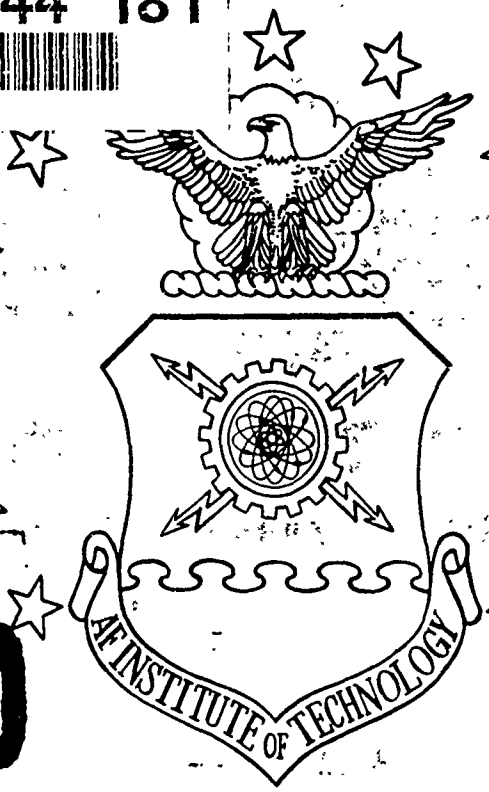


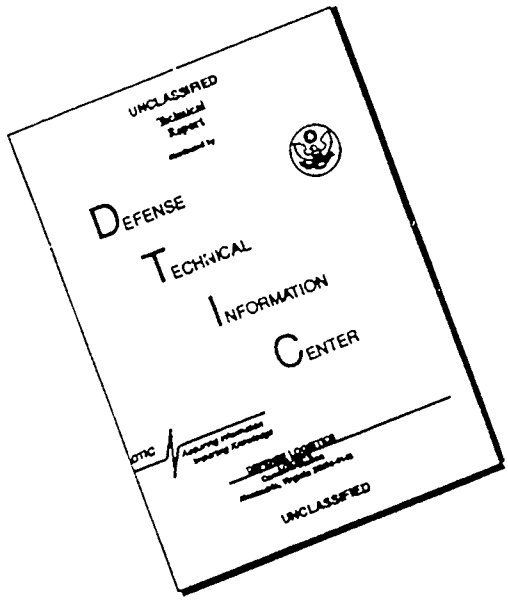
DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISCLAIMER NOTICE



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

AFIT/GIR/LSR/91D-10

AN INVESTIGATION INTO THE USE OF COMPUTER-ASSISTED INSTRUCTION TO PRESENT BASIC ENGLISH GRAMMAR CONCEPTS

THESIS

Frank E. Jones, Capt, USAF

AFIT/GIR/LSR/91D-10

The views expressed in this thesis are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

AFIT/GIR/LSR/91D-10

AN INVESTIGATION INTO THE USE OF COMPUTER-ASSISTED INSTRUCTION TO PRESENT BASIC ENGLISH GRAMMAR CONCEPTS

THESIS

Presented to the Faculty of the School of Systems and Logistics

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Information Resource Management

Frank E. Jones, B.A., M.S.

Captain, USAF

December 1991

Acknowledgements

I extend my deepest appreciation to Dr. Charles Fenno, who served in name as my thesis advisor, but who in reality played a much greater role. From our earliest conversations concerning research, through the idea for and development of this project, and culminating with the current document, Dr. Fenno consistently provided key guidance, thoughtful suggestions, and just the right amount of prodding to keep me motivated and constantly focused in the right direction.

I thank Dr. Kirk Vaughan for inviting me to address his students in COMM 310 course and solicit volunteers for my field test.

I thank the following professors who served as subject-matter experts for the research and whose expert opinions greatly improved my own understanding of English grammar: from the Department of Communication and Organizational Science, School of Systems and Logistics, AFIT: Dr. Charles Fenno, Lt Col Tom Huguley, Maj John Stibravy, Dr. Freda Stohrer, Dr. Robert Weaver; and Dr. Kirk Vaughan; and from the Department of English, University of Dayton: Dr. Peter Aarons, Dr. Eugene August, Brother James Boehlein, Dr. Alex Cameron, Dr. Joyce Durham, Dr. Cheryl Fenno, Mr. James Fisher, Mr. Sam Johnson, Dr. Joseph Patrouch, Ms. Madeline Riley, Mrs. Ellie Stockum, Dr. Alex Tuss, Ms. Sara Wieland, and Dr. Stephen Wilhoit.

Finally, I gratefully acknowledge the two most important people in my life: Suzann and Kevin Jones. Thank you for your unending patience when I seemed to ignore you so many times over the past year. I could not have made it through this project without your love, your understanding, and your kindness. I am eternally thankful to both of you.

Trademark Acknowledgements

DIALOG is a registered trademark of Dialog Information Services, Inc.

IBM PC is a trademark of International Business Machines Corporation.

Technojock's Turbo Toolkit is a trademark of Technojock Software, Inc.

Turbo C is a registered trademark of Borland International.

Turbo Pascal is a registered trademark of Borland International

Table of Contents

List of Figures

Abstract

The purpose of this research was to develop a computer-assisted instruction (CAI) tutorial that would address basic grammatical concepts.

The research consisted of four phases. The first phase was a review of applicable CAI literature, including theories, instructional and programming techniques, and previous research findings. The second phase was a survey of communication professors to determine the grammatical errors commonly committed by students. The third phase involved development of the tutorial content and the program software. The resulting tutorial presents basic grammar in terms of the three building elements of language (words, phrases, clauses) and the five functions they perform (naming, showing action/condition/state, modifying, connecting, adding filler) to create sentences. The program software, written in Turbo Pascal, employs a hypertext environment for its computer-user interface. The final phase was a preliminary field test of the program to evaluate its execution and hypertext interface.

The project resulted in a functional, stand-alone CAI tutorial that addresses basic grammatical concepts. The hypertext program code can be used to create and display any hypertext document. The study recommends that further research be conducted to broaden the scope of the tutorial and to evaluate its effectiveness as an instructional tool.

AN INVESTIGATION INTO THE USE OF COMPUTER-ASSISTED INSTRUCTION TO PRESENT BASIC ENGLISH GRAMMAR CONCEPTS

I. Introduction

Background

The School of Systems and Logistics at the Air Force Institute of Technology (AFIT) is "the Air Force's graduate school of technical management" (11:158). Students entering graduate programs at the school have already completed their undergraduate education. Occasionally, however, students arrive who need remedial training in basic skills to prepare themselves for the rigors of graduate-level education. Currently, the school offers remediation in mathematics, computer skills, and communication, through non-credit, four-week undergraduate courses scheduled prior to the graduate program.

The communication course is a

> refresher course in basic aspects of written expression, including grammar, mechanics, usage, style, logic, and organization. Course work consists of class discussion and self-study exercises. (11:178)

The course employs traditional classroom instruction by a professor, reading assignments from textbooks and handouts, and several written assignments. To meet its objectives, the course meets five days a week throughout the four-week summer short term (49).

The current approach to the communication course has two inherent limitations. First, the four-week period provides inadequate time to allow coverage of all of the fundamental writing skills required for effective communication. The rapid pace of the course does not allow

time for the instructor to dwell on all topics to the extent necessary for each student to master the areas. Grammar, an important element of the course, is especially difficult to cover without excluding other subject areas. Second, the course does not allow for transportability. That is, the interactive portions of instruction take place entirely within the classroom. There is no means by which students may devote non-classroom hours to interactive study. Further, a transportable tool could be forwarded to incoming students to use as a refresher prior to their arrival at AFIT.

Based on these limitations, it is evident that the school would benefit from an alternative method of teaching the basic grammar sections of its undergraduate communication course.

One potential alternative to traditional classroom instruction is computer-assisted instruction (CAI). Matta and Kern have observed that CAI "is becoming an increasingly popular technique for education" (37:77). The advantages of CAI include consistency of instruction each time the lesson is offered, student-paced lessons, immediate feedback, and active student involvement (7:4-5). Research has shown that CAI can be an effective supplement to instruction in the Air Force. Sivley, for example, studied the effects of using CAI in a base supply course and concluded "that CAI--in terms of learning levels and time savings--is a feasible method of instruction" (42:59).

Research Objective

The purpose of this research was twofold: first, to determine the types of grammatical errors commonly committed by students in their written communication, and second, to develop a CAI tutorial covering

fundamental grammar, the effectiveness of which would be evaluated by students in AFIT's refresher course in written communication. If a successful CAI tutorial could be developed, it would provide two important benefits: First, it would benefit the students who use it by helping them to improve their writing skills. Second, it would ease the classroom intensiveness of the current course structure.

The following research objectives were established to guide the research:

1) Determine what CAI theories, techniques, and research findings are already available.

2) Identify the grammatical errors commonly committed by students in their written communication.

3) Develop a CAI tutorial that will help students to better understand English grammar concepts.

4) Assess the effectiveness of the tutorial by providing it to students in AFIT's refresher course in fundamental written communication and evaluating their feedback.

The first objective was achieved through a literature search and review. The second objective was reached by first consulting both textbooks and then surveying professors of communication at AFIT and at the University of Dayton. The third objective involved the development and verification of a computer program in a higher-order language. The fourth objective was achieved by distributing the tutorial to volunteers in the refresher course and soliciting their feedback through a survey and quiz of their understanding of the concepts presented in the tutorial.

Assumptions

Several assumptions were made at the outset of this project. First, it was assumed that the appropriate writing style is dictated by

the purpose of the communication intended, and further, that the purpose of communication determines whether or not grammatical irregularities are acceptable. For this project, errors and patterns of writing are addressed in terms of expository, utilitarian prose, rather than prose intended to entertain. Second, the project assumes that an understanding of grammatical concepts is necessary for effective writing. In other words, a writer must know how the language operates in order to construct communication for specific audiences and purposes. Third, the project assumed that the CAI tutorial would be used as a supplement providing further, more in-depth instruction in written communication, and that, while it is an effective tool for its intended purpose, it is not intended to replace a complete written communication course.

Scope and Limitations

The scope of this research is limited in several areas. The research is limited to the application of CAI to the specific area of concepts of grammar. It is not intended to address the overall effectiveness of CAI in other applications. The scope is also limited in terms of hardware and software requirements. The software will be written specifically for IBM PC-compatible microcomputers, which are the standard computers at the Air Force Institute of Technology. This limits the amount of computer memory available to the software. If a mainframe computer were employed, memory availability would be much greater.

Another limitation is that, as computer technology continues to advance, the software developed for the current research could, in time, become obsolete. A related issue is that the software used in this

research project is written in a higher-order language, using Borland's Turbo Pascal. Software written with Turbo Pascal can be distributed without licensing fees (3:2). Other available programming packages were dismissed as cost prohibitive. Kemmer-Richardson et al. compare the various authoring tools available and the relative costs involved (28:78-95).

II. Methodology

Introduction

This chapter addresses the steps taken to achieve the four research objective statements listed in Chapter 1. It describes the procedures used to perform the literature review, the process of surveying the professors of communication to determine grammatical errors commonly committed by students in their written communication, the coding of the program software, and the evaluation of the program by the students in AFIT's refresher course.

Objective Statement 1

> Determine what CAI theories, techniques, and research findings are already available.

This objective was reached through a review of the literature pertaining to these subjects. Sources included journals, theses, and books, which were consulted for applicable information. The areas of concentration included descriptions of CAI, advantages and disadvantages, computer-user interaction, and computer programming. The results of the literature review and their application to this project are contained in Chapter 3.

Objective Statement 2

> Identify the grammatical errors commonly committed by students in their written communication.

Research Objective 2 was met through a series of steps. First, four sources were reviewed to form a basis from which to develop a survey instrument. Two of these sources were writing handbooks published for college students (25; 32). The third source was a class

handout from the Air Force Institute of Technology (AFIT) course COMM 687, Theory and Practice of Professional Communication (15). The fourth source of preliminary material was the course work from AFIT's refresher course, COMM 310, Fundamentals of Written Communication (48). From these sources, a list of eight common grammatical errors was compiled (see Appendix A).

From this list of grammatical errors, a questionnaire was developed for distribution to two groups of communication professors who served the function of subject-matter experts for the project. The first group, consisting of seven professors from the Department of Communication and Organizational Sciences, School of Systems and Logistics, AFIT, provided input from their perspective in dealing directly with AFIT students. The second group, consisting of sixteen professors and teachers from the Department of English, University of Dayton, provided additional insight from the perspective of a civilian undergraduate school. The questionnaire and the data collected are at Appendix B. The data are analyzed in Chapter 3.

The questionnaire consisted of four sections. In the first section, the respondent was presented with a variety of grammatical errors and asked to evaluate their frequency in student writing and the extent to which they degrade communication. The purpose of this section was to determine the specific content of the CAI grammar tutorial.

In the second section of the questionnaire, the respondent was asked to indicate the order of instruction most appropriate for teaching grammatical concepts. The purpose of this section was to determine an appropriate order for sequencing the content of the CAI grammar tutorial.

The third section of the questionnaire solicited three other inputs. First, the respondent was asked to indicate the number of years he or she had been teaching Communication/English/Composition. This would establish the relative authority of the subject-matter experts. Next, the respondent was asked indicate the student textbook he or she thought was the best for helping students learn grammatical concepts. This would provide the researcher with additional resources from which to draw the course content.

The final section of the questionnaire solicited other comments or suggestions from the respondent.

The information gathered from the subject-matter experts was compiled and analyzed to determine the grammatical errors commonly committed by students. Additionally, this information was useful in the development of the CAI grammar tutorial. The results of this survey were used as a starting point for the development of the CAI grammar tutorial.

Objective Statement 3

> Develop a CAI tutorial that will help students to better understand English grammar concepts.

This objective consisted of two distinct facets: development of the tutorial contents and coding of the program software.

Development of the Tutorial Content. The content of the tutorial was developed apart from the program coding. The content development phase included a series of review iterations involving the researcher, the research advisor, and the course director of CCMM 310, Fundamentals of Written Communication.. The skeleton of the tutorial was developed using notecards as a reference tool. Since the tutorial was a branching

tutorial, each notecard represented a separate branch and level of information, and the notecards were arranged to form a storyboard of the tutorial's contents. From these notecards, a complete "paper version" of the content was written. The paper version consisted of the actual screens that the user of the tutorial would see during execution of the computer program, annotated to indicate key information necessary for writing the code for the computer program. The tutorial screens are at Appendix C.

Coding the Program Software. The two most common methods by which to develop CAI software for the microcomputer are to write the program using a higher-order language or to use a CAI authoring system (28:78,83). As mentioned in Chapter 1, the tutorial for the current project used the higher-order language Turbo Pascal. This decision was reached based on two factors. First, instruction in the use of this software was included as part of the researcher's educational program at AFIT and, as such, was an available resource to the researcher. Second, software written and compiled using the Borland Turbo Pascal environment may be distributed without licensing fees (3:2).

Coding of the software consisted of converting the paper version of the tutorial into the CAI version. The computer program was written by the researcher, after extensive consulting of several sources (4; 5; 45). The Borland books provided the guidance necessary to write computer programs in Turbo Pascal. The TechnoJock software provided several important procedures used in the project, including menus and other screen-design features of the final product. Software developed using both the Borland software and the TechnoJock software may be used

and distributed without incurring licensing fees from those companies (3:2; 45:iii).

Since the project involved a branching tutorial, the program's computer-user interface was developed in a hypertext environment, which allowed the user to select options from menus and to branch to desired sections of the tutorial at will. The tools necessary to code hypertext within a computer program were found in an article by Gessner (20). Gessner provided these tools to his readers in order to "demystify hypertext systems, and to show you how to take advantage of the hypertext paradigm in your own programs" (20:22). Gessner's "hypertext editor" was modified for use in the development of the text of the tutorial. His "hypertext engine" was modified and used within the computer program to display the hypertext contents of the tutorial. Gessner encourages programmers to freely use, modify, or enhance his source code (20:33).

Verification of the software was performed iteratively throughout the process, with debugging of each portion of the program as it was developed. After the tutorial was completed, it was verified to ensure the tutorial performed correctly. The final source code of the computer programs for the tutorial is at Appendix D.

Objective Statement 4

> Assess the effectiveness of the tutorial by providing it to students in AFIT's refresher course in fundamental written communication and evaluating their feedback.

To achieve this objective, volunteers were solicited from AFIT's course, CCMM 310, Fundamentals of Written Communication. Thirty-six of the 40 students enrolled in the course during the Summer Short Term,

1991, volunteered and were provided with a copy of the tutorial. The students were asked to make use of the tutorial during the final two weeks of the course. After the course ended, a survey was distributed to the volunteers to obtain their inputs and to evaluate how well they understood the concepts contained in the tutorial. The survey and the data collected are contained in Appendix E. The results of the survey are analyzed in Chapter 3.

Summary

This chapter described the process through which the four objective statements were addressed during the research project. It discussed the literature review, the survey of subject-matter experts, the development of the CAI tutorial, and the evaluation of the tutorial. The findings and results of these actions are addressed in Chapter 3.

III. Results and Analyses

Introduction

This chapter presents the results of the research project and analyses of the data collected. Specifically, it addresses these four areas:

1) the results of the literature review,
2) the data collected from the questionnaire given to subject-matter experts,
3) the results of the tutorial program development phase, and
4) the data collected from the program evaluation.

Results of the Literature Review

This section relates the findings of the literature review and their application to the current project. The literature review concentrated on two areas: computer-assisted instruction (CAI) and programming a branching tutorial.

Computer-Assisted Instruction. Matta and Kern assert that "the computer has entered the classroom as a common tool in education. Computer-aided instruction is becoming an increasingly popular technique for education" (37:77). The discussion of CAI addresses the following areas:

1) CAI terminology,
2) learner interaction with CAI,
3) advantages and disadvantages of CAI,
4) types of CAI,
5) development,
6) computer-user interface,

7) screen design,

8) validation of CAI, and

9) previous CAI research.

Terminology. Numerous terms are used throughout the literature relating to the use of computers in education. These terms include "computer-assisted (or -aided) instruction," "computer-managed instruction," and "computer-based instruction." Additionally, for each term, the word "instruction" is often replaced with the word "training." Although these various terms are related, each describes a different facet of the subject.

Computer-assisted (or -aided) instruction (CAI) is defined by Kemner-Richardson et al., as "the use of the communications and storage capabilities of a computer to provide the direct presentation of instructional materials and/or provision of practice to the learner" (28:9). In CAI, the computer is "the prime deliverer of the instruction" (6:16). Orwig adds that in CAI, "the learner and the computer are in direct communication with each other" (40:3).

Computer-managed instruction (CMI) involves the use of a computer to perform administrative tasks related to education, including "testing, scheduling, allocating resources, collecting student data, and providing status reports" (28:13). CMI can be used to support either CAI or other non-CAI training approaches.

Computer-based instruction (CBI) and computer-assisted instruction (CAI) are similar. Godfrey and Sterling see the difference between the two as one of attitude. "Learners in CBT environments tend to be paid employees, and there is a far greater emphasis on efficient mastery of objectives and on comparison of costs with traditional training" (21:4).

Others define CBI in broader terms as the combination of CAI and CMI (28:11).

APPLICATION TO THE PROJECT: The tutorial developed during this research most closely follows the description of the term "computer-assisted instruction" (CAI), so that term will be used throughout this report.

CAI-Learner Interaction. Thompson and Jorgensen offer three models of interactions possible in the learning environment. These models describe the levels of interaction that CAI can require of its user. At the lowest level is reactive interaction, where the interaction is limited to "routine choices requiring only short-term memory or simplistic cognitive manipulations of the material" (46:24). The next level, proactive interaction, requires learners to "construct and deduce principles from their own actions and experiences" (46:24). At this level, the computer acts as a "facilitator or 'mirror'--allowing for learner initiated activities and self-monitoring" (46:24). The third level, the true interactive level, combines both of the previous levels, "spiralling between information transmission and simpler levels of learner manipulation of the material and activities that are grounded in the learner's own evolving understandings of the experiences" (46:25). Thompson and Jorgensen conclude that proactive and interactive approaches to learning "suggest ways to be more responsive to and foster critical understanding on the part of the learner" (46:26).

Jones and Steinberg offer four insights into computer-assisted instruction and its effects on learning (27:201). First, there is a distinct difference between reading and listening. CAI presents its information primarily in written form, while human learning is both

aural and visual. Without the aural aspects, the learner loses the intonations and emphasis found in speech. For example, they offer the following two questions:

"YOU are going to buy that?"

"You are going to buy THAT?"

Even though they are worded the same, these questions take on different meanings, depending on the emphasis given to various words. "Although typographical cues such as capitalization can provide some of this extra information, they are generally not as effective as oral cues" (27:201).

The next insight offered by Jones and Steinberg is the importance of explicit messages. In CAI, there is no human instructor to provide clarification. Thus, any instructions necessary to complete the program must be clearly stated and understandable. "Everything that is implicit [outside of the CAI environment] must be made explicit [within CAI]" (27:201).

Third, the CAI program should not implement procedures or processes that are contrary to user expectations or conventions, as this will tend to hinder learning. For example, software programs often allow the user to press the <Esc> key to interrupt a currently running process or procedure; if the user expects the <Esc> key to perform this function, use of this key for other purposes would be contrary to the user's expectations. If it is necessary to violate this practice, the procedure should be explained explicitly (27:201).

Fourth, to be effective, the CAI program should provide a framework for understanding that is tailored to the intended user's background knowledge. Unfamiliar jargon or technical terms should be avoided. If such terms must be used, they should be sufficiently

defined or explained. Further, "the organization of concepts and labels should also fit into the user's framework of understanding" (27:210). Outlines and hierarchical structures, for example, are both methods of structuring information that fit into most users' frameworks of understanding.

Burke outlines two types of CAI design considerations for the computer-learner interaction: physical designs, which are concerned with the paths the user follows through the lesson, and logical designs, which are concerned with how the material is presented (6:95-97).

Physical designs are divided into six groups, some of which may be combined. Linear designs present information in the same linear order each time they are used. Spiral designs spiral through the material, covering a different aspect of it each time--for example, respiration across several organisms, then reproduction across the organisms (6:96). Branching designs provide alternate tracks through the material and allow for personalized instruction. Multi-track design offers different levels of presentation, from abstract to detailed. Regenerative design generates a different set of problems each time the program is run. Adaptive design uses the data that is accumulated as a basis for self-improvement of the lesson (6:97).

Logical designs, dealing with how the lesson presents the material, are divided into five groups. In didactic design, information is presented, then the user is asked questions about it. In discovery design, the program helps the user reach insights on his or her own. In EGRUL, the logic leads from examples (EG) to rules (RUL) about the particular information (6:98). RULEG, conversely, leads from rules to examples. In a fading design, interactions between the program and the

user decrease as the lesson progresses. This design is "particularly useful for content which has to be memorized" (6:99).

APPLICATION TO THE PROJECT: The tutorial developed during this research took advantage of the points raised above. The four insights presented by Jones and Steinberg (27:201) were closely observed. First, visual cues, including capitalization and underlining, are used as replacements for the aural cues found in speech. Second, procedures for using the tutorial are kept simple and are explicitly displayed on every screen. Third, necessary keystrokes, including cursor movement using the cursor keys and item selection using the <Enter> key are consistent with established computer interface standards. Fourth, the tutorial is structured in a hierarchical framework that allows users to select the depth of information that fits their particular needs.

Burke's two CAI design concerns, physical and logical, were both considered (6:95-97). The physical design of the tutorial is a branching design which allows users to personalize the instruction to meet their needs. The logical design of the tutorial is "RULEG" (6:99), leading from rules about concepts to examples of the concepts.

Advantages and Disadvantages. Several advantages of CAI over instructor-based instruction are commonly cited. First, CAI provides consistent training. The same material is presented each time the CAI program is used. "Success no longer depends on the competency of individual instructors" (7:4).

The second advantage is that with CAI, the user is able to proceed at his or her own pace (50:19). The individualized nature of CAI allows the user to match the lesson to his or her own learning pace (40:5). By permitting the user to set the pace, CAI ensures the proper amount of

time is allotted for learning. "Overtraining and undertraining aren't issues when the learner sets the pace" (7:4).

A third advantage is the immediate and personalized feedback that can be offered through CAI (50:19; 7:4; 40:6). Since the lesson in CAI involves interaction between the user and the computer, it can monitor user responses and remediate or advance the current user as appropriate based on his or her needs. Further, CAI ensures "guaranteed mastery. Trainees cannot complete the lesson until they have successfully demonstrated the required skill or knowledge" (7:5).

The fourth advantage of CAI is the reduced training time required. According to Wher, CAI "generally takes 40 percent less time than instructor-led training" (50:19). Research into this area will be addressed below in the Previous CAI Research section.

Matta and Kern suggest the following benefits of CAI over instructor-led training: "students learn faster and more comfortably; students can be presented with richer and more complex problems; and [CAI provides] better measurement of student progress" (37:7-78).

There are also inherent limitations to CAI. Orwig addresses the following three limitations (40:7-8). First, a computer is NOT a human. The computer "cannot replicate all the subtle attributes of a well-trained, ambitious instructor" (40:7). Second, the CAI program is text-based and thus highly verbal. As such, it lends itself to highly verbal instructional tasks and, conversely, is not as efficient with nonverbal instruction. Third, development of CAI is a time-consuming task. The benefits of a computer-assisted program must balance the efforts taken to produce it.

APPLICATION TO THE PROJECT: The CAI tutorial developed during this research has several potential advantages. By providing students with the CAI tutorial, instructors can devote more classroom time to other, more detailed concepts that require direct instructor-student interaction. Further, the tutorial allows users to set their own pace through the material and concentrate on those areas that meet their needs.

The tutorial avoided some of the disadvantages addressed by Orvig (39:7-8). First, it does not attempt to replace instructor-led instruction in grammar, but rather supplements that instruction. Second, since grammar is a text-based and verbal area of instruction, CAI is indeed an appropriate method for delivering the instruction. However, development time for the tutorial became an issue for this project during the validation phase of development.

Types of CAI. Computer-assisted instruction can take many forms. Generally, the various forms of CAI are grouped into the following types:

1) informational,

2) drill-and-practice,

3) tutorial,

4) simulation, and

5) intelligent CAI (29:19-24; 40:11-13).

Informational CAI is the most basic form of CAI. It is a one-way process of providing information to support the instructional process. Kemner-Richardson et al. suggest that "information CAI can be used as a dynamic tool, with information such as helpful hints or lessons learned added by instructors at any time" (28:19).

Drill-and-practice CAI adds a dimension of limited interface into CAI. It is "used to practice and refine an emerging concept" (40:13) through repetition of question-and-answer modules. Drill-and-practice is "often provided at the subtask level" (28:21) to reinforce portions of a larger concept. It emphasizes basic knowledge through review and reinforcement (34:27).

In tutorial CAI, the computer presents information about a concept and then tests the user's understanding of it. It emphasizes knowledge acquisition and comprehension (34:27). "Immediate feedback messages and schemes . . . are, in good measure, what make the lesson a tutoring experience" (28:21).

The types of CAI tutorials include linear and branching tutorials. Linear tutorials are sequential lessons used to teach an entire concept. Each student who uses a linear tutorial follows the same path through the material from beginning to end (40:13). Branching tutorials are also used to teach an entire concept, but these "involve branching to accommodate a variety of learners" (40:13). Branching tutorials are often presented in a hypertext-like environment, allowing the user to access related information at will. Hypertext is discussed below in the section on programming a tutorial.

Simulations are computer-based alternatives to real-life experiences that are used to build knowledge and skills of the user (40:11). Simulations are well-suited for situations when they can help to reduce the costs, time, or potential hazards of the actual process being modeled (28:23).

A final form of CAI is intelligent CAI (ICAI). ICAI, which is still in the development stages, is intended to model the "one-on-one

interaction of an instructional dialogue" (28:24). ICAI encompasses three components that must be imbedded within it: subject-matter expertise, an understanding of the user's current knowledge level, and the rules for conducting an instructional experience. At the present time, "ICAI is not a fully mature technology" (28:24).

APPLICATION TO THE PROJECT: The branching tutorial was adopted for this project. The user follows the tutorial's branches through the program's hypertext interface.

CAI Development. Steinberg suggests a three-phase plan for the development of CAI: initial planning, unit planning, and lesson completion and unit integration (44:20).

In the initial planning phase, the target population must first be characterized. Burke refers to this process as the "front-end analysis" (6:40). The developer must determine who will use the CAI, what prerequisite skills are necessary, and at what level it will be targeted. Also in the initial phase, goals must be developed for the user, computer, and each lesson; expected outcomes must be anticipated; and evaluation measures must be selected. Burke labels this step "outcome specification" (6:41).

In the unit planning phase, each individual unit is planned, developed, and evaluated. Response and feedback options are developed and the computer-user interface factors are developed and refined as necessary (44:21). Burke suggests the development of flow charts as a method of ensuring the lesson flows as intended. A system flow chart should give the "big picture" of the tutorial and a detailed flow chart should give an in-depth description of each lesson (6:47). Each lesson should be created in a paper version first, to ensure the material is

properly presented, and then each lesson should be coded, tested, and debugged (6:50). An effective means of validating the final lesson is to use a pre-test/post-test method (6:53).

Orwig asserts that the most successful technique for creating the paper version of a lesson is to develop story boards on 4"x6" cards. This "allows you to visualize the actual content and flow of the instruction before you begin committing it to a programming language" (40:49). Figure 1 illustrates the format that Orwig suggests for the story board.

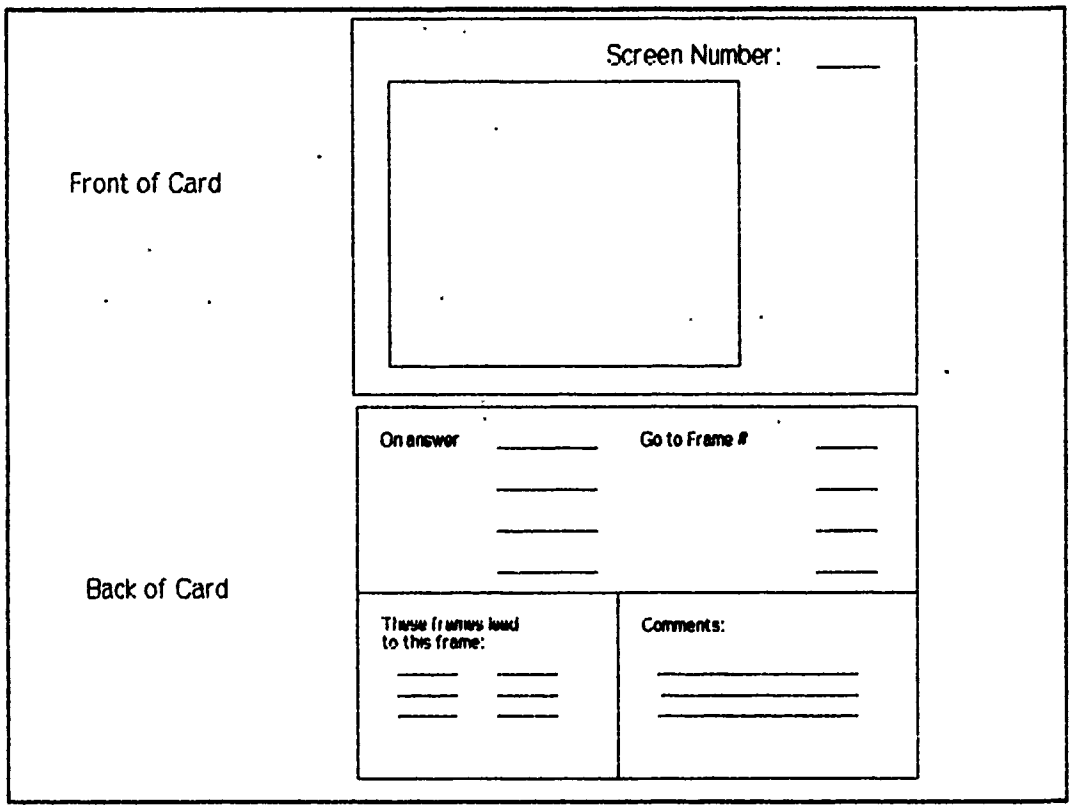


Figure 1. Story Board for CAI Development (40:50)

In Steinberg's third phase of CAI development, lesson completion and unit integration, the final "management and human factors decisions" are made (44:21). Introduction, transition, and conclusion screens are

developed, evaluation measures are developed and inserted into the program, and the lessons are integrated into a complete product. Finally, the complete product is evaluated, revised, and documented.

APPLICATION TO THE PROJECT: The development of the tutorial followed Steinberg's three-phase plan of initial planning, unit planning, and lesson completion (44:20). The development process will be discussed in detail below in the section on program development.

Computer-User Interface. The computer-user interface is the method through which the computer and the user communicate with each other. The design of the computer-user interface is an important consideration in developing CAI. Jones and Steinberg cite its importance: "In a CAI lesson, you want a student to devote full attention to learning and not be encumbered by the mechanics, such as entering responses and moving about" (27:203).

Williams et al. evaluated five categories of human design factors that have been identified by Hamel and Clark (51:1-2).

1) Brevity: the interface "should minimize the amount of information to be attended to" (51:1).

2) Consistency: The format and screen locations of information should be kept constant, so the user will know what is expected.

3) Flexibility: The user should be able to determine the pace and sequence to fit his or her needs.

4) Compatibility: Information should be presented in a manner expected by the user.

5) Responsiveness: The user should be offered appropriate and timely feedback. Also, the program should help the user to understand how the program works.

Williams et al. concluded from the results of their research that "each of these categories . . . significantly contributed to the ease of use" (51:17).

Jones and Steinberg offer three tools that will enhance computer-user interface. First, reasonable keys should be used to facilitate communications and avoid frustration. For example, use "T and F," rather than "1 and 2," to represent "True and False" (27:202). Second, they suggest consistent use of keys. The same keys should be used throughout the program to represent the same function (27:203). For example, if the <F1> key for "help" in one portion of the program, it should not be used for a different function elsewhere in the program. Third, they suggest adequate screen displays be used. The fact that something is on the screen does not necessarily mean the user will see it. "Arrange the screen so that different kinds of information are presented in clearly identifiable, separate locations" (27:203).

APPLICATION TO THE PROJECT: Through its hypertext environment, the tutorial's computer-user interface is kept simple. All actions are performed using only the cursor keys, the <Enter> key, the <Pg Up> key, and the <Esc> key. Further, the actions performed by these keys are consistent throughout the program. Finally, the function of each of these keys is listed at the bottom of every screen.

Screen Design. The essential element of the computer-user interface is the screen design. Hannafin and Hooper define screen design as "the purposeful organization of presentation stimuli in order to influence how students process information" (23:156).

Hannafin and Hooper present a framework for evaluating screen design decisions, addressing "the five functional requirements of

instruction: Retrieving, Orienting, Presenting, Encoding, and Sequencing (ROPES)" (23:159).

These five factors must be considered and evaluated to ensure effective screen design decisions have been made. Retrieving is concerned with the transfer of the information from the computer to the user. The screen design should attempt to "optimize integration of lesson content within [the user's] existing schemata" (23:159). The screen design should also help Orient the user within the program. If the screens are kept consistent throughout, the user will have a better idea of what to expect and what is expected of him or her (23:160). The Presentation aspects of screen design are "manifested throughout the other phases of the model" (23:159) and are thus not discussed separately. The screen design should help the user to Encode the information from short-term to long-term memory. This can be facilitated through outlines, headings, and segmented text units (23:161). The final factor, Sequencing, suggests that the rate of presentation should be tailored to the abilities of the current user and to the cognitive complexity of the material (23:162). These five functional requirements of instruction, contained within the ROPES model, should each be considered and evaluated before finalizing decisions about screen design.

The basic unit of screen design is the frame. Orwig defines the frame as "a single page of the instructional program. For CAI, the frame is one 'screen' of the program" (40:41). The frames of a CAI program are identified by their particular function. Selfe lists six general types of frames (41:49-51). Introductory frames set the tone for the lesson by providing initial information about the program,

including its purpose and expectations. Selfe suggests that introductory frames should be kept to a minimum (41:49). Orientation frames locate the current lesson, showing where it lies within the framework of the entire program. Help frames are used to explain commands, menus, keystrokes, and other features. These frames should be optional, but accessible as needed (41:49). Menu frames provide the options available to the user. Instruction frames contain the material that is to be taught. Orwig refers to these frames as the "heart of the instruction" (40:41). Exercise and testing frames evaluate the user's retention of the material and ensure the objectives of the lesson have been reached.

The personal computer offers programmers the option of including graphics (or pictures) as part of the CAI program, in addition to standard text. Dossett and Konczak reviewed three studies which examined the effectiveness of adding graphics to text-only CAI programs (13:41-43). Based on their evaluation of the results of those studies, they concluded that "when acquiring factual knowledge is the objective, adding visuals to CAI has little benefit" (13:43).

Jones and Steinberg offer three suggestions for designing the display of frames. First, the display should draw attention to important messages by setting them apart from the rest of the text (27:109). Second, the display should be laid out in such a way as to encourage the student to view the entire frame (27:111). This helps to prevent missing key information. Third, The material should be arranged in a manner that will help the user understand and remember the information (27:112). Jones and Steinberg offer the following as their "cardinal principle" of frame displays: "Keep the display uncluttered.

The lesson should be displayed so as to facilitate learning and not to impeded it in any way" (27:112).

APPLICATION TO THE PROJECT: The factors addressed above were considered throughout the tutorial development phase of this project. The design of the tutorial is discussed in detail below in the section on development of the tutorial content.

Validation of CAI. Validation is the important follow-up step after a CAI program has developed. Burke states

> It is imperative that your lesson be validated; that is, tested with a group of students representative of your intended student population and then revised until it works properly. (6:53)

Burke suggests that the CAI program can be used to validate itself if it includes a pretest and posttest. The pretest should test comprehensively for knowledge of all the material in the program. "Remember, you are not testing the student, you are testing the lesson" (6:53). To properly test for validity, the posttest should bear a one-to-one relationship with the pretest; however, it should not be identical to the pretest.

When validating a CAI program, the important factor to consider is its training effectiveness. According to Dossett and Monczak, "Training effectiveness is the degree to which training objectives are realized" (13:41). Matta and Kern offer three measures of training effectiveness that have been applied in previous research into CAI validation (37:82). "Achievement" is the degree to which the user understands the material that was presented. This can be tested immediately upon completion of the program. "Retention" is the degree to which the user retains the

material and should be tested after a delay in time. "Student attitude" is a measure of the users' feelings about their experience with the CAI.

An effective way to validate CAI is to field test the program. Burke suggests that "field testing should be conducted under conditions which are as realistic as possible" (6:54). Selfe identifies seven facets of CAI that should be validated through field testing:

1) Instructional Objectives: How well are the objectives met?
2) Lesson Attractiveness: How did the users "like" it?
3) Ease of Use: How easy is it to start, use, and navigate?
4) Classroom Fit: Does it fit into the class program?
5) Content: How do experts evaluate the content of the CAI?
6) Biases: Does the language of the CAI contain any biases?
7) Surface Features: Does the CAI contain good grammar, usage, and correct spelling? (41:121-22)

APPLICATION TO THE RESEARCH: Initial validation of the tutorial was achieved through field testing it with students in AFIT's refresher course in fundamental written communication. The seven facets of CAI identified by Selfe were addressed in the validation process. This process will be discussed in detail below in the section on program evaluation.

Previous CAI Research. The literature contains numerous studies into the effectiveness of various facets of CAI.

In an often-cited meta-analysis of CAI, Kulik et al. evaluated the results of 24 CAI studies. They concluded that "CBE [computer-based education] has basically positive effects on adult learners" (31:248). Their most significant finding was in the area of instruction time. They found that out of 13 studies reporting times, 12 of them reported

that CAI lessons were completed "on the average in about 71 percent of the time for conventional teaching methods" (31:249).

Fryer conducted an extensive literature review with two goals: "To compile information about CAI into a single source . . . [and] . . . to study current CAI technology in order to determine the suitability of CAI to Air Force Civil Engineering" (18:1-1). His findings led him to five conclusions (18:6-1,2). First, the literature indicated that CAI "as an instructional methodology, was a viable method of instruction under the right condition" (18:6-1). Second, the literature did not contain sufficient information relating to the costs of CAI. Third, a variety of environments exist for the development of CAI, including higher order languages and authoring systems. Fourth, the most important factors involved in choosing hardware and software for CAI "are the lesson objectives and the future needs of the instructor and student" (18:6-2). And fifth, few present applications meet the needs of Civil Engineering training.

Colorado conducted a critical assessment of CAI research. He found that the research designs used most often in the literature include case studies and correlational, causal, comparative, and experimental studies. He suggested that other possible research designs include research reviews, meta-analyses, and evaluation studies of particular systems (10:227).

Colorado's assessment of these various research designs led him to conclude that their flaws fall into three categories. The first category are "problems related to the general organization and clarity of the actual research report" (10:228). These problems are generally caused by poor research of the literature and poorly stated procedures.

The second category contains "problems related to internal and external validity" (10:227). These result from non-random selection of subjects, questionable use of volunteers, the Hawthorne effect (the subjects' awareness of the test affects the results), and the novelty effect (the novelty of the CAI affects the results). The third category contains "problems related to the CAI research approach and to the definition of the problem" (10:228). He cites numerous factors that can influence and confound CAI research. These include

1) whether the CAI is supplemental or the primary training method,

2) whether the CAI is developed from a higher-order language or from an authoring system,

3) what type of CAI is used,

4) what the subject matter was, and

5) what the duration was.

Colorado found researchers who "were not careful enough to differentiate among these factors, thus allowing for the possibility of alternate explanations to their research findings" (10:230).

Grabe et al. conducted two replicated studies to evaluate the effects of allowing volunteer students to supplement their studies (educational psychology) with CAI study material. In the experiments, the four chapters of course material were divided into two chapters supplemented with CAI and two chapters without CAI. The authors then compared the results of the effectiveness of the CAI-supplemented chapters against the other chapters using an examination covering all four chapters. In both studies, they found that "students performed better on the chapters they were able to study with the aid of the computer-assisted technique" (22:112).

Dossett and Hulverson conducted an experiment comparing three groups of Air Force personnel learning basic electronics: non-CAI trained, individual-CAI trained and two-person-CAI trained (12:553). They found that both CAI groups had mean training times significantly lower (37 percent and 49.2 percent, respectively) than the non-CAI trained group (12:555). They concluded from their study that "a staff of specialists in educational psychology and computer technology is not required for the development of effective CAI training materials" (12:558). A further conclusion was that "extremely low student-instructor contact time" can help with limitations of space and instructors (12:558).

Sivley conducted an experiment to assess the feasibility of developing a CAI program for the Wright-Patterson Air Force Base Supply Squadron's Equipment Custodian Management Block III Course (42:2). He compared examination results of a group trained using the CAI course with a group trained by an instructor. Sivley found that the group trained using CAI required 67 percent less total training time to complete the course (42:57). Additionally, the students trained with the CAI course averaged seven percentage points higher on the examination than the students in the non-CAI course (42:58). From his findings, he concluded that "CAI--in terms of learning levels and time savings--is a feasible method of instruction . . . [for the equipment custodian course]" (42:59).

Wher cites a study by the Comptroller of the Currency, where an instructor-led course for assistant bank examiners was replaced with a CAI course. Examiners who had taken the old course were asked to take the new CAI course and to rank their preference. On a scale of 1 to 5,

with "1" representing a preference for the instructor-led course and "5" representing the CAI course, the average rating was 4.2, which suggests that the CAI course was preferred over the instructor-led course (50:18).

Neapolitan conducted an experiment to test the effectiveness of "CAI software which requires little or no instructions to run, does not tie up the computer for very long and teaches the student without help from the instructor" (38:493). He designed an easy-to-run tutorial program that presented information, then asked questions about the information presented. If incorrect answers were given, the program gave the student the correct information. After completing the tutorial, the subjects were given a test. The experiment consisted of two groups of 40 subjects. One group was given the tutorial, while the other was given the written equivalent. Both were tested on the material and given a subjective questionnaire to gauge their feelings about the subject and the teaching method. Neapolitan found that the test scores of the subjects with the CAI tutorial were significantly better (at the .05 significance level) than the test scores of the other group (38:494). He further found that the CAI group had "substantially higher levels of feeling subjectively that they understood the theory and that the teaching method was effective" (38:495).

APPLICATION TO THE PROJECT: The most significant impact of the previous CAI research on the current research is that it provided a conceptual framework in which to develop the tutorial. The various factors that Colorado suggests lead to flaws in CAI research (10:228) were considered throughout the process: for example, the tutorial was developed as a supplemental tool, rather than a primary training method.

As such, the research was similar to the study conducted by Grabe et al. (22:112). Likewise, the tutorial's software was developed so that it requires very little instructions, similar to the "stand alone" research conducted by Neapolitan (38:493).

Programming a Branching Tutorial. Galagan cites Gery, whom she calls "a leading critic and advocate of CBT" (19:74), on the structure of CAI: "An ideal control structure . . . would be a relational database, where every piece of information and all the relationships associated with it are accessible" (19:75). The users should be able to move among the related information without having to return to the main menu to find their place again. In other words, the ideal structure for CAI would seem to be a structure provided by a programming approach known as hypertext.

"The essence of hypertext is the dynamic linking of concepts, allowing the reader to follow preferences instantaneously and to be in control" (2:158). This movement from one topic to another related topic distinguishes hypertext from traditional, linear text documents and tutorials which are structured from top to bottom.

Hypertext has potential applications in expert systems, decision support systems, and particularly training. Anderson-Inman suggests, "The exploratory nature of hypertext documents not only facilitates learning but also increases the reader's awareness of a personally effective approach to the learning process" (1:53).

The following discussion of hypertext will include

1) a description of hypertext,

2) advantages and disadvantages of hypertext,

3) avoiding the disadvantages, and

4) programming hypertext.

The review was limited to hypertext for microcomputers (although other applications are discussed in their historical perspective). Also, the broader topic of hypermedia, which links other media, "such as graphs, video, spreadsheets, animation, and voices" (43:33), is addressed only by implication.

Description of Hypertext. As stated earlier, hypertext dynamically links concepts, allowing the reader to follow preferences instantaneously and to be in control (2:158). The reader can view a particular reference and be returned to the original place in the document. Oppenheim offers Christopher Locke's definition: "Hypertext allows you to explicitly link information that is conceptually related" (39:87).

Franklin describes hypertext as associative because "a single word or an entire document can be linked to another word or group of words in any way that is meaningful to the user" (16:37). Hypertext is "an Associative Information Management System . . . that augments human associative thought and memory" (16:37).

Hypertext is distinguished from traditional text documents because it adds a third dimension to the document. Traditional text documents are read two-dimensionally, from beginning to end. Hypertext enables a third dimension through its use of linked concepts, normally hidden, that can be accessed at will by the user (17:96).

Two key features of the hypertext document are links and threads. By selecting a particular link, the user is presented the associative information. Carr refers to the links as "hot buttons," which lead the

user to more information about the selected concept (9:8). Anderson-Inman describes three types of links. Referential links connect a specific concept with related concepts; organizational links provide a hierarchical structure; and key-word links scan the document for the same key word (:.:21-2). As the user activates various links and moves throughout the document, the system creates a thread which marks the path taken. This allows the user to back up using the same path, or alternatively, to back up part way and take another path (9:8).

Advantages and Disadvantages of Hypertext. Hypertext has several advantages over traditional text documents. Most importantly, it allows the user to adapt it to his or her specific needs. Since the links are only activated upon request, the user who does not need the detailed information can continue without accessing it (9:9). Further, hypertext allows documents to be viewed from many different perspectives. "Readers can browse the information according to their level of familiarity and interest" (8:60).

Hypertext offers an alternative to "tables of contents, indexes, cross-references, see references, footnotes, and glossaries" (30:63). It allows the user to instantly access information for which he or she would otherwise have to search. "It is this element of variety, of flexibility, of alternative approaches to our normal print environment that is so appealing" (30:63).

Marchionini sees advantages in the new approaches to learning that hypertext offers. As "an enabling rather than directive environment, offering unusually high levels of learner control . . . it offers new ways of learning how to learn" (35:9). Further, it "offers the potential to alter the roles of teachers and learners and the crucial

interaction between them" (35:9). Teachers and students will learn from each other through the varied interpretations of the information.

The nature of hypertext, both as a computer application and as a non-linear document, leads to several potential problems that authors must consider.

The biggest problem the designer faces is the potential for disorientation (1:27; 35:11; 43:39). Since hypertext documents do not have the familiar signposts of tables of contents and indexes, users who are new to the hypertext environment "may get lost in hyperspace as a result" (2:161-2). The number of links and potential paths could lead to disorientation if the designer does not provide a backbone, or hierarchical structure to the document.

Bevilacqua refers to the potential for "cognitive overload" from the variety of paths and choices offered. A navigation problem may occur if the user attempts to follow too many paths at once. "This problem is not new with hypertext, though computerized access does add a sometimes overwhelming dimension to it" (2:161).

A drawback to user-control is that the user "may choose to allocate his time in inefficient ways" (9:10). Anderson-Inman suggests research be conducted comparing guided studying and learner-controlled studying, especially for elementary and secondary education (1:27).

Manes perceives the major problem with hypertext to be the creation of the initial links (33:91). Carr agrees: "All of the links have to be programmed in . . . a time-consuming task for the developer, but if the link isn't programmed in, it doesn't exist" (9:8).

Printing a hypertext document presents problems that the traditional text document does not face. Two-dimensional documents can be

printed just as they are read, from beginning to end. The hypertext document cannot be printed as easily, because the links and paths do not follow any particular order. When printing a hypertext document, the writer must decide for the reader how much associative information should be included and where it should be located in the text (17:98).

Avoiding the Disadvantages of Hypertext. Horton suggests several solutions to the "lost in hyperspace" problem of hypertext (26:21). First, the document should have a "home" or base topic, from which the rest of the document stems, and a simple action to return to that topic. This provides a familiar starting point to the users, to which they can return as necessary. Second, when the user is ready to return to previous screens in the document, there should be an automatic capability to backtrack through those screens. This relieves the users of the task of determining from which screens they had come. Third, topics should be numbered and accessed by those numbers. Horton suggests that it is easier for users to remember numbers than words. Fourth, the document should consist of predetermined paths or trails down which the user will read. While the users should be allowed the capability to branch to related topics, a predetermined, backbone structure will help the user traverse the document more efficiently. Finally, the backbone structure should be a grid, hierarchy, or sequence, rather than a web-like structure. The former structures help to prevent the users from becoming lost in the document, whereas, in a web-like structure, the users could become confused as to their current location in the document.

APPLICATION TO THE PROJECT: Horton's suggestions for avoiding the "lost in hyperspace" problem of hypertext (26:21) were all considered in

the program development phase of the tutorial. "Home topics" are represented in the form of menus from which the users branch out and to which they return. Automatic backtracking through previous screens is achieved by simply pressing the <Pg Up> key. And finally, the backbone of the program is the hierarchical structure of a branching tutorial.

Programming Hypertext. The most difficult aspect of programming hypertext is the development of the mechanism which allows the user to jump from screen to screen through a series of links in the text. A variety of methods to program this capability have been offered in the literature. The following discussion will address two of the methods.

King presents a method by which the links are established only when the document is displayed (29:34). His method, developed for the Turbo C programming language, involves associating a key phrase with each concept. When this key phrase appears in other concepts, the program automatically creates a link to the key phrase's concept. This approach is useful if the document will be continually expanded, since links are not preprogrammed into the document. The approach "allows you to build up your hypertext document either by adding topics when you need them or when you find that certain words need elucidation" (29:34). This allows the author to concentrate on writing the document rather than managing the document's links (29:34).

Gessner suggests another method for developing a hypertext document (20). In Gessner's method, developed for the Turbo Pascal programming language, every screen in the document can be accessed by associating its page number with a key word on any other page. This association is a hypertext link which is embedded within the document at

the time the document is created. Using this approach, the document can continually be expanded, simply by adding new pages.

The programming of a hypertext document involves two distinct processes. The first process consists of creating the document and identifying the links. The second process consists of displaying the document for the user. In King's approach, the document is created in a word processor. Each topic is distinguished by a form feed command followed by its key phrase. When the document is displayed, using King's hypertext display program, links are created whenever the program encounters a key phrase in the text. In Gessner's approach, the document is created using a hypertext editor which allows the user to identify key words and link them to othe. pages in the document. The document is then displayed with the links already created.

APPLICATION TO THE PROJECT: The tutorial for this research was developed following Gessner's approach to hypertext, primarily because it was developed for Turbo Pascal, which was the programming language chosen for this project. Additionally, the hypertext editor and display programs were readily adaptable to the requirements of this project.

Literature Review: Summary. The information covered in this literature review had a direct influence on the remainder of the research. The CAI literature provided insight into the structure, development, and validation of CAI software. It also pointed to hypertext as the most useful non-linear programming technology with which to develop a branching tutorial. The hypertext literature provided an in-depth understanding on the advantages of hypertext and the pitfalls to avoid. It further provided the necessary knowledge to

develop the computer program code for the hypertext portions of the tutorial.

Survey of Subject-Matter Experts

Before developing a tutorial to help students to better understand English grammar, it was necessary to consult experts who would have a direct input into the subject matter. Specifically, the professors of communication at the Air Force Institute of Technology (AFIT) have first-hand knowledge of the types of grammatical errors commonly committed by AFIT students in their written communication. Additionally, to broaden the perspective of the research, professors in the Department of English, University of Dayton (UD), were consulted for their input based on their experience with errors commonly committed by undergraduate students in their written communication. This direct input from experts in the area of written communication would provide the research with a strong base of information from which to develop the tutorial.

Questionnaires were distributed to seven professors of communication at AFIT and sixteen professors of communication at UD. Six of the seven AFIT professors responded to the questionnaire and fifteen of the sixteen UD professors responded. The questionnaire and the aggregate responses to each question are at Appendix B. The responses are shown totalled for each institution and combined for an overall total.

Frequency and Severity of Errors. This section of the questionnaire contained several examples for each of eight types of common grammatical errors compiled from four sources (See Appendix A).

These eight errors are

1) sentence fragment,

2) fused sentence,

3) comma splice,

4) subject-verb agreement error,

5) pronoun reference error,

6) misplaced modifier,

7) illogical verbal modifier, and

8) parallelism error.

The purpose of this section was to allow the subject-matter experts to provide direct input into the type, frequency, and severity of the errors commonly committed by students in their written communication.

The questionnaire asked the respondents to indicate, for each example, how often the error occurs in students' writing. The questionnaire provided a five-point scale ranging from "never" to "very frequent" from which to choose. Respondents were also asked to indicate the severity of the error, with severity defined as "how seriously the error degrades effective communication."

The responses given by the subject-matter experts indicate that all of the eight errors are committed by students in their written communication. Of the eighteen examples of the errors given, only one example (2a-fused sentence) received an occurrence response of "never." This response was given by one AFIT professor. Additionally, of these eighteen examples, only one example (8b-parallelism) was labelled "not a problem." This response was given by one UD professor. The remainder of the occurrence responses ranged between "seldom" and "very frequent." The remainder of the severity responses ranged between "minor" and "very

severe." As expected, the AFIT professors, whose students are at the graduate level, indicated a lower frequency of the errors than the UD professors, whose students are at the undergraduate level. The severity responses of both groups were similar.

APPLICATION TO THE RESEARCH: The purpose of this section of the questionnaire was to provide the research with a base of information from which to develop the tutorial. The relative importance of each example was determined by combining the frequency and severity data as follows:

1) MAJOR IMPORTANCE: total frequency responses of "often and very often" multiplied by total severity responses of "severe" and "very severe"

2) MODERATE IMPORTANCE: total frequency responses of "occasionally" multiplied by total severity responses of "moderate"

3) LEAST IMPORTANCE: total frequency responses of "never" and "seldom" multiplied by total severity responses of "not a problem" and "minor"

The results of this compilation are shown at Figure 2. As the figure shows, all of the examples are of either major or moderate importance. Thus, each type of error should receive attention in the tutorial.

Chronology of Teaching. In this section of the questionnaire, the respondents were asked to indicate the order of instruction most appropriate for the listed grammatical concepts. The list included the eight grammatical errors, plus "parts of speech" and "clause recognition." The average responses and standard deviations of those responses are shown in Appendix B. The order suggested by the average responses is shown in Figure 3. The experts all agreed that the first grammatical concept that should be taught is "parts of speech." Although the order of the individual errors varied slightly, the next

	Strong Importance	Moderate Importance	Least Importance
Example 1a	42	<u>55</u>	6
Example 1b	<u>66</u>	36	6
Example 2a	42	<u>66</u>	0
Example 2b	<u>90</u>	56	8
Example 3a	<u>120</u>	84	2
Example 3b	<u>112</u>	38	0
Example 4a	<u>98</u>	66	0
Example 4b	<u>91</u>	36	16
Example 4c	78	<u>126</u>	10
Example 4d	16	<u>28</u>	22
Example 5a	<u>150</u>	21	2
Example 5b	<u>90</u>	44	8
Example 6a	12	<u>104</u>	7
Example 6b	48	<u>55</u>	15
Example 7a	<u>168</u>	63	0
Example 7b	88	<u>96</u>	0
Example 8a	88	<u>108</u>	0
Example 8b	45	<u>110</u>	4

Strong Importance = (Often + Very Often) X (Severe + Very Severe)

Moderate Importance = Occasionally X Moderate

Least Importance = (Never + Seldom) X (Not a problem + minor)

The high score for each example is underlined.

Figure 2. Combination of Frequency and Severity Data

two major areas in the chronology were clause recognition and sentence structure. In all cases, problems with modifiers and parallelism were placed last in the chronology.

AFIT Professors	UD Professors	Combination
Parts of Speech	Parts of Speech	Parts of Speech
Sentence Fragment	S-V Agreement	S-V Agreement
Clause Recognition	Sentence Fragment	Sentence Fragment
Fused Sentence	Clause Recognition	Clause Recognition
S-V Agreement	Pronoun Reference	Fused Sentence
Comma Splice	Fused Sentence	Pronoun Reference
Pronoun Reference	Comma Splice	Comma Splice
Verbal Modifier	Verbal Modifier	Verbal Modifier
Misplaced Modifier	Misplaced Modifier	Misplaced Modifier
Parallelism	Parallelism	Parallelism

Figure 3. Suggested Teaching Chronology

APPLICATION TO THE PROJECT: During the tutorial content development phase of the research, the suggested chronology was considered and played an important role in narrowing the focus of the tutorial. Parts of speech, subject-verb agreement, clause recognition, and sentence structure are each covered in the tutorial. Due to the time constraints of the project, the last three areas in the chronology suggested by the subject-matter experts (illogical verbal modifiers, misplaced modifiers, and parallelism) were excluded from this version of the tutorial.

Teaching Experience of the Experts. This question asked the respondents to indicate the number of years they have taught communications, English, or composition. This information was requested in order to establish the relative authority of the subject-matter experts. The responses to this question were:

1) AFIT Professors: Average experience of 19.33 years

2) UD Professors: Average experience of 18.92 years

3) Combined: Average experience of 19.05 years

APPLICATION TO THE RESEARCH: The professors and teachers participating in this phase of the research averaged over 19 years of experience teaching communications, English, or composition. Hence, their responses could be considered expert opinions in the area of grammatical concepts.

Suggested Textbooks. This question was asked to determine the textbooks that the respondents suggest are most appropriate for helping students learn grammatical concepts. The textbooks suggested by the subject-matter expert are listed in Appendix B. The most commonly suggested textbooks were Simon & Schuster (47), Harbrace (25), and Prentice-Hall (36).

APPLICATION TO THE PROJECT: The Harbrace (25) and Prentice-Hall (36) books were among the sources used to develop the questionnaire distributed to the subject-matter experts. Their appearance on the list of suggested textbooks validates that decision. The Simon & Schuster text (47) was not initially consulted, but was subsequently obtained and used as a reference during the development of the tutorial.

Survey of Subject-Matter Experts: Summary. The survey of subject-matter experts provided the research with an authoritative base of information from which to begin development of the tutorial. With an average of over 19 years experience in communication, English, or composition, the professors and teachers who responded to the questionnaire provided important insights into the instruction of grammatical concepts. The textbooks suggested by the respondents were valuable references used throughout the development of the content of the tutorial.

Development of the Tutorial

The tutorial development phase of the research consisted of two distinct facets: development of the tutorial content and coding of the program software.

Development of the Tutorial Content. Development of the tutorial content began immediately after the data had been collected from the subject-matter experts and followed Steinberg's three-phase plan of initial planning, unit planning, and lesson completion (44:20). Steinberg's plan was described above in the section on CAI development.

Initial Planning. Initial planning for the tutorial first led to the development of a plan consisting of a series of lessons designed to address each of the areas identified by the subject-matter experts. In this plan, the following lessons would be included:

1) Parts of Speech,
2) Clause Recognition and Sentence Structure,
3) Sentence Structure Errors,
4) Subject-Verb Agreement,
5) Pronoun Reference,
6) Problems With Modifiers, and
7) Parallelism.

Based on this plan, the first three of these lessons were written, using the textbooks suggested by the subject-matter experts as reference tools. After evaluating these first three lessons, it became evident that the method of instruction found in the reference books and subsequently transferred into the lessons failed to deal properly with the circularity and interconnectedness of the concepts of grammar. For example, to understand what a clause is, one must understand the

relationship between a subject and verb, but the concept of a relationship between a subject and verb can only be understood in terms of clauses. So rather than dealing with these two concepts as separate parts of the tutorial, they must somehow be taught simultaneously.

Another problem that surfaced during this evaluation was that, by following the traditional method of grammar instruction as found in the resource textbooks, the tutorial would assume that each student already had a basic level of understanding of grammar, since each lesson required the user to have some understanding of the fundamental concepts behind it. It would not be useful as a tool to introduce grammar concepts.

An additional concern with this first plan was that it failed to exploit the many benefits that instruction in a hypertext environment can achieve. The hypertext environment allows users to jump from one concept to another related concept as necessary for better understanding of the concepts.

This evaluation of the first plan for the tutorial indicated that the traditional method of instruction should not be used in the tutorial. Instead, the tutorial would start with the simplest concepts of grammar and build upward to the more difficult concepts. Additionally, rather than forcing the user to memorize definitions and lists of examples of the various grammatical concepts, it would teach users to recognize concepts and patterns based on two ideas: the building elements in grammar and the functions that can be performed by these building elements.

In English grammar, there are three building elements that are used to form sentences:

1) words,

2) phrases, and

3) clauses.

These three building elements perform the following five functions in English grammar:

1) naming things,

2) showing action or existence of things,

3) modifying (changing, qualifying, or renaming) things,

4) connecting other words or groups of words, and

5) adding filler to sentences.

The grammar tutorial would, thus, be written in such a way that it helps its users to recognize the three building elements and the five functions that they perform, and to understand how sentences are formed from the building elements. To guide this new approach, Faulkner's text was added a reference since his textbook takes a functional approach to grammar (14).

Unit Planning. The unit planning phase consisted of developing an in-depth outline for the tutorial. The backbone structure of the tutorial is an inverted tree that begins at the top and branches out to other ideas as the user descends the tree. Figure 4 illustrates the basic backbone structure of the tutorial developed during the unit planning phase.

The user selects one of the four options from the main menu. The main menu suggests to first-time users that they should review the areas sequentially, beginning with the introduction. The second branch presents the five functions that are performed in language. The third branch presents the three building elements and further branches out

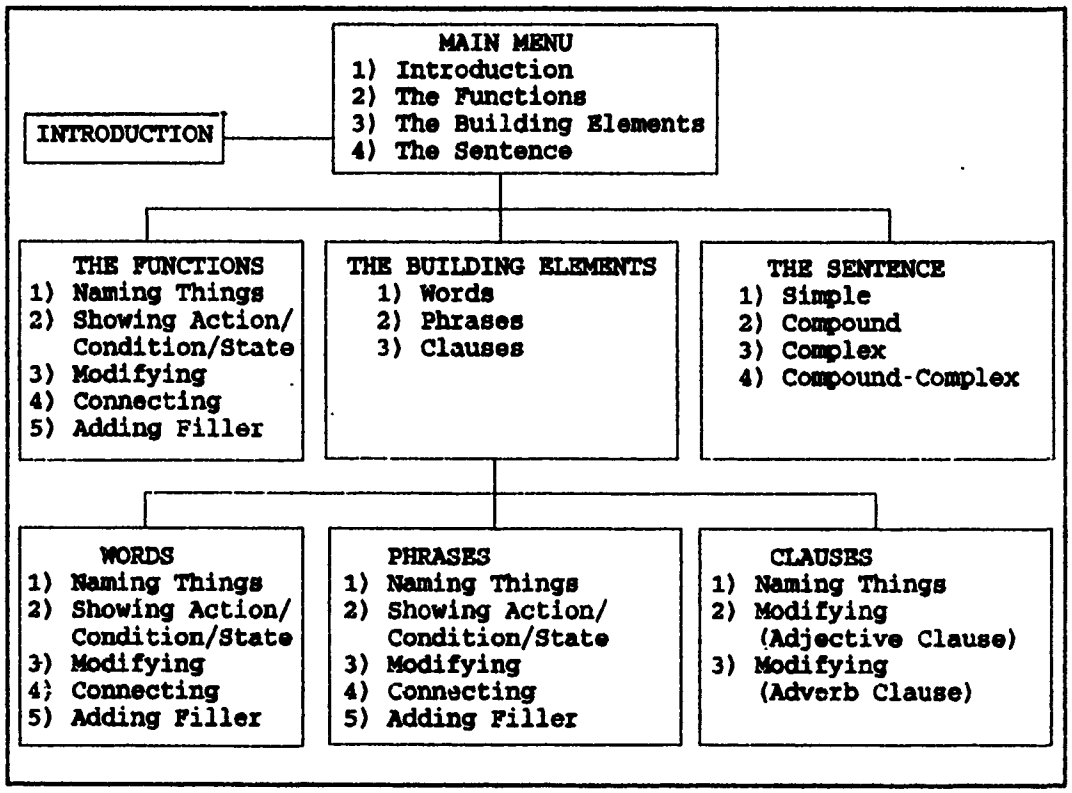


Figure 4: Backbone Structure of the Tutorial

into how each element performs the various functions. The final branch shows how the building elements are used to form complete sentences. The hypertext environment allows the users to move up and down the tree structure to suit their particular needs and to jump from an area in one branch to an area in another branch as necessary to understand the concepts being presented.

During the unit planning phase, the hierarchical structure of the tutorial was first created by using individual index cards to represent each branch in the tree. Next, the concepts contained on the index cards were expanded to include in-depth coverage and examples of the concepts. Once the concepts had been expanded to their final form, the lesson completion phase began.

Lesson Completion. In the lesson completion phase, the various concepts were broken down into the actual screens on which they would appear. Following Burke's suggestion (6:50), a complete, screen-by-screen paper version of the tutorial was developed. This paper version of the tutorial consisted of computer-generated "notecards" similar to Orwig's story board (40:50) as shown in Figure 1 above. This version of the tutorial is at Appendix C. During the lesson completion phase, the tutorial was entitled, "The Grammar Lesson."

Development of the Program Software. Development of the program software required two activities: converting the paper version of the tutorial into a hypertext document and developing a program which would display that hypertext document and contain the user interface.

The hypertext document was created using a modified version of the hypertext editor presented by Gessner (20:26). Gessner's editor was modified to allow for the larger text screen required by the tutorial. The Turbo Pascal code for the modified hypertext editor, a procedure called "help_editor," is contained in the second program listed in Appendix D. The hypertext editor allows for limited text editing, but its unique feature is the ability to insert hypertext links directly into the document. Links are inserted by placing the cursor on the desired key word, pressing <F2>, and indicating the screen to which that key word is linked. Key words appear in yellow text, while the remainder of the text appears in white. Since the hypertext editor saves each screen of the document as a separate record in a file, the number of possible screens in a document is limited only by the memory capabilities of the machine on which it is executed. "The Grammar Lesson" consists of seven instruction screens and 137 lesson screens.

The program which displays the hypertext and contains the user interface was also written using Turbo Pascal. This program, called "Grammar," is the first program listed in Appendix D. The structure of the program is shown in Figure 5.

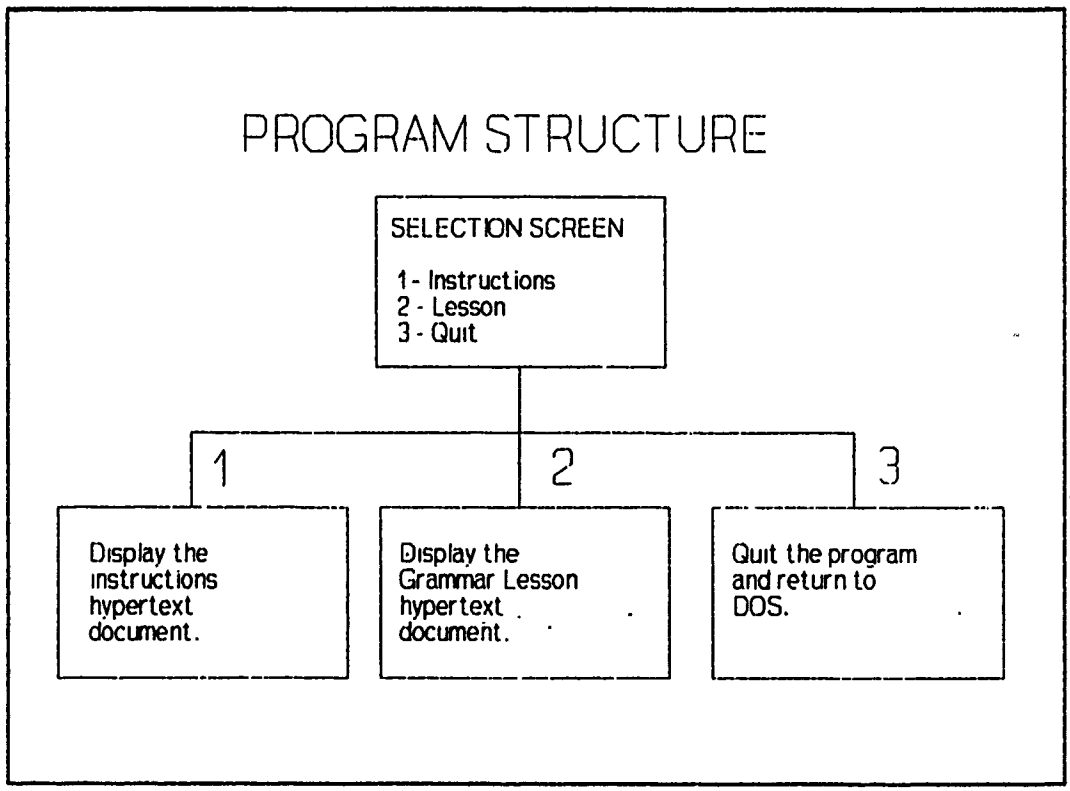


Figure 5: "Grammar" Program Structure

When the user executes the program, the introductory screen appears. After the user presses <Enter>, the selection screen appears. Both of these screens, as well as the instructional and lessons screens, are shown at Appendix C. To make a selection, the user moves the highlight bar to the desired option, using the cursor keys, and presses <Enter> with the desired option highlighted. If the user selects "Instructions" or "The Grammar Lesson," the program retrieves the appropriate hypertext document and displays it as shown in Figure 6.

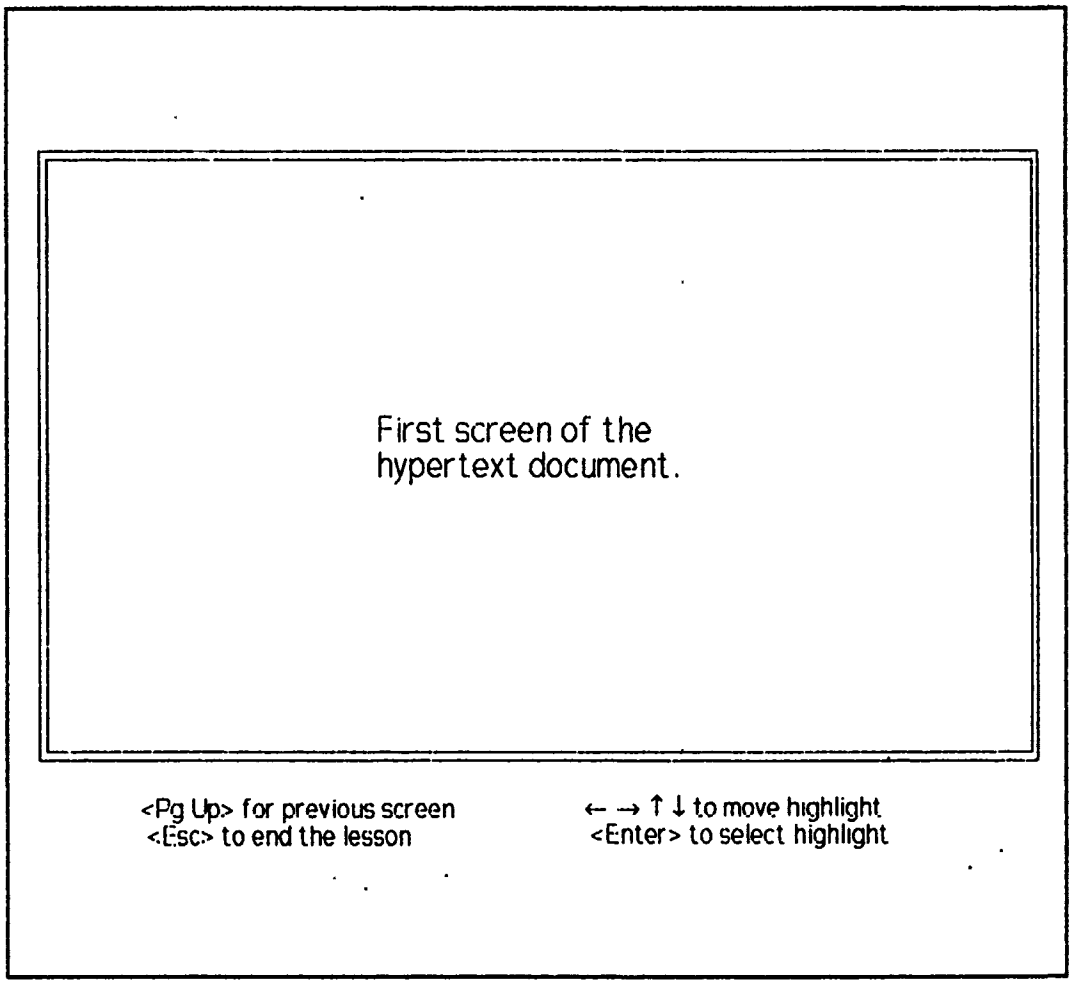


Figure 6: Hypertext Display Screen

In the hypertext document, normal text appears in white, key words appear in yellow, and the highlight bar is red. The user moves from screen to screen in the hypertext documents by moving the highlight bar to a key word, and selecting it by pressing <Enter>. The program automatically displays the screen associated with the selected key word. To return to the previous screen, the user presses the <Pg Up> key. To leave the hypertext document and return to the program's selection menu, the user presses the <Esc> key. As shown in Figure 6, the keys necessary to perform each of these actions are listed at the

bottom of each screen. A more detailed description of the procedures is shown in the "Instructions" screens in Appendix C.

Program Evaluation

The final step in any software development project is to determine how well the software works. One author asserts that commercial software developers devote approximately 50 percent of total software development time to software testing (24:1). Due to the time constraints of this project, however, an extensive testing program was unfeasible. Instead, a preliminary field test was conducted to provide initial feedback concerning program execution and tutorial content. The goal of this preliminary field test was to address the fourth research objective:

> Assess the effectiveness of the tutorial by providing it to students in AFIT's refresher course in fundamental written communication and evaluating their feedback.

To conduct the field test, volunteers were solicited from students enrolled in the AFIT summer short term course COMM 310, Fundamentals of Written Communication. Of the 40 students enrolled in the course, 36 students volunteered to participate in the field test. At the end of the first week of the four-week course, each volunteer received a copy of the prototype tutorial software, and two weeks later, each volunteer received a user-evaluation survey to complete. Of the 36 surveys distributed, 17 were completed and returned. The survey instrument and the students' responses are at Appendix E.

The research objective was to assess the effectiveness of the tutorial. For the purpose of this project, effectiveness was defined in three ways:

1) program execution,

2) lesson content, and

3) ability to teach the grammar concepts.

Unfortunately, the results of the field test could not be used to conduct an in-depth analysis of the effectiveness of the tutorial. This limitation is based on three factors.

First, as mentioned above, the time constraints of the project did not allow for an extensive period of user testing. Instead of testing the tutorial through a complete software testing program, the project was limited to a preliminary field test of the software. The field test consisted of a survey which was divided into three parts:

1) program execution,

2) lesson contents,

3) sample quiz.

The survey instrument and the students' responses are at Appendix E.

The second factor limiting the analysis of the field-test results was the low response rate to the survey. Thirty-six students received a copy of the tutorial, but less than half of them (only seventeen) completed and returned the survey. This greatly limited the strength of inferences that could be drawn from the responses. The specific limitations of inferences will be discussed in the sections below which address each section of the survey.

The third limiting factor was the lack of a controlled environment for the sample quiz portion of the survey. Due to the time constraints, a pre-test/post-test experiment was unfeasible. This limited the inferences that could be made based on the results of the quiz. Additionally, since the volunteers reviewed the software and completed

the survey on their own, the results of the sample quiz cannot be used as an effective measurement of how much knowledge they gained directly from the tutorial.

Because of these three limiting factors, the effectiveness of the software could not be completely assessed. In spite of this limitation, however, the field test did provide valuable information for the research project in the areas of program execution and lesson content. As a result of the field test, potential enhancements to the software were identified and subsequently added to the final version of the software. These enhancements will be described in the sections below.

The following paragraphs describe each sections of the survey and summarize the results in the form of findings. For each correctable finding, the actions taken to resolve the finding are presented.

Program Execution. The program execution section of the survey contained software coding-related questions. The purpose of this section was to provide information concerning the physical structure of the program, including the initial start-up, the menu structure, and navigation through the hypertext structure of the program. The following findings represent an assessment of the software based strictly on the responses of the survey participants; generalization to a population was not necessary in these circumstances.

FINDING 1: The program startup procedures were effective.

All users reported that the program initialized correctly from the DOS prompt.

FINDING 2: When users pressed the cursor keys while viewing screens without highlighted key words, the program aborted.

Two users identified this discrepancy in the software. When a user views a screen with no highlighted words, that is, a screen with no hypertext links (see, for example, Lesson Screen 10, in Appendix C), the cursor keys had no intended function. However, if a user pressed a cursor key while viewing this type of screen, the program would abort. This discrepancy was corrected and no longer occurs in the latest version of the software program.

> Finding 3: The screen colors used in the program effectively allowed users to distinguish between normal text and key words.

In the program, normal text appears white on the blue background, and key words appear in yellow on the blue background. This distinction is important because the key words are the hypertext links to other screens. All of the users indicated that they were able to distinguish between normal text and key words.

> Finding 4: The users understood the procedures for moving from one screen to another in the tutorial; however, they occasionally lost their sense of where they were in the program.

This finding illustrates two distinct aspects to reading a hypertext document:

1) moving from screen to screen through the physical structure of the document and

2) maintaining a sense of one's current location in the document.

Of the 17 respondents, 14 found movement from screen to screen "not at all confusing," while three found it "somewhat confusing." However, the responses were much different when the users were asked if they ever lost their sense of where they were in the program. Only four users replied that they never lost their sense of direction; five users indicated that they seldom did; and eight users indicated that they

occasionally did. These figures suggest that the users apparent ease in moving from page to page throughout the hypertext document does not ensure their ability to maintain a sense of their location in the document. To help alleviate this problem, an index of key terms, accessible from the Main Menu, has been added to the tutorial. With this enhancement, and the "hot key" enhancement described below, users can now move quickly to and from desired areas in the document with a minimum of keystrokes.

> Finding 5: The current method of paging up through previous screens should be replaced with a "hot key" that will automatically return the user to the Main Menu.

With the prototype version of the software, the user could only move between linked screens. If the user had traversed down a particular path of the lesson, he or she had to page up through each previously viewed screen in order to return to the beginning of the path. This method was cumbersome, and several of the users suggested a "hot key" that would automatically return the user to the Main Menu. Based on this input, the program code was modified to allow the user to jump immediately to the Main Menu from anywhere in the document simply by pressing the <Home> key.

Lesson Contents. This section of the survey contained questions related to the lesson's presentation of basic grammar concepts. The purpose of the section was to allow students who are currently enrolled in a grammar fundamentals course to comment on the content of the tutorial. For the reasons described earlier, inferences about a larger population cannot be drawn from the limited responses to this section of the survey. Instead, the following findings are presented based solely on the responses to the survey.

Finding 6: The users were able to understand the concepts as presented in the tutorial.

Eight users indicated that they had never studied English grammar in terms of functions and building elements as presented in the tutorial; four users were unsure; and only five indicated that they had studied grammar in terms of functions and building elements. All seventeen users replied that they understood the lesson's presentation of the two concepts, with ten users indicating that they understood the concepts "quite well."

Finding 7: The tutorial helped its users achieve a better understanding of the English language.

Sixteen of the users indicated that the tutorial helped them to improve their understanding of the language. Additionally, 16 of the 17 users indicated that the tutorial helped them recognize functions, building elements, and properly written sentences. One user indicated that the program would help him or her to recognize the functions and properly written sentences, but not the building elements.

Finding 8: The users found the tutorial useful in relation to their current fundamentals of written communication course.

Fourteen of the users replied that the tutorial was useful to them in relation to the course. The three users who found it "not at all useful" in relation to the course stated that this was because they did not review the tutorial until after they had completed the course. Several users suggested that the tutorial could have been more useful if they had been able to devote sufficient time to it. One user suggested that the tutorial should be sent to new students before they arrive at AFIT so that they can review basic grammar concepts prior to their arrival.

Sample Quiz. The final section of the survey was a sample quiz of the concepts presented in the tutorial. The purpose of the quiz was not to test how well the users had learned the concepts contained in the tutorial; that type of testing was unfeasible under the time constraints of this preliminary field test of the software. Rather, the purpose of the quiz was to gauge the users' comfort with the terminology and their ease in locating information within the tutorial.

The survey instructed the users to attempt to answer the questions from memory first, but to refer to the tutorial if necessary. For each question, the respondents were asked to indicate whether they had answered the question from memory or by referring to the tutorial. The sample quiz was completed by 15 of the 17 respondents. They remaining two users indicated that they were unable to devote the time necessary to prepare for a quiz of the concepts. The results of the sample quiz are shown in Appendix C.

The inferences that can be drawn from this portion of the survey are very limited. The users completed the quiz on their own, and no controls were in place to monitor their use of the tutorial and completion of the quiz. Findings based on the sample quiz are limited to the following:

> Finding 9: Traditional definitions of grammatical concepts which were learned in early schooling may persist.

This finding is illustrated in the responses to question 25 of the survey which asked, "According to the lesson, what do adverbs modify?" The desired answer to this question, according to the lesson, is that adverbs can modify anything other than naming elements. Only five users responded correctly to this question, and four of these users referred

to the tutorial to find the correct answer. The remaining ten users answered the question from memory, and all them responded that "adverbs modify verbs and adjectives." This traditional definition of adverbs is only partially correct, since adverbs can also modify other adverbs--for example, "quite" in the sentence "We played quite well." Also, in some cases an adverb can modify an entire sentence--for example, "obviously" in the sentence "Obviously, we won the game." The tutorial attempts to simplify the concept of adverbs by presenting an adverb as an element that modifies any other than a naming element; in other words, it modifies anything that is not modified by an adjective. The fact that the majority of respondents gave the more traditional definition of adverbs could suggest that

1) the students understood the concept as it was presented in the program, but repeated the traditional definition from rote, or

2) the program did not adequately convey the information as intended.

Both of these reasons are possible, but the field test did not provide an adequate means to distinguish between them. A solution to this problem could be to emphasize explicitly to the users those areas that vary from the traditional definitions they may have learned in their early schooling.

Finding 10: The tutorial was helpful as a reference tool for the users while they completed the sample quiz.

For every question, the ratio of correct answers to incorrect answers was higher for those who referred to the tutorial than for those who answered from memory. The users who took the time to refer to the tutorial as a reference were much more likely to respond with the correct answer than those who did not refer to the tutorial.

Finding 11: Those users who referred to the tutorial did not find it very easy to locate specific concepts within the tutorial.

Seven of the users found it at least "somewhat easy" to locate the answers in the tutorial; three found it "not very easy;" but only two found it "very easy." One respondent suggested that a menu of key words should be included to simplify the process of finding particular concepts within the tutorial. Based on this finding, the tutorial has been enhanced to include an index of key concepts that is accessible directly from the Main Menu. This improvement should greatly alleviate the difficulty encountered by the users in locating specific concepts.

Summary

This chapter presented the results of the research project and analyzed the data collected throughout the process. The information was presented in four sections:

1) Results of the Literature Review,

2) Survey of Subject-Matter Experts,

3) Development of the Tutorial, and

4) Program Evaluation.

These sections correspond to the four research objectives identified in Chapter 1. Conclusions will be presented in the next chapter.

IV. Conclusions and Recommendations

Introduction

As stated in Chapter 1, the purpose of this research project was twofold:

1) to determine the types of grammatical errors commonly committed by students in their written communication and

2) to develop a CAI tutorial that would present basic grammatical concepts to help students avoid those grammatical errors.

To guide the research, four research objectives were established:

1) determine what CAI theories, techniques, and research findings are already available,

2) identify the grammatical errors commonly committed by students in their written communication,

3) develop a CAI tutorial that will help students to better understand English grammar concepts, and

4) assess the effectiveness of the tutorial by providing it to students in AFIT's refresher course in fundamental written communication and evaluating their feedback.

This chapter will address each of the research objectives and evaluate how they were achieved. The chapter ends with overall conclusions and recommendations for further research in the area of this project.

Conclusions About Research Objective 1

DETERMINE WHAT CAI THEORIES, TECHNIQUES, AND RESEARCH FINDINGS ARE ALREADY AVAILABLE

This purpose of this research objective was to provide background information regarding CAI in general and also to provide information concerning appropriate methods of programming a branching tutorial.

CAI Literature. The information about CAI collected during the literature review had direct application to the project because it provided important information concerning the following aspects of CAI:

1) CAI terminology,
2) learner interaction with CAI,
3) advantages and disadvantages of CAI,
4) types of CAI,
5) development of CAI,
6) computer-user interface,
7) screen design,
8) validation of CAI, and
9) previous CAI research.

Each of these areas guided specific aspects of the development process used through the remainder of the project. The manner in which each area was applied is addressed in detail in Chapter 3.

Computer-Programming Literature. The computer-programming information collected during the literature review likewise played an important role in the research project.

First, the literature pointed to hypertext as an appropriate environment for computer-user interface for a branching tutorial. The purpose of a branching tutorial is to allow the user to select the sequence of instruction according to his or her needs. A hypertext environment not only provides the interface tools necessary to handle the branching function, but it also simplifies the process by reducing movement through the tutorial to a "point-and-click" process involving a minimum of keystrokes.

Second, the literature review supported the decision to use the higher-order language, Turbo Pascal, as the development environment for the tutorial. This decision allowed the tutorial to be developed without the use of an expensive hypertext authoring tool. Additionally, the resulting software can stand alone, unlike hypertext programs developed with some commercially-available hypertext authoring tools which require additional software to run the programs developed.

The decision to develop the tutorial following Gessner's approach to programming hypertext (20.) proved worthwhile. His hypertext editor and display programs were readily adaptable to the requirements of this project. Further, since the program was written in a higher-order language, corrections and enhancements to the program software, which were identified during the preliminary field test of the software, were easily incorporated into the program by making the necessary changes to the program source code and re-compiling the program.

Conclusions About Research Objective 2

IDENTIFY THE GRAMMATICAL ERRORS COMMONLY COMMITTED BY STUDENTS IN THEIR WRITTEN COMMUNICATION

The purpose of this research objective was to provide the project with a strong base of information from which to develop the tutorial. The objective was successfully achieved by surveying communication professors at both AFIT and the University of Dayton, who served the role of subject-matter expert for the project. The two groups provided a broad spectrum of experience, both military and civilian, from which to draw conclusions about common grammatical errors. The survey supported the initial hypothesis that the most common grammatical errors committed by students in their written communication are

1) sentence fragment,

2) fused sentence,

3) comma splice,

4) subject-verb agreement,

5) pronoun reference error,

6) misplaced modifier,

7) illogical verbal modifier, and

8) parallelism.

While the initial intent of the project was to address each of these areas, due to time constraints, the tutorial excluded the latter three areas, but included parts of speech and clause recognition, which were also recommended by all of the subject-matter experts.

Conclusions About Research Objective 3

DEVELOP A CAI TUTORIAL THAT WILL HELP STUDENTS TO BETTER UNDERSTAND ENGLISH GRAMMAR CONCEPTS

This research objective was the most time-consuming phase of the research project. It involved two distinct facets: development of the tutorial content and coding of the program software.

Development of the Tutorial Content. As mentioned in the previous section, the survey of subject-matter experts provided important information for inclusion in the program. The decision to forego the traditional method of grammar instruction, as detailed in Chapter 3, and to adopt a different method allowed the project to exploit the benefits of instruction in a hypertext environment.

The content of the tutorial begins with the simplest concepts and builds upwards to more difficult concepts. By focusing on the two concepts of language--the building elements and the functions they

perform--the tutorial avoids the traditional method of forcing students to memorize definitions and lists of examples. The goal of the tutorial is to provide a basic understanding of these two concepts. The tutorial's effectiveness in achieving this goal will be addressed below.

Development of the Program Software. Once the tutorial content was finalized, the program software to edit and display the content as a hypertext document needed to be developed. This phase of development relied greatly on the model provided by Gessner (20). His code, written in Turbo Pascal, was easily adapted to the requirements of this project. The discrepancies and enhancements identified in the preliminary field test were subsequently added to the prototype program code, resulting in a program that effectively displays and allows movement throughout the hypertext grammar tutorial.

Conclusions About Research Objective 4

> ASSESS THE EFFECTIVENESS OF THE TUTORIAL BY PROVIDING IT TO STUDENTS IN AFIT'S REFRESHER COURSE IN FUNDAMENTAL WRITTEN COMMUNICATION AND EVALUATING THEIR FEEDBACK

The purpose of this research objective was to evaluate three areas of the grammar tutorial software:

1) the execution of the computer program,

2) the hypertext environment used in the program, and

3) the tutorial's effectiveness as a learning tool.

As discussed in Chapter 3, the first two facets of the evaluation were achieved by the preliminary field test of the computer software. The third area, however, could not be assessed by the current research project.

Program Execution. The preliminary field test of the software and the resulting improvements to the software code support the conclusion that execution of the program was effective. The only serious problem identified by the field-test participants (the cursor keys causing an abort in some cases) was corrected by a modification of the source code.

Hypertext Environment. The field test of the software identified several potential enhancements to the program that would make it easier for users to move about the tutorial without losing their sense of where they are in the document. These enhancements--a "hot key" that automatically returns the user to the Main Menu and an index that allows the user to jump directly to key concepts in the tutorial--have been added to the latest version of the software.

Effectiveness as a Learning Tool. As discussed in Chapter 3, this facet of evaluation was not achieved during this project. To properly assess the tutorial's effectiveness as a learning tool, a controlled experiment should be conducted, including a pre-test/post-test design which would assess the user's knowledge prior before and after using the tutorial. The preliminary field test conducted during this project did not follow the type of experimental design necessary to achieve this purpose. Instead, its usefulness was primarily in the areas addressed in the previous two sections.

Overall Conclusions

This research project resulted in a grammar tutorial that has been field-tested to assess the effectiveness of its software and its hypertext environment. The hypertext interface that the tutorial employs was enhanced to incorporate the suggestions of participants in

the field test of the software. The content of the tutorial was developed based on the information collected from 23 subject-matter experts--communication professors from two different institutions. The tutorial is ready to be evaluated in an experiment to assess its effectiveness as a learning tool.

The program that was developed during this research project supports that conclusion that a branching CAI tutorial, using hypertext for its computer-user interface, can be developed in a higher-order language. The hypertext programs developed for the grammar tutorial are transparent to the contents of the hypertext document being edited or displayed; hence, the same programs could be used to create and display hypertext documents in other text-based areas of instruction.

Recommendations for Further Research

The following recommendations are offered based on the conclusions drawn from this research project:

Recommendation 1. The program developed during this project has been field tested to evaluate its program execution and its hypertext interface. Before it is released to students, however, it should be evaluated for its effectiveness as a learning tool.

Recommendation 2. The tutorial developed during this project covers basic grammar concepts; it did not cover the grammatical errors highlighted by the survey of subject-matter experts. Future research should include an expansion of, or supplement to, the tutorial which would cover these additional areas of grammar. Other material deemed appropriate by the course director of AFIT's Fundamentals of Written Communication course, COMM 310, could also be included.

Recommendation 3. The software developed for this project can create and display any hypertext document. This tutorial covered basic English grammar concepts. Future research should address the possibilities of developing hypertext documents to supplement other areas of instruction.

Summary

This chapter restated the four research objectives which guided this research project and described how effectively these objectives were achieved. The first three objectives were met successfully:

1) the literature review provided a valuable base of information about CAI;

2) the subject-matter expert survey provided the necessary information from which to develop the tutorial content; and

3) the tutorial development phase resulted in a prototype version of the grammar tutorial.

The fourth objective, effectiveness of the tutorial, was only partially achieved. The execution of the program and its hypertext interface were both assessed and improved based on the results of the preliminary field test of the software. The tutorial's effectiveness as a learning tool, however, could not be evaluated due to the time limitations of the current project. Such an evaluation should take place before the tutorial is distributed to students.

Several recommendations for further research were also offered:

1) evaluation of the effectiveness of the tutorial as a learning tool,

2) expansion of the contents of the tutorial to include other areas of grammar, and

3) development of tutorials in other areas of instruction, using the editing and displaying programs developed in this project.

Appendix A: Glossary of Common Grammatical Errors

1. SENTENCE FRAGMENT

COMM 687: A word or group of words which is punctuated as a sentence but which fails to meet the minimum sentence requirement of at least one clause. (15:1)

HARBRACE: A group of words beginning with a capital letter and ending with a period. Although written as if it were a sentence, a fragment is only part of a sentence--such as a phrase or subordinate clause. (25:28)

PRENTICE HALL: Any group of words that is set off as a sentence but that lacks a subject, a verb, or an independent clause. (32:62)

EXAMPLES: Because we submitted the paperwork yesterday.
Which will ensure we receive the parts on time.

2. FUSED SENTENCE

COMM 687: Occurs when two independent clauses are joined with no punctuation and/or coordinating conjunction between them. (15:1)

HARBRACE: No punctuation between main clauses. (25:34)

PRENTICE HALL: Two grammatically complete thoughts with no separating punctuation. (32:70)

EXAMPLES: The student completed the assignment early it did not meet the objectives.
The student completed the assignment early however, it did not meet the objectives.

3. COMMA SPLICE or COMMA FAULT

COMM 687: Occurs when two independent clauses are connected only with a comma. (15:1)

HARBRACE: Only a comma between main clauses. (25:34)

PRENTICE HALL: Placing a comma between two main clauses without a coordinating conjunction. (32:69)

EXAMPLES: The student completed the assignment early, it did not meet the objectives.
The student completed the assignment early, however, it did not meet the objectives.

4. SUBJECT-VERB AGREEMENT ERROR

COMM 687: Occurs when a subject and a verb fail to agree in number or person. (15:2)

HARBRACE: Make the verb agree in number with its subject. (25:62)

PRENTICE HALL: Every verb should agree in number with its subject. (32:75)

EXAMPLES: The classrooms on the third floor has no heat.
Everyone have received the latest bulletin.

5. PRONOUN REFERENCE ERROR

COMM 687: Any problem in the relationship of a pronoun and its antecedent. (15:2)

HARBRACE: Make a pronoun refer unmistakably to its antecedent. (25:285)

PRENTICE HALL: Each pronoun should refer to a single antecedent. (32:86)

EXAMPLES: The student cannot find the instructor's book and now he needs it.
She met with the colonel every morning before her staff meeting.

6. MISPLACED PARTS

COMM 687: [MISPLACED MODIFIERS (MM)] A modifier that creates ambiguity because of its location. This error can be corrected by moving the modifier nearer to the unit it needs to modify. (15:3)

HARBRACE: Avoid needless separation of related parts of the sentence . . . Place modifiers near the words they modify. (25:269)

PRENTICE HALL: (MIS PT) Word order . . . is the principal way to keep many modifiers attached to the words they modify. (32:100)

EXAMPLES: One group only completed the first assignment.
(Versus: Only one group completed the first assignment.)
Major Johnson even works after 1630.
(Versus: Major Johnson works even after 1630.)

7. DANGLING MODIFIER

COMM 687: Occurs when a modifying word or phrase has nothing in the sentence to modify. The error often occurs when the sentence begins with the modifier. (15:4)

HARBRACE: Verbal phrases that do not refer clearly and logically to another word or phrase in the sentence. (Hodges et al.:272)

PRENTICE HALL: (DLG) Has nothing to modify because the word it logically should modify is not present in its sentence. (32:108)

EXAMPLES: While attempting to complete the project, the power went out in the building.
Although not responsible for the malfunction, the colonel reprimanded the lieutenant.

8. PARALLELISM

COMM 687: Occurs when a writer establishes a series or pattern in a sentence and then violates the pattern without apparent reason. (15:4)

HARBRACE: Balance nouns with nouns, prepositional phrases with prepositional phrases, main clauses with main clauses, and so on. (25:275)

PRENTICE HALL: When you coordinate two or more elements in a sentence, readers expect you to . . . state them in the same grammatical form. (32:235)

EXAMPLES: The professor expects the students to show up for class, pay close attention, and turning in all assignments.
The clerk purchased new ribbons for all printers, new labels for all folders, and paperclips.

Appendix B: Subject-Matter Expert Questionnaire and Responses

This appendix consists of the questionnaire distributed to teachers of communication at the Air Force Institute of Technology (AFIT) and University of Dayton (UD). Aggregate responses to each question are listed immediately following the question and are shown broken down by institution and by overall total.

PART I. GRAMMATICAL ERRORS

This section contains examples of eight types of grammatical errors. For each type of error, several examples are given, with relevant portions underlined for emphasis. (If you observe *stylistic problems* that are *unrelated to the underlined grammatical error*, please note them in the comments sections.) Using the five-point scales provided, please circle your responses to

1) indicate the relative frequency with which you find each error in students' expository writing and

2) indicate the severity of each error. "Severity" means how seriously the error degrades effective communication.

---

1. SENTENCE FRAGMENT (a group of words punctuated as a sentence but lacking an independent clause)

EXAMPLE 1A: *Because we submitted the paperwork yesterday.*

OCCURRENCE:	1-Never	2-Seldom	3-Occasionally	4-Often	5-Very Frequent
RESPONSES: AFIT:	0	4	1	0	0
UD:	0	2	10	3	0
TOTAL:	0	6	11	3	0

SEVERITY:	1-Not a Problem	2-Minor	3-Moderate	4-Severe	5-Very Severe
RESPONSES: AFIT:	0	0	1	3	1
UD:	0	1	4	6	4
TOTAL:	0	1	5	9	5

EXAMPLE 1B: *For example, psychology, sociology, and anthropology.*

OCCURRENCE:		1-Never	2-Seldom	3-Occasionally	4-Often	5-Very Frequent
RESPONSES:	AFIT:	0	3	2	0	0
	UD:	0	3	4	5	1
	TOTAL:	0	6	6	5	1

SEVERITY:		1-Not a Problem	2-Minor	3-Moderate	4-Severe	5-Very Severe
RESPONSES:	AFIT:	0	0	0	4	1
	UD:	0	1	6	3	3
	TOTAL:	0	1	6	7	4

2. FUSED SENTENCE (in a compound sentence, two independent clauses joined without semicolon, colon, or comma and coordinating conjunction)

EXAMPLE 2A: *The student completed the assignment early ___ it did not meet the objectives.*

OCCURRENCE:		1-Never	2-Seldom	3-Occasionally	4-Often	5-Very Frequent
RESPONSES:	AFIT:	1	2	1	0	1
	UD:	0	3	10	2	0
	TOTAL:	1	5	11	2	1

SEVERITY:		1-Not a Problem	2-Minor	3-Moderate	4-Severe	5-Very Severe
RESPONSES:	AFIT:	0	0	2	2	1
	UD:	0	0	4	5	6
	TOTAL:	0	0	6	7	7

EXAMPLE 2B: *The student completed the assignment early ___ however, it did not meet the objectives.*

OCCURRENCE:		1-Never	2-Seldom	3-Occasionally	4-Often	5-Very Frequent
RESPONSES:	AFIT:	0	1	2	1	1
	UD:	0	3	5	6	1
	TOTAL:	0	4	7	7	2

SEVERITY:		1-Not a Problem	2-Minor	3-Moderate	4-Severe	5-Very Severe
RESPONSES:	AFIT:	0	0	3	1	1
	UD:	0	2	5	3	5
	TOTAL:	0	2	8	4	6

3. COMMA SPLICE (two independent clauses joined only by a comma)

EXAMPLE 3A: *The student completed the assignment early, ___ it did not meet the objectives.*

OCCURRENCE:	1-Never	2-Seldom	3-Occasionally	4-Often	5-Very Frequent
RESPONSES: AFIT:	0	0	4	0	1
UD:	0	1	3	6	5
TOTAL:	0	1	7	6	6

SEVERITY:	1-Not a Problem	2-Minor	3-Moderate	4-Severe	5-Very Severe
RESPONSES: AFIT:	0	1	2	2	0
UD:	0	1	6	5	3
TOTAL:	0	2	8	7	3

EXAMPLE 3B: *The student completed the assignment early, ___ however, it did not meet the objectives.*

OCCURRENCE:	1-Never	2-Seldom	3-Occasionally	4-Often	5-Very Frequent
RESPONSES: AFIT:	0	0	1	3	1
UD:	0	0	3	5	7
TOTAL:	0	0	4	8	8

SEVERITY:	1-Not a Problem	2-Minor	3-Moderate	4-Severe	5-Very Severe
RESPONSES: AFIT:	0	1.5	1.5	2	2
UD:	0	1	8	2	3
TOTAL:	0	2.5	9.5	4	5

4. SUBJECT-VERB AGREEMENT ERROR (when the subject and verb of a clause do not agree in number or person)

EXAMPLE 4A: *The location of the conferences have not been announced.*

OCCURRENCE:	1-Never	2-Seldom	3-Occasionally	4-Often	5-Very Frequent
RESPONSES: AFIT:	0	2	4	0	0
UD:	0	1	7	6	1
TOTAL:	0	3	11	6	1

SEVERITY:	1-Not a Problem	2-Minor	3-Moderate	4-Severe	5-Very Severe
RESPONSES: AFIT:	0	0	2	1	2
UD:	0	0	4	8	3
TOTAL:	0	0	6	9	5

EXAMPLE 4B: *Neither the dean nor the professor were available for comment.*

OCCURRENCE:	1-Never	2-Seldom	3-Occasionally	4-Often	5-Very Frequent
RESPONSES: AFIT:	0	3	2	1	0
UD:	0	1	2	8	4
TOTAL:	0	4	4	9	4

SEVERITY:	1-Not a Problem	2-Minor	3-Moderate	4-Severe	5-Very Severe
RESPONSES: AFIT:	0	2	2	0	1
UD:	0	2	7	4	2
TOTAL:	0	4	9	4	3

EXAMPLE 4C: *The criteria for passing the course was listed in the syllabus.*

OCCURRENCE:	1-Never	2-Seldom	3-Occasionally	4-Often	5-Very Frequent
RESPONSES: AFIT:	0	0	2	3	1
UD:	0	2	4	5	4
TOTAL:	0	2	6	8	5

SEVERITY:	1-Not a Problem	2-Minor	3-Moderate	4-Severe	5-Very Severe
RESPONSES: AFIT:	0	2	3	1	0
UD:	0	3	7	2	3
TOTAL:	0	5	10	3	3

EXAMPLE 4D: *The coach, who is in excellent condition, run five miles a day.*

OCCURRENCE:	1-Never	2-Seldom	3-Occasionally	4-Often	5-Very Frequent
RESPONSES: AFIT:	0	4	2	0	0
UD:	0	7	7	1	0
TOTAL:	0	11	9	1	0

SEVERITY:	1-Not a Problem	2-Minor	3-Moderate	4-Severe	5-Very Severe
RESPONSES: AFIT:	0	1	0	2	2
UD:	0	1	2	6	6
TOTAL:	0	2	2	8	8

---

5. PRONOUN REFERENCE ERROR (any problem between a pronoun and its antecedent)

EXAMPLE 5A: *Jim cannot find the book that Bill brought and now he needs it.*

OCCURRENCE:	1-Never	2-Seldom	3-Occasionally	4-Often	5-Very Frequent
RESPONSES: AFIT:	0	2	3	1	0
UD:	0	0	4	8	1
TOTAL:	0	2	7	9	1

SEVERITY:	1-Not a Problem	2-Minor	3-Moderate	4-Severe	5-Very Severe
RESPONSES: AFIT:	0	1	2	2	1
UD:	0	0	1	9	3
TOTAL:	0	1	3	11	4

EXAMPLE 5B: *John wanted to be a chemist because it is an honorable profession.*

OCCURRENCE:	1-Never	2-Seldom	3-Occasionally	4-Often	5-Very Frequent
RESPONSES: AFIT:	0	0	3	3	0
UD:	0	2	1	11	1
TOTAL:	0	2	4	14	1

SEVERITY:	1-Not a Problem	2-Minor	3-Moderate	4-Severe	5-Very Severe
RESPONSES: AFIT:	0	2	4	0	0
UD:	0	2	7	6	0
TOTAL:	0	4	11	6	0

6. MISPLACED MODIFIER (a modifier that creates ambiguity because of its location in the sentence)

EXAMPLE 6A: *The hermit only wanted to be left alone.*
(Versus: *The hermit wanted only to be left alone.*)

OCCURRENCE:	1-Never	2-Seldom	3-Occasionally	4-Often	5-Very Frequent
RESPONSES: AFIT:	0	1	1	2	2
UD:	0	0	7	4	4
TOTAL:	0	1	8	6	6

SEVERITY:		1-Not a Problem	2-Minor	3-Moderate	4-Severe	5-Very Severe
RESPONSES:	AFIT:	0	3	3	0	0
	UD:	0	4	10	1	0
	TOTAL:	0	7	13	1	0

EXAMPLE 6B: *The woman walked the dog wearing a beautiful evening gown.*

OCCURRENCE:		1-Never	2-Seldom	3-Occasionally	4-Often	5-Very Frequent
RESPONSES:	AFIT:	0	2	4	0	0
	UD:	0	3	7	3	1
	TOTAL:	0	5	11	3	1

SEVERITY:		1-Not a Problem	2-Minor	3-Moderate	4-Severe	5-Very Severe
RESPONSES:	AFIT:	0	2	1	2	1
	UD:	0	1	4	6	3
	TOTAL:	0	3	5	8	4

7. ILLOGICAL VERBAL MODIFIER (a verbal modifier that tries to modify a noun or pronoun that is not contained in the modified expression--sometimes called a dangling modifier)

EXAMPLE 7A: *While attempting to complete the project, the power failed throughout the building.*

OCCURRENCE:		1-Never	2-Seldom	3-Occasionally	4-Often	5-Very Frequent
RESPONSES:	AFIT:	0	0	3	3	0
	UD:	0	0	6	8	1
	TOTAL:	0	0	9	11	1

SEVERITY:		1-Not a Problem	2-Minor	3-Moderate	4-Severe	5-Very Severe
RESPONSES:	AFIT:	0	0	3	3	0
	UD:	0	0	4	9	2
	TOTAL:	0	0	7	12	2

EXAMPLE 7B: *Unable to finish our midterm examination, the professor asked whether we needed more time.*

OCCURRENCE:		1-Never	2-Seldom	3-Occasionally	4-Often	5-Very Frequent
RESPONSES:	AFIT:	0	0	3	2	0
	UD:	0	0	9	6	0
	TOTAL:	0	0	12	8	0

SEVERITY:		1-Not a Problem	2-Minor	3-Moderate	4-Severe	5-Very Severe
RESPONSES:	AFIT:	0	1	2	2	0
	UD:	0	0	6	7	2
	TOTAL:	0	1	8	9	2

8. PARALLELISM ERROR (two or more elements presented in a series or pattern but not following the same grammatical form)

EXAMPLE A: *The professor expects the students to complete all assignments, pay close attention during lectures, and, to the best of their ability, attending every class session.*

OCCURRENCE:		1-Never	2-Seldom	3-Occasionally	4-Often	5-Very Frequent
RESPONSES:	AFIT:	0	0	3	2	0
	UD:	0	0	6	6	3
	TOTAL:	0	0	9	8	3

SEVERITY:		1-Not a Problem	2-Minor	3-Moderate	4-Severe	5-Very Severe
RESPONSES:	AFIT:	0	0	4	1	0
	UD:	0	0	8	4	3
	TOTAL:	0	0	12	5	3

EXAMPLE B: *Either the student completes the assignment or fails the course.*

OCCURRENCE:		1-Never	2-Seldom	3-Occasionally	4-Often	5-Very Frequent
RESPONSES:	AFIT:	0	0	3	2	0
	UD:	0	1	7	4	3
	TOTAL:	0	1	10	6	3

SEVERITY:		1-Not a Problem	2-Minor	3-Moderate	4-Severe	5-Very Severe
RESPONSES:	AFIT:	0	1	3	1	0
	UD:	1	2	8	4	0
	TOTAL:	1	3	11	5	0

PART II. CHRONOLOGY OF INSTRUCTION

Please indicate the order of instruction you feel is most appropriate for the following areas of grammar. In other words, if you were going to help a student master the following grammatical concepts, in what order would you teach them? Use "1" for first, "2" for second, et cetera.

RESPONSES:

Area of Instruction	Expert Category	Average	Standard Deviation
Parts of Speech	AFIT	1.00	0.00
	UD	2.17	2.34
	Combination	1.78	1.99
Clause Recognition	AFIT	3.17	2.19
	UD	4.38	2.47
	Combination	4.00	2.45
Sentence Fragment	AFIT	3.00	0.63
	UD	3.86	2.26
	Combination	3.63	2.01
Pronoun Reference	AFIT	6.33	1.11
	UD	5.00	1.46
	Combination	5.40	1.50
Fused Sentence	AFIT	4.00	0.58
	UD	5.07	2.63
	Combination	4.75	2.28
Misplaced Modifier	AFIT	8.20	2.04
	UD	7.07	1.94
	Combination	7.37	2.03
Illogical Verbal Modifier	AFIT	8.17	1.21
	UD	6.93	1.62
	Combination	7.30	1.62
Comma Splice	AFIT	6.00	1.83
	UD	6.14	3.20
	Combination	6.10	2.86
Subject-Verb Agreement	AFIT	5.80	0.98
	UD	2.71	3.33
	Combination	3.53	1.60
Parallelism	AFIT	8.83	1.07
	UD	8.54	2.24
	Combination	8.63	1.95

PART III. OTHER INFORMATION

1. How many years have you taught Communication/English/Composition?

RESPONSES:

Category	Expert Average Experience (in years)
AFIT	19.33
UD	18.92
COMBINED	19.05

2. Which student textbook do you think is the best for helping students learn grammatical concepts?

RESPONSES:

AFIT	Handbook of Current English, Corder Warner Series Harbrace College Handbook Any handbook (Harbrace or Rinehart) Prentice-Hall Handbook for Writers No single preference (Harbrace or Rinehart)
UD	"Any handbook is only as good as the instructor." McGraw-Hill College Handbook Simon & Schuster Handbook for Writers, 2d Ed. No preference "Most are pretty weak to me." "I'm not sure I've found one." Simon & Schuster Handbook for Writers, 2d Ed. Harbrace College Handbook Simon & Schuster for classes; A Writer's Reference, Hacker (Bedford/St Martin's) for refresher Prentice-Hall Handbook for Writers "Haven't found one yet!" Prentice-Hall or Simon & Schuster
Note: Three UD professors did not respond to this question.	

3. Do you use standard abbreviations to mark grammatical errors? (Y/N)

If yes, please supply the abbreviations you use for the following:

Area	Responses*	Most Common
Sentence Fragment	AFIT: Frag Frag Frag Frag Frag Frag UD: SF Fr Frag Frag Frag Frag Frag Frag	Frag
Pronoun Reference	AFIT: Ref Agr Ref PnRef ProRef UD: Pr Pro Ref Ref Ref Pro Ref Ref	Ref
Fused Sentence	AFIT: FS Fus FS Fus UD: FS R/O Fused FS R/O Fus Fused R/O	FS R/O Fus
Misplaced Modifier	AFIT: —→ MM MM Mod Error UD: MM MM MM M.M. Mod	MM
Illogical Verbal Modifier	AFIT: Dang Mod Dg Dangl Mod DM Dang UD: DM DM DM DM DM	DM
Comma Splice	AFIT: CS CS CS C.S. CS UD: Splice R/O CS CS R/O CS C.S. CS CS	CS
Subject-Verb Agreement	AFIT: S-V S/V Agr S/V Agr UD: S/V Agr S-V Agr S-V S/V Agr S/V Agr	S/V
Parallelism	AFIT: // Par // Parallelism UD: // Paral // //ism Parallel	//
Others AFIT:	Subordination Sub; Active Voice av; Word Choice wc	
UD:	Delete ϕ; Insert; Wrong Word ww; Spelling sp; Punctuation p; Word Choice wc; Usage Usage; Possession Poss	

* Note: Not all respondents had abbreviations for every category.

PART IV. OTHER COMMENTS AND SUGGESTIONS

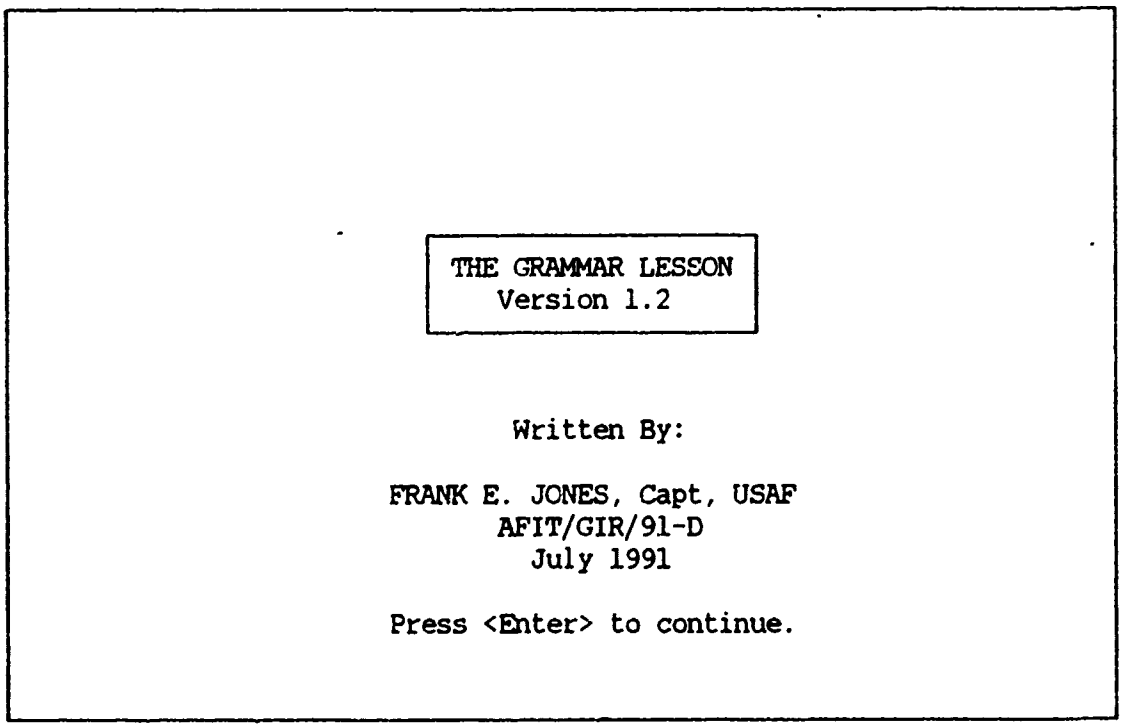
Thank you very much for your participation in this survey. The knowledge that you and other experts supply will greatly benefit future students who are learning grammatical concepts. Please return this questionnaire in the attached envelope to

Capt Frank Jones
AFIT/LSG
Wright-Patterson AFB OH 45433-6583

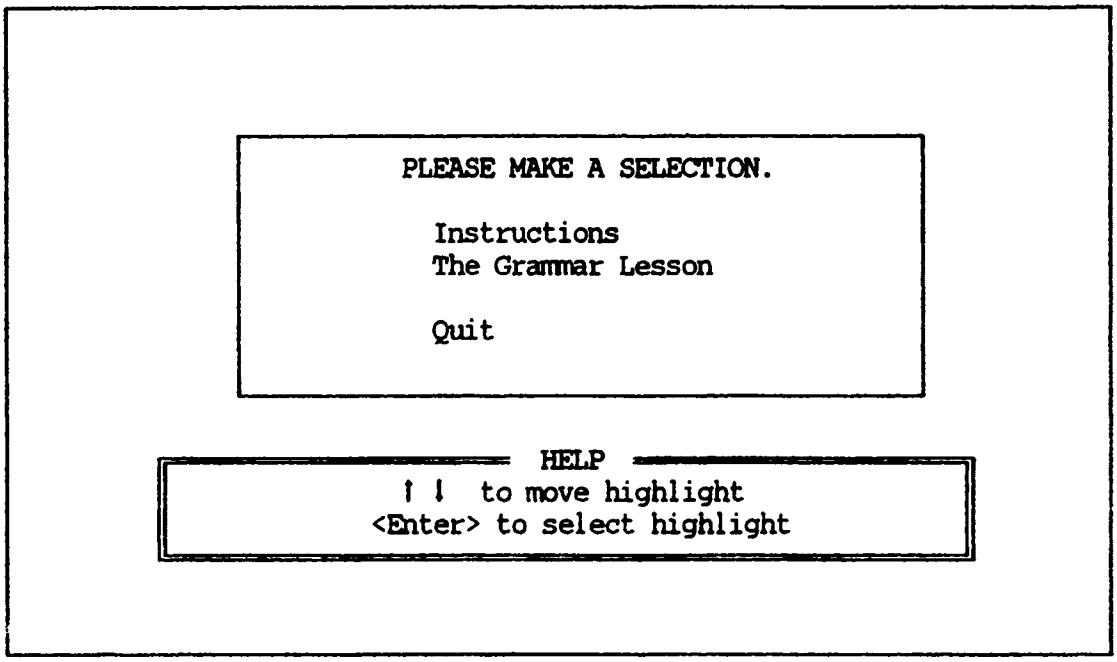
Appendix C: Grammar Lesson Program Screens

This appendix contains the screens from the latest version of the Grammar Lesson program. It includes the enhancements to the tutorial added as a result of the field test, as discussed in Chapter 3.

The program consists of an introduction screen, a selection screen, seven instruction screens, and 137 lesson screens. In the CAI version of the tutorial, the key words (the hypertext links) appear in yellow text, while normal text appears white. In this paper version of the tutorial, each hypertext link is identifiable by a bracketed number immediately following it. The number in brackets is the screen number to which the key word is linked. In the CAI tutorial, this link is transparent to the user, since the program automatically finds the linked screen when a key word is selected.



INTRODUCTION SCREEN



SELECTION SCREEN

.[1] INSTRUCTIONS (NEXT)[2]

1. This lesson employs a concept known as "hypertext." To move from screen to screen, you will "select" key words from the current screen. (The key words appear in yellow.) To select one of these key words, you will use the arrow keys to move the highlight bar to the word and press <Enter>.

2. To bring yourself back to a previous screen after you have selected a key word, press the <Pg Up> key.

3. To return to the Main Menu from anywhere, press the <Home> key.

4. When a new screen appears, the highlight bar is positioned on the "dot" in the upper left-hand corner. If the word "(NEXT)" appears in the upper right-hand corner, it means that there is a "follow-on" screen to the current screen. You should always select "(NEXT)" after you are finished with the current page.

To continue, move the highlight bar to "(NEXT)" and press <Enter>.

INSTRUCTIONS: Screen 1

.[2] INSTRUCTIONS (cont.) (NEXT)[7]

(This is the second screen of instructions. Notice that the highlight bar is once again positioned on the dot.)

5. The key words on a screen may appear in a menu format:

1[3] - Selection One
2[4] - Selection Two
3[5] - Selection Three

or they may appear within a sentence:

This is an example of a key word[6] that appears within a sentence.

6. Use the arrow keys to move the highlight bar around this page. Try selecting some of the key words.

Select "(NEXT)" when you are ready to continue.

INSTRUCTIONS: Screen 2

SELECTION ONE

You selected "1" from the sample menu.

Press <Pg Up>.

INSTRUCTIONS: Screen 3

SELECTION TWO

You selected "2" from the sample menu.

Press <Pg Up>.

INSTRUCTIONS: Screen 4

SELECTION THREE

You selected "3" from the sample menu.

Press <Pg Up>.

INSTRUCTIONS: Screen 5

KEY WORD

You selected the sample "word" in the sentence.

Press <Pg Up>.

INSTRUCTIONS: Screen 6

INSTRUCTIONS (cont.)

7. Notice that, on this page, there is no highlight bar. That means this page has no key words to select.

8. When a screen has no key words, you should use the <Pg Up> key to return to the previous screen.

9. When you are ready to quit the lesson, press the <Esc> key.

10. If you are unsure of what keys to use, look to the bottom of the screen, where the keys are listed.

11. Press <Esc> if you are finished with these instructions
or
Press <Pg Up> to return to the previous screen
or
Press <Home> to return to the first screen.

INSTRUCTIONS: Screen 7

.[1] MAIN MENU

Move the highlight to the area you would like to review and press <Enter>. To return to this screen from anywhere in the lesson, press <Home>.

NOTE: If you are reviewing this lesson for the first time, you should begin with the introduction and follow the remaining sequence. Use the Quick Index to jump immediately to a specific area.

0[139] - QUICK INDEX OF KEY CONCEPTS

1[2] - INTRODUCTION

2[4] - THE FUNCTIONS IN GRAMMAR

3[32] - THE BUILDING ELEMENTS IN GRAMMAR

4[123] - THE SENTENCE

LESSON: Screen 1

.[2] INTRODUCTION (NEXT)[3]

When we communicate with each other, we try to do so in a language that we both have in common. One measure of the effectiveness of communication is the extent to which writers convey the exact meaning of what they are trying to say. One way to ensure that this exactness is conveyed is to structure our use of the language so that it adheres to certain rules about the language that meet the expectations of the reader. Such a system of rules about language is known as its "grammar."

The rules of grammar describe how words and groups of words should be linked together to form sentences, so that we are able to communicate our thoughts to each other.

This grammar lesson is designed to help you come to a better understanding of how sentences can be written so that they are as effective as possible. This lesson will help you recognize the two basic features of the English grammar--functions and building elements--and how they work together to form sentences.

LESSON: Screen 2

In English, there are five functions that words and groups of words can perform within a sentence. Words or groups of words can

1) name things
2) show action or existence of things
3) change, qualify, or rename things
4) connect words or groups of words
5) add filler to groups of words

These five functions of English grammar can be performed by any of the building elements in the language. The building elements are

1) words
2) phrases
3) clauses

This lesson will discuss the five functions and the three building elements. It will then show how sentences are formed by combining the building elements.

To return to the Main Menu, Press <Home>.

LESSON: Screen 3

[4] FUNCTIONS

The five functions that words and groups of words can perform are:

1[5] - Naming things
2[11] - Showing action or existence of things
3[17] - Changing, qualifying, or renaming things
4[24] - Connecting words or groups of words
5[28] - Adding filler to groups of words

This section will describe each of these functions and present examples of them. You should review the examples to see how the functions are performed within them.

To review a function, select it from the list above.

To return to the MAIN MENU, press <Home>.

LESSON: Screen 4

.[5] FUNCTION: Naming Things (NEXT)[6]

The naming function can be performed a word or by groups of words. To decide whether or not a word or group of words is performing this function, ask yourself "what" or "who?"

In the following examples, single words or groups of words that perform the naming function are underlined. As you read each example, see how the underlined words name things.

1. Pilots fly.

   Who flies? -> pilots

2. Pilots fly airplanes.

   Fly what? -> airplanes

LESSON: Screen 5

.[6] FUNCTION: Naming Things (cont.) (NEXT)[7]

3. Riding is a good form of exercise.

   What is a good form of exercise? -> riding

4. Riding a bicycle is a good form of exercise.

   What is a good form of exercise? -> riding a bicycle

5. To ride a bicycle is good for you.

   What is good for you? -> to ride a bicycle

LESSON: Screen 6

```
.[7]                FUNCTION: Naming Things (cont.)          (NEXT)[8]

6.  Riding a bicycle is good exercise.
                               --------

    Riding a bicycle is good what?  ->  exercise

7.  The test was easy.
        ----

     What was easy?  ->  test

8.  We knew that the test was easy.
    --

    Who knew?  ->  we
```

LESSON: Screen 7

```
.[8]                FUNCTION: Naming Things (cont.)          (NEXT)[9]

9.  We knew that the test was easy.
            ----------------------

    Knew what? ->  that the test was easy

10.  We knew that the test was easy.
                   --------

     What was easy?  ->  the test

11.  That the test was easy was obvious from the many perfect scores.
     ----------------------

     What was obvious?  ->  that the test was easy
```

LESSON: Screen 8

.[9] FUNCTION: Naming Things (cont.) (NEXT)[10]

12. Only the selection panel knows who was selected chairperson.

Knows what? -> who was selected chairperson

LESSON: Screen 9

FUNCTION: Naming Things
REVIEW

1 - The function of naming things can be performed by individual words or by groups of words.

2 - One way to decide if a word or group of words is performing this function is to ask the question "Who?" or "What?"

To return to the FUNCTIONS MENU, press <Pg Up> six times.

LESSON: Screen 10

.[11] FUNCTION: Showing Action or Condition/State (NEXT)[12]

Individual words or groups of words can be used to show action by or to named things or can be used to show the condition or state existence of named things.

To identify a word or group of words that shows action, ask questions like: Does what? Did what? Had done what? Will do what? Is what? Was what? Had been what? Will be what?

To identify a word or group of words that shows existence, ask yourself if it gives existence or describes the state or condition of a naming element.

In the following examples, individual words and groups of words that show action or existence are underlined. Review each example and ask yourself the questions shown to see how the underlined words or groups of words show action or existence.

LESSON: Screen 11

.[12] FUNCTION: Showing Action/Condition/State (cont.) (NEXT)[13]

1. Pilots fly.

   Pilots do what? -> fly

2. The pilot flew the airplane.

   The pilot did what? -> flew

3. The pilot is flying the airplane.

   The pilot is what? -> is flying

LESSON: Screen 12

```
.[13]    FUNCTION:  Showing Action/Condition/State (cont.)  (NEXT)[14]

4.  The pilot has flown over 100 different airplanes.
             ---------

    The pilot has done what?  ->  has flown

5.  The students entered the classroom.
                 -------

    The students did what?  ->  entered

6.  The students are entering the classroom.
                 ------------

    The students are what?  -> are entering
```

LESSON: Screen 13

```
.[14]    FUNCTION:  Showing Action/Condition/State (cont.)  (NEXT)[15]

7.  The students had entered the classroom early today.
                 -----------

    The students had done what?  ->  had entered

8.  The students will be leaving school early today.
                 ---------------

    The students will be what?  -> will be leaving

9.  The students were very enthusiastic at today's assembly.
                 ----

    Condition or state?  -> were
```

LESSON: Screen 14

.[15] FUNCTION: Showing Action/Condition/State (cont.) (NEXT)[16]

10. The classroom seemed unusually cold this morning.

Condition or state? -> seemed

11. The audience grew restless in spite of their patience.

Condition or state? -> grew

12. The audience remained standing throughout the performance.

Condition or state? -> remained

LESSON: Screen 15

FUNCTION: Showing Action/Condition/State
REVIEW

1 - Individual words or groups of words can be used to show action that is performed by or received by named things, or they can be used to show existence of named things.

2 - To determine if a word or group of words is performing this function, ask the following kinds of questions:

Does what? Did what? Had done what? Will do what? Is what? Was what? Had been what? Will be what?

To return to the FUNCTIONS MENU, press <Pg Up> six times.

LESSON: Screen 16

.[17] FUNCTION: Changing, Qualifying, or Renaming (NEXT)[18]

Individual words or groups of words can be used to change, qualify, or rename other words or groups of words. This function is called the "modifying" function.

Modifiers can be identified by asking any of the following questions:

Where? Why? How? When? How many? Which one? What kind?

In the following examples, individual words or groups of words that change, qualify, or rename other words or groups of words are underlined. As you read each example, see how the underlined modifiers answer the questions shown.

LESSON: Screen 17

.[18] FUNCTION: Modifying (cont.) (NEXT)[19]

1. The deer entered the pasture.

(No modifiers)

NOTE: In this lesson, the articles[74] (a, an, the) are classified as "fillers" rather than modifiers.

2. The young deer entered the pasture.

Which deer? -> the YOUNG deer

3. The young deer entered the beautiful pasture.

Which pasture? -> the BEAUTIFUL pasture

LESSON: Screen 18

.[19] FUNCTION: Modifying (cont.) (NEXT)[20]

4. The young deer cautiously entered the beautiful pasture.

How? -> cautiously

5. The young deer entered the beautiful pasture slowly.

How -> slowly

6. The professor teaches economics.

(No modifiers)

.[20] FUNCTION: Modifying (cont.) (NEXT)[21]

7. Our professor teaches economics.

Which professor? -> OUR professor

8. Our favorite professor teaches economics.

Which of our professors? our FAVORITE professor

9. The professor that we think is best teaches economics.

Which professor? -> the professor THAT WE THINK IS BEST

LESSON: Screen 20

.[21] FUNCTION: Modifying (cont.) (NEXT)[22]

10. The professor that teaches economics has arrived.

Which professor? -> the professor THAT TEACHES ECONOMICS

11. Our professor, Dr. Johnson, teaches three courses.

Which professor? -> Dr. Johnson

12. The professor teaches three courses.

How many courses? -> three

LESSON: Screen 21

.[22] FUNCTION: Modifying (cont.) (NEXT)[23]

13. The professor teaches the course whenever her schedule allows.

When? -> whenever her schedule allows

14. We arrived early because we wanted to get good seats.

Why? -> because we wanted to get good seats

15. We arrived early to get good seats.

Why? -> to get good seats

LESSON: Screen 22

FUNCTION: Modifying
REVIEW

1 - Individual words or groups of words can be used to change, qualify, or rename other words or groups of words.

2 - Words or groups of words that perform this function are called modifiers.

3 - Modifiers can be identified by asking questions like:

Where? Why? How? When? How many? Which one? What kind?

To return to the FUNCTIONS MENU, press <Pg Up> six times.

LESSON: Screen 23

.[24] FUNCTION: Connecting Words or Groups of Words (NEXT)[25]

Individual words or groups of words can be used to connect other words or groups of words. Through this connection, they signal a relationship between the things that they connect.

In the following examples, individual words or groups of words that connect other words or groups of words are underlined. As you read each example, see how the connections are made and how the connections say something about the relationship between the connected words.

1. Pilots and co-pilots fly airplanes.
---

2. Pilots can fly airplanes or helicopters.
--

3. The students will have a review today and take the test tomorrow.

LESSON: Screen 24

.[25] FUNCTION: Connecting (cont.) (NEXT)[26]

4. You may take that course either this quarter or next quarter.
------ --

5. I enjoyed that class so I am taking its follow-on course.
--

6. We can take this course after we have taken its prerequisite.
-----

7. We took the test today rather than waiting until tomorrow.
----

8. The lecture was not only entertaining but also informative.
-------- --------

9. John is a full-time student, and he works full time.

LESSON: Screen 25

.[26] FUNCTION: Connecting (cont.) (NEXT)[27]

10. We reviewed our notes while we waited for the professor.
-----

11. While we waited for the professor, we reviewed our notes.
-----

12. The lightbulb in the kitchen needs to be replaced.
--

13. They live in the house on the corner.
-- --

14. He gave the first-place trophy to the winners.
--

15. The students entered the auditorium before the guest speaker.

LESSON: Screen 26

FUNCTION: Connecting
REVIEW

1 - Individual words or groups of words can be used to connect other words or groups of words.

2 - When the connection is made, it says something about the relationship between the things joined.

To return to the FUNCTIONS MENU, press <Pg Up> four times.

LESSON: Screen 27

.[28] FUNCTION: Adding Filler (NEXT)[29]

If a word or group of words does not perform any of the other four functions (naming, showing action/existence, modifying, or connecting), then it is a "filler."

Fillers can be used for several purposes as the following examples show. In each example, the filler word or group of words is underlined. As you read each example, see how the fillers fail to perform any of the other four functions mentioned above.

1. Ouch--that hurt!

   This filler is used to show emotion.

LESSON: Screen 28

.[29] FUNCTION: Adding Filler (cont.) (NEXT)[30]

2. Oh my! I wish I would have remembered my notebook.

   This filler is also used to show emotion.

3. The book was very interesting.

   This filler signals that a naming element will follow.

4. Dr Johnson is away at a scheduling conference.
   -

   This filler signals that a naming element will follow.

LESSON: Screen 29

.[30] FUNCTION: Adding Filler (cont.) (NEXT)[31]

5. There are several students in the classroom.

   This filler is necessary as a place-holder because of the order of the other words. It can be avoided by rearranging the other words:

   "Several students are in the classroom."

6. It is important to follow directions.
   --

   This filler is a place-holder. It can be avoided by rearranging the other words:

   "To follow directions is important."

LESSON: Screen 30

FUNCTION: Adding Filler
REVIEW

1 - Sometimes words or groups of words do not name things, show action or existence of things, modify things, or connect things.

2 - These words or groups of words are fillers. Fillers can

-- show emotion,

-- signal that a naming thing follows, or

-- serve as place-holders.

To return to the FUNCTIONS MENU, press <Pg Up> four times.

LESSON: Screen 31

. [32] BUILDING ELEMENTS (NEXT)[33]

The first part of this lesson described the five functions that words or groups of words can perform. These five functions are:

1 - Naming things
2 - Showing action/existence of things
3 - Changing, qualifying, or renaming things
4 - Connect words or groups of words
5 - Add filler to groups of words

In this part of the lesson, we will look more closely at the words and groups of words that perform these five functions. We will call them the "building elements" of English grammar.

The building elements in English are:

1 - Words
2 - Phrases
3 - Clauses

LESSON: Screen 32

.[33] BUILDING ELEMENTS (cont.) (NEXT)[34]

1) A WORD is a symbol that represents a concept or idea.

   EXAMPLES: desk, Sarah, Dayton, small, happy, in, but, the

2) PHRASES are the next larger building element. A phrase has the following characteristics:

   1 - It contains more than one word.

   2 - It performs a single function within a larger structure.

   3 - If it contains both a naming element and an action/existence element, they do not operate in a subject/verb[98] relationship.

   EXAMPLES: in the hallway
   are going
   exercising three times a week
   to get our attention

LESSON: Screen 33

.[34] BUILDING ELEMENTS (cont.) (NEXT)[35]

3) CLAUSES are the next larger building element. If a group of words DOES contain a naming element and an action/existence element that operate in subject/verb[98] relationship, it is a clause.

   EXAMPLES: pilots fly
   the students will graduate next month
   when the test is completed
   the book that I borrowed from the library
   although we knew the material

LESSON: Screen 34

.[35] BUILDING ELEMENTS MENU

The remainder of this section will cover each of the building elements in more detail. Select the element you would like to review from the list below.

1[36] - WORDS
2[77] - PHRASES
3[97] - CLAUSES

After you are familiar with the three building elements of grammar and how they perform the five functions of grammar, you should return to the MAIN MENU and select Item 4 - THE SENTENCE. That section will show you how the building elements are put together for effective communication.

To return to the MAIN MENU, press <Pg Up> four times.

LESSON: Screen 35

.[36] BUILDING ELEMENTS: WORDS

The most basic unit in language is the word. A word is a symbol that represents a concept or idea. In effective communication, it is important that we select the right word for the concept or idea that we are trying to identify. Words perform each of the five functions of English grammar. Words can:

1[37] - name things
2[51] - show action or existence of named things
3[60] - change, qualify, or rename things
4[68] - connect words or groups of words
5[73] - add filler to groups of words

To review how words are used to perform each functions, select its number from the list above.

To return to the Elements Menu, press <Pg Up>.

LESSON: Screen 36

.[37] WORDS: Naming Function (NEXT)[38]

Words can function as the names of people, places, things, or ideas. There are three properties of words that perform the naming function:

1[40] - Person
2[41] - Number
3[42] - Gender

The naming function can be performed by three types of words:

- Nouns
- Pronouns
- Gerunds

NOUNS are words that are used to give actual names to things

EXAMPLES: table, student, Dayton, thunder, leadership

For more detailed information on nouns, select here -> NOUNS[43]

LESSON: Screen 37

.[36] WORDS: Naming Function (NEXT)[39]

PRONOUNS are words that take the place of nouns. The noun that is replaced by a pronoun is called the pronoun's ANTECEDENT. (The antecedent should normally precede the pronoun so the reader can more easily identify what noun has been replaced.

EXAMPLES OF ANTECEDENTS AND PRONOUNS:

John	->	he
Susan	->	she
(the writer)	->	I
(the reader)	->	you
John and Susan	->	they
book	->	it
JOHN'S book	->	HIS book
SUSAN'S book	->	HER book

For more detailed information on pronouns select here -> PRONOUNS[44]

LESSON: Screen 38

WORDS: Naming Function (cont.)

GERUNDS fall into a special category of naming words. Gerunds appear as if they are action-showing words, but are used to name concepts or ideas. In the following examples, notice how the word appearing alone seems to show action, but within a group of words, it performs the naming function.

ALONE	WITHIN A GROUP OF WORDS
Smoking	Smoking is unhealthy. (What is? -> smoking)
Running	Running is a healthy form of exercise. (What is? -> running)
Broiled	Broiled is a good way to eat steak. (What is? -> broiled)

To return to the WORDS MENU, press <Pg Up> three times.

LESSON: Screen 39

PROPERTIES OF NAMING WORDS: Person

PERSON refers to the relationship among the writer, the reader, and other named things in a group of words:

a) First person: the writer(s)
Examples: I, me, we, us

b) Second person: the reader(s)
Example: you

c) Third person: things written about
Examples: he, she, it, they, desk

Press <Pg Up>

LESSON: Screen 40

PROPERTIES OF NAMING WORDS: Number

NUMBER refers to the number of things named:

a) Singular: Indicates one person or thing
Examples: I, you, desk

b) Plural: Indicates more than one person or thing
Examples: we, you, they, students, books

c) Collective: Singular words that represent groups of things
Examples: family, class, committee

Press <Pg Up>

LESSON: Screen 41

PROPERTIES OF NAMING WORDS: Gender

GENDER refers to the sex of the named thing(s):

a) Masculine: person or animal of the male sex
Examples: man, boy, rooster, bull, he

b) Feminine: person or animal of the female sex
Examples: woman, girl, hen, cow, she

c) Common: a thing that is named without indicating its sex
Examples: child, dog, someone

d) Neuter: a thing that is neither male nor female
Examples: book, table, leadership, thunder

Press <Pg Up>

LESSON: Screen 42

WORDS: NAMING FUNCTION -- NOUNS

Nouns can be further broken down into the following categories:

COMMON NOUN: general people, places, or things
EXAMPLES: student, textbook, city, automobile

PROPER NOUN: specific people, places, or things
EXAMPLES: Bill, Dr. Johnson, Dayton, American, Chrysler

COMPOUND NOUN: two or more words that function together as a single unit and are treated as a single word
EXAMPLES: mother-in-law, Air Force, University of Dayton, Hudson River

COLLECTIVE NOUN: singular words that represent groups of things
EXAMPLES: family, class, council, committee

Press <Pg Up>

LESSON: Screen 43

.[44] WORDS: NAMING FUNCTION -- PRONOUNS (NEXT)[45]

Pronouns can be further broken down into several categories:

PERSONAL PRONOUN: replaces a noun that names the writer, reader, or other thing written about. The noun replaced is called the pronoun's ANTECEDENT. In English, the personal pronouns are:

	SINGULAR	PLURAL
FIRST PERSON:	I (me, my)	we (us, ours)
SECOND PERSON:	you (you, yours)	you (you, yours)
THIRD PERSON:	he, she, it (him, her, his, hers, it, its)	they (them, theirs)

LESSON: Screen 44

.[45] PRONOUNS (cont.) (NEXT)[46]

DEMONSTRATIVE PRONOUN: points out its antecedent

In English there are four demonstrative pronouns:

SINGULAR: this -> This is my book.
that -> That is my book.

PLURAL: these -> These are my books.
those -> Those are my books.

(NOTE: These four words can also be used as adjectives[60]).

LESSON: Screen 45

.[46] PRONOUNS (cont.) (NEXT)[47]

INDEFINITE PRONOUN: refers to nonspecific things:

SINGULAR: each, none, some, all, one, either, neither, anyone, everyone, someone, anybody, everybody, somebody, nobody, anything, everything, something, nothing

PLURAL: none, both, few, some, several, many, all

NOTE: There is a difference between the singular and plural uses of the words NONE, SOME, and ALL. For things that are uncountable quantities (soup, sand, pie) use a singular verb. For things measured in NUMBER, use a plural verb.

None (some, all) of the pie was eaten.

None (some, all) of the students were absent.

LESSON: Screen 46

.[47] PRONOUNS (cont.) (NEXT)[48]

POSSESSIVE PRONOUN: shows possession

SINGULAR

first person:	mine	->	The book is mine.
second person:	yours	->	The book is yours.
third person:	his	->	The book is his.
	hers	->	The book is hers.

PLURAL

first person:	ours	->	The book is ours.
second person:	yours	->	The book is yours.
third person:	theirs	->	The book is theirs.

LESSON: Screen 47

.[48] PRONOUNS (cont.) (NEXT)[49]

REFLEXIVE PRONOUN: refers to the subject[98] of a group of words and receives the action of the subject. These are words that end in "self" (singular) or "selves" (plural).

EXAMPLES:

I taught myself to juggle.
They congratulated themselves on their victory.

INTENSIVE PRONOUN: emphasizes the identity of a naming element. The intensive pronoun serves as an appositive[62]. These words also end in "self" (singular) or "selves" (plural).

EXAMPLES:

You should do the work yourself.
We did the work ourselves.

LESSON: Screen 48

.[49] PRONOUNS (cont.) (NEXT)[50]

RECIPROCAL PRONOUN: refers to individual parts of a plural antecedent
The reciprocal pronouns are:

each other -> They exchanged notes with each other.
one another -> Students should help one another.

RELATIVE PRONOUN: introduce noun_clauses[114] and adjective_clauses[117]

EXAMPLES: that -> The team that wins will get the prize.
who (whom) -> Who will win is a foregone conclusion.
whose -> The student whose name is called will answer.

INTERROGATIVE PRONOUN: introduce a question

EXAMPLES: what -> What did the professor say?
who -> Who submitted the purchase order?
whom -> John asked, "Whom shall we call?"
which -> Which class are you takin...

LESSON: Screen 49

PRONOUNS (cont.)

NUMERAL PRONOUN: replaces an antecedent with its number

EXAMPLES:

(The professor gave the students the test results.)

One received a perfect score.

Two were above average.

The professor assigned failing grades to four.

Press <Pg Up> seven times.

LESSON: Screen 50

.[51] WORDS: SHOWING ACTION OR CONDITION/STATE (NEXT)[52]

Words that are used to show action or condition/state of named thing are called VERBS.

Verbs that show action are called ACTION VERBS.

EXAMPLES: students STUDY
pilots FLY
donkeys KICK

Verbs that show condition/state are called LINKING VERBS. Words that are commonly used as linking verbs include be, feel, seem, appear, grow, become, and remain.

EXAMPLES: they APPEAR anxious
we FEEL prepared for the test

The distinction between action verbs and linking verbs will be discussed further in the section on clauses[103].

LESSON: Screen 51

.[52] WORDS: VERBS (cont.) (NEXT)[53]

The appropriate verb form to use for a particular action or existence depends on three properties: tense, person, and number. Here are descriptions and examples of each of these properties.

TENSE refers to the time of the action or existence. We will discuss six tenses in English grammar.

SIMPLE TENSES show the relation of the time of the action/existence to the time that the thought is communicated.

Present Tense: the action/existence occurs at the same time as it is communicated ("we finish")

Past Tense: the action/existence occurs prior to the communication ("we finished")

Future Tense: the action/existence occurs after the communication ("we will finish")

LESSON: Screen 52

.[53] WORDS: VERBS (cont.) (NEXT)[54]

PERFECT TENSES select one of three points in time as a point of reference and show that the action/condition occurred prior to that time.

Present Perfect Tense:	the point of reference is the present time ("we have finished")
Past Perfect Tense:	the point of reference is in the past ("we had finished before yesterday")
Future Perfect Tense:	the point of reference is in the future ("we will have finished by tomorrow")

LESSON: Screen 53

.[54] WORDS: VERBS (cont.) (NEXT)[55]

The **PERSON** and **NUMBER** of a verb must match the person and number of the thing(s) acting/existing, as described in the section on properties[37] of naming words.

WRONG	RIGHT
they is present	they are present
he run every day	he runs every day
they seems happy	they seem happy

LESSON: Screen 54

.[55] WORDS: VERBS (cont.) (NEXT)[56]

The fundamental form of a every verb is its infinitive. The infinitive is always a two-word structure: the word "to" plus the basic verb word. This basic verb word is called the basic infinitive form.

EXAMPLES: to be to study to eat to take to give

The following examples will conjugate (show the various forms of) the verb "TO JUMP."

LESSON: Screen 55

.[56] VERB TENSES (NEXT)[57]

The PRESENT TENSE is normally formed by using the basic infinitive form of the verb. In THIRD PERSON SINGULAR, an "s" is added to the basic infinitive form.

	SINGULAR	PLURAL
1st Person	I jump	we jump
2nd Person	you jump	you jump
3rd Person	he/she/it jumps	they jump

The PAST TENSE of most verbs is formed by adding "ed", "d", or "t" to the basic infinitive form of the verb. A few verbs form the past tense by changing an internal vowel (run --> ran). If you are unsure of the proper past tense of a verb, refer to a dictionary.

1st Person	I jumped	we jumped
2nd Person	you jumped	you jumped
3rd Person	he/she/it jumped	they jumped

LESSON: Screen 56

.[57] VERB TENSES (cont.) (NEXT)[58]

The FUTURE TENSE is formed by preceding the basic infinitive form of the verb with the word "will."

	SINGULAR	PLURAL
1st Person	I will jump	we will jump
2nd Person	you will jump	you will jump
3rd Person	he/she/it will jump	they will jump

The PRESENT PERFECT TENSE is normally formed by preceding the past tense form of the verb with "have." In THIRD PERSON SINGULAR, the past tense form is preceded by the word "has."

1st Person	I have jumped	we have jumped
2nd Person	you have jumped	you have jumped
3rd Person	he/she/it has jumped	they have jumped

LESSON: Screen 57

.[58] VERB TENSES (cont.) (NEXT)[59]

The PAST PERFECT TENSE is formed by preceding the past tense form of the verb with the word "had."

	SINGULAR	PLURAL
1st Person	I had jumped	we had jumped
2nd Person	you had jumped	you had jumped
3rd Person	he/she/it had jumped	they had jumped

The FUTURE PERFECT TENSE is formed by preceding the past tense form of the verb with the words "will have."

1st Person	I will have jumped	we will have jumped
2nd Person	you will have jumped	you will have jumped
3rd Person	he/she/it will have jumped	they will have jumped

LESSON: Screen 58

WORDS: VERBS (cont.)

Conjugating verbs in English is often complex because many verbs have irregular forms. For further details about verbs and verb conjugation, you should refer to the front of a dictionary or to a writer's handbook.

To return to the WORDS MENU, press <Pg Up> nine times.

LESSON: Screen 59

.[60] WORDS: MODIFYING (NEXT)[61]

Words that are used to change, qualify, or rename things are called modifiers.

1. A word that modifies a naming element is called an ADJECTIVE[63]. Adjectives are recognizable because they answer the questions:

   Which one? What kind? How many?

   EXAMPLES: the YOUNG deer (Which deer? -> young)
   BLUEBERRY pie (What kind? -> blueberry)
   THREE books (How many? -> three)

   For more detailed information about adjectives, select the word "adjective" above.

LESSON: Screen 60

.[61] WORDS: MODIFYING (cont.) (NEXT)[62]

2. A word that modifies anything other than a naming element is called an ADVERB[66]. Adverbs are often recognizable because they answer the questions: How? When? Where? Why?

EXAMPLES: ran QUICKLY (How? -> quickly)
SILENTLY approached (How? -> silently)

Adverbs are often formed by adding "-ly" to adjectives:

The LOUD student answered the question. (adjective)
The student answered the question LOUDLY. (adverb)

There is a special form of adverb called a conjunctive adverb. This concept will be discussed in the section on sentences[126].

For more detailed information about adverbs, select the word "adverb" above.

LESSON: Screen 61

WORDS: MODIFYING (cont.)

3. A special category of modifier is the APPOSITIVE. An appositive is a word that immediately follows something performing a naming function and renames the same thing. An appositive is normally set off by commas.

EXAMPLES: My good friend, John, is coming to visit.

Our first class, biology, was cancelled.

To return to the WORDS MENU, press <Pg Up> three times.

If you jumped here from Intensive Pronouns, press <Pg Up> only once.)

LESSON: Screen 62

.[63] ADJECTIVES (NEXT)[64]

There are several categories of adjectives:

DESCRIPTIVE ADJECTIVE: used to show quality or intensity

EXAMPLES: the BRIGHT light
the LARGER classroom

PROPER ADJECTIVE: formed from proper nouns

EXAMPLES: the AMERICAN flag
their ACHILLES heel

DEMONSTRATIVE ADJECTIVE: used to point out particular things. In English, there are four demonstrative adjectives:

this that these those

EXAMPLES: this book, that book, these books, those books

LESSON: Screen 63

.[64] ADJECTIVES (cont.) (NEXT)[65]

INTERROGATIVE ADJECTIVE: used to introduce a question about a named thing. In English there are three interrogative adjectives:

what, which, whose

EXAMPLES: WHAT course are you taking?
WHICH professor teaches it?
WHOSE books are those?

NUMERAL ADJECTIVE: indicates number or numerical order. They are

Number: one, two, three, four, etc
Numeric order: first, second, third, fourth, etc

EXAMPLES: one book, two students, three professors
first arrival, second period, third base

LESSON: Screen 64

ADJECTIVES (cont.)

INDEFINITE ADJECTIVE: indicates naming elements in a general way but does not definitely identify them. Indefinite adjectives include:

each	every	neither	much	any	some
both	few	several	many	all	no

EXAMPLES: Many students attended the study session.

The professor advised several students to attend.

Press <Pg Up> three times.

LESSON: Screen 65

.[66] ADVERBS (NEXT)[67]

An adverb can modify anything other than a naming element. (Naming elements are modified by adjectives.)

Adverbs often answer the questions: How, when, where, or why?

EXAMPLES: she answered slowly
------ (modifies the verb "answered")

the very loud noise
---- (modifies the adjective "loud")

I studied yesterday
--------- (modifies the verb "studied")

they climbed down
---- (modifies the verb climbed)

LESSON: Screen 66

ADVERBS (cont.)

MORE EXAMPLES:

a rather plain meal
  ------ (modifies the adjective "plain")

he sang very well
      ---- (modifies the adverb "well")

they shouted rather loudly
            ------ ------
("loudly" modifies "shouted")
("rather" modifies "loudly")

Adverbs that answer the question "why" form a special group called adverb clauses[120]. These will be discussed in the clause section.

Press <Page Up> twice.

LESSON: Screen 67

.[68] WORDS: CONNECTING FUNCTION (NEXT)[69]

Some words act as connectors between other words or groups of words.

1 - One type of connecting word is the COORDINATING CONJUNCTION. A coordinating conjunction connects two or more of the building elements of the same structure. In English, there are seven coordinating conjunctions:

and but for nor or so yet

EXAMPLES: John OR Susan
over the river AND through the woods
the students nor their instructors
I wanted to buy the car, BUT I could not..

LESSON: Screen 68

.[69] WORDS: CONNECTING FUNCTION (cont.) (NEXT)[70]

2 - Sometimes the elements connected by a coordinating conjunction are preceded by another word, for example:

EITHER John OR Susan
BOTH over the river AND through the woods
NEITHER the students NOR their instructors

These pairs of connecting words are known as CORRELATIVE CONJUNCTIONS.

LESSON: Screen 69

.[70] WORDS: Connecting Function (cont.) (NEXT)[71]

3 - A special group of connectors are called prepositions. A preposition connects a naming element to a group of words in such a way that it shows a relationship (such as place or time) between the naming element and the group of words.

EXAMPLES: the lamp -> the table
The lamp ON the table

("On" is a preposition that shows the relationship between "the lamp" and "the table.")

I drove my car -> the bridge.
I drove my car ACROSS the bridge.

("Across" is a preposition that shows the relationship between "drove my car" and "the bridge.")

For a list of common prepositions, select here -> PREPOSITION[72]

LESSON: Screen 70

WORDS: CONNECTING FUNCTION (cont.)

4 - A fourth type of connecting word is called a SUBORDINATING[106] CONJUNCTION. This will be discussed in the section on clauses.

5 - A fifth type of connecting word is called a CONJUNCTIVE[126] ADVERB. This will be discussed in the section on sentences.

To return to the WORDS MENU, press <Pg Up> four times.

LESSON: Screen 71

COMMON PREPOSITIONS

Here is a list of common prepositions:

about	before	during	near	until
above	behind	except	of	up
across	below	excluding	on	upon
after	beneath	following	over	via
against	beside	from	since	with
along	between	in	through	within
amid	beyond	including	to	without
among	by	inside	toward	
at	despite	into	under	
as	down	like	underneath	

NOTE: In rare cases, a preposition may consist of more than one word (in spite of, because of, due to)

Press <Pg Up>.

LESSON: Screen 72

WORDS: FILLER FUNCTION

Four of the functions that words can perform are:

1 - naming things
2 - showing action or existence
3 - Modifying (changing, qualifying, or renaming)
4 - connecting words or groups of words

Some words do not perform any of these functions. The function of these words is to provide "filler" to groups of words. There are three types of such filler words:

1[74] - ARTICLES
2[75] - EXPLETIVES
3[76] - INTERJECTIONS

To review each of these filler words, select it from the list above.

To return to the WORDS MENU, press <Pg Up>.

LESSON: Screen 73

WORDS: FILLER FUNCTION -- Articles

In English, nouns [43] are often preceded by small words that signal to the reader that a noun is to follow. There are three words in English that perform this filler function:

a an the

EXAMPLES:	A book	AN easel	THE professor
	A student	AN automobile	THE classroom

The articles "a" and "an" are called indefinite articles because they do not identify a specific noun. The article "the" is called a definite article because it does identify a specific noun.

(NOTE: The articles are sometimes labelled adjectives[60] because they qualify the nouns that follow them.)

Press <Pg Up>.

LESSON: Screen 74

WORDS: FILLER FUNCTION -- Expletives

Sometimes words are used as filler in thoughts or ideas that contain the existence function. These words, called EXPLETIVES, are like pronouns since they are used to introduce sentences. But unlike a pronoun, an expletive disappears if the words are placed in a different order. Two words are often used to perform this function: "it" and "there."

EXAMPLES: a. It is necessary to pass this class.
--
Rearranged: To pass this class is necessary.

b. There are several students in this class.

Rearranged: Several students are in this class.

Press <Pg Up>.

LESSON: Screen 75

WORDS: FILLER FUNCTION -- Interjections

Some words are used to convey surprise or other emotions. Words that perform this filler function are called INTERJECTIONS. If used alone, interjections are usually followed by an exclamation point. If joined with other words, they are usually set off with a comma.

EXAMPLES: Oh, did you say something to me?
--

Ouch! I hit my thumb with the hammer!

Well, that explains things.

(NOTE: Interjections are not normally used in expository writing.)

Press <Pg Up>.

LESSON: Screen 76

.[77] BUILDING ELEMENTS: PHRASES (NEXT)[78]

Words by themselves are basic building elements in language. But words are seldom used alone. The next larger building element in language is the PHRASE.

A phrase has the following characteristics:

1 - It contains more than one word.
2 - It performs a single function within a larger structure.
3 - If it contains both a naming element and an action/existence element, they do not operate in a subject/verb[98] relationship.

EXAMPLES: in the hallway
are going
exercising three times a week
to get our attention

LESSON: Screen 77

.[78] BUILDING ELEMENTS: PHRASES (cont.)

A phrase can perform any of the five functions of grammar:

1[79] - Naming things
2[84] - Showing action or existence
3[86] - Modifying (Changing, qualifying, or renaming)
4[95] - Connecting words or groups of words
5[96] - Adding filler

To review each function of phrases, select it from the list above.

To return to the BUILDING ELEMENTS MENU, press <Pg Up> twice.

LESSON: Screen 78

PHRASES: NAMING FUNCTION

A phrase performs the naming function if it answers the questions:

"Who?" or "What?"

There are two types of phrases that commonly perform this function:

1[80] - GERUND PHRASE
2[82] - INFINITIVE PHRASE

To review each of these naming functions of phrases, select it from the list above.

To return to the PHRASES MENU, press <Pg Up>.

LESSON: Screen 79

.[80] PHRASES: NAMING FUNCTION -- Gerund Phrase (NEXT)[81]

A GERUND PHRASE consists of a GERUND[39] and any of its objects or modifiers. The object of a gerund is the word or words that receive the "action" of the gerund in the phrase.

EXAMPLES:

a. Walking your dog will keep both of you healthy.
   ------- --------
   GERUND + OBJECT = GERUND PHRASE

   (What keeps you both healthy? -> walking your dog)

b. Proofreading your work helps you prevent errors.
   ------------ ---------
   GERUND + OBJECT = GERUND PHRASE

   (What helps prevent errors? -> proofreading your work)

LESSON: Screen 80

GERUND PHRASES (cont.)

MORE EXAMPLES:

c. Exercising three times a week will help your heart.
---------- ------------------
GERUND + MODIFIER = GERUND PHRASE

(What helps your heart? -> exercising three times a week)

d. One way to get their attention is speaking softly.
-------- ------
GERUND + MODIFIER = GERUND PHRASE

(What will get their attention? -> speaking softly)

Press <Pg Up> twice.

LESSON: Screen 81

.[82] PHRASES: NAMING FUNCTION -- Infinitive Phrase (NEXT)[83]

Another type of phrase that performs the naming function is the INFINITIVE PHRASE. The infinitive phrase consists of an infinitive[55] and its objects or modifiers. The object of an infinitive is the word or words that receive the "action" of the infinitive.

EXAMPLES:

a. To publish their book was their biggest ambition.
---------- ----------
INFINITIVE + OBJECT = INFINITIVE PHRASE

(What was their ambition? -> to publish their book)

LESSON: Screen 82

INFINITIVE PHRASES (cont.)

MORE EXAMPLES:

b. They encouraged the fans to cheer enthusiastically.
-------- ---------------

INFINITIVE + MODIFIER = INFINITIVE PHRASE

(What did they encourage? -> to cheer enthusiastically)

c. To learn their lessons completely is their goal.
-------- ------------ ----------

INFINITIVE + OBJECT + MODIFIER = INFINITIVE PHRASE

(What is their goal? -> to learn their lessons completely)

Press <Pg Up> twice.

LESSON: Screen 83

.[84] PHRASES: SHOWING ACTION/CONDITION/STATE (NEXT)[85]

The second function that phrases can perform is to show action or existence of named things.

The phrases that perform this function are called VERB PHRASES.

In some verb tenses[52], the action must be shown by more than one word. In these cases, a verb phrase is used. Note that in each verb phrase, there are two parts:

1 - A form of the verb TO BE or the verb TO HAVE.

2 - Another word ending in either "-ing" or "-ed" (or a non-regular equivalent). The "-ing" word is called a PRESENT PARTICIPLE. The "-ed" word is called a PAST PARTICIPLE.

LESSON: Screen 84

PHRASES: SHOWING ACTION/CONDITION/STATE (cont.)

EXAMPLES:

a. We are taking the final examination next week.
   ----

b. By the end of the day, we had purchased everything we wanted.
                              ----

c. When he takes the examination, he will have studied for days.
                                   ----

d. The manager has been seeking a raise for several months.
                 ----

e. The ball was kicked by the punter.

To return to the PHRASES MENU, press <Pg Up> twice.

LESSON: Screen 85

PHRASES: MODIFYING FUNCTION

When building elements perform the modifying function, they change, qualify, or rename a word or group of words. The following types of phrases can be used to perform the modifying function:

1[87] - PARTICIPIAL PHRASE
2[90] - PREPOSITIONAL PHRASE
3[92] - INFINITIVE PHRASE
4[94] - APPOSITIVE PHRASE

To review each of these modifying functions of phrases, select it from the list above.

To return to the PHRASE MENU, press <Pg Up>.

LESSON: Screen 86

.[87] PHRASES: MODIFYING FUNCTION -- Participial Phrase (NEXT)[88]

A PARTICIPIAL PHRASE functions like an adjective[60]. It modifies a naming element. It consists of a participle[84] and any objects or modifiers of the participle. The object of a participle is the word or words that receive the "action" of the participle.

EXAMPLES:

a. The students taking calculus will have a quiz today.
------ --------

PARTICIPLE + OBJECT
(PARTICIPIAL PHRASE modifying "students")

b. The tattered page, torn from its book, was blown away.
---- ------------

PARTICIPLE + MODIFIER
(PARTICIPIAL PHRASE modifying "page")

LESSON: Screen 87

.[88] PARTICIPIAL PHRASES (cont.) (NEXT)[89]

c. Clearing her throat, she caught our attention.
-------- ----------

PARTICIPLE + OBJECT
(PARTICIPIAL PHRASE modifying "she")

d. Slowly entering the auditorium, he captivated the audience.
------ -------- -------------

MODIFIER + PARTICIPLE + OBJECT
(PARTICIPIAL PHRASE modifying "he")

LESSON: Screen 88

PARTICIPIAL PHRASES (cont.)

A special type of participial phrase is the **ABSOLUTE PHRASE**, which modifies an entire thought or idea. It consists of a naming element, a verb participle without a helping_verb[84], and any modifiers.

EXAMPLES:

a. The dog welcomed us home, its tail wagging eagerly.

NAMING ELEMENT + PARTICIPLE + MODIFIER = ABSOLUTE PHRASE

b. The movie having ended late, we went straight home.

NAMING ELEMENT + PARTICIPLE + MODIFIER = ABSOLUTE PHRASE

c. The students cheering loudly, the graduation ceremony ended.

NAMING ELEMENT + PARTICIPLE + MODIFIER = ABSOLUTE PHRASE

Press <Pg Up> three times.

LESSON: Screen 89

.[90] PHRASES: MODIFYING FUNCTION -- Prepositional Phrase (NEXT)[91]

A **PREPOSITIONAL PHRASE** functions either like an adjective (modifying a naming element) or like an adverb (modifying an action/existence element). It consists of a preposition and any objects and modifiers of the object.

EXAMPLES:

a. The lamp on the table is broken.

(like an adjective, since it modifies the word "lamp")

b. The cars raced down the street.

(like an adverb, since it modifies the word "raced")

LESSON: Screen 90

PREPOSITIONAL PHRASES (cont.)

NOTE: The object of a preposition may itself be modified by another prepositional phrase.

c. They lived in the house on the corner.

("on the corner is like an adjective, modifying the word "house")

("in the house on the corner is like an adverb modifying the word "lived")

Press <Pg Up> twice.

LESSON: Screen 91

.[92] PHRASES: MODIFYING FUNCTION -- Infinitive Phrase (NEXT)[93]

An INFINITIVE PHRASE can act either like an adjective (modifying a naming element) or like an adverb (modifying an action/ existence element). A infinitive phrase consists of an infinitive[55] and any objects and modifiers.

EXAMPLES:

a. He made every effort to influence his students.

INFINITIVE + OBJECT =INFINITIVE PHRASE
(like an adjective modifying the word "effort")

b. They knew this was their last chance to ask questions.

INFINITIVE + OBJECT = INFINITIVE PHRASE
(like an adjective modifying the word "chance")

LESSON: Screen 92

INFINITIVE PHRASE (cont.)

c. They waved to get our attention.
------ ----------

INFINITIVE + OBJECT = INFINITIVE PHRASE
(like an adverb modifying the word "waved")

d. They studied every night to prepare properly for the test.
---------- -------- -----------

INFINITIVE + MODIFIER + MODIFIER = INFINITIVE PHRASE
(like an adverb modifying the word "reviewed")

Press <Pg Up> twice.

LESSON: Screen 93

PHRASES: MODIFYING FUNCTION -- Appositive Phrase

An APPOSITIVE PHRASE is a group of words performing a naming function that immediately follows something else performing a naming function and renames the same thing. An appositive phrase is normally set off by commas.

EXAMPLES:

a. Dr. Williams, our favorite professor, will retire next year.
----

b. They recommended the house specialty, spaghetti and meatballs.
----

c. The exercise, running in place, was very strenuous.

Press <Pg Up>.

LESSON: Screen 94

PHRASES: CONNECTING FUNCTION

Phrases can be used to connect words or groups of words together, signalling the relationship between the connected elements.

1 - The CORRELATIVE_CONJUNCTIONS[69] are sometimes referred to as phrases because they come in pairs.

2 - Phrases can also serve as SUBORDINATING CONJUNCTIONS[106]. These will be discussed in the lesson on clauses.

To return to the PHRASES MENU, press <Pg Up>.

LESSON: Screen 95

PHRASES: ADDING FILLER

Like words, phrases can be used to convey surprise or other emotions. These phrases are called INTERJECTIONS. If used alone, interjections are usually followed by an exclamation point. If joined with other words, they are usually set off by a comma.

EXAMPLES:

a. Good grief! Why did we do that?
   ----

b. Oh my, what a great movie that was!

(NOTE: Interjections are not normally used in expository writing.)

To return to the PHRASES MENU, press <Pg Up>.

LESSON: Screen 96

.[97] BUILDING ELEMENTS: CLAUSES (NEXT)[98]

The first two building elements of grammar are words[36] and phrases[77]. In the previous sections, we have reviewed these two elements and shown how they perform the five functions of grammar:

1 - Naming things
2 - Showing action or existence
3 - Modifying (Changing, qualifying, or renaming)
4 - Connecting words or groups of words
5 - Adding filler

The goal of communication is to convey complete thoughts or ideas. Words and phrases alone are unable to do this. To convey a complete thought or idea, we must have the final building element:

THE CLAUSE.

LESSON: Screen 97

.[98] BUILDING ELEMENTS: CLAUSES (cont.) (NEXT)[99]

A clause is a group of words that requires two things:

1 - A naming element, which is the thing about which we are trying to communicate something. In a clause, this is called the SUBJECT. Any building element that performs a naming function can serve as a subject..

2 - An element that gives action or existence to the subject. This is the clause's VERB. The verb may be either a word or a phrase.

All clauses will contain both a subject and a verb. They may also contain elements performing any of the other functions, including other naming elements, modifiers, connectors, and fillers, but to be a clause, they must contain, as a minimum, one subject and one verb.

LESSON: Screen 98

.[99] BUILDING ELEMENTS: CLAUSES (cont.) (NEXT)[100]

Review the following groups of words and see which are clauses and which are not:

a. pilots fly
   ------ ---
   subject + verb => CLAUSE

b. pilots fly airplanes
   ------ ---
   subject + verb => CLAUSE

c. pilots often fly airplanes
   ------ ---
   subject + verb => CLAUSE

LESSON: Screen 99

.[100] BUILDING ELEMENTS: CLAUSES (cont.) (NEXT)[101]

d. many pilots often fly airplanes every day
        ------ ---
        subject + verb => CLAUSE

e. although pilots fly airplanes
          ------ ---
          subject + verb => CLAUSE

f. pilots flying airplanes
   ------ ------
   subject + participle => NOT A CLAUSE (participle is not a verb)

g. athletes are healthy
   -------- ---
   subject + verb => CLAUSE

LESSON: Screen 100

.[101] BUILDING ELEMENTS: CLAUSES (cont.) (NEXT)[102]

h. many athletes are healthy
   -------- ---
   subject + verb => CLAUSE

i. even though many athletes are healthy
                 -------- ---
                 subject + verb => CLAUSE

j. athletes who are healthy
           --- ---
       subject + verb => CLAUSE

k. the healthy athletes on the skiing team

           no verb => NOT A CLAUSE

LESSON: Screen 101

.[102] BUILDING ELEMENTS: CLAUSES (cont.) (NEXT)[103]

l. the students finished their examination early
   -------- --------
   subject + verb => CLAUSE

m. because the students finished their examination early
           -------- --------
           subject + verb => CLAUSE

n. the students finishing their examination early
   -------- ---------
   subject + participle => NOT A CLAUSE (participle is not a verb)

LESSON: Screen 102

.[103] BUILDING ELEMENTS: CLAUSES (cont.) (NEXT)[104]

As you can see from the preceding examples, clauses come in many forms. Regardless of the form, every clause contains a subject and a verb. If we look at all clauses, we find that their subjects and verbs fall into one of three possible patterns:

1[108] - SUBJECT + VERB
("pilots fly")
2[109] - SUBJECT + VERB + OBJECT
("pilots fly airplanes")
3[112] - SUBJECT + LINKING VERB + COMPLEMENT
("pilots are courageous")
("pilots are the heroes")

You may review each of these patterns by selecting its number above.

Each pattern contains at least the elements listed. They may also contain modifiers and filler elements, but these are not part of the basic pattern. They are "free" and do not change the basic pattern.

LESSON: Screen 103

.[104] BUILDING ELEMENTS: CLAUSES (cont.) (NEXT)[105]

Thus far in this section, we have looked at quite a few clauses. We have learned that all clauses contain both a subject and a verb. But not all clauses convey complete thoughts. Consider the following two examples:

a. the committee held a meeting

b. after the committee held a meeting

Both of these are clauses; both contain the same subject (committee) and the same verb (held). But only the first one conveys a complete thought. Something about Example B keeps it from conveying a complete thought.

Press <Pg Up> several times and re-read the preceding examples and notice which clauses convey complete thoughts and which clauses do not.

LESSON: Screen 104

.[105] BUILDING ELEMENTS: CLAUSES (cont.) (NEXT)[106]

If a clause conveys a complete thought, it is called an INDEPENDENT CLAUSE. Each of the three subject/verb patterns can produce an independent clause:

1 - "pilots fly"
2 - "pilots fly airplanes"
3 - "pilots are courageous"

Each of these examples conveys a complete thought. But see what happens when certain other elements are added to them:

1 - "if pilots fly"
2 - "while pilots fly airplanes"
3 - "because pilots are courageous"

These examples are still clauses, but they no longer convey complete thoughts.

LESSON: Screen 105

.[106] BUILDING ELEMENTS: CLAUSES (cont.) (NEXT)[107]

1 - "if pilots fly"
2 - "while pilots fly airplanes"
3 - "because pilots are courageous"

Each of these now contains a connector that takes away the completeness of the thought. In other words, these connectors make the clause subordinate to some other clause in order to complete the thought. These connectors are called SUBORDINATING CONJUNCTIONS.

If a clause does not convey a complete thought by itself, it is called a SUBORDINATE CLAUSE (or a DEPENDENT CLAUSE, since it depends on some other clause to complete the thought). When the incomplete thoughts have been connected to independent clauses, the result is a new complete thought.

1 - "if pilots fly" . . . "they are happy"
2 - "while pilots fly airplanes" . . . "they are doing their jobs"
3 - "because pilots are courageous" . . . "they often become heroes"

LESSON: Screen 106

BUILDING ELEMENTS: CLAUSES (cont.)

A subordinate clause is a building element, just like a word or a phrase. It can perform one of the following functions of grammar:

1[114] - Naming function: A Noun Clause
2[115] - Modifying function: An Adjective Clause
3[116] - Modifying function: An Adverb Clause

For a description of each of these functions of a subordinate clause, select its number from the list above.

This concludes the discussion of the final building element:

THE CLAUSE

Press <Home> to return to the MAIN MENU.

LESSON: Screen 107

SUBJECT + VERB

This pattern is: "Something did something." More formally, the pattern is:

SUBJECT + VERB (plus any modifiers and/or fillers)

EXAMPLES: pilots fly
------ ---

although pilots often fly
------ ---

while pilots are flying
------ ----------

Note that even though modifiers and fillers are sprinkled into the clause, the pattern remains the same.

Press <Pg Up>.

LESSON: Screen 108

.[109] SUBJECT + VERB + OBJECT (NEXT)[110]

This pattern is: "Something did something to something else." (The "something else" is called the object.) This pattern is:

SUBJECT + VERB + OBJECT (plus any modifiers and/or fillers)

EXAMPLES: pilots fly airplanes
---------- --- ---------

although pilots often fly their airplanes
---------- --- ---------

the students studied their notes carefully
-------- ------- -----

since the students had studied their notes carefully
-------- ---------- -----

LESSON: Screen 109

.[110] SUBJECT + VERB + OBJECT (cont.) (NEXT)[111]

Note that even though modifiers and fillers are sprinkled into the clauses, the pattern remains the same.

Consider this clause: the professor handed the book to me
--------- ------ ----
subject verb object

This type of clause will often be restructured as follows:

the professor handed me the book
--------- ------ ----
subject verb object

The clause still follows the same pattern, but the prepositional phrase "to me" has been repositioned and shortened to "me" by itself. Note that the action is still received by "the book," which thus remains the object. Since "me" is no longer part of a prepositional phrase, it takes a new name, the INDIRECT OBJECT.

LESSON: Screen 110

SUBJECT + VERB + OBJECT (cont.)

To test for an indirect object, see if it can be moved after the verb and follow either "to" or "for." Review the following examples:

a. she wrote her senator a letter
   ---------- (indirect object)
   she wrote a letter to her senator
                ---------- (prepositional phrase)

b. he tossed the tailback the ball
   ---------- (indirect object)
   he tossed the ball to the tailback
                ---------- (prepositional phrase)

c. they bought the speaker a gift
   ---------- (indirect object)
   they bought a gift for the speaker
                ---------- (prep phrase)

Press <Pg Up> three times.

LESSON: Screen 111

.[112] SUBJECT + LINKING VERB + COMPLEMENT (NEXT)[113]

This pattern is: "Something is something."

The pattern consists of a subject and a condition/state verb, followed either a naming element or a modifier. If a naming element follows, it is called a NOUN COMPLEMENT. If a modifier follows, it is called an ADJECTIVE COMPLEMENT. The pattern is:

SUBJECT + LINKING VERB + COMPLEMENT (plus any modifiers or fillers)

EXAMPLES:

a. the car is blue
   --- -- ----
   S + LV + ADJECTIVE COMPLEMENT

b. the car is a Buick
   --- -- -----
   S + LV + ADJECTIVE COMPLEMENT

LESSON: Screen 112

SUBJECT + LINKING VERB + COMPLEMENT

c. although the car remained a lemon
--- -------- -----
S + LV + NOUN COMPLEMENT

d. his car, a Buick, was expensive
--- --- ---------
S + LV + ADJECTIVE COMPLEMENT

e. if the car had been a Buick
--- -------- -------
S + LV + NOUN ADJECTIVE COMPLEMENT

Note that even though modifiers and fillers are sprinkled into the clauses, the pattern remains the same.

Press <Pg Up> twice.

LESSON: Screen 113

.[114] SUBORDINATE CLAUSE: As a Naming Element (NEXT)[115]
THE NOUN CLAUSE

Subordinate clauses function as building elements, just like words and phrases do. One function that subordinate clauses can perform is the naming function. Subordinate clauses that perform this function are called NOUN CLAUSES.

Noun clauses, like all subordinate clauses, usually begin with words that make them dependent by taking away the completeness of the thought. Common subordinating conjunctions are:

what who that when why which

Noun clauses are used just like any other naming element within another clause. They can serve as subjects of other clauses, objects, and complements.

LESSON: Screen 114

.[115] SUBORDINATE CLAUSE: As a Naming Element (NEXT)[116]
THE NOUN CLAUSE (cont.)

In each of the following examples, note that each underlined element is a subordinate clause and that it serves as a naming element.

EXAMPLES:

a. what we say in class is protected by academic freedom

(subject of another, independent clause)

b. the coach knows who will be the starting quarterback

(object of the verb "knows")

c. our hope is that everyone will pass this examination

(noun complement)

LESSON: Screen 115

SUBORDINATE CLAUSE: As a Naming Element
THE NOUN CLAUSE (cont.)

d. the professor assured me that my answer was correct

(object of the verb "assured")

e. when the class is rescheduled will depend on the teacher

(subject of a clause)

f. the students chose which course was their favorite

(object of the verb"chose")

SUMMARY: A subordinate clause can serve as a naming element within another clause. This type of clause is called a NOUN CLAUSE.

Press <Pg Up> three times.

LESSON: Screen 116

.[117] SUBORDINATE CLAUSE: As a Modifier of Naming Elements (NEXT)
THE ADJECTIVE CLAUSE [118]

Subordinate clauses function as building elements, just like words and phrases do. One function that subordinate clauses can perform is to modify naming elements. Subordinate clauses that perform this function are called ADJECTIVE CLAUSES. An adjective clause performs as a single element just like an adjective[60].

An adjective clause, like all subordinate clauses, usually begins with a word that make the clause dependent by taking away the completeness of the thought. Common subordinating conjunctions are:

who	that	which	whose
what	when	where	whom

LESSON: Screen 117

.[118] SUBORDINATE CLAUSE: As a Modifier of Naming Elements (NEXT)
THE ADJECTIVE CLAUSE (cont.) [119]

In each of the following examples, note that each underlined element is a subordinate clause and that it performs like an adjective.

EXAMPLES:

a. the course that Dr. Jones teaches is interesting

(modifies "course")

b. my advisor is the professor whose office was just remodeled
---
(modifies "professor")

c. the speaker whom we just heard is from the mayor's office

(modifies "speaker")

LESSON: Screen 118

SUBORDINATE CLAUSE: As a Modifier of Naming Elements
THE ADJECTIVE CLAUSE (cont.)

d. The books which I just bought were very expensive.

(modifies "books")

e. The reason why the payment was late is given in my letter.

(modifies "reason")

SUMMARY: A subordinate clause can serve as a modifier of a naming element. This type of subordinate clause acts like an adjective and is called an ADJECTIVE CLAUSE.

Press <Pg Up> three times.

LESSON: Screen 119

.[120] SUBORDINATE CLAUSE: As a Modifier of Verbs and Adverbs (NEXT)
THE ADVERB CLAUSE [121]

Subordinate clauses function as building elements, just like words and phrases do. One function that subordinate clauses can perform is to modify elements other than naming elements. Subordinate clauses that perform this function are called ADVERB CLAUSES. An adverb clause performs as a single element just like an adverb[61].

Adverb clauses usually answer one of these questions:

How? When? Why? Under what conditions?

In each of the following examples, note that each underlined element is a subordinate clause and that it performs like an adverb.

LESSON: Screen 120

.[121] SUBORDINATE CLAUSE: As a Modifier of Verbs and Adverbs (NEXT)
THE ADVERB CLAUSE (cont.) [122]

EXAMPLES:

a. before the class ended, we were able to ask questions

(When?)

b. they won the game because they were the most prepared

(Why?)

c. if the water continues to rise, we must evacuate

(Under what conditions?)

d. since we were late for the movie, we decided to go home

(Why?)

LESSON: Screen 121

SUBORDINATE CLAUSE: As a Modifier of Verbs and Adverbs
THE ADVERB CLAUSE (cont.)

e. he shortened his speech so that we could ask questions.

(Why?)

f. they lost the game even though they were the better team

(Under what condition?)

g. he is so tall that his head hits the ceiling

(How tall?)

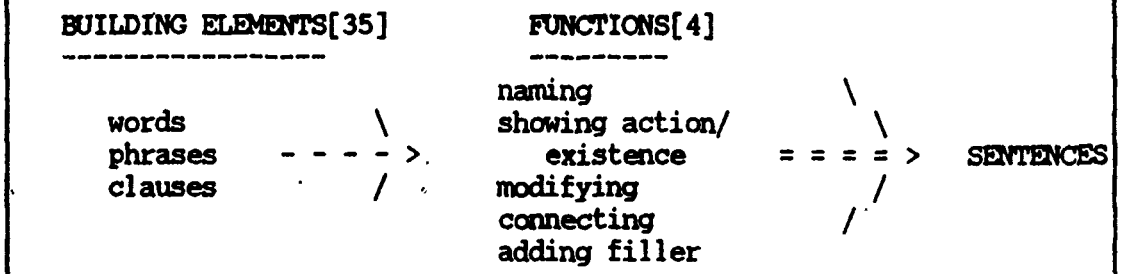
SUMMARY: Some subordinate clauses modify elements other than naming elements. This type of subordinate clause is called an ADVERB CLAUSE.

Press <Pg Up> three times.

LESSON: Screen 122

.[123] SENTENCES (NEXT)[124]

In the first two sections of this lesson, we looked at the five basic functions performed by grammatical elements and the three building elements that can perform those functions. The next step is to join these concepts to form sentences:
(NOTE: To review either of these areas, select it below.)



A SENTENCE is a grammatical unit that communicates one complete thought. The minimum requirement for a sentence is one independent_clause[105].

LESSON: Screen 123

.[124] SENTENCES (cont.) (NEXT)[125]

An independent clause is a building element that conveys a complete thought. To convert it to a sentence, capitalize the first letter of the clause and put a period at the end of it. See how each of the four clause patterns can become a sentence:

INDEPENDENT CLAUSE: pilots fly

SENTENCE: Pilots fly.

INDEPENDENT CLAUSE: pilots fly airplanes

SENTENCE: Pilots fly airplanes.

INDEPENDENT CLAUSE: pilots are courageous

SENTENCE: Pilots are courageous.

LESSON: Screen 124

.[125] SENTENCES (cont.) (NEXT)[126]

Sentences of this type, consisting of only one independent clause, are called SIMPLE SENTENCES. Other types of sentences may contain more than one clause. There are four sentence types that can be formed from clauses:

SENTENCE TYPE	NUMBER OF INDEPENDENT CLAUSES	NUMBER OF SUBORDINATE CLAUSES
Simple[129]	one	none
Compound[131]	two or more	none
Complex[133]	one	one or more
Compound-Complex[137]	two or more	one or more

For discussion about each type of sentence, select it from the list above.

LESSON: Screen 125

.[126] SENTENCES (cont.) (NEXT)[128]

Two connectors that join clauses are the coordinating_conjunction[68] and the subordinating_conjunction[106]. Another connector is the CONJUNCTIVE ADVERB. When conjunctive adverbs are used, the clauses must either be combined as a compound sentence (or compound-complex), or be written as two separate sentences. Conjunctive adverbs include:

accordingly	also	anyway	as a result
besides	consequently	finally	furthermore
hence	however	incidently	indeed
instead	likewise	meanwhile	moreover
nevertheless	nonetheless	now	otherwise
still	suddenly	then	therefore
thus			

The conjunctive adverbs should not be confused with subordinating conjunctions which cause a clause to become a subordinate clause. A clause that contains a conjunctive adverb remains an independent clause. Select here for examples of proper usage -> EXAMPLES[127].

LESSON: Screen 126

EXAMPLES OF PROPER USAGE OF CONJUNCTIVE ADVERBS

a. Thus, the results of the experiment appeared valid.

b. The lecture seemed to last forever; finally, it ended.

c. The students were late for class. Consequently, they missed some important information.

d. The student decided, however, to spend the weekend studying.

e. Suddenly, flames leaped from the scientist's beaker.

Press <Pg Up>.

LESSON: Screen 127

SENTENCES (cont.)

This concludes the section on sentences.

To return to the MAIN MENU, press <Home>.

LESSON: Screen 128

.[129] SENTENCE TYPES: The Simple Sentence (NEXT)[130]

A SIMPLE SENTENCE contains only one independent clause and no subordinate clauses. The independent clause may include modifiers and fillers. Do not be confused by words and phrases that perform these functions.

EXAMPLES:

a. The professor spoke very clearly.
   --------- -----
   subject verb

b. The students studied their notes in preparation for the test.
   -------- ------- ----- ---
   subject verb object

LESSON: Screen 129

SENTENCE TYPES: The Simple Sentence (cont.)

c. This classroom is always the coldest room in the building.
   --------- -- ---- ----
   subject linking noun
   verb complement

d. Everyone cast his or her vote in the election.
   -------- ---- ----
   subject verb object

e. Both students and teachers were pleased with the results.
   ---------- ---- -------
   subject linking adjective
   verb complement

Press <Pg Up> twice.

LESSON: Screen 130

.[131] SENTENCE TYPES: The Compound Sentences (NEXT)[132]

A COMPOUND SENTENCE contains two or more independent clauses, but no subordinate clauses. The independent clauses are usually joined by a comma and coordinating_conjunction[68]. Remember, in English there are seven coordinating conjunctions:

and but for nor or so yet

In special cases the independent clauses can be joined by a semicolon, a dash, or a colon.

EXAMPLES:

a. The long winter finally ended, and the flowers began to bloom.
---------- ----------

LESSON: Screen 131

SENTENCE TYPES: The Compound Sentence (cont.)

b. During halftime, the fans visited the concession stand, the teams
---------- ---------
rested, and the coaches developed a plan for the second half.
------ ----------

c. The quarter has ended; however, we still must take the final exam.
---------- ----------

d. The audience gasped--the trapeze artist had nearly fallen!
---------- ----------

e. We wanted to visit the museum, but we did not have enough time.
---------- ----------

Press <Pg Up> twice.

LESSON: Screen 132

.[133] SENTENCE TYPES: The Complex Sentence (NEXT)[134]

A COMPLEX SENTENCE consists of one independent clause and one or more subordinate clause.

Like a simple sentence, a complex sentence consists of one independent clause, following one of the three patterns[103] of clauses that we have discussed:

SUBJECT + VERB (+ ANY MODIFIERS)

SUBJECT + VERB + OBJECT (+ ANY MODIFIERS)

SUBJECT + LINKING VERB + COMPLEMENT (+ ANY MODIFIERS)

Unlike a simple sentence, though, in a complex sentence, at least one of the building elements in the sentence is a subordinate clause. A noun clause can serve as subject or object. An adjective clause or an adverb clause can serve as a modifier.

LESSON: Screen 133

.[134] SENTENCE TYPES: The Complex Sentence (cont.) (NEXT)[135]

EXAMPLES:

(NOTE: When a subordinate clause precedes an independent clause, a comma normally separates them.)

a. They asked questions when the professor had finished the lecture.
---- ----- --------- --
S + V + OBJECT + MODIFIER (adverb clause)

b. After the performance has started, patrons may not enter
---------------------------------- ------- -------------
MODIFIER (adverb clause) + SUBJECT + VERB

the auditorium, even if they have reserved seats.
-------------- ------------------------------
+ OBJECT + MODIFIER (adverb clause)

LESSON: Screen 134

.[135] SENTENCE TYPES: The Complex Sentence (cont.) (NEXT)[136]

c. We knew that the examination would not be very difficult.
-- ---- --
S + V + OBJECT (noun clause)

d. The car that you purchased arrived yesterday.
------- ----------------- ------- ---------
SUBJECT + MODIFIER + VERB + MODIFIER.
(adjective clause)

e. Even though we had studied, we were concerned about the test.
------------------------- -- ---- --------- -------------
MODIFIER + S + V + ADJECTIVE + MODIFIER
(adverb clause) COMPLEMENT

LESSON: Screen 135

.[136] SENTENCE TYPES: Complex Sentences (cont.)

f. Our truck, which was just painted, has a new scratch on the door.
--------- -------------------- --- ------------ -----------
SUBJECT + MODIFIER + V + OBJECT + MODIFIER
(adjective clause)

g. They are the students who were elected to the student council.
---- --- ----------- ---------------------------------------
S + LV + NOUN + MODIFIER
COMPLEMENT (adjective clause)

h. We told the professor that our project would be completed today.
-- ---- ------------ --
S + V + INDIRECT + OBJECT
OBJECT[110] (noun clause)

Press <Pg Up> four times.

LESSON: Screen 136

.[137] SENTENCE TYPES: The Compound-Complex Sentence (NEXT)[138]

The compound-complex sentence is potentially the most complicated of all sentence structures:

1 - It is similar to a compound_sentence[131] because it must have at least two independent clauses.

2 - It is similar to a complex_sentence[133]] because somewhere in the structure there will be at least one subordinate clause.

EXAMPLES:

a. Although the students were prepared for the test, they found it very difficult, and they asked the professor for more time to complete it.

(SUBORDINATE CLAUSE + INDEPENDENT CLAUSE + INDEPENDENT CLAUSE)

LESSON: Screen 137

SENTENCE TYPES: The Compound-Complex Sentence (cont.)

b. The runners exchanged words of encouragement before the race began; after the race ended, they were too tired to speak.

(INDEPENDENT CLAUSE + SUBORDINATE CLAUSE + INDEPENDENT CLAUSE)

c. Since there were no more questions, the professor dismissed us, and I went to the library.

(SUBORDINATE CLAUSE + INDEPENDENT CLAUSE + INDEPENDENT CLAUSE)

d. I disagree with some of his views; however, I will vote for him if he decides to run for office next year.

(INDEPENDENT CLAUSE + INDEPENDENT CLAUSE + SUBORDINATE CLAUSE)

Press <Pg Up> twice.

LESSON: Screen 138

.[139] QUICK INDEX OF KEY CONCEPTS

LESSON: Screen 139

Appendix D: The Grammar Lesson Program Code in Turbo Pascal

1. The following code is the main program for the Grammar Lesson:

```
program GRAMMAR;  {This is the main program for the Grammar Lesson}

uses crt, fastttt5, menuttt5,winttt5, mod_ht;
{
This program uses units from Technojock's Turbo Toolkit (tm), which is a
collection of Turbo Pascal procedures and functions from Technojock
Software, Inc.  It will be referred herein as TTT.

MOD_HT is a modification of the hypertext program by Richard Gessner in
"Building a Hypertext System," Dr. Dobb's Journal, 15(6):22-33 (Sep 90).
                                                                    }
var main_menu : menu_record;  { menu_record is the structure for menus}
    tut_menu : menu_record;  { from TTT}
  choice : integer;
  ch : integer;
  error : integer;

procedure intro;  {This is the introduction screen}
  begin
    offcursor;
    fbox(1,1,80,25,yellow,blue,2);
    writecenter(8,yellow,blue,'THE GRAMMAR LESSON');
    writecenter(9,yellow,blue,'Version 1.2');
    box(29,7,50,10,yellow,blue,1);
    writecenter(13,yellow,blue,'Written By:');
    writecenter(15,yellow,blue,'FRANK E. JONES, Capt, USAF');
    writecenter(16,yellow,blue,'AFIT/GIR/91-D');
    writecenter(17,yellow,blue,'July 1991');
    writecenter(20,lightred,blue,' Press <Enter> to continue. ');
    readln;
    fbox(1,1,80,25,yellow,blue,0);
  end;

begin {GRAMMAR}

  intro;

  menu_set(main_menu);                    { This sets up the main menu.}
  with main_menu do
    begin
      heading1 := 'PLEASE MAKE A SELECTION.';
      colors[1] := yellow;
      colors[2] := red;
      colors[3] := yellow;
      colors[4] := blue;
      colors[5] := lightblue;
      boxtype := 2;
```

```
    margins := 15;
    allowesc := false;
    addprefix := 0;
    topic[1] := 'Instructions';
    topic[2] := 'The Grammar Lesson';
    topic[3] := ' ';
    topic[4] := 'Quit';

    totalpicks := 4;
  end;
repeat
  choice := 1;
  fbox(1,1,80,25,yellow,blue,0);
  fbox(5,21,75,24,lightblue,blue,2);
  Writecenter(21,yellow,blue,' HELP ');
  Writeat(25,22,yellow,blue,'   '+chr(24)+' '+chr(25)+'   to move
      highlight');
  Writeat(25,23,yellow,blue,' <Enter> to select highlight ');

  displaymenu(MAIN_MENU,FALSE,CHOICE,ERROR); { This displays the menu}
  case choice of                             { and carries out the
     1 : begin                               { choice.              }
           fbox(1,1,80,25,yellow,blue,0);
           do_help('INSTRUCT.HYP',1,1);
         end;
     2 : begin
           fbox(1,1,80,25,yellow,blue,0);
           do_help('LESSON.HYP',1,1);
         clrscr;
         end;
     4 : begin
           fbox(1,1,80,25,yellow,blue,2);
           writecenter(13,yellow,blue,'Thank you for using the Grammar
                Lesson!');
           writecenter(20,lightred,blue,'Press <Enter>.');
           readln;
           clrscr;
           oncursor;
           exit;
         end;
  end;
 until true = false;
end.
```

2. The following code is the MOD_HT program which contains the hypertext editing and displaying units:

```
{$V-} {$F+} {$O+}   { Written By:  Rick Gessner, 1989. }
                    { Modified By:  Frank Jones, 1991. }
Unit MOD_HT;

Interface

PROCEDURE Help_Editor(FileName: String);
PROCEDURE Do_help(FileName: String; GoPage,HomePage: Word);

Implementation

Uses Crt,fastttt5,winttt5;
CONST HelpColor        : Array[False..true] of Byte =
         ( blue*16+White,                          { for normal text}
           blue*16+Yellow);                      { for hot-link text.}
           NormalColor      : Byte = blue*16+White; {draw screen info.}
           BoldColor        : Byte = red*16+yellow;  { for select bar.}
           Header           : String[50] = 'Grammar Tutor';
           MaxLinesPerPage  = 20;
           MaxLineWidth     = 70;
           PGUP    = 'I'; PGDN    = 'Q'; UpArrow = 'H'; {Edit keys}
           DnArrow = 'P'; LArrow  = 'K'; RArrow  = 'M';
           ESC     = #27; HomeKey = 'G'; EndKey  = 'O';
           RETURN  = ^M;  BkSpc   = #8;  NULL    = #0;
           Tab     = #9;  F2      = '<'; DelKey  = 'S';
Type  HelpRecord = Record  {The main structure for our hypertext files}
      HelpLines : Array[1..MaxLinesPerPage] of String[100];
      end; {String length MUST be > than MaxLineWidth to store
            hot-links!}
Var            HelpRec    : HelpRecord;
               HelpFile   : File of HelpRecord;
     Alt,Ctrl,CommandKey : Boolean;

FUNCTION Make_String(Ch   : Char; Size : Integer) : String;

Var S: string;
Begin
  S[0] := Chr(Size);            { Set length byte = SIZE.          }
  FillChar(S[1],Size,Ch);       { Fill the string with chr(CH).    }
  Make_String:= S;              { and return the string as function}
end; {Make String }             { value.                           }

PROCEDURE Draw_Box(topx,topy,botx,boty: Byte; Color,Width: byte);

Type BoxPos = (TopL,TopR,BotL,BotR,Top,Bot,LSide,RSide);
Var Y       : Integer;
Const  Boxchar : Array[1..2,TopL..RSide] of char =
    (( 'Z','?','@','Y','D','D','3','3'), {ASCII chars, single line box }
    ( 'I',';','H','<','M','M',':',':')); {ASCII chars, double line box }
```

```
Begin
  TextAttr:=Color;
  If Not (Width in [1,2]) then Width:=1; { Sure width value is OK? }
     Gotoxy(TopX,TopY);    { First, draw the top line of the box...}
     Write(BoxChar[Width,TopL]+Make_String(BoxChar[width,top],
           BotX-TopX-1)+BoxChar[Width,TopR]);
     For Y:=TopY+1 to BotY-1 do
         Begin        { Second, draw the middle lines of the box...}
           Gotoxy(TopX,Y);
           Write(BoxChar[Width,LSide],BoxChar[Width,RSide]:
                 BotX-TopX);
         end;
     GotoXY(TopX,BotY);  { Third, draw the bottom line of the box.}
     Write(BoxChar[Width,BotL]+Make_String(BoxChar[width,top],
           BotX-TopX-1)+BoxChar[Width,BotR])
end; {Draw Box}

FUNCTION Read_KeyBoard: Char;   {Routine to get keystrokes from user}

Const  CtrlMask  = $04;
       AltMask   = $08;
Var    KBDFlag   : Byte Absolute $0040:$0017;

Begin
  Read_KeyBoard:=ReadKey;
  CommandKey :=((KBDFlag AND AltMask)<>0) or ((KBDFlag AND
  CtrlMask)<>0);
  ALT  :=(KBDFlag AND AltMask)<>0; CTRL := (KBDFlag AND CtrlMask)<>0;
  If KeyPressed Then
     Begin
       Read_Keyboard := ReadKey; {In case user pressed modified key}
       CommandKey := True;
     end;
end; {Read_Keyboard}

PROCEDURE Show_HelpLine(X,Y,StartBold,EndBold: Integer; Var Line:
               String);

Var I,J: Integer;

  PROCEDURE Write_Char(Ch: Char);

  Begin
    If Ord(Ch)>127 then Ch:=Chr(Ord(Ch)-128); {Clear high bit}
       If Ord(Ch)>27 then Write(Ch) else Inc(i);
    end;
Begin
    TextAttr:=HelpColor[False];
    Window(X,Y,59,Y); ClrEOL; Window(1,1,80,25); {Prepare for output}
    Gotoxy(X,Y); I:=1;
    While I<=Length(Line) do                    {Do each char in line}
      Begin
        TextAttr:=HelpColor[Ord(Line[i])>128];    {Set proper color}
```

```
        If I in [StartBold..EndBold] then TextAttr:=BoldColor;
           Write_Char(Line[i]); Inc(i);
      end;
end; {Show helpline}

PROCEDURE Show_Help_Page(X,Y: Integer; Var HelpRec: HelpRecord);

Var I: Integer;

Begin
  Window(X+1,Y+1,X+56,Y+MaxLinesPerPage+1); ClrScr; Window(1,1,80,25);
  For I:=1 to MaxLinesPerPage do
      Show_HelpLine(X,Y+I,0,0,HelpRec.HelpLines[I]);
end; {Show help page}


FUNCTION Determine_Actual_Line_Pos(Var Line: String; LinePos : Integer):
             Integer;
Var I,J: Integer;  {Convert visual edit column to char. position,}
Begin              {by skipping over embedded hot links.        }
  I:=0; J:=1;
  While (J<=Length(Line)) and (I<>LinePos) do
    Begin
      Inc(I); {else Inc(J,2);
      Inc(J);
    end;
  Determine_Actual_Line_Pos:=J;
end; {Determine actual line pos}

FUNCTION Link_Count(Var Line: String): Integer;

Var I,Count: Integer;  { Returns 2*#nulls in line, used to convert }
Begin                  { from actual byte pos. to visual byte pos., }
  Count:=0;            { during data input. }
  For I:=1 to Length(Line) do
    If Line[i]=Null then Inc(Count,2);
       Link_Count:=Count;
end; {Link count}

FUNCTION Input_HelpPage(X,Y: Byte; Var AHelpRec: HelpRecord): Char;

Var   Ch          : Char;   { Main editing routine in this system.   }
      PageNum     : Byte;   { It is really just a page-oriented line }
      I,J,                  { editor that knows how to jump over     }
      LinePos,              { two-byte hot-links.  NOTE:  If you     }
      RealLinePos,          { add editing options, take the embedded }
      LineNum     : Integer;{ hot-links into account!              }

PROCEDURE Delete_Linked_Char(Var Line: String; LinePos: Integer);
           Var I,J: Integer;
Begin
  LinePos:=Pred(Determine_Actual_Line_Pos(Line,LinePos));
  If Ord(Line[LinePos])>127 then  {Were on a linked item}
```

```
      begin
        I:=LinePos;
        While ((Ord(Line[I-1])>127) and (I>1)) do Dec(i);
          J:=LinePos;                        {Next find end of link}
          While ((Ord(Line[J+1])>127) and (I<Length(Line))) do Inc(J);
          Delete(Line,LinePos, {Delete all of item + link if necc.}
            1+(2*Ord(J=I)));
       end;
end; {Delete linked char}

Begin
  Show_Help_Page(X,Y,AHelpRec);  {Display this page }
  LinePos:=1; RealLinePos:=1;    {Now do a little init stuff.}
  LineNum:=1;
  With AHelpRec do               {Now enter main edit loop...}
    Repeat
      Show_HelpLine(X,Y+LineNum,0,0,HelpLines[LineNum]);
      Gotoxy(X+LinePos-1,Y+LineNum);
      Repeat Ch:=Read_KeyBoard Until Ch <> Null;
      If CommandKey then
         Case Ch of

^Y     : If RealLinePos<=Length(HelpLines[LineNum]) then
         Begin    { ^Y = Delete to end of line. }
           If (RealLinePos=1) then HelpLines[LineNum]:=''
           else
             Begin
               While HelpLines[LineNum,RealLinePos]<>Null do
                 Delete(HelpLines[LineNum],RealLinePos,1);
                 If HelpLines[LineNum,RealLinePos]=Null then
                    Delete(HelpLines[LineNum],RealLinePos+2,255)
             end;
        end;
F2 : Begin { F2 = Add/Remove hot-link.}
       J:=RealLinePos;
       While (j>0) and (HelpLines[LineNum,j]<>' ') do
         Dec(j);
         Inc(j);
         If Ord(HelpLines[Linenum,j]) in [28..127] then
            Repeat {Now get a valid page # to jump to...}
            Gotoxy(3,24); Write('Link Page: ');
            Readln(PageNum);
            Gotoxy(3,24); ClrEOL;
            Until (PageNum>0) and (PageNum<256);
            While (HelpLines[LineNum,j]<>' ') and
            (j<=Length(HelpLines[LineNum])) and
            (HelpLines[LineNum,j]<>Null) do
              Begin
               HelpLines[LineNum,j]:=Chr(Ord(HelpLines[LineNum,j])+128);
               Inc(j);
              end;
            If Ord(HelpLines[LineNum,J-1]) in [28..127] then
               Delete(HelpLines[LineNum],J,2) else
```

```
              Insert(Null+Chr(PageNum),HelpLines[LineNum],j);
            end;
LArrow : If RealLinePos>1 then {Move left one line}
            Begin
              Dec(linePos);
              RealLinePos:=
            Pred(Determine_Actual_Line_Pos(HelpLines[LineNum],LinePos));
            end;
RArrow : If RealLinePos<=Length(HelpLines[LineNum]) then
                     Begin          {Move right one line}
                       Inc(LinePos);
                       If RealLinePos<Length(HelpLines[LineNum]) then
                          Inc(RealLinePos,1+Ord(HelpLines[LineNum,
                             RealLinePos+1] = Null)*2)
                       else Inc(realLinePos)
                       end;
DnArrow : If LineNum<MaxLinesPerPage then
            Begin             {Move down one line.}
            Inc(LineNum);
            If LinePos<=Length(HelpLines[LineNum]) then
RealLinePos:=Pred(Determine_Actual_Line_Pos(HelpLines[LineNum],LinePos))
            else
              Begin
                RealLinePos:=Succ(Length(HelpLines[LineNum]));
                LinePos:=RealLinePos-Link_Count(HelpLines[LineNum]);
              end;
            end;
UpArrow : If LineNum>1 then      {Move up 1 line.}
            Begin
              Dec(LineNum);
              If LinePos<=Length(HelpLines[LineNum]) then
RealLinePos:=Pred(Determine_Actual_Line_Pos(HelpLines[LineNum],LinePos))
              else
                Begin
                  RealLinePos:=Succ(Length(HelpLines[LineNum]));
                  LinePos:=RealLinePos-Link_Count(HelpLines[LineNum]);
                end;
              end;
HomeKey : Begin              {Move to 1 char. in line.}
            LinePos:=1;
            RealLinePos:=LinePos;
          end;
EndKey : Begin               {Move to end of line.}
           RealLinePos:=Succ(Length(HelpLines[LineNum]));
           LinePos:=RealLinePos-Link_Count(HelpLines[LineNum]);
         end;
DelKey : If (RealLinePos<=Length(HelpLines[LineNum])) then
           Begin                 {Delete a character.}
             If (HelpLines[LineNum,RealLinePos]) in [' '..'}']
                then Delete(HelpLines[LineNum],RealLinePos,1) else
                Delete_Linked_Char(HelpLines[LineNum],LinePos);
RealLinePos:=Pred(Determine_Actual_Line_Pos(HelpLines[LineNum],LinePos))
           end;
```

```
            end else
Case Ch of
Return: If LineNum<MaxLinesPerPage then  {Move down 1 line}
            Begin
             Inc(LineNum); LinePos:=1; RealLinePos:=1;
            end;
Tab : Begin    {Tab right 10 chars.}
       If RealLinePos+10<=Length(HelpLines[LineNum])+1 then
          Inc(RealLinePos,10) else
          RealLinePos:=Length(HelpLines[LineNum])+1;
          LinePos:=RealLinePos-Link_Count(HelpLines[LineNum]);
        end;
BkSpc  : If RealLinePos>1 then  {Backspace }
           Begin
             If HelpLines[LineNum,RealLinePos-1] in [' '..'}'] then
                Begin
                  Delete(HelpLines[LineNum],RealLinePos-1,1);
                  Dec(RealLinePos);
                  Dec(LinePos)
                end else
                  Begin
                    Delete_Linked_Char(HelpLines[LineNum],LinePos-1);
                    Dec(LinePos);
RealLinePos:=Pred(Determine_Actual_Line_Pos(HelpLines[LineNum],LinePos))
;
                  end;
                end;
' '..'}' : If Length(HelpLines[LineNum])<MaxLineWidth then
              Begin     {Insert a valid Ascii char.}
                If (Ord(HelpLines[LIneNum,RealLinePos])>127) and
                   (RealLinePos<=Length(HelpLines[Linenum])) then
                     Ch:=Chr(Ord(Ch)+128);
                     Insert(Ch,HelpLines[LineNum],RealLinePos);
                     Inc(RealLinePos);
                     Inc(LinePos); Ch:=#255;
                end;
           end;
  Until CH in [ESC,PGUp,PgDn]; {ESC=Quit;PGUp=Prev page;PgDn=Next Page.}
  Input_HelpPage:=Ch;
end; {Input helppage}
    FUNCTION Read_Helprec(Var AHelpRec: HelpRecord;
RecNum:Integer):Integer;

Var I : Integer;
Begin
  FillChar(AHelprec,SizeOf(AHelprec),0); {$I-} { Hyperdata file read   }
  If FileSize(HelpFile)<RecNum then exit;      { routine.  Just enough }
     Seek(helpfile,RecNum-1);                  { error checking to be  }
     Read(helpfile,AHelpRec);                  { considered safe.      }
     Read_HelpRec:=IOResult; {$I+}
   end; {Read helprec}

FUNCTION Write_HelpRec(Var AHelpRec: HelpRecord;
```

```
RecNum:Integer):Integer;
Begin {$I-}
 Seek(helpfile,RecNum-1);          {Hyperdata file write routine.}
 Write(helpfile,AHelpRec); {$I+}   {This routine contains just  }
 Write_HelpRec:=IOresult;          {enough error checking to be}
end; {Write helprec}                   {considered safe.}

FUNCTION Open_HelpFile(FileName: String): Integer;
Var result: Integer;
Begin
  Assign(HelpFile,FileName); {$I-}  {Opens hyperdata file specified}
  Reset(HelpFile);                  {as "FileName".  If the file   }
  result:=IOResult;                 {doesnt exist, then it will be }
  If Result=2 then                  {created.                      }
     Begin                          {Error checking is limited, but}
       ReWrite(HelpFile);           {enough to be safe.            }
       Result:=IOResult;
     end;
  Open_HelpFile:=Result;
end; {open helpfile}

PROCEDURE Help_Editor(FileName: String);
Var I,HelpRecNum: Integer;
AHelpRec     : HelpRecord;
Ch           : Char;
Result       : Integer;
Begin
  Result:=Open_HelpFile(FileName);     {Open the specified file.}
  If Result=0 then                     {Continue only if no error.}
     Begin
       TextAttr := NormalColor;
       HelpRecNum:=1;
       Gotoxy(40-(Length(Header) div 2),3); Writeln(Header);
       Gotoxy(4,2); Writeln('File: ',FileName);
       Repeat
         fbox(1,4,73,25,lightblue,blue,2);
         Gotoxy(4,3); Writeln(' Reading  ');
         Result:=Read_HelpRec(AHelpRec,HelpRecNum);
         Gotoxy(4,3); Writeln('Page: ',HelpRecNum:3);
         Ch:=Input_HelpPage(3,4,AHelpRec);
         Result:=Write_HelpRec(AHelpRec,HelpRecNum);
         Gotoxy(4,3); Writeln(' Writing  ');
         Case Ch of
            PgUp :  begin
                      fbox(1,4,73,25,lightblue,blue,2);
                      If helpRecNum>1 then Dec(HelpRecNum);
                    end;
            PgDn : begin
                     fbox(1,4,73,25,lightblue,blue,2);
                     If HelpRecNum < 255 then Inc(HelpRecNum);
                   end;
         end;
       Until Ch=ESC;
```

```
    end else  {Report the opening error...}
   Writeln('ERROR: ',Result,' opening ',FileName,'. Unable to
           continue.');
 {$I-} Close(HelpFile); Result:=IOresult; {$I+}
end; {Help editor}

FUNCTION Find_Next_Link( Var X,Y: Integer; EndX,EndY: Integer;
Var AHelpRec: HelpRecord): Boolean;
Var OrigX,OrigY,Col,                {Recursive routine used to find a }
    Row,StartCol,StopCol: Integer;  {hot-link on the page after the   }
Begin                               {current page position (X,Y).     }
  Find_Next_Link:=False; {First, look from current pos to end of page.}
  For Row:=Y to EndY do
      Begin
        If Row = Y then if Col = X then exit;
        If Row<>Y then StartCol:=1 else StartCol:=X;
           STOPCOL:= LENGTH(AHELPREC.HELPLINES[ROW]);
           If AhelpRec.HelpLines[Row]<>'' then
              For Col:=StartCol to StopCol do
           If (AHelpRec.HelpLines[Row,Col]=Null) then
              Begin
                Find_Next_Link:=True;
                X:=Col; Y:=Row;
                Exit;  {make a quick getaway!}
              end;
           end;
           {ok, search from top of page to the startpos}
           If X+Y>2 then
              Begin
                Col:=1; Row:=1;
                If Find_Next_link(Col,Row,Pred(X),Y,AHelpRec) then
                   Begin
                     X:=Col; Y:=Row; Find_Next_Link:=true;
                   end;
               end;
         end; {find next link}

FUNCTION Find_Prev_Link( Var X,Y: Integer; EndX,EndY: Integer;
Var AHelpRec: HelpRecord): Boolean;
Var OrigX,OrigY,Col,                {Recursive routine used to find a }
    Row,StartCol,StopCol: Integer;  {hot-link on the page prev. to the}
Begin                               {current page pos. (X,Y).         }
  Find_Prev_Link:=False; {First, look from current pos to top of page. }
  For Row:=Y downto 1 do
      Begin
        If Row = Y then if Col = X then exit;
        StopCol:=1;
        If Row<>Y then StartCol:=Length(AhelpRec.HelpLines[Row])
        else StartCol:=X;
        If AhelpRec.HelpLines[Row]<>'' then
           For Col:=StartCol downto StopCol do
               If (AHelpRec.HelpLines[Row,Col]=Null) then
                   Begin
```

```
                    Find_Prev_Link:=True;
                    X:=Col; Y:=Row;
                    Exit;  {make a quick getaway!}
                  end;
              end;
              {ok, search from bottom of page to the startpos}
              Row:=MaxLinesPerPage;
              Col:=Length(AHelpRec.HelpLines[Row]);
              If Find_Prev_link(Col,Row,Succ(X),Y,AHelpRec) then
                 Begin
                    X:=Col; Y:=Row; Find_Prev_Link:=true;
                 end
         end;
      end; {find prev link}

PROCEDURE Do_Help(FileName: String; GoPage,HomePage: Word);
Const XPos = 3;
      YPos = 1;
      Color    : Byte = Black*16+White;   {This is hypertext engine.  }
      MaxStackSize = 50;                  {This routine used to read  }
                                          {and navigate through data  }
Type  StackRec = Record                   {file, "FILENAME".          }
      Page : Byte;                        {GoPage is starting         }
      Row,                                {page to display; HomePage  }
      Col  : Integer;                     {is used to specify an main }
                                          {index (or home) page.  }
Var   Result  : Integer;
      Stack   : Array[0..MaxStackSize] of StackRec;
      AHelpRec: HelpRecord;
      Ch      : CHar;
      StackLvl: Byte;
      StartCol: Integer;
      Linked,
      Load    : Boolean;
      Count   : Integer;

FUNCTION Pop_Stack: Byte;  {Pop the top page info (Stack) record}
Begin
  If StackLvl>1 then
     Begin
       Dec(StackLvl);
       Load:=True;
     end;
  Pop_Stack:=StackLvl;
end; {pop stack}

FUNCTION Push_Stack(PageNum: Byte): Byte;
Begin                          {Push a page info (stack) record.}
  Inc(StackLvl);
  Stack[StackLvl].Page:=PageNum;
  Stack[StackLvl].Col:=1;
  Stack[StackLvl].Row:=1;
  Push_Stack:=StackLvl;
```

```
end; {push stack}

Begin
  offcursor;
  If GoPage=0 then GoPage:=1;       {Make sure GoPage is valid.}
     Result:=Open_HelpFile(FileName);
  If Result=0 then
     Begin
       Load:=true;
       TextAttr :=Color;
       box(xpos,ypos,xpos+72,ypos+maxlinesperpage+2,lightblue,blue,2);
       FillChar(Stack,SizeOf(Stack),0);
       StackLvl := 0;
       If HomePage in [1..255] then StackLvl:=Push_Stack(HomePage);
          If (GoPage in [1..255]) and (GoPage<>HomePage) then
            StackLvl:=Push_Stack(GoPage);
            GotoXY(XPos+29-(Length(Header) div 2),YPos);
            Writeln(Header);
            Repeat
             With Stack[StackLvl] do
                Begin
                  If Load then  {System needs new hyperdata file page.}
                     Begin
                       Result:=Read_HelpRec(AHelpRec,Page);
       fbox(xpos,ypos,xpos+72,ypos+maxlinesperpage+2,lightblue,blue,2);
                     Show_Help_Page(XPos+1,YPos,AHelpRec);
                     If StackLvl>1 then
                        Writeat(10,24,yellow,blue,'<Pg Up> for
                                previous screen');
                        Writeat(10,25,yellow,blue,'<Esc> to end the
                                lesson');
                        Writeat(40,24,yellow,blue,chr(26)+'
              '+chr(27)+' '+chr(24)+' '+chr(25)+' to move highlight');
                        Writeat(40,25,yellow,blue,'<Enter> to select
                                highlight');
                        Linked:=Find_Next_Link(Col,Row,80,
                                MaxLinesPerPage,AHelpRec);
                        Load:=False;
                      end;
                      If Linked then { Show the hot-link } do it.}
                         Begin
                           StartCol := Col;
                           While
                          Ord(AHelprec.HelpLines[Row,StartCol-1])>127
                            do Dec(StartCol);
                     Show_HelpLine(XPos+1,YPos+Row,StartCol,Pred(Col),
                                   AHelpRec.HelpLines[Row]);
                         end;
                      Repeat Ch:=Read_KeyBoard until Ch<>Null;
           Show_HelpLine(XPos+1,YPos+Row,0,0,AHelpRec.HelpLines[Row]);
                     Case Ch of   {Now handle navigation...}
                       RArrow, DnArrow, Tab    : Begin
                         Inc(Col);
```

```
                                Linked:= Find_Next_Link(Col,Row,80,
                                        MaxLinesPerPage,AHelpRec);
                                end;
                              Return : If Linked then
                                Begin
                                  Load:=true;
                                    If (StackLvl>1) and
                                       (Stack[StackLvl-1].Page=
                                        Ord(AHelpRec.HelpLines[Row,Col+1]))
                                    then StackLvl:=Pop_Stack else
                                  StackLvl:= Push_Stack(Ord(AHelpRec.HelpLines
                                             [Row,Col+1]));
                                    end;
                              LArrow, UpArrow : Begin
                               Dec(Col);
                               Linked:=Find_Prev_Link(Col,Row,1,1,AHelpRec);
                               end;
                              PgUp : StackLvl:=Pop_Stack;
                              Homekey : Begin
                                          For Count := 1 to StackLvl do
                                              StackLvl := Pop_Stack;
                                        End;
                          end;
                        end;
            Until Ch=ESC;
          end else
               Writeln('ERROR: ',Result,' opening ',FileName,'. Unable
                        to continue.');
  {$I-} Close(HelpFile); result:=IOResult; {$I+}
end; {do help}

Begin {No init code required}
end.
```

Appendix E: Program Evaluation Survey and Responses

Thank you for participating in this evaluation of the prototype program, "The Grammar Lesson." Your honest feedback and evaluation of the program will benefit future students attending AFIT. This questionnaire will allow you to provide feedback in two important areas: program execution and lesson content. You will also be given a short "sample quiz" on concepts presented in "The Grammar Lesson" to evaluate the effectiveness of the program's instruction.

Please place an "X" in the appropriate space to answer each of the following questions:

PROGRAM EXECUTION

1. When you typed "GRAMMAR" from the DOS prompt, did the program start?

   RESPONSES: Yes: 17
   No: 0

2. Did the software ever "crash" while you were using it?

   RESPONSES: Yes: 2
   No: 17

   If "Yes," please describe the circumstances:

   RESPONSES: Both crashes occurred when the users pressed the cursor keys while on a screen without highlighted words.

3. Did you have any difficulty making a selection from the first menu?

   RESPONSES: Yes: 0
   No: 17

   If "Yes," please explain: (No responses)

4. Were you able to distinguish between normal text and key words?

   RESPONSES: Yes: 17
   No: 0

5. While reviewing the lesson, did you have any difficulty moving the highlight bar and selecting an option?

   RESPONSES: I found it very confusing: 0
   I found it somewhat confusing: 3
   I found it not at all confusing: 14

6. COMMENTS ABOUT HIGHLIGHTING AND SELECTING:

RESPONSES: Several users found it inconvenient to press <Pg Up> repeatedly to back up through previous screens. Others responded favorably to the ease of locating and highlighting key words. Several users commented that they had no problems with highlighting and selecting.

7. "The Grammar Lesson" uses a hypertext environment. Hypertext is sometimes criticized because users can lose their sense of where they are in the program. Did you ever lose your sense of where you were in this program?

RESPONSES:		
	Never:	4
	Seldom:	5
	Occasionally:	8
	Often:	0
	Very Often:	0

8. Did you ever have difficulty returning to previous screens?

RESPONSES:		
	Never:	10
	Seldom:	5
	Occasionally:	2
	Often:	0
	Very Often:	0

9. COMMENTS CONCERNING PROGRAM EXECUTION:

RESPONSES: Six users suggested that a single keystroke to return to the main menu would be helpful. One user commented that the messages telling the user how many times to press <Pg Up> were very helpful. Another user cited the simplicity and conciseness of program execution.

LESSON CONTENTS

10. Had you ever studied English grammar in terms of functions and building elements, as presented in this lesson?

RESPONSES:		
	Yes:	5
	I'm not sure:	4
	No:	8

11. How well did you understand the lesson's presentation of the two concepts?

RESPONSES:		
	Quite well:	10
	Somewhat:	7
	Slightly:	0
	Not at all:	0

12. How much did "The Grammar Lesson" help you to improve your understanding of how the language works?

RESPONSES:

Greatly:	0
Somewhat:	11
Slightly:	5
Not at all:	1

13. How useful was "The Grammar Lesson" in relation to your COMM 310 course?

RESPONSES:

Very Useful:	2
Somewhat Useful:	8
Slightly Useful:	4
Not at all Useful:	3

14. How well did "The Grammar Lesson" help you to be able to recognize the functions performed by words and groups of words?

RESPONSES:

Greatly:	0
Somewhat:	12
Slightly:	5
Not at all:	0

15. How well did "The Grammar Lesson" help you to be able to recognize the building elements of the language?

RESPONSES:

Greatly:	2
Somewhat:	10
Slightly:	4
Not at all:	1

16. How well did "The Grammar Lesson" help you to be able to recognize properly written sentences?

RESPONSES:

Greatly:	1
Somewhat:	10
Slightly:	6
Not at all:	0

17. COMMENTS ABOUT THE LESSON CONTENTS:

RESPONSES: Two users found the content too basic and the examples too simplistic. Three users commended the simplicity of the content and the examples. One user was unable to devote enough time to the tutorial.

SAMPLE QUIZ OF "THE GRAMMAR LESSON" CONCEPTS

The following questions relate to the content of "The Grammar Lesson." You may refer to the lesson to answer the questions, but to help me understand how well you retained what was presented, **please attempt to answer each question from memory first.** After each question, please indicate if you answered the question from memory or if you referred to the lesson.

18. "The Grammar Lesson" shows how three building elements perform the five functions of language. In your own words (or in the program's words), list the building elements and the functions of language:

ANSWER: Building Elements: words, phrases, clauses
Functions: naming things, showing action/existence, modifying, connecting, adding filler

RESPONSES:		Correct	Incorrect
	Answered from memory:	4	3
	Referred to the lesson:	8	0

19. Which of the five functions of language can WORDS perform?

ANSWER: all five functions

RESPONSES:		Correct	Incorrect
	Answered from memory:	8	1
	Referred to the lesson:	6	0

20. Which of the five functions of language can PHRASES perform?

ANSWER: all five functions

RESPONSES:		Correct	Incorrect
	Answered from memory:	7	1
	Referred to the lesson:	7	0

21. Which of the five functions of language can CLAUSES perform?

ANSWER: naming, modifying

RESPONSES:		Correct	Incorrect
	Answered from memory:	1	5
	Referred to the lesson:	5	4

22. According to the lesson, which function of language is performed by the articles (a, an, the)?

ANSWER: adding filler

RESPONSES:		Correct	Incorrect
	Answered from memory:	8	1
	Referred to the lesson:	6	0

23. What are the six verb tenses described in the lesson?

ANSWER: past, present, future, past perfect, present perfect, future perfect

		Correct	Incorrect
RESPONSES:	Answered from memory:	2	4
	Referred to the lesson:	9	0

24. According to the lesson, what do adjectives modify?

ANSWER: naming elements

		Correct	Incorrect
RESPONSES:	Answered from memory:	11	0
	Referred to the lesson:	4	0

25. According to the lesson, what do adverbs modify?

ANSWER: everything other than naming elements

		Correct	Incorrect
RESPONSES:	Answered from memory:	1	10
	Referred to the lesson:	4	0

26. According to the lesson, what are the two requirements of a clause?

ANSWER: naming element and action/existence element in a subject/verb relationship

		Correct	Incorrect
RESPONSES:	Answered from memory:	6	2
	Referred to the lesson:	7	0

27. According to the lesson, what are the three patterns of clauses?

ANSWER: SUBJECT + VERB
SUBJECT + VERB + OBJECT
SUBJECT + LINKING VERB + COMPLEMENT

		Correct	Incorrect
RESPONSES:	Answered from memory:	0	6
	Referred to the lesson:	9	0

28. According to the lesson, what is the minimum requirement for a sentence?

ANSWER: one independent clause

		Correct	Incorrect
RESPONSES:	Answered from memory:	3	7
	Referred to the lesson:	5	0

FOR THE FOLLOWING EXAMPLES, PLEASE INDICATE THE TYPE OF BUILDING ELEMENT UNDERLINED AND THE FUNCTION IT IS PERFORMING:

29. We went backstage to congratulate them <u>after</u> the performance.

ANSWER: word, connecting

		Correct	Incorrect
RESPONSES:	Answered from memory:	8	4
	Referred to the lesson:	3	0

30. We congratulated them <u>after they finished their performance</u>.

ANSWER: clause, modifying

		Correct	Incorrect
RESPONSES:	Answered from memory:	5	6
	Referred to the lesson:	3	1

31. The performers <u>were entertaining</u> the audience throughout the show.

ANSWER: phrase, showing action

		Correct	Incorrect
RESPONSES:	Answered from memory:	13	1
	Referred to the lesson:	1	0

32. <u>Entertaining audiences</u> was a joy for the performers.

ANSWER: phrase, naming element

		Correct	Incorrect
RESPONSES:	Answered from memory:	3	9
	Referred to the lesson:	3	0

33. <u>Entertaining audiences world wide</u>, they enjoyed performing.

ANSWER: phrase, modifying

		Correct	Incorrect
RESPONSES:	Answered from memory:	3	9
	Referred to the lesson:	2	1

34. The performance <u>that we attended</u> was entertaining and informative.

ANSWER: clause, modifying

		Correct	Incorrect
RESPONSES:	Answered from memory:	5	6
	Referred to the lesson:	3	1

35. They knew <u>that they entertained their audiences every night</u>.

ANSWER: clause, naming element

		Correct	Incorrect
RESPONSES:	Answered from memory:	7	6
	Referred to the lesson:	2	0

36. There will be over five hundred people in the audience tonight.

ANSWER: word, adding filler

		Correct	Incorrect
RESPONSES:	Answered from memory:	7	8
	Referred to the lesson:	0	0

37. We will try to arrive early for tonight's performance.

ANSWER: phrase, naming element

		Correct	Incorrect
RESPONSES:	Answered from memory:	7	7
	Referred to the lesson:	1	0

38. We will try to arrive early, even though we are working late today.

ANSWER: phrase, connecting

		Correct	Incorrect
RESPONSES:	Answered from memory:	7	6
	Referred to the lesson:	2	0

39. If you referred to the lesson for any of these questions, how easily were you able to locate the answers in the lesson?

RESPONSES:	Very Easily:	2
	Somewhat Easily:	7
	Not Very Easily:	3
	Didn't refer to lesson:	3

OTHER COMMENTS

40. Recommendations for improving "The Grammar Lesson":

RESPONSES:
1. a menu of key words to simplify finding concepts
2. more interaction or quizzes to reinforce learning
3. a reference on each screen to help orient the user
4. fewer keystrokes to return to menus
5. automatically highlight "NEXT" when a new screen appears, rather than making the user move to it

41. Other general comments or suggestions:

RESPONSES:
1. Software should be sent to incoming AFIT students so they can review basic grammar concepts before arriving at the school.
2. excellent screens--not too much information, clean easy to read
3. More time should have been allotted for testing. (NOTE: 14 days were allotted)
4. a good tool for future reference
5. The breakdown and simplification of content was helpful.

Bibliography

1. Anderson-Inman, Lynne. "Electronic Studying: Information Organizers to Help Students to Study 'Better' not 'Harder,' Part II," The Computing Teacher, 16: 21-29, 53 (June 1989).

2. Bevilacqua, Ann F. "Hypertext: Behind the Hype," American Libraries, 20: 158-62 (February 1989).

3. Borland International. "No-Nonsense License Agreement," distributed with Turbo Pascal User's Guide. Scotts Valley CA: Borland International, 1989.

4. -----. Turbo Pascal Reference Guide. Scotts Valley CA: Borland International, 1989.

5. -----. Turbo Pascal User's Guide. Scotts Valley CA: Borland International, 1989.

6. Burke, Robert L. CAI Sourcebook. Englewood Cliffs NJ: Prentice-Hall, Inc., 1982.

7. Burleson, Sandra K. "Computer-Assisted Instruction," Business and Economic Review, 35(1): 3-7 (October-December 1988).

8. Byles, Torrey. "A Context For Hypertext: Some Suggested Elements of Style," Wilson Library Bulletin, 63: 60-62 (November 1988).

9. Carr, Clay. "Hypertext: A New Training Tool?" Educational Technology, 28: 7-11 (August 1988).

10. Colorado, Rafael J. "Computer-Assisted Instruction Research: A Critical Assessment," Journal of Research on Computing in Education, 20(3): 226-33 (Spring 1988).

11. Department of the Air Force. AFIT Catalog, 1989-1991. Washington: US Government Printing Office, 1989.

12. Dossett, Dennis L. and Patti Hulverson. "Increased Technical Training Efficiency: Peer Training Via Computer Assisted Instruction," Journal of Applied Psychology, 68: 552-558 (November 1983).

13. Dossett, Dennis L. and Lee J. Konczak. "New and Improved, or Just New?" Training and Development Journal, 39(7): 41-44 (July 1985).

14. Faulkner, Claude W. Writing Good Sentences: A Functional Approach To Sentence Structure, Grammar, and Punctuation (Revised Edition). New York: Charles Scribner's Sons, 1973.

15. Fenno, Charles R. Class handout distributed in COMM 687, Theory and Practice of Professional Communication. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, July 1990.

16. Franklin, Carl. "Hypertext Defined and Applied," ONLINE, 13: 37-49 (May 1989).

17. -----. "Hypertext on the PC: Guide, Version 1," Database, 11: 95-100 (February 1988).

18. Fryer, Capt Richard A. Computer-Assisted Instruction and Its Application to Air Force Civil Engineering. MS Thesis, AFIT/GEM/LSM/87S-7. School of Systems and Logistics, Wright-Patterson AFB OH, September 1987 (AD-A187915).

19. Galagan, Patricia. "Computers and Training: Allies or Enemies?" Training and Development Journal, 41(4): 73-76 (April 1987).

20. Gessner, Rick. "Building a Hypertext System," Dr. Dobb's Journal, 15(6): 22-33 (September 1990).

21. Godfrey, David and Sharon Sterling. The Elements of CAL: The How-To Book on Computer Aided Learning. Reston VA: Reston Publishing Company, Inc, 1982.

22. Grabe, Mark et al. "An Evaluation of Computer Assisted Study in Controlled and Free Access Settings," Journal of Computer-Based Instruction, 16(3): 110-16 (Summer 1989).

23. Hannafin, Michael J. and Simon Hooper. "An Integrated Framework for CBI Screen Design and Layout," Computers in Human Behavior, Robert D. Tennyson, editor, 5(3): 155-165 (1989).

24. Hetzel, Bill. The Complete Guide to Software Testing (2d Edition). Wellesley MA: QED Information Sciences, Inc, 1988.

25. Hodges, John C. et al. Harbrace College Handbook (11th Edition). San Diego: Harcourt Brace Janovich, Publishers, 1990.

26. Horton, William. "Is Hypertext the Best Way to Document Your Product? An Assay for Designers," Technical Communication: Journal of the Society for Technical Communication, 38(1):20-35 (February 1991).

27. Jones, A. Russell and Esther R. Steinberg. "Interface Design: An Educational Perspective," Journal of Systems and Software, 11(3): 199-204 (March 1990).

28. Kemner-Richardson, Sue et al. Computer-Assisted Instruction: Decision Handbook. May 1982-December 1982. Contract F3361581C-0007. Denver CO: Denver Research Institute, University of Denver, April 1985 (AD-A154646).

29. King, Todd. "A Self-Referential Hypertext Engine," Dr. Dobb's Journal, 15(6):34-38 (June 1990).

30. Kinnell, Susan K. "Hypertext on the PC: Guide, Version 2," Database, 12: 62-66 (August 1989).

31. Kulik, C-L. C. et al. "The Effectiveness of Computer-Based Adult Education: A Meta-Analysis," Journal of Educational Computing Research, 2: 235-252 (1989).

32. Leggett, Glenn. Prentice Hall Handbook for Writers (10th Edition), edited by Richard S. Beal. Englewood Cliffs NJ: Prentice Hall, 1988.

33. Manes, Stephen. "Hypertext: A Breath of Air Freshener," PC Magazine, 6: 91, 95 (June 1987).

34. Manion, Mary H. "CAI Modes of Delivery and Interaction: New Perspectives for Expanding Applications," Educational Technology, 25(1): 25-27 (January 1985).

35. Marchionini, Gary. "Hypermedia and Learning: Freedom and Chaos," Educational Technology, 28: 8-12 (November 1989).

36. Marius, Richard, and Harvey S. Wiener. The McGraw-Hill College Handbook. New York: McGraw-Hill Book Company, 1985.

37. Matta, Khalil F. and Gary M. Kern. "A Framework for Research in Computer-Aided Instruction: Challenges and Opportunities," Computers and Education, 13(1): 77-84 (1989).

38. Neapolitan, Jerry. "A Test of Simple Computer-Assisted Instructional Software," Teaching Sociology, 17(4): 493-96 (October 1989).

39. Oppenheim, Pam. "The Latest Word on Text-Based Management Systems," National Productivity Review, 9: 83-88 (Winter 1989/90).

40. Orwig, Gary W. Creating Computer Programs for Learning. Reston VA: Reston Publishing Company, Inc, 1983.

41. Selfe, Cynthia. Computer-Assisted Instruction in Composition: Create Your Own. Houghton: Michigan Technological University, 1986.

42. Sivley, 1st Lt Michael A. Assessing the Feasibility of Computer-Assisted Instruction for the Standard Base Supply Systems Equipment Management Course. MS Thesis, AFIT/GLM/LSR/89S-54. School of Systems and Logistics, Wright-Patterson AFB OH, September 1989 (AD-A215071).

43. Smith, Karen E. "Hypertext--Linking to the Future," ONLINE, 12: 32-40 (March 1988).

44. Steinberg, Esther R. Teaching Computers To Teach. Hillside NJ: Lawrence Erlbaum Associates, Publishers, 1984.

45. TechnoJock's Turbo Toolkit. Houston TX: TechnoJock Software, Inc, 1990.

46. Thompson, James G. and Sally Jorgensen. "How Interactive is Instructional Technology? Alternative Models for Looking at Interactions Between Learners and Media," Educational Technology, 19: 24-26 (February 1989).

47. Troyka, Lynn Q., ed. Simon & Schuster Handbook for Writers (Second Edition). Englewood Cliffs NJ: Prentice Hall, 1990.

48. Vaughan, David K. "Review of Grammar, Mechanics, Usage, and Style." Handout for COMM 310, Fundamentals of Written Communications. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, June 1990.

49. -----, course director. Syllabus for COMM 310, Fundamentals of Written Communications. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, June 1990.

50. Wher, Joe. "Instructor-Led or Computer-Based: Which Will Work Best for You?" Training and Development Journal, 42(6): 18-21 (June 1988).

51. Williams, Kent E. et al. An Evaluation of Characteristics Contributing Towards Ease of User-Computer Interface in a Computer-Aided Instruction Exercise. Technical Report TR87-030. Naval Training Systems Center, Orlando FL, December 1987 (AD-A199458).

Vita

Captain Frank E. Jones was born on 17 November 1961 in Savannah, Georgia. In 1979, he graduated from Charlotte High School, Punta Gorda, Florida, and in 1980, he enlisted in the United States Air Force. In 1982, he was released from active duty to attend the University of Florida as a member of the Air Force Reserve Officer Training Corps (AFROTC). He graduated with honors in 1984, earning the degree Bachelor of Arts in Religion. In 1985, he was a distinguished graduate of AFROTC and received a regular commission as a second lieutenant in the United States Air Force. His first assignment was to Lowry Air Force Base, Colorado, where he served as executive officer, 3405th Student Group, and headquarters squadron section commander, 3460th Technical Training Group. His next assignment was to Elmendorf Air Force Base, Alaska, where he served as squadron section commander, 962d Airborne Warning and Control Squadron. In 1988, he was a distinguished graduate of Squadron Officer School at Maxwell Air Force Base, Alabama. In 1989, he graduated with honors from Chapman College, receiving the degree Master of Science in Human Resource Management and Development. He entered the School of Systems and Logistics, Air Force Institute of Technology, in May 1990.

Permanent Address: 300 Beverly Ave
Port Charlotte FL 33952

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1991	Master's Thesis

4. TITLE AND SUBTITLE
AN INVESTIGATION INTO THE USE OF COMPUTER-ASSISTED INSTRUCTION TO PRESENT BASIC ENGLISH GRAMMAR CONCEPTS

5. FUNDING NUMBERS

6. AUTHOR(S)
Frank E. Jones, Capt, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GIR/LSR/91D-10

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The purpose of this research was to develop a computer-assisted instruction (CAI) tutorial that would address basic grammatical concepts. The research consisted of four phases. The first phase was a review of applicable CAI literature, including theories, instructional and programming techniques, and previous research findings. The second phase was a survey of communication professors to determine the grammatical errors commonly committed by students. The third phase involved development of the tutorial content and the program software. The resulting tutorial presents basic grammar in terms of the three building elements of language (words, phrases, clauses) and the five functions they perform (naming, showing action/condition/state, modifying, connecting, adding filler) to create sentences. The program software, written in Turbo Pascal, employs a hypertext environment for its computer-user interface. The final phase was a preliminary field test of the program to evaluate its execution and hypertext interface. The project resulted in a functional, stand-alone CAI tutorial that addresses basic grammatical concepts. The hypertext program code can be used to create and display any hypertext document. The study recommends that further research be conducted to broaden the scope of the tutorial and to evaluate its effectiveness as an instructional tool. (25)

14. SUBJECT TERMS	15. NUMBER OF PAGES
Computer-Aided Instruction, User-Computer Interface, Computer Programs, English Language, Grammar, Hypertext, Theses.	194
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

AFIT Control Number AFIT/GIR/LSR/91D-10

AFIT RESEARCH ASSESSMENT

The purpose of this questionnaire is to determine the potential for current and future applications of AFIT thesis research. Please return completed questionnaires to: AFIT/LSC, Wright-Patterson AFB OH 45433-6583.

1. Did this research contribute to a current research project?

   a. Yes b. No

2. Do you believe this research topic is significant enough that it would have been researched (or contracted) by your organization or another agency if AFIT had not researched it?

   a. Yes b. No

3. The benefits of AFIT research can often be expressed by the equivalent value that your agency received by virtue of AFIT performing the research. Please estimate what this research would have cost in terms of manpower and/or dollars if it had been accomplished under contract or if it had been done in-house.

   Man Years _____ $_____

4. Often it is not possible to attach equivalent dollar values to research, although the results of the research may, in fact, be important. Whether or not you were able to establish an equivalent value for this research (3 above), what is your estimate of its significance?

   a. Highly Significant b. Significant c. Slightly Significant d. Of No Significance

5. Comments

_____ _____

Name and Grade Organization

_____ _____

Position or Title Address