED THE LETTERS B A C
1 9.50 Q
2IF FOR ANY REASON YOU HAVE TO LEAVE THE TERMINAL, PLEASE TYPE\
2                          <FINISHED\
2THE '<' KEY IS LOCATED ON THE BOTTOM ROW OF KEYS.  IT IS THE
     INTRO                      JTL


2UPPER CASE COMMA (,).  WHEN YOU RETURN TO THE TERMINAL, PLANIT
2WILL RETURN YOU TO THE SAME POINT WITHIN THE INSTRUCTIONAL
2LESSON.  NOW.....

110.00 Q
2WHEN YOU ARE READY FOR THE FIRST
2PLANIT LESSON -- TYPE 'GO'
3A GO
4-A F:HERE IS THE LESSON C:GOTO PLANIT1
```

FREQUENCY OF FRAME WORKSHEET ERRORS BY STUDENT

ERROR	STUDENT S1	S16	S3	S4	S2	S10	S13	S5	S8	S9	S15	S7	Total Incidents by Error
Missing Entries													
1. Frame type											9		9
2. Required group(s) for frame type										1	14		15
3. Group number (any group)		1									16	1	18
4. Correct answer + symbol (where required)										1		5	6
5. Statement to initialize counter							1	1	1	2			5
6. Statement to increment counter							5			5			10
7. Group 3 or Group 4 tag(s)											1	15	16
8. Required action statement or command	1			1		2	3		4	5		20	36
Incorrect Entries													
9. Correct entry in wrong group						1							1
10. Incorrect frame type for intent						1			2		2	3	8
11. Incorrect use of lineskip control symbol					1								1
12. Improper Group 3 construction for intent	2	3					6		1	1		5	18
13. Correct answer + symbol in wrong position			1										1
14. Wrong syntax for Group 3 or Group 4 tag				2									2
15. Use of number tag in M-frame			1										1
16. Incorrect syntax for KEYWORD or PHONETIC					2		2			1			5
17. Incorrect syntax to increment counter									9	3			12
18. Incorrect positioning of counter actions			1		8	4	1	3					17
19. Incorrect syntax or choice of action command			1	1		1	4		2	5	4		18
20. Incorrect line order of action commands		1		2		6							9
21. Incorrect spacing of tags/action commands							4	5		2			11
22. Illegal choice of printable Group 4 character	1												1
23. Improper procedure for line continuation			1		2	15	1	16	11	2			48
24. Improper relational for decision logic intent				1	1								2
25. Improper syntax of decision statement				1	1			3		6			11
TOTAL ERRORS = (NUMERATOR OF SKILL RATIO)	4	5	5	8	15	30	27	28	30	34	46	49	
			Highly Acceptable				Acceptable				Unacceptable		