An experimental, computerized, ESL lesson on two-word verbs using mnemonics and imagery

Kevin Loyd Quinn
Iowa State University

Follow this and additional works at: http://lib.dr.iastate.edu/rtd

Part of the Bilingual, Multilingual, and Multicultural Education Commons, English Language and Literature Commons, and the First and Second Language Acquisition Commons

Recommended Citation
Quinn, Kevin Loyd, "An experimental, computerized, ESL lesson on two-word verbs using mnemonics and imagery" (1984). Retrospective Theses and Dissertations. 35.
http://lib.dr.iastate.edu/rtd/35

This Thesis is brought to you for free and open access by Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
An experimental, computerized, ESL lesson on two-word verbs
using mnemonics and imagery

by

Kevin Loyd Quinn

A Thesis Submitted to the
Graduate Faculty in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF ARTS

Major: English

Iowa State University
Ames, Iowa
1984
## TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>RATIONAL FOR THE LESSON</td>
<td>9</td>
</tr>
<tr>
<td>DESIGN OF THE LESSON</td>
<td>11</td>
</tr>
<tr>
<td>TESTING AND EVALUATION</td>
<td>13</td>
</tr>
<tr>
<td>Teacher Feedback</td>
<td>13</td>
</tr>
<tr>
<td>Student Feedback</td>
<td>14</td>
</tr>
<tr>
<td>Process Observation</td>
<td>17</td>
</tr>
<tr>
<td>Pre- and Post-Testing</td>
<td>18</td>
</tr>
<tr>
<td>Portfolio Evaluation</td>
<td>20</td>
</tr>
<tr>
<td>CONCLUSIONS</td>
<td>22</td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td>24</td>
</tr>
<tr>
<td>APPENDIX 1 SCREEN DISPLAYS</td>
<td>28</td>
</tr>
<tr>
<td>APPENDIX 2 PRE- AND POST-TESTS</td>
<td>67</td>
</tr>
<tr>
<td>APPENDIX 3 QUESTIONNAIRE</td>
<td>72</td>
</tr>
<tr>
<td>APPENDIX 4 COMPUTER CODE</td>
<td>74</td>
</tr>
</tbody>
</table>
INTRODUCTION

The thesis project I have completed has been the development and evaluation of a computer-assisted instruction (CAI) lesson for students of English as a Second Language (ESL). In particular, I have considered the value of an application of mnemonics in such a CAI lesson. Mnemonics is the system of memorizing words or information by linking a fact to be learned to concrete visual images.

Mnemonics has a long history. According to Paivio (1971), it was developed around 500 B.C. by Simonides, who used imagery as the major memory mechanism for remembering speeches and word lists. Simonides, unfortunately, did not record the specifics of his memory technique, but others, such as Cicero (a contemporary of Simonides), did:

Persons so desiring to train this faculty of memory must select places and form mental images of the things they wish to remember and store those things in those places, so that the order of the places is associated with the things themselves and we shall employ the places and images respectively as a wax writing--tablet and the letters written on it (Yates 1972, p. 2).

Simonides' method became known as the "Loci-system" of mnemonics, because the method relied on places and locations to retrieve information.

Soon, this early method of memorizing became commonly used, and a number of variations of the earlier rules for places and images were developed. One of the earliest attempts at variation was made by Quintilian (around 500 B.C.), who used not only an "architectural" method (i.e., items to be memorized were paired with the locations of a
room), but also used an "imaginary walk" method, in which he imagined himself taking a walk and seeing houses, trees and other "signposts." The "to-be-remembered items are imagined at these prominent locations and retrieved by retracing the route" (Paivio 1971, p. 161). Another variation was that of Peter Ramus in the sixteenth century. According to Paivio (1971), he used a method combining spatial memorization and an abstract, alphabetically-coded symbol system.

During Ramus's time, mnemonics were quite popular. Many theologians, orators and authors relied on spatially-organized mnemonic systems, or variations of them, to provide a way to remember words and ideas. Albert Magnus and Thomas Aquinas advocated using mnemonics to remember abstract concepts related to ethics and morality.

Such a use of the imagery mnemonic may have been expressed rather generally in both the visual arts and the writing and rhetoric of the period. Grotesque figures symbolizing the virtues and vices, such as charity and envy, presumably reflected the ancient instruction to form images that are striking and novel. Verbal pictures may have been similarly drawn by the preachers of the day for the same practical end of providing images that would serve as reminders of the points of a sermon on sin (Paivio 1971, p. 161).

Others during this period (the Hermetic philosophers, Camillo, Bruno and Fludd) applied mnemonics to astrology and the occult (Paivio, 1971).

By the seventeenth century, mnemonics had become so popular that some had begun to make a living lecturing on the "art" of mnemonics and demonstrating its effectiveness. One of the foremost of these professional mnemonists was Gregor von Feinagle, who lectured in Europe on mnemonics in the nineteenth century. Feinagle used "an elaborate com-
bination of the method of places and images and a number-alphabet-word code" like Ramus's (Paivio 1971, p. 167). Feinagle is also noteworthy in that he was one of the first to apply a mnemonic system to language learning.

Also during this time, several new forms of mnemonics were increasingly practiced. Two of the important ones are the rhyming system and the "Hook" system. The rhyming system mnemonic device involved two steps: pairing numbers with a set group of easily-visualized concrete nouns (e.g., "one-bun," "two-shoe," "three-tree," etc.) and making associations, usually visual, by linking the concrete noun (e.g., "bun") with the item to be memorized. If the memorizer needed to remember the name of a certain species of mouse, for example, he or she would visualize that mouse eating a bun (a bun is the first concrete noun on the list). That item and its imagery cue (the mouse and the bun) would be stored in the first position on the list. Additional items would be stored in a sequential list.

A second form, called the "Hook" method, was based on an interaction between rhyme and imagery, which, according to Paivio and Desrochers (1981), was invented by Pierre Herigon in 1694. The Hook method involves the use of a number-letter alphabet (e.g., 1-T, 2-N, 3-M, 4-R, etc.) which can serve as "hooks" for other words. The letters associated with each number stand for words (e.g., T = tea, N = Noah, M = moose, R = room, etc.). Paivio (1971) suggests that by associating a word to be learned with one of the "hook" words, which is already memorized, a connection can be made and the new word remembered.
The popularity of mnemonics has been sustained into the 20th century, particularly in the application of mnemonics to the learning of a second language vocabulary. Two of the foremost advocates of applying mnemonics to learning a foreign vocabulary are Lorayne and Lucas (1974), who suggest learning foreign words by linking them with pictureable English words that sound like those foreign words. Their method, called the "Link" method, is designed to form a linked list by which the first item will recall the second, and so on. Others to apply associational strategies to learning vocabulary are Cohen and Aphek (1979). They suggest using as many mnemonic devices as possible and letting the student choose the strategy that best works.

But until approximately the time Lorayne and Lucas's book was published, little experimental data had been compiled concerning the application of mnemonics to second language learning. However, since then, a few experiments have been undertaken to determine the efficacy of mnemonics for second-language learning. Two major research efforts stand out.

The first of these involves the use of a method called the "Keyword technique." This technique, which was developed by Atkinson and Raugh (1974), is a "peg" system, meaning that, like the Hook technique and the "one-bun, two-shoe" methods, it attaches words to be memorized to an already memorized peg of some type.

Atkinson (1975, p. 828) said he developed the Keyword technique for foreign language classes because he feels it is difficult to find classroom instructional situations in which mnemonic aids are truly useful. "In Cicero's time, these aids may have had some usefulness,
but in this age of cheap memory devices (including pencil and paper),
the value of mnemonic aids is questionable. An exception may be the
Keyword method." The Keyword technique involves learning a foreign
word by selecting a native language word that sounds like all or part
of the foreign word and is easy to visualize. For example, for
speakers of English, the Spanish word for beach, "playa," which sounds
like "play," could be hooked to an image of people playing on a beach.

Atkinson (1975, p. 16) also suggested using CAI, together with
imagery mnemonics, in foreign language learning:

> It is interesting to speculate on the potentials of the Keyword method in a foreign language curriculum. One possibility is that the Keyword method could be used in a specialized computerized program, supplementing an introductory course in grammar/composition. The purpose of this lesson would be to provide the student with an individualized procedure for rapidly expanding vocabulary acquisition.

Atkinson, in fact, has used computers in many of his experiments.

Atkinson and Raugh (1974) conducted four experiments using the
Keyword technique and CAI with Stanford University students learning
Spanish. Two experiments compared groups using the Keyword technique
to groups using rote or rehearsal learning. A third experiment
compared a group using the Keyword method with a group allowed to use
any memory technique other than mnemonics. A fourth experiment
compared a group using the Keyword technique with a group which could
choose whatever technique they preferred: the Keyword technique or
their own method. In all experiments, the Keyword technique proved to
be highly effective compared to the control group. In the best
results, the Keyword group scored 88 percent correct, while the control
group scored 28 percent. Students in both groups were given a total of 15 minutes to learn 60 vocabulary items using either the Keyword method or rote memorization. Testing immediately followed the learning of the vocabulary items.

Similarly, Atkinson and Raugh (1974) experimented using the Keyword technique with students learning Russian. They wanted to determine if the Keyword technique would work for languages other than a somewhat familiar Romance language like Spanish, which has many English cognates. In this computer-controlled study, 120 words were learned in sets of 40 on successive days. A computer system presented the Russian words through headphones while also displaying the English translation and keyword. A 72 percent retention rate after one month was recorded for the students learning the vocabulary through the Keyword technique.

In addition to the convincing data which have come from his research, Atkinson (1975) reports several impressions as a result of experiments on how to improve the application of mnemonics. He has discovered that new users perform better if keywords are provided for them. However, those who were experienced in mnemonics performed better if they were given the opportunity to provide their own keywords. This result has been confirmed by several other experimenters. (See Pressley, et al., 1980 and Levin, et al., 1979.) However, despite the innovative methods used, Atkinson's Keyword method has some limitations.

One of the severest criticisms of the Keyword method, and one of the most recent, came from Hall, et al. (1981), who concluded that the value of the Keyword method was limited. These researchers found that
when keywords were supplied by the experimenter, the gain of the Keyword group was similar or somewhat inferior to that of the control group, both in translation from the native language to the target language (English) and in translation from the target language to the native language. The Keyword group was clearly inferior to the control group when students were required to generate keywords, and showed no better retention than the control group after a week. The only condition in which the Keyword group scored better results than the control group was when words to be learned were presented successively, rather than simultaneously, with the experimenter controlling the pace of the experiment. But when students were allowed to study all items simultaneously, at their own pace, using their own memorization methods, they performed better than an experimenter-paced group using the Keyword method. This last finding seems to support the method of learner-controlled learning suggested by Cohen and Aphek (1979).

Other researchers to criticize the Keyword method include Paivio and Desrochers (1981), who feel that there are several problems with the Keyword technique which need to be addressed: the Keyword technique seems to be valuable only in an experimental setting rather than in a classroom setting (although Atkinson had claimed the Keyword technique would work in a classroom setting just as well as in a laboratory); the Keyword technique relies on external cues to retrieve vocabulary from memory, instead of already learned foreign words; and the Keyword technique seems to be somewhat ineffective with expressive or productive skills. In addition, keyword mnemonics seem to work best for those who are either young (elementary and junior high school), or
inexperienced beginners.

Paivio and Desrochers (1981) still view the Keyword technique as valuable, yet they have seen a need for another mnemonic device for use with language learners. Thus they have applied the Hook technique to second language learning. Instead of using a keyword chosen from the native language to serve as a hook for the to-be-learned vocabulary, these researchers chose to use a mnemonic from the target language.

One experiment to apply the Hook technique was done by Desrochers (1980). He used an "extended" Hook technique in which the subjects coded the gender of French nouns by adding a natural gender tag (e.g. "man" or "woman") into a mnemonic image. He reported "substantial" gains of the group using the Hook method over the gains by the control group.

In another experiment, Paivio and Desrochers (1981, p. 314) compared their mnemonic method and rote learning. They found "the imagery condition was far superior to the rote association condition on all trials" when applied to 96 French words.

In addition to these studies, which focused on second-language learners, there are other recent studies which generally confirm the guidelines for applications of mnemonics established by Simonides, Cicero and their contemporaries. (See Miller, 1956; Middleton, 1887, and Mandler, 1967.) Mnemonics has the best results if the imagery is vivid, if it is well-organized, if it involves interaction between the item to be learned and memory cues, if it is "natural" and if it is generated by a user trained in effective mnemonics.
RATIONALE FOR THE LESSON

In view of the work done in mnemonics, in particular in CAI, I decided to apply it to ESL. I wanted to design and evaluate a lesson on the memorization of the meaning of two-word verbs (verb plus particle combinations, such as "take on," "take off," etc.) for Iowa State University’s two ESL programs: the Intensive English and Orientation Program, for students whose scores on the Test of English as a Foreign Language (TOEFL) are below 500, and the post-admission ESL program, for students who have scored above 500 on TOEFL but are not sufficiently prepared to write compositions of native speaker quality.

My reason for choosing to do a lesson on two-word verbs is that they are confusing words for even advanced ESL learners because their meaning is illogical and unpredictable. For example, "to turn out" is a two-word verb, yet it does not usually mean "to turn" (change directions) plus "out"; it usually means "to attend" or "to produce." One difficulty with a lesson covering two-word verbs, however, is that there are more than 30 "families," or groupings, of two-word verbs. Thus, I selected the "take family" because its members are very frequently used by native speakers of English. The most common members of the family, which are the members covered in the computer lesson, are "take after," "take back," "take down," "take for," "take in," "take off," "take on," "take out," "take over," and "take up." Some of these words have more than one meaning ("take down" has two meanings, "take off" has three meanings, "take out" has two meanings and "take up" has two meanings). Thus, there are a total of 15 meanings presented for
In designing this computer lesson, I decided to use Cohen's and Aphek's (1979) suggestion that language learners be provided a variety of strategies to choose from. Thus, I presented the following associational strategies: a picture showing the meaning of the two-word verb (e.g., one of the pictures appearing in the computer lesson shows a picture of an airplane taking off as one of the meanings of the verb "take off"); rebuses (a symbol representing a word--e.g., the word "up" is represented by an arrow pointing upward); colors (the computer system I used allows for the use of seven colors); a short definition of the word, and a sentence or paragraph showing a larger context for the word. Students are told that colors, words and rebuses all provide possible associational strategies, and that they may choose any or all of the strategies.
DESIGN OF THE LESSON

The lesson has four parts: an introductory section, an explanation of the mnemonic strategies to be used in the lesson, a presentation of the two-word verb family with accompanying mnemonic devices and a practice section.

The introductory section of the lesson contains a brief overview of the grammar of two-word verbs. It explains the separability and inseparability of two-word verbs and the fact that two-word verbs can be used in all tenses. This section also contains a diagnostic test. It may be inserted or deleted as the teacher desires. For the purpose of evaluating the pilot program, the diagnostic test was removed from the lesson, and paper and pencil pre- and post-tests were administered to determine how much a student benefited from the lesson. (See Appendix 2.) After evaluation of the program, the diagnostic test will be inserted.

The explanation of the mnemonic strategy section lists the strategies to be used in the lesson and displays the rebuses and colors to be used to associate meaning with each two-word verb.

The presentation section follows with a full screen display for each two-word verb. Each screen display shows a figure or object depicting the meaning of the two-word verb, a one- or two-word definition, a context and the colors and the rebus for it.

The practice section of the lesson has two subsections which allow the student to apply what has been memorized earlier in the lesson.

Several other important features were included in this lesson.
The student is allowed to break out of either of the two practice subsections at any time he or she wants to return to the lesson's directory and review earlier material. Also, during the practice section of the lesson, the student is allowed to stop work at any time. In addition, the first practice subsection has fewer than one-third of the total practice items, so that a student may receive feedback before beginning the second subsection.
Throughout the development and implementation of this lesson, I was continually evaluating and testing it to increase its effectiveness. This evaluation was based on five principles of CAI lesson evaluation suggested by Schreck (1978). Each is described below, and its application to the present lesson is discussed. The evaluation methods are listed in the approximate order I used them to evaluate the lesson.

Teacher Feedback

According to Schreck, teacher feedback is one of the most valuable evaluation methods. Teachers observe the lesson in progress and the student response to it, making comments and suggestions. Teacher feedback provides more reliable, objective and knowledgeable feedback than student evaluation of lessons. Teachers, by definition, should have a better idea of what constitutes an effective lesson than students would. They have spent time developing and evaluating lessons, and they have an understanding of what is relevant and useful for a particular teaching task. One concern, however, with teacher feedback is that the time needed to evaluate lessons properly is considerable, so that it is often difficult to obtain an adequate number of comments.

Nevertheless, I decided to have as many teachers in ISU's ESL and IEOP programs as possible evaluate the lesson. Throughout the fall and winter, six teachers observed the lesson and provided comments to me. Much of the feedback received during this time concerned modifications of lesson content to improve its "naturalness," its correctness and its readability (smoothness of transitions and conciseness).
Student Feedback

Student feedback is the student response, either written or oral, concerning the value of the lesson. The rationale for this method is obvious. Since the students are the ones who are actually trying to learn, then they should be the ones who can tell best how effective it is. To an extent, this is true—if the students are open, uninhibited, and willing to respond. However, according to Schreck, there have been questions as to whether student feedback would reflect popularity of a particular teacher, and hence, any lesson developed by him/her. In addition, there have been questions over how to measure student responses and how to properly administer "instruments" that would be sensitive to student input.

Here, despite the doubts raised, it seemed that student evaluation would be beneficial, if for no other reason than to discover how well the students liked the lesson. Therefore, from the conception of the lesson through the time of its implementation, I asked for feedback. During the development stage, the feedback was informal, but after the lesson had been completed, the feedback was mostly formal, in the form of answers to a questionnaire. (See Appendix 3.) During development of the lesson, the most common comment I received from students concerned the speed of the lesson when a great number of people were using the system. One problem that resulted was that when the computer was slow, some of the students would, out of frustration, repeatedly press the "return" key, which normally would cause the lesson to move to the next display. Unfortunately, while the terminal did not respond quickly, it did "remember" how many times the "return" key had been pressed. When
the terminal did respond, it would process several of the "return" commands and display several screens before the student had a chance to read or see everything displayed. The main problem in this situation is with the demands placed on the central computer which runs the lesson. I later learned of some additional code to insert in this lesson that would alleviate the problem of build-up of keyboard commands. However, at least in the near future, little is being done to improve the computer's processing speed at peak use times. Overall, however, students seemed to enjoy the concept of the lesson, i.e., using pictures, colors, definitions and symbols as memory aids. (The main computer code for the lesson is listed in Appendix 4.)

Twenty-seven students completed the computer lesson. Soon after this, they filled out my questionnaire with questions concerning what strategy they used most frequently, what they liked and disliked about the lesson, and how they would rate various aspects of the lesson on a scale from 1 to 10. Comments from the questionnaire were positive. One student wrote, "I recommend this lesson to all the foreign students, because I think that all of them have problems with two-word verbs." Another student wrote, "We have fun besides studying English." Other favorable comments concerned the variety of mnemonic strategies presented, the immediate feedback in the practice section, the chance to practice what had been learned, the opportunity to take as much time as needed and the clarity of the explanations provided.

Students also suggested improvements for the lesson. Several students said they were confused by the numerous colors used for displaying the questions. One student suggested using just one color for
the questions, while having the symbols in different colors.

Others were annoyed by how long it took to plot the pictures on the screen. Students also mentioned that to them the pictures were either unclear or lacked value as a mnemonic device, and one student suggested having better pictures. It is also interesting to note that the questionnaire results show that the full screen pictures received the lowest average value rating (7.47 on a scale from 1-10) among the categories of "Organization," "Pictures," "Explanations" and "Overall." None of the students indicated that full-screen pictures were the most useful mnemonic strategy for them. There are many possible reasons why students found the pictures to be relatively less useful: cultural bias, unclear instructions, difficulty of forming mental images of the pictures, lack of quality in the pictures and/or a reluctance to practice mnemonics. I suspect that, to some extent, all of these were factors.

This reluctancy to use mnemonics apparently was not limited to the pictures, however. When students were asked which memory device proved most useful, the majority indicated that short definitions of the two-word were the most useful. The next most common favorite was the use of sentence contexts, followed by a tie between color and symbol mnemonics. These results are in line with earlier observations. Paivio (1971) observed that many students are reluctant to use mnemonics, perhaps because of the familiarity of the rote memorization methods which they have been using for their entire lives with considerable success. Thus, instead of trying something new, students rely on a proven method of learning.
One other interesting finding is that a few students used a variety of mnemonic strategies or combinations of strategies. While most students chose to rely solely on either short definitions or sentence contexts, four chose combinations of several of the devices. The numbers here are too small to make any inferences; but it seems that, at least for some of the students, the availability of a variety of mnemonic devices did help them. Thus, Cohen and Aphek's (1979) suggestion that a variety of mnemonic strategies should be offered seems to be supported by student feedback.

Process Observation

Process observation is simply observing the interaction between the teacher (in this case the computer) and the student while an actual lesson is in progress. The rationale behind process observation is that it can explain how the techniques used in a lesson work and what else a student may learn in addition to the stated objectives of the lesson.

One advantage of process observation is that it is easy. A drawback to process observations, however, is the possible interference between student and observer. Schreck (p. 59) terms this a "reactive" effect. "Watching a teacher teach is an artificial undertaking if one is not part of the class ... for this reason, observations of learning introduce bias which somehow must be taken into account when interpreting findings."

Schreck listed two means of process observation which could be applied to the evaluation of CAI lessons: a) "over the shoulder" observation and b) an internal reporting program on the part of the
computer which would record the student's answers and feedback given. It seemed valuable to attempt both of these methods. During development of the lesson, I watched ten students and six teachers use the lesson. During these observations I noticed a number of errors or inconsistencies in the lesson. On one occasion, I noticed that computer code was being displayed along with some of the graphics displays. On another occasion, the graphics seemed unusually slow—a problem due to the computer displaying one picture repeatedly instead of just once.

In addition to "over the shoulder" observation, the computer system at Iowa State University also provides a means of reporting some information about the student's interaction with the computer, such as the amount of time on the system, and whether the lesson was completed. This feedback proved to be of little value to me. However, one interesting fact which I noted from the computer's reports was that, on the average, women took longer to complete the lesson. Perhaps the women were more careful or concerned about their performance than the men.

**Pre- and Post-Testing**

Schreck (p. 44) refers to Hayes, Lambert and Tucker (1967), stating that the most widely-used method for evaluating language programs is pre- and post-testing, which has the advantage of providing a concrete measure of what has been learned.

However, as Schreck observes, a number of criticisms have been made against pre- and post-tests. Schreck cites Stake (1973) in addressing the concern that pre- and post-tests assume the measurability of outcomes at the time of completion of instruction, or even
at any time after that. Schreck (p. 12) also cites Savignon (1972), who states that a pre- and post-test, if it is discrete point, may actually be testing something less than the skill required for effective communication. "The language proficiency test may be said to measure linguistic competence while actual use of the language for communication requires communicative competence." Thus, care should be exercised in interpreting the results in pre- and post-testing.

However, here this means of evaluation seemed useful, since only 10 words would be tested, and the knowledge of two-word verbs, which have definite meanings, can be measured by discrete point tests. For these reasons, pre- and post-tests were administered as part of the evaluation of the final version of the lesson.

The pre- and post-tests were constructed to contain "natural" discourse, i.e., items which might occur in everyday, informal speech. The tests were also constructed to contain comparable questions. In addition, the two tests were given to a group of students from another section of the same ESL course as the students who took the computer lesson. The pre-test was then given two days before the post-test, with no intervening instruction. A paired t-test indicated no significant difference between the scores on the pre- and post-tests, and thus provided further evidence that the tests are roughly equivalent in difficulty.

The scores on the two tests for the twenty-seven students who completed the final version of the lesson showed a gain of approximately 5.22 points (out of a possible 16) between pre-and-post-tests. A one-tailed, correlated t-test comparing these means indicates that the
gain is significant at \( p < .001 \). (See Table 1.)

<table>
<thead>
<tr>
<th>Variable</th>
<th>No. of Cases</th>
<th>Mean</th>
<th>Standard Deviation</th>
<th>Standard Error</th>
<th>(Difference) Mean</th>
<th>Standard Deviation</th>
<th>Standard Error</th>
<th>T Value</th>
<th>Degrees of Freedom</th>
<th>2-Tail Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Post</td>
<td>27</td>
<td>13.1852</td>
<td>2.497</td>
<td>0.480</td>
<td>5.2222</td>
<td>2.900</td>
<td>0.558</td>
<td>9.36</td>
<td>26</td>
<td>0.000</td>
</tr>
<tr>
<td>Pre</td>
<td></td>
<td>7.9630</td>
<td>2.738</td>
<td>0.527</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Thus, the lesson appears to be valuable as a learning device.

Portfolio Evaluation

A portfolio evaluation, as described by Patricia T. Scheyer and Robert E. Stake (1978), is a long-term, continuing effort to improve computer lessons. Essentially, this method of evaluation is a "self-study"; i.e., participants in a portfolio evaluation self-study "start with at least a general idea or group of questions and look for any evidence which would define or explain how to modify the program" (Schreck, p. 83). A file of "good" and "bad" points of a program is created, continually revised and reviewed, and used as a guideline to direct that and any future programs' development.

There are several difficulties with this method, however. Schreck (p. 91), in implementing a portfolio observation in the evaluation of the Plato computer system at the University of Illinois at Champaign-Urbana, noted that "there seems to be disagreement among responding teachers on many of the points raised." As a result, few conclusions could be reached about what would be effective for the Plato CAI in the future. In addition, such a method of evaluation assumes the availability of teachers. It also assumes their closeness to the computer...
system hardware and software, and their awareness of what is desirable
and practicable in a computer lesson.

The Portfolio method did not seem as valuable in this situation as
the first four methods. Therefore, I decided not to use it for the
purpose of evaluating this pilot lesson. Perhaps as the quantity of
CAI lessons for ESL increases, portfolio evaluation will be a more
valuable tool.
CONCLUSIONS

After completing this project, I have no doubts that carefully constructed CAI is a valuable auxiliary tool in teaching. In particular, students seem to enjoy CAI and to make gains as a result of such a lesson. It is less clear, however, what conclusions can be drawn as to the role of mnemonics in this lesson. Mnemonics may have helped, but it is hard to define how, and to what degree the lesson's mnemonics were used. Some students did indicate that they preferred one learning strategy over another, while some used several strategies, seemingly without a preference. Thus, I think it is valuable to include a variety of learning aids, even if only a small number of students employ them.

Therefore, in future lessons of this type, I would recommend using short definitions, sentence contexts, colors (but in a more carefully controlled manner) and symbols. I would not recommend pictures because of the amount of time it took to produce them and the somewhat negative feedback students made concerning them. I would also recommend more extensive evaluation, with larger numbers of students and a longer-term examination of the efficacy of CAI in combination with mnemonics.

It also would be interesting to test the relative value of the mnemonic devices, in an effort to understand how and why each of the particular mnemonics was used and how each helped the students. In particular, it would be interesting to isolate each of the mnemonic strategies used in this lesson in order to determine which is the most effective. A related experiment to this could be done with a view to
answering the questions of "why" and "how" students apply their choice of mnemonics to learning a second language. Further study could also be done to determine which combination of the mnemonic strategies using this lesson produces the most improvement. Other mnemonic systems, such as Paivio's adaptation of the "Hook" method, could be used experimentally, in conjunction with CAI, to determine in what areas and to what extent mnemonics and CAI can effectively aid second language learning. Such experiments could also also provide information about the effectiveness of mnemonics in CAI compared to more traditional teaching.
BIBLIOGRAPHY


Atkinson, Richard C. and Fletcher, John D. "Teaching Children to Read with a Computer." Reading Teacher, 25, No. 4 (January, 1972), 319-327.


Levin, Joel R.; Pressley, Michael; McCormick, Christine B.; Miller, Gloria E.; Shriberg, Linda K. "Assessing the Classroom Potential of the Keyword Method." Journal of Educational Psychology, 71, No. 5 (1979), 583-594.


Martin, Anne V.; McChesney, B.; Whalley, E.; Derlin E. Guide to


Miller, G. A. "The Magical Number Seven, Plus or Minus Two." Psychological Review (1956) 63, 81-87.


Pressley, Michael; Levin, Joel R.; Hall, James; Miller, Gloria; Berry, Jill K. Journal of Experimental Psychology: Human Learning and Memory, 6, No. 2 (1980), 163-173.


APPENDIX 1

SCREEN DISPLAYS

Note: This Appendix contains a complete, typical display of what a student would see as he used the lesson. This section shows how each screen display is developed.
Two-word verbs

by Kevin Quinn
A lesson for intermediate to advanced English-as-a-second-language learners

One Moment please...loading keys
Students of English as a Second Language often find that two-word English verbs are difficult. Two-word verbs are often idioms, which must be **MEMORIZED**.

Here is an example:

"to turn out" is a two-word verb.
It does **NOT** usually mean to turn (change directions) + out.

In the sentence below, "to turn out" means "to attend."

No one turned out for the meeting. (No one attended the meeting.)

In this lesson, we will look at one family of two-word verbs --

the family of verbs using **TAKE**.
Ten common members of the "TAKE" family are:

To take after
To take back
To take down
To take for
To take in
To take off
To take on
To take out
To take over
To take up

In this lesson, you will learn the meanings of several of these verbs by using definitions, pictures, and color.

(press RETURN to continue)
Before going further, it would be good to understand several key grammatical points about two-word verbs.

(press RETURN to continue)

A two-word verb is made up of two parts: a verb and a particle. Particles include the words that also act as prepositions: "off," "on," "in," "after," etc.
A two-word verb is made up of two parts: a verb, and a particle. Particles include the words that also act as prepositions: "off," "on," "in," "after," etc.

Here are some examples of two-word verbs:

<table>
<thead>
<tr>
<th>VERB</th>
<th>PARTICLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>take</td>
<td>off</td>
</tr>
<tr>
<td>call</td>
<td>up</td>
</tr>
<tr>
<td>get</td>
<td>in</td>
</tr>
</tbody>
</table>

Two-word verbs also often come in groupings, or families. That is, one member of a two-word verb family looks just the same as another except for their particles. It is important to remember each family has the same main verb.

Example: "Take on" and "Take off"

One problem with two-word verbs is that they may be separable or inseparable.

An inseparable verb must keep verb and particle next to each other. An example is "take after."

He took after his father. She took after her mother. Neither took after their grandparents.
One problem with two-word verbs is that they may be separable or inseparable.

An inseparable verb must keep verb and particle next to each other. An example is "take after."

He took after his father. She took after her mother. Neither took after their grandparents.

A separable two-word verb, on the other hand, may have words between its verb and its particle. An example is "take back."

He took the television back to the store.

There are two other points to remember:

**Point One:**

The two-word verbs in this lesson (with the exception of "take after,") are all separable verbs. When you use these verbs you can either

keep the verb and the particle together

OR

separate them.

When you use inseparable verbs you must

keep the verb and the particle together.

**Point Two:**

Two-word verbs may be used in past, present, or future tense.

(press RETURN to continue)
In this section, you will see:

1. A symbol for each particle used in this lesson.

2. Some suggestions on how to use the colors, pictures and definitions coming later in this section.

3. Some pictures to help you remember the meanings of the verbs.

But first, the symbols...

(press RETURN to continue)
Please note that the green circle in the "back" symbol is not on the front, but on the back of the symbol.

These are the symbols for the particles used in this lesson.
Seven of the particles have one color each.
Three of the particles have two colors:
"down" purple and white
"for" red and white
"over" green and white

And now, here's how to work through the memorization section...

There are six ways to 
MEMORIZE each of these verbs:
1. Look at the definition of the verb.

There are six ways to 
MEMORIZE each of these verbs:
1. Look at the definition of the verb.
2. Look at the color of the words. In your mind, try to connect the color with the verb and its meaning.
There are six ways to memorize each of these verbs:

1. Look at the definition of the verb.

2. Look at the color of the words. In your mind, try to connect the color with the verb and its meaning.

3. Look at the symbol for the particle. Try to connect the symbol with the verb and its meaning.

4. Read the paragraph to see how the verb is used.

5. Look at the picture which shows the action of the verb. Try to connect the picture with the verb and its meaning.

(press RETURN to continue)
There are six ways to MEMORIZE each of these verbs:

1. Look at the definition of the verb.
2. Look at the color of the words. In your mind, try to connect the color with the verb and its meaning.
3. Look at the symbol for the particle. Try to connect the symbol with the verb and its meaning.
4. Read the paragraph to see how the verb is used.
5. Look at the picture which shows the action of the verb. Try to connect the picture with the verb and its meaning.
6. Imagine a picture which shows the action of the verb. Try to connect it with the verb and its meaning.

(press RETURN to continue)

Now the lesson itself will begin....

(There are 10 two-word verbs to study in the lesson. Some of them have more than one meaning.)

(press RETURN to continue)

**Take for**

I saw Professor Thomas carrying a ladder and tool box into his house. I took him for a repairman.

(Take 4)

(The number four (4) might help you remember the particle "for.")

Take for = to mistake for (someone or something)

(press RETURN to continue)
He took the handkerchief out of his pocket to wipe the sweat from his forehead.

When John passed the insurance sign, he remembered to take out some insurance.

Dudley's Insurance
"We insure with a smile!"
The student took down the whole lecture so that he wouldn’t forget the important points.

She took down the blue curtains in the kitchen. She wanted to wash them that day.
Take up

The huge desk took up most of the space in the bedroom. It was twice as big as a normal desk.

(take up)

Take up = to fill (space)

(press RETURN to continue)

Take up

It was a difficult topic to take up in a meeting. Few people understood it.

(Take up)

Take up = to discuss

This morning we are going to take up the topic of computer science in education.

(press RETURN to continue)
The green mountains were so amazing that it was hard to take in their beauty.

"Take in" can also refer to unpleasant news.
**Take in**

The green mountains were so amazing that it was hard to take in their beauty.

(take \[\square\] )

take in = to appreciate or understand

"Take in" can also refer to unpleasant news.

It was hard to take in the news of President Kennedy's death.

(press RETURN to continue)

---

**Take off**

Someone had taken the lid off the pot of hot breakfast cereal. By the time I got up, the cereal was cold.

(take \[\square\] )

take off = to remove (from) (with objects)

(press RETURN to continue)
He took off his coat and hung it on the coat rack. It was too hot to wear it.

To take off = to remove (used with clothes)

John takes after his brother Mark. They have the same majors, and they look the same.

Take after = to resemble
**Take back.**

The man put the parts of the broken toy in a box. He took it back to the complaints desk.

**(take box)**

(green circle is on box back)

Take back = to return

(press RETURN to continue)

---

**Take on**

Mr. Friedrich took on 30 more workers at the factory last week.

**(take job)**

take on = to hire, to employ

(job application form)

(press RETURN to continue)
Take over
Judge Thompson took over the court when Judge Anderson retired last year.

(take over = to assume responsibility (for))

Take off
Bill's dad was in the jet plane that just now took off from the airport.

(take off = to depart (with flying))
1. Introduction
2. Two-word verbs — hints and helps
3. A family of two-word verbs
4. Exercises on two-word verbs
5. Quit all work

Press ± ± to move to your choice.
Press RETURN to choose the green item.
The red * means the item is completed.
Now comes the practice....

(press RETURN to continue)

In each example:

1.) Decide the meaning of the sentence.
2.) Decide which particle shows that meaning.
   Use the color of the sentence to help you remember.
3.) Type in the missing particle.

If you make a mistake, the computer will give you the symbol and definition for the particle. The incorrect answer will be erased. You will get two tries. Instead of typing in a particle, you can also press the PF4 key to quit the lesson or the PF2 key to return to the menu.

(press RETURN to continue)

In each example:

1.) Decide the meaning of the sentence.
2.) Decide which particle shows that meaning.
   Use the color of the sentence to help you remember.
3.) Type in the missing particle.

If you make a mistake, the computer will give you the symbol and definition for the particle. The incorrect answer will be erased. You will get two tries. Instead of typing in a particle, you can also press the PF4 key to quit the lesson or the PF2 key to return to the menu.

He took the suitcases OUT to the car. Then he sat down to rest.

Incorrect. Try again.

Take + this particle means
"to get (a loan or insurance)" or
"to move or remove (from inside)."
In each example:

1.) Decide the meaning of the sentence.
2.) Decide which particle shows that meaning.
   Use the color of the sentence to help you remember.
3.) Type in the missing particle.

If you make a mistake, the computer will give you the symbol and definition for the particle. The incorrect answer will be erased. You will get two tries. Instead of typing in a particle, you can also press the PF4 key to quit the lesson or the PF2 key to return to the menu.

Because it was dark, George couldn't see very well.
He took the shirt __ for Jan.

Right. (press RETURN to continue)

In each example:

1.) Decide the meaning of the sentence.
2.) Decide which particle shows that meaning.
   Use the color of the sentence to help you remember.
3.) Type in the missing particle.

If you make a mistake, the computer will give you the symbol and definition for the particle. The incorrect answer will be erased. You will get two tries. Instead of typing in a particle, you can also press the PF4 key to quit the lesson or the PF2 key to return to the menu.

"Mr. Wallace is out of town for a vacation. Mr. Smith has taken __ for him until he comes back. Would you like to talk to Mr. Smith?"

Right. (press RETURN to continue)

Take this particle means "to assume responsibility (for)."

In each example:

1.) Decide the meaning of the sentence.
2.) Decide which particle shows that meaning.
   Use the color of the sentence to help you remember.
3.) Type in the missing particle.

If you make a mistake, the computer will give you the symbol and definition for the particle. The incorrect answer will be erased. You will get two tries. Instead of typing in a particle, you can also press the PF4 key to quit the lesson or the PF2 key to return to the menu.

He took his sweat pants __ and stepped to the starting line of the race. His muscles tensed as he waited for the starter's gun.

Right. (press RETURN to continue)
In each example:

1. Decide the meaning of the sentence.
2. Decide which particle shows that meaning.
   Use the color of the sentence to help you remember.
3. Type in the missing particle.

If you make a mistake, the computer will give you the symbol and definition for the particle. The incorrect answer will be erased. You will get two tries. Instead of typing in a particle, you can also press the PF4 key to quit the lesson or the PF2 key to return to the menu.

His secretary took >_____ everything that he said. Then she used her notes to prepare a report for the meeting.

↑ Take + this particle means "to write" or "to remove (from the wall)."

In each example:

1. Decide the meaning of the sentence.
2. Decide which particle shows that meaning.
   Use the color of the sentence to help you remember.
3. Type in the missing particle.

If you make a mistake, the computer will give you the symbol and definition for the particle. The incorrect answer will be erased. You will get two tries. Instead of typing in a particle, you can also press the PF4 key to quit the lesson or the PF2 key to return to the menu.

We didn't have time to discuss this issue completely last week. We'll have to take it >up_____ at the next meeting.

A CORRECT answer is given above. Press RETURN to continue.

↑ Take + this particle means "to discuss" or "to fill (space)."
In each example:

1.) Decide the meaning of the sentence.
2.) Decide which particle shows that meaning.
    Use the color of the sentence to help you remember.
3.) Type in the missing particle.

If you make a mistake, the computer will give you the symbol and definition for the particle. The incorrect answer will be erased. You will get two tries. Instead of typing in a particle, you can also press the PF4 key to quit the lesson or the PF2 key to return to the menu.

Sam: "Which of your parents do you think you take ___?"
Ralph: "What do you think?"
Sam: "I think you resemble your father."

Incorrect. Try again.

Take + this particle means

→ "to resemble."

The news of the murder was so shocking I couldn't take it ___ all at once. I had to read the story several times.

Right. (press RETURN to continue)

Take + this particle means "to appreciate" or "to understand." (It can refer to both pleasant and unpleasant things.)
In each example:

1.) Decide the meaning of the sentence.
2.) Decide which particle shows that meaning.
    Use the color of the sentence to help you remember.
3.) Type in the missing particle.

If you make a mistake, the computer will give you the symbol and definition for the particle. The incorrect answer will be erased. You will get two tries. Instead of typing in a particle, you can also press the PF4 key to quit the lesson or the PF2 key to return to the menu.

You can take the sweater back to Target if it doesn't fit. They will return your money.

Right. (press RETURN to continue)

Take + this particle means "to return."

In each example:

1.) Decide the meaning of the sentence.
2.) Decide which particle shows that meaning.
    Use the color of the sentence to help you remember.
3.) Type in the missing particle.

If you make a mistake, the computer will give you the symbol and definition for the particle. The incorrect answer will be erased. You will get two tries. Instead of typing in a particle, you can also press the PF4 key to quit the lesson or the PF2 key to return to the menu.

John: "Mr. Smith hired ten new employees yesterday."
Jim: "Did he? Why did he have to take >on_____ so many?"

Right. (press RETURN to continue)
That was the last question in this section...

to see how you did...

press RETURN.

You have 3 so far out of 10 possible. If you scored less than eight in this section, it might be good to review the earlier lesson. If you want to return to the menu, at the next >______,
press F2. To continue with the practice, press EXIT.
In each example:

1.) Decide the meaning of the sentence.
2.) Decide which particle shows that meaning.
3.) Type in the missing particle.

If you make a mistake, the computer will give you the symbol and definition for the particle. The incorrect answer will be erased. You will get two tries.

Instead of typing in a particle, you can also press the PF4 key to quit the lesson or the PF2 key to return to the menu.

Hundreds of airplanes take >off___ from Chicago’s O’Hare airport everyday. It’s the busiest airport in the midwest.

Right. (press RETURN to continue)

In each example:

1.) Decide the meaning of the sentence.
2.) Decide which particle shows that meaning.
3.) Type in the missing particle.

If you make a mistake, the computer will give you the symbol and definition for the particle. The incorrect answer will be erased. You will get two tries.

Instead of typing in a particle, you can also press the PF4 key to quit the lesson or the PF2 key to return to the menu.

All department stores take >on___ temporary workers for the Christmas season.

Right. (press RETURN to continue)

* Take + this particle means
  
  "to hire" or
  "to employ."
In each example:

1.) Decide the meaning of the sentence.
2.) Decide which particle shows that meaning.
   Use the color of the sentence to help you remember.
3.) Type in the missing particle.

If you make a mistake, the computer will give you the symbol and definition for the particle. The incorrect answer will be erased. You will get two tries. Instead of typing in a particle, you can also press the PF4 key to quit the lesson or the PF2 key to return to the menu.

I’ve never taken anything to Sam’s Appliance Shack. Everything I’ve bought from him has worked well.

Right. (press RETURN to continue)

In each example:

1.) Decide the meaning of the sentence.
2.) Decide which particle shows that meaning.
   Use the color of the sentence to help you remember.
3.) Type in the missing particle.

If you make a mistake, the computer will give you the symbol and definition for the particle. The incorrect answer will be erased. You will get two tries. Instead of typing in a particle, you can also press the PF4 key to quit the lesson or the PF2 key to return to the menu.

When he reached the mountaintop, he looked at the valley below. The view was so scenic he had to take it in slowly.

Right. (press RETURN to continue)

Take *this* particle means "to appreciate" or "to understand." (It can refer to both pleasant and unpleasant things.)
In each example:

1.) Decide the meaning of the sentence.
2.) Decide which particle shows that meaning.
   Use the color of the sentence to help you remember.
3.) Type in the missing particle.

If you make a mistake, the computer will give you the symbol and definition for the particle. The incorrect answer will be erased. You will get two tries. Instead of typing in a particle, you can also press the PF4 key to quit the lesson or the PF2 key to return to the menu.

He took the lid >off_ the jar of peanut butter. Then he made himself a sandwich.

Right. (press RETURN to continue)

Take + this particle means "to remove (from)" or "to depart" (with flying).

In each example:

1.) Decide the meaning of the sentence.
2.) Decide which particle shows that meaning.
   Use the color of the sentence to help you remember.
3.) Type in the missing particle.

If you make a mistake, the computer will give you the symbol and definition for the particle. The incorrect answer will be erased. You will get two tries. Instead of typing in a particle, you can also press the PF4 key to quit the lesson or the PF2 key to return to the menu.

At the end of the school year, students in the dormitories must take >down_ any posters or pictures hanging on the walls of their rooms.

Right. (press RETURN to continue)
In each example:

1. Decide the meaning of the sentence.
2. Decide which particle shows that meaning.
   Use the color of the sentence to help you remember.
3. Type in the missing particle.

If you make a mistake, the computer will give you the symbol and definition for the particle. The incorrect answer will be erased. You will get two tries. Instead of typing in a particle, you can also press the PF4 key to quit the lesson or the PF2 key to return to the menu.

Mary takes after her mother. They both like to ride horses and garden.

Right. (press RETURN to continue)

Take + this particle means "to resemble."

In each example:

1. Decide the meaning of the sentence.
2. Decide which particle shows that meaning.
   Use the color of the sentence to help you remember.
3. Type in the missing particle.

If you make a mistake, the computer will give you the symbol and definition for the particle. The incorrect answer will be erased. You will get two tries. Instead of typing in a particle, you can also press the PF4 key to quit the lesson or the PF2 key to return to the menu.

He took an automobile insurance policy to help him pay for engine repairs and body work in case his car was damaged.

Right. (press RETURN to continue)
In each example:

1. Decide the meaning of the sentence.
2. Decide which particle shows that meaning.
   Use the color of the sentence to help you remember.
3. Type in the missing particle.

If you make a mistake, the computer will give you the symbol and definition for the particle. The incorrect answer will be erased. You will get two tries. Instead of typing in a particle, you can also press the PF4 key to quit the lesson or the PF2 key to return to the menu.

Sony Television Company guarantees that you can take your Sony television for most routine repairs within one year after it was purchased.

right. Press RETURN to continue.

In each example:

1. Decide the meaning of the sentence.
2. Decide which particle shows that meaning.
   Use the color of the sentence to help you remember.
3. Type in the missing particle.

If you make a mistake, the computer will give you the symbol and definition for the particle. The incorrect answer will be erased. You will get two tries. Instead of typing in a particle, you can also press the PF4 key to quit the lesson or the PF2 key to return to the menu.

Some jet airliners are huge. The largest one takes more space than two football fields.

A CORRECT answer is given above. Press RETURN to continue.

up. Take + this particle means "to discuss" or "to fill (space)."
In each example:

1.) Decide the meaning of the sentence.
2.) Decide which particle shows that meaning.
   Use the color of the sentence to help you remember.
3.) Type in the missing particle.

If you make a mistake, the computer will give you the
symbol and definition for the particle. The
incorrect answer will be erased. You will get two tries.
Instead of typing in a particle, you can also press the PF4
key to quit the lesson or the PF2 key to return to the menu.

It's easy to take Connie >for___ Amy. They look a lot alike
because they are twins.

Right. (press RETURN to continue)

4. Take + this particle means "to mistake for
   (someone or something)."

In each example:

1.) Decide the meaning of the sentence.
2.) Decide which particle shows that meaning.
   Use the color of the sentence to help you remember.
3.) Type in the missing particle.

If you make a mistake, the computer will give you the
symbol and definition for the particle. The
incorrect answer will be erased. You will get two tries.
Instead of typing in a particle, you can also press the PF4
key to quit the lesson or the PF2 key to return to the menu.

A pocket dictionary doesn't take >up____ much space,
and it doesn't weigh much. For these reasons, many foreign
students bring pocket dictionaries to class.

Right. (press RETURN to continue)
In each example:

1.) Decide the meaning of the sentence.
2.) Decide which particle shows that meaning.
   Use the color of the sentence to help you remember.
3.) Type in the missing particle.

If you make a mistake, the computer will give you the symbol and definition for the particle. The incorrect answer will be erased. You will get two tries. Instead of typing in a particle, you can also press the PF4 key to quit the lesson or the PF2 key to return to the menu.

---

The sunrise was so beautiful I had a difficult time taking
>in____ the sight.
(*** There are 5 left ***)

Right. (press RETURN to continue)

---

Why don’t you take that paragraph >out____ of the composition?
It isn’t related to the main idea.

Right. (press RETURN to continue)
In each example:

1.) Decide the meaning of the sentence.
2.) Decide which particle shows that meaning.
   Use the color of the sentence to help you remember.
3.) Type in the missing particle.

If you make a mistake, the computer will give you the symbol and definition for the particle. The incorrect answer will be erased. You will get two tries. Instead of typing in a particle, you can also press the PF4 key to quit the lesson or the PF2 key to return to the menu.

Many students take >out loans while they go to college. Otherwise, they wouldn't have money to pay for school.

Right. (press RETURN to continue)

Take + this particle means
"to get (a loan or insurance)" or
"to move or remove (from inside)."

In each example:

1.) Decide the meaning of the sentence.
2.) Decide which particle shows that meaning.
   Use the color of the sentence to help you remember.
3.) Type in the missing particle.

If you make a mistake, the computer will give you the symbol and definition for the particle. The incorrect answer will be erased. You will get two tries. Instead of typing in a particle, you can also press the PF4 key to quit the lesson or the PF2 key to return to the menu.

William Harbeck has all the qualifications we listed for this job. Let's take him >on_____.

Right. (press RETURN to continue)
In each example:

1.) Decide the meaning of the sentence.
2.) Decide which particle shows that meaning.
   - Use the color of the sentence to help you remember.
3.) Type in the missing particle.

If you make a mistake, the computer will give you the symbol and definition for the particle. The incorrect answer will be erased. You will get two tries. Instead of typing in a particle, you can also press the PF4 key to quit the lesson or the PF2 key to return to the menu.

I worked on the project for two weeks. Then Bill took over. He did a better job than I did.

Right. (press RETURN to continue)

There are many "help wanted" advertisements for companies who want to take on computer scientists. But there aren't many companies who want to hire a historian.

Right. (press RETURN to continue)
In each example:

1.) Decide the meaning of the sentence.
2.) Decide which particle shows that meaning.
   Use the color of the sentence to help you remember.
3.) Type in the missing particle.

If you make a mistake, the computer will give you the
symbol and definition for the particle. The
incorrect answer will be erased. You will get two tries.
Instead of typing in a particle, you can also press the PF4
key to quit the lesson or the PF2 key to return to the menu.

Flight 26 took >off_ twenty minutes late today. The stormy
weather in the morning kept all planes from flying.

Right. (press RETURN to continue)

In each example:

1.) Decide the meaning of the sentence.
2.) Decide which particle shows that meaning.
   Use the color of the sentence to help you remember.
3.) Type in the missing particle.

If you make a mistake, the computer will give you the
symbol and definition for the particle. The
incorrect answer will be erased. You will get two tries.
Instead of typing in a particle, you can also press the PF4
key to quit the lesson or the PF2 key to return to the menu.

The army took >over_ the government, the industries, and the radio
station of the tiny country.

Right. (press RETURN to continue)
In each example:

1.) Decide the meaning of the sentence.
2.) Decide which particle shows that meaning.
   Use the color of the sentence to help you remember.
3.) Type in the missing particle.

If you make a mistake, the computer will give you the symbol and definition for the particle. The incorrect answer will be erased. You will get two tries. Instead of typing in a particle, you can also press the PF4 key to quit the lesson or the PF2 key to return to the menu.

When Sam heard his father was dead, he was so shocked he could hardly take in the news. It took him days to recover.

Right. (press RETURN to continue)

Take + this particle means "to appreciate" or "to understand." (It can refer to both pleasant and unpleasant things.)

In each example:

1.) Decide the meaning of the sentence.
2.) Decide which particle shows that meaning.
   Use the color of the sentence to help you remember.
3.) Type in the missing particle.

If you make a mistake, the computer will give you the symbol and definition for the particle. The incorrect answer will be erased. You will get two tries. Instead of typing in a particle, you can also press the PF4 key to quit the lesson or the PF2 key to return to the menu.

The employee wanted an increase in his salary, but he was afraid to take up this issue with his boss.

Right. (press RETURN to continue)
That was the last question...

to see how you did...

press RETURN.

You have 25 total in the practice section. There were 32 possible. If you scored less than 24 of the total, it might be valuable to review the material. To return to the menu, press (EXIT).
APPENDIX 2

PRE-AND POST-TESTS
Directions: Choose the best word to fill in the blank. Use this list.

after, in, out, back, off, over, down, on, up

1. When Kathy heard that her father had had a heart attack, the news was hard for her to take _________.
2. Before Theodore washed the windows, he took _________ the curtains so they wouldn't get wet.
3. He took the cap _________ the pen and began to write the essay for the English 100C class.
4. After the ceremony, the new king took _________ the throne which the old king had held for 20 years.
5. The farmer took _________ three new laborers to work in his soybean fields. Their first job was to pull weeds.
6. We'll take two-word verbs _________ next week. We'll discuss 15-20 of the most common ones.
7. The instructor is very young-looking. It's not hard to take _________ a student. Many of his students look older than he does.
8. The Seiko Watch Company guarantees it will take _________ any watch which loses more than thirty seconds a month.
9. Jim takes _________ his mother. He has the same friendliness and cheerfulness that she does.
10. When he got married, he took _________ a life insurance policy. He wanted to provide some income for his wife in case he died.
11. The new snow on the mountains was so bright and spectacular that it was hard to take _________.
12. "You might want to take the important points _________," the English teacher said to her students. "You will need this information for the test."
13. He worked for about an hour. Then the sun came out and it began to get hot, so he took _________ his shirt.
14. The new city office building takes _________ an entire city block. It is the largest building in the city.
15. While he was trying to read, his dog kept barking till finally he gave up and took it _________ for a walk.
16. Bill watched his father's airplane taxi down the runway and take _________.
Pre-test Score Inventory

<table>
<thead>
<tr>
<th>Item Number</th>
<th>Particle Used (% of total, N = 32*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>af** bk dn fr in of on ot ov up m</td>
</tr>
<tr>
<td>2</td>
<td>0.0 6.3 0.0 6.3 25.0 3.1 6.3 0.0 25.0 28.1 0</td>
</tr>
<tr>
<td>3</td>
<td>0.0 0.0 40.6 0.0 0.0 46.9 0.0 6.3 3.1 0.0 0.0</td>
</tr>
<tr>
<td>4</td>
<td>3.1 0.0 0.0 3.1 0.0 65.6 9.4 15.6 3.1 0.0 0.0</td>
</tr>
<tr>
<td>5</td>
<td>0.0 3.1 3.1 3.1 0.0 3.1 0.0 3.1 81.3 0.0 0.0</td>
</tr>
<tr>
<td>6</td>
<td>3.1 18.8 18.8 6.3 37.5 3.1 0.0 15.6 0.0 15.6 0</td>
</tr>
<tr>
<td>7</td>
<td>0.0 0.0 6.3 43.8 18.8 0.0 15.6 3.1 12.5 6.3 3</td>
</tr>
<tr>
<td>8</td>
<td>3.1 12.5 0.0 59.4 12.5 0.0 0.0 3.1 0.0 3.1 6</td>
</tr>
<tr>
<td>9</td>
<td>6.3 68.8 3.1 3.1 3.1 0.0 0.0 0.0 12.5 3.1 0</td>
</tr>
<tr>
<td>10</td>
<td>46.9 3.1 0.0 3.1 15.6 9.4 12.5 9.4 12.5 9.4 0</td>
</tr>
<tr>
<td>11</td>
<td>3.1 3.1 0.0 9.4 12.5 0.0 6.3 3.1 9.4 53.1 0</td>
</tr>
<tr>
<td>12</td>
<td>3.1 3.1 3.1 0.0 9.4 34.4 5.0 6.3 6.3 18.8 0</td>
</tr>
<tr>
<td>13</td>
<td>0.0 6.3 46.9 0.0 3.1 0.0 0.0 15.6 6.3 9.4 0</td>
</tr>
<tr>
<td>14</td>
<td>0.0 0.0 0.0 0.0 0.0 94.1 3.1 3.1 0.0 0.0 0.0</td>
</tr>
<tr>
<td>15</td>
<td>0.0 0.0 6.3 0.0 0.0 0.0 3.1 87.5 0.0 3.1 0</td>
</tr>
<tr>
<td>16</td>
<td>3.1 0.0 0.0 0.0 3.1 64.4 6.3 0.0 0.0 0.0 0</td>
</tr>
</tbody>
</table>

*Please note that, although there are 32 pre-tests calculated in this analysis, only twenty-seven students completed both pre- and post-tests and did the lesson in the allotted time. Also note that percentages may not add up to be 100% due to rounding error.

**af = after  bk = back  dn = down  fr = for  in = in of = off  on = on  ot = out  ov = over  up = up  m = miscellaneous
Directions: Choose the best word to fill in the blank. Use this list.

after  in  out
back  off  over
down  on  up

1. The sunset over the ocean was beautiful. I stood there looking at it for fifteen minutes to take it all ________.

2. Zach took the old painting ________ so that he could hang a new one in its place.

3. The jet airplane quickly taxied down the runway. Then it took ________ with a roar.

4. When the next secretary general takes ________ leadership of the United Nations, he will face a very dangerous world situation.

5. The construction company took ________ fourteen workers to help build the apartment building. If they have good weather the work should be finished before winter.

6. "We'll take ________ a new topic next week," the teacher said. "But first, let's finish studying two-word verbs."

7. It's easy to take Larry ________ Tom. Both have blond hair, blue eyes and dark beards.

8. If your radio doesn't work, take it ________ to the store you brought it from. They will repair it without cost.

9. Henry takes ________ his dad. They both like fishing, playing basketball, and reading.

10. Mandy washed the dishes for her mother. Then she took the trash ________.

11. I usually take ________ everything a professor writes on the blackboard. I've found it helps me study better on tests.

12. He took the cap ________ the tube of toothpaste. Then he squeezed the tube in the middle.

13. The Pentagon takes ________ almost a square mile of space. It is the largest building in Washington, D.C.

14. He took ________ an accident insurance policy when he began to work for the construction company. He was planning ahead in case he was injured.

15. The newspaper reporter took ________ his shoes and put his feet up on his desk. It had been a hard day.

16. When Alf received an "F" in his engineering class, the news was hard for him to take ________. So he decided to go to the gym to relax.
### Post-test Score Inventory

<table>
<thead>
<tr>
<th>Item Number</th>
<th>Particle Used (% of total, N = 33*)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>af** bk dn fr in of on ot ov up m</td>
</tr>
<tr>
<td>1</td>
<td>0.0 0.0 3.0 0.0 67.6 3.0 3.0 3.0 21.2 0.0 0</td>
</tr>
<tr>
<td>2</td>
<td>0.0 3.0 81.8 0.0 0.0 12.1 0.0 3.0 0.0 0.0 0.0 0</td>
</tr>
<tr>
<td>3</td>
<td>0.0 0.0 0.0 0.0 0.0 97.0 0.0 0.0 0.0 0.0 3.0 0</td>
</tr>
<tr>
<td>4</td>
<td>0.0 0.0 0.0 3.0 0.0 0.0 0.0 3.0 0.0 93.9 0.0 0</td>
</tr>
<tr>
<td>5</td>
<td>0.0 0.0 6.1 0.0 15.2 0.0 67.6 6.1 0.0 9.1 0</td>
</tr>
<tr>
<td>6</td>
<td>0.0 0.0 0.0 3.0 6.1 0.0 15.2 9.1 12.1 54.5 0</td>
</tr>
<tr>
<td>7</td>
<td>12.1 0.0 0.0 81.8 0.0 0.0 0.0 0.0 6.1 0.0 0</td>
</tr>
<tr>
<td>8</td>
<td>0.0 90.9 3.0 0.0 0.0 0.0 0.0 0.0 0.0 3.0 3.0 0</td>
</tr>
<tr>
<td>9</td>
<td>78.8 0.0 0.0 3.0 0.0 0.0 3.0 12.1 0.0 3.0 0</td>
</tr>
<tr>
<td>10</td>
<td>0.0 3.0 6.1 0.0 0.0 9.1 3.0 75.8 0.0 3.0 0</td>
</tr>
<tr>
<td>11</td>
<td>0.0 0.0 90.9 0.0 3.0 0.0 3.0 0.0 0.0 3.0 0</td>
</tr>
<tr>
<td>12</td>
<td>0.0 0.0 0.0 0.0 0.0 90.9 3.0 9.1 0.0 0.0 0</td>
</tr>
<tr>
<td>13</td>
<td>0.0 0.0 0.0 3.0 6.1 0.0 3.0 3.0 21.2 63.6 0</td>
</tr>
<tr>
<td>14</td>
<td>3.0 0.0 0.0 0.0 9.1 0.0 18.1 45.5 6.1 21.2 0</td>
</tr>
<tr>
<td>15</td>
<td>0.0 0.0 0.0 0.0 0.0 93.9 0.0 6.1 0.0 0.0 0</td>
</tr>
<tr>
<td>16</td>
<td>0.0 3.0 0.0 3.0 78.8 0.0 9.1 0.0 3.0 6.1 0</td>
</tr>
</tbody>
</table>

*Please note that, although there are 33 post-tests calculated in this analysis, only twenty-seven students completed both pre- and post-tests and did the lesson in the allotted time. Also note that percentages may not add up to be 100% due to rounding error.

**af = after  bk = back  dn = down  fr = for  in = in
of = off  on = on  ot = out  ov = over  up = up
m = miscellaneous
1. Rate this computer lesson on a scale from 1-10 (1 is low, 10 is high).
   Organization 1 2 3 4 5 6 7 8 9 10
   Pictures 1 2 3 4 5 6 7 8 9 10
   Explanations 1 2 3 4 5 6 7 8 9 10
   Overall 1 2 3 4 5 6 7 8 9 10

2. Would you recommend this lesson to someone else? Why?

3. What would you like to see changed in this lesson? What bothered you about this lesson?

4a. What did you like about this lesson?

4b. Would you like a lesson on another two-word verb family?

5. What other topics would you like to see covered by computer lessons?

6. Approximately how long did you spend on this lesson? How many questions did you get correct in the practice section?

7. Which memory device in this lesson was the most useful? (circle one)
   full-screen pictures
   colors
   short definitions (ex.: take off = "to remove")
   sentence contexts (ex.: The airplane took off at 4:30 this afternoon.)
   symbols for the particles (ex.: take ↑)
APPENDIX 4

COMPUTER CODE

Note: This Appendix contains the computer code which ran the lesson I developed. Much of the credit for this code should go to Jim Augspurger, a graduate of Iowa State University with a major in Computer Science.
Lesson VERBS
Define LineType :STRING
controls logic.
;
TextLine :STRING
;
Rcount :INTEGER
read from file.
;
RightAns 5 :STRING
ans for 1 question.
Template :STRING,RESTART
;
RFBcount :INTEGER
feedbacks.
;
RightFB 5 :STRING
each right answer.
;
Wcount :INTEGER
read from file.
;
WrongAns 5 :STRING
ans from 1 question.
;
WFbcount :INTEGER
read from file.
;
WrongFB 5 :STRING
each wrong answer.
;
NewValue :INTEGER
;
Spacing :INTEGER
between text on the screen.
;
MaxTries :INTEGER
student is given answer.
;
StartLine :INTEGER
postion of text.
;
Margin :INTEGER
indent from left screen boundary.
;
BaseLine :INTEGER
on screen.
;
Temp :INTEGER
number read from screen.
;
FileLine :INTEGER
file- used for error pointer.
;
Error : STRING
msg to display.
;
ScreenCoor :INTEGER
writing on screen.
;
PromptLine 5 :STRING
replotting for wrong answers.
;
QueryState :INTEGER
if ">" is inline.
;;; ColorState :INTEGER
;; Qcount :INTEGER
a student response.
;; LineFacet :STRING
textline.
;; PromptCoor : INTEGER
pending queries.
;; CatchAll : STRING
unspecified wrong answer.
;; Quest : INTEGER
query in a line.
;; MatchFound : BOOLEAN
found.
;; SkipQueSym : BOOLEAN
I :INTEGER
EraseLine : STRING
erasing a message up to 80 characters.
;; HueTable : INTEGER
and converted to integers thru this table.
;; PromptColor : INTEGER
student response color.
;; PromptLineColor : INTEGER
following the prompt symbol.
;; CurrentColor : INTEGER
written in this color.

Attempt : INTEGER
TotalAttempt : INTEGER
TotalQuestion : INTEGER
NumberWrong : INTEGER
numright : integer
numok : integer
FirstRun : BOOLEAN,RESTART
Debug1 : STRING
Debug2 : STRING
Debug3 : STRING
GiveAnsFB : STRING
Currentpic : STRING
Qmode : STRING
InstructorMode : BOOLEAN
DebugMode : BOOLEAN
FileName : STRING
RFBcolor : INTEGER
WFBcolor : INTEGER
Selector : INTEGER,RESTART
RespLen : INTEGER,RESTART
AveAnLen : INTEGER,RESTART
TolQuers : INTEGER,RESTART
RpCt5 : INTEGER,RESTART
Rplt5 : INTEGER,RESTART
practice : boolean

$ Number of queries pending
$ Unique component of
$ Position of prompt for
$ Feedback to catch any
$ Looping index for each
$ Set to true if answer is
$ String of blanks for
$ Table of HueTable read in
$ Color of prompt and
$ Color of prompt line
$ Color of text to be
$ Tells total number wrong.
$ One of three variables indicating
$ Tells if practicing or not.
diagtest:boolean
not.
practice2:boolean
not.
jumpptomabolean
DEFINE PF1,PF2,PF3,PF4:STRING
ASSIGN PF1:=CHAR(27)+"OP"
ASSIGN PF2:=CHAR(27)+"CQ"
ASSIGN PF3:=CHAR(27)+"OR"
ASSIGN PF4:=CHAR(27)+"OS"
prompt ">
do setup("KP1")
italic 0
Mode Replace

;******************************************************************************;
; Initialize and assign strings to variables
;******************************************************************************;
assign Debug1 := " => RIGHT <= "
assign Debug2 := " => WRONG <= "
assign Debug3 := " => CATCH-ALL <= "
assign InstructorMode :=false
assign DebugMode : = FALSE
assign GiveAnsFB := ".
RETURN to continue."
assign SkipQueSym:=FALSE
Assign Rcount := 0
assign RFBcount := 0
assign Wcount := 0
assign WFBcount := 0
assign Qcount := 0
assign Quest := 1
assign Spacing := 100
assign MaxTries := 3
assign Startline := 04
assign Margin := 4
assign BaseLine := 00
assign EraseLine := "
For i:= 1,5
assign RightFB i :=" right. Press RETURN to continue."
assign WrongFB i := "incorrect. (try again.)"
EndFor
assign RFBcolor := GREEN
assign WFBcolor := RED
assign Qcount := 0

;******************************************************************************;
; Assigning colors to an array for use in data files
;******************************************************************************;
assign HueTable "BLACK" := 0
assign HueTable "DARK" := 0
assign HueTable "BLUE" := 1
assign HueTable "RED" := 2
assign HueTable "MAGENTA" := 3
assign HueTable "PURPLE" := 3
assign HueTable "VIOLET" := 3
assign HueTable "GREEN" := 4
assign HueTable "CYAN" := 5
assign HueTable "YELLOW" := 6
assign HueTable "WHITE" := 7
assign PromptLineColor := Yellow
assign PromptColor := Red
assign ScreenCoor := StartLine
assign CurrentColor := 7

; Sets green highlight on menu to the first selection for the first run.

IF FirstRun<>TRUE
   assign Selector:=1
   assign FirstRun:=TRUE
   assign Template:="l234567890"
EndIF

; Initializes counter variables for query sections.

assign TotalAttempt:=0
assign TotalQuestion:=0
assign NumberWrong := 0
assign numright:=0
assign numcorrect:=0
assign numok:=0

; Start of the screen displays

Slide "intr2.pic"
do  keychar("load")
size 1
charset "standard"
fcolor cyan
at 2300
write One Moment Please...loading keys
MLOAD "M","MENUBRD2.PIC"
assign Error:="" 
ERASE

; Sets up a file of student responses

LOG FULL
Open "info.dat",3,write
put 3,"Name Account-Name Course Date Time"
put 3,Name,"+Accname+","+Course+","+Date+","+Time
put 3,""
loop Error=""
   assign jumptoman:=false
   do Menukey(selector,Template)
   LOG FULL
endloop
put 3,"Logged out at time= "+TIME+" Date = "+DATE
put 3,="=================================================================="
include XDriyer.dal
include SelectJ.dal
include SelectO.dal
include SelectC.dal
include SelectU.dal
include Menukey.dal
unit exit
erase slide "quits2.pic"
do pf4exit
delson

Unit CheckVal(command,tvalue)

;**********************************************************************
; ; Unit CheckVal displays an error message if there are too many ; ; numbers following a command read in a data file, (ex.: "$ERASE-RANGE:"
; ; Also, this unit displays an error message with a missing "$START-QUERY:"
; ; command in the data file that the main program is reading.
; ;**********************************************************************

Define command:string
tvalue:integer
If tvalue > 5
assign Error:= "TOO MANY VALUES FOLLOWING "+command+" OR MISSING
$START-QUERY:: ON LINE= "
Do RptError
EndIf

Unit ReadLine

;**********************************************************************
; ; This unit displays an error message in case the end-of-data-file ; ; symbol is missing.
; ;**********************************************************************

GET 5,TextLine
If ioreult <> 1
assign Error := "I/O ERROR: POSSIBLE MISSING END OF FILE SYMBOL (¢) on line= "
Do RptError
Else
assign FileLine := FileLine + 1
assign LineType := SUBSTR(TextLine,1,1)
EndIf

Unit ReadNum

;**********************************************************************
; ; This unit displays an error message in case a number is missing ; ; after a data file command which requires a number.
GET 5, textline
If IOresult <> 1
   assign Error := "MISSING NUMBER IN COLUMN ONE ON LINE FOLLOWING $COMMAND: LAST LINE PROCESSED = "
   Do RptError
Else
   assign FileLine := FileLine + 1
   assign LineType := "0"
   assign NewValue := (INT(Number(TextLine))) * 100
EndIf

Unit RptError

;**********************************************************************
; This is the master message printing unit. It takes all the messages
; assigned to the error variables and outputs them, along with
; information about where the error is and what kind of error it is.
; Also, this unit halts the program. (unit PF4 exit)
;**********************************************************************

Define MesgLen: INTEGER

ERASE 1800;2483
Mode Overlay
mode normal
Fcolor Green
at 1918
write LESSON TERMINATED : ERROR IN INSTRUCTOR'S FILE
BOX 2000;2383:+1
assign MesgLen := LEN(Error+CHAR(FileLine)+" ")
Italics -27
at 2141-INT(MesgLen/2)
write <s,Error>> <<s,FileLine>> TYPE= <<s,LineType>>
ITALICS 0
at 2334
write PRESS RETURN
PAUSE
Do PF4Exit

Unit Parser

;**********************************************************************
; This is the unit which parses all the "$COMMAND:"s written into
; the data files processed by this program.
;**********************************************************************

LOOP lineType = "$"
   TEST UPPER(TextLine)
   Value "$INSTRUCTOR-MODE:"
do ReadLINE
assign InstructorMode:=TRUE
if SkipQueSym=TRUE
  assign SkipQueSym:=FALSE
else
  assign SkipQueSym:=TRUE
endif
do Readline
Value "$RIGHT:"
  Do ReadLine
  LOOP LineType <> "$"
    assign Rcount := Rcount +1
    Do CheckVal("$RIGHT:",Rcount)
    assign RightAns Rcount := TextLine
  Do ReadLine
ENDLoop
Value "$OK-FEEDBACK:"
  Do ReadLine
  LOOP LineType <> "$"
    assign RFBcount := RFBcount +1
    Do CheckVal("$OK-FEEDBACK:",RFBcount)
    assign RightFB RFBcount := TextLine
  Do ReadLine
ENDLoop
Value "$WRONG:"
  Do ReadLine
  LOOP LineType <> "$"
    assign Wcount := Wcount +1
    Do CheckVal("$WRONG:",Wcount)
    assign WrongAns Wcount := TextLine
  Do ReadLine
ENDLoop
Value "$NO-FEEDBACK:"
  Do ReadLine
  LOOP LineType <> "$"
    assign WFBcount := WFBcount +1
    Do CheckVal("$NO-FEEDBACK:",WFBcount)
    assign WrongFB WFBcount := TextLine
  Do ReadLine
ENDLoop
Value "$PICFILE:"
  Do ReadLine
  Assign Currentpic:=TextLine
  Do ReadLine
Value "$QMODE:"
  Do ReadLine
  Assign Qmode:=Upper(Textline)
  Do ReadLine
Value "$LINE-SPACING:"
  Do ReadNum
    assign Spacing := NewValue
  Do ReadLine
Value "$MAXTRIES:"
  Do ReadNum
    assign MaxTries := INT(NewValue/100)
  Do ReadLine
Value "$START-LINE:"
  Do ReadNum
    assign StartLine := NewValue + Margin
  Do ReadLine
Value "$MARGIN;"
... Do ReadNum
... assign StartLine := StartLine - Margin + INT(NewValue/100)
... assign Margin := NewValue
... Do ReadLine
Value "$PRESSRET;"
... Do readline
... Do pressret
Value "$PAUSE;"
... Do ReadLine
... PAUSE
Value "$ERASE-UP;"
... Do ReadNum
... assign temp := BaseLine
... assign BaseLine := BaseLine - NewValue
... If BaseLine < 0
... ... assign BaseLine:=0
... Endif
... ERASE BaseLine;temp + 84
... assign ScreenCoor := BaseLine + Margin
... Do ReadLine
Value "$ERASE-DOWN;"
... Do ReadNum
... ERASE 00; NewValue+83
... assign BaseLine := 00
... assign ScreenCoor := StartLine
... Do ReadLine
Value "$ERASE-RANGE;"
... Do ReadNum
... assign temp := NewValue
... Do ReadNum
... ERASE Temp;NewValue +83
... assign BaseLine := Temp
... assign ScreenCoor := Temp + Margin
... Do ReadLine
Value "$ERASE;"
... ERASE
... assign BaseLine := 00
... assign ScreenCoor := StartLine
... Do ReadLine
Value "$ITALICS;"
... Do ReadNum
... Italics INT(NewValue/100)
... Do ReadLine
Value "$PCOLOR;"
... Do ReadLine
... assign CurrentColor := HueTableTextLine
... Fcolor CurrentColor
... Do ReadLine
Value "$FCOLOR;"
... Do ReadLine
... assign CurrentColor := HueTableTextLine
... Fcolor CurrentColor
... Do ReadLine
Value "$FONT;"
... Do ReadLine
... CharSet Textline
Value "$FONT-STD;"
... CharSet "STANDARD"
Value "$PROMPT-COLOR;","$PCOLOR;"
... Do ReadLine
... assign PromptColor := HueTableTextLine
... Do ReadLine
Value "$ANSWER-COLOR;","$ACOLOR;"
... Do ReadLine
... assign PromptLineColor := HueTableTextLine
83

```
Do ReadLine
  Value "\$CATCH-ALL:" 
  Do ReadLine
    assign CatchAll := TextLine
  Do ReadLine
  Value "\$START-QUERY:" 
  If Qcount < Quest
    assign Error := "QUERY NOT FOUND OR MISSING SYMBOL (> ) IN TEXT PRIOR TO LINE="
  Else
     Do RptError
     Else
      Do ProcessQ 
        if jumptoman
          assign quest:=quest+1
          return 
        endif
      assign Quest := Quest + 1
      If Quest > Qcount 
        assign Quest := 1
      assign Qcount := 0
      Endif
      assign Rcount := 0
      assign Wcount := 0
      assign RFBcount := 0
      assign WFBcount := 0
      EndIf
     Else
      assign Error := "INVALID \$COMMAND: " + TextLine + " ERROR ON LINE="
     Do RptError 
     END Test
END Loop
RETURN

Unit MainLoop

;************************************************************************
; This unit reads the data file to be displayed until the end-of-file 
; marker is read. If a "$COMMAND" is read, Unit MainLoop calls the 
; parser which then routes and directs the processing of the data 
; file.
;************************************************************************
ERASE
assign ScreenCoor:=StartLine
assign BaseLine:=00
do Readline
  LOOP LineType <> "¢"
    Test LineType
      Value "$" 
      . Do Parser 
      . if jumptoman
      . return
    endif
  Other
    assign QueryState := INSTRING(TextLine,">")
    if SkipQueSym=TRUE
```

assign QueryState:=0
endif
assign ColorState := INSTRING(TextLine," ")
If (QueryState < ColorState) AND (QueryState>0)
  Do SetQLoc
Else
  If (ColorState < QueryState) AND (ColorState>0)
    Do SetColor
  Else
    If QueryState<>0
      do SetLoc
    Else
      if ColorState<>0
        do SetColor
      Else
        at ScreenCoor
        write <s,TextLine>
      endif
    endif
  EndIf
assign BaseLine := BaseLine + Spacing
If BaseLine > 2500
;********************************************************************
If the screen display is too long to fit on the screen, an error
message is displayed.
;********************************************************************
assign Error := "SCREEN IS FULL. YOU MUST ERASE
BEFORE PRINTING LINE= 
  Do RptError
EndIf
;********************************************************************
The following two lines set the screen location to the location
desired on the screen.
;********************************************************************
assign ScreenCoor := BaseLine + Margin
Do ReadLine
EndLoop
EndTest
RETURN
Unit SetColor

;**********************************************************************
; This unit processes all the instructions, both in the main program
; and in the data files, which display colors. It also displays
; the keychar font.
;**********************************************************************

define ColorStr :STRING

LOOP ColorState <> 0
  Assign LineFacet := SUBSTR(TextLine,1,ColorState-1)
  assign ColorStr := SUBSTR(TextLine,ColorState+1,1)
  Fcolor CurrentColor
  at ScreenCoor
  write <<s,LineFacet>>
  assign ScreenCoor := Where + 1
  assign TextLine := SUBSTR(TextLine,ColorState+3,LEN(TextLine)-ColorState-2)
  assign ColorState := INSTRING(TextLine,"")
  if SkipQueSym<>TRUE
    assign QueryState := INSTRING(TextLine,">")
  endif
  Test ColorStr
  Value "D","d"
  . Assign CurrentColor := 0
  value "F","f"
  . CharSet "KEYCHAR.FNT"
  Value "B","b"
  . Assign CurrentColor := 1
  Value "P","p"
  . Assign CurrentColor := 2
  Value "M","m"
  . Assign CurrentColor := 3
  Value "G","g"
  . Assign CurrentColor := 4
  Value "H","h"
  . Assign CurrentColor := 5
  Value "Y","y"
  . Assign CurrentColor := 6
  Value "W","w"
  . Assign CurrentColor := 7
  Value "/","
  . Italics 0
  Value "/"
  . Italics -27
EndTest
Fcolor CurrentColor
If (ColorState=0) AND (QueryState=0)
  at ScreenCoor
  write <<s,TextLine>>
Else
  If ((QueryState>ColorState) AND (ColorState=0)) OR (((QueryState<ColorState) AND (QueryState>0))
    do SetQLoc
  EndIf
EndIf
EndLoop

Unit ClrMesg

;**********************************************************************
; This unit clears any error messages written on the screen.
;**********************************************************************
Fcolor Dark
at Baseline
write <<s,EraseLine>>
Unit SetMesg(MesgColor,Message)

;***********************************************************************
; This unit sets writes a message on the screen.
;***********************************************************************
Define MesgColor:INTEGER
Message :STRING
Fcolor MesgColor
at Baseline
write <<s,Message>>

Unit ProcessQ

;***********************************************************************
; This unit processes the query-judging unit.
; It directs the incorrect and correct feedback student's
; receive.
;**********************************************************************
Define Basis :STRING
Induction :STRING
OKsemaphor :BOOLEAN
Rindex :INTEGER
Windex :INTEGER
Attempt :INTEGER
NthSeek :INTEGER
ScreenLoc :INTEGER
UnderLine :STRING

if jumptoman
    return
endif
assign ScreenLoc := PromptCoor Quest

;***********************************************************************
; Initialize query-formatting variables.
;***********************************************************************
assign UnderLine := PromptLine Quest
assign QLength := LEN(UnderLine)
assign Attempt := 0
when string,pf4,exit
when string,pf2,retunit

;***********************************************************************
; Writes underline after "->" query symbol.
;***********************************************************************
$QHEADER
size 1
LOG FULL
Fcolor PromptLineColor
at ScreenLoc + 1
write <<s,UnderLine>>
Fcolor PromptColor
assign NthSeek := 0
assign Attempt:=Attempt+1

;***********************************************************************
; The only query in the entire program....
;***********************************************************************
QUERY ScreenLoc

specs exact
If NthSeek = 0
;****************************************************************
; If this is the first attempt at the question, these
; variables are assigned these values.
;*****************************************************************
End If
RightV Induction
assign NthSeek := NthSeek + 1
If (NthSeek > 1) AND (OKsemaphor = TRUE)
   . Do ClrMesg
   . Do SetMesg(GREEN,RightFB Rindex - 1)
   . if (attempt=1) and (diagtest)
      . assign numright:=numright+1
   . endif
   . if (attempt=1) and (practice)
      . assign numcorrect:=numcorrect+1
   . endif
   . if (attempt=1) and (practice2)
      . assign numok:=numok+1
   . endif
   JUDGE STOP
Else
   . If OKsemaphor = TRUE
      . assign Induction := RightAns Rindex
      . assign Rindex := Rindex + 1
   . . JUDGE AGAIN
   Else
      JUDGE NO
      . If Attempt > MaxTries
         . . Do ClrMesg
         . . Do SetMesg(PromptLineColor,GiveAnsFB)
         . . . . at ScreenLoc + 1
         . . . . write <<s,UnderLine>>
         . . . . at ScreenLoc + 1
         . . . . write <<s,RightAns 1 >>
         . . . . JUDGE STOP
      . Else
         . If (Attempt=1) and (UPPER(Qmode)="PICTURES")
            . . slide Currentpic
            . . mode replace
            . . BRANCH 0=0,$QHEADER
            . Else
               . Do ClrMesg
               . Do SetMesg(RED,WrongFB Windex - 1)
               . . . . . . . . . . BRANCH 0=0,$QHEADER
               . Endif
      . EndIf
WrongV Basis
If Rindex <= Rcount
   . assign Induction := RightAns Rindex
   . assign Rindex := Rindex + 1
   . . JUDGE AGAIN
Else
   . assign OKsemaphor := FALSE $$ All right ans tested
   . If Windex <= Wcount
assign Induction := WrongAns Windex
assign Windex := Windex + 1
JUDGE AGAIN

Else

If InstructorMode = true

If (RESPONSE="F") OR (RESPONSE="f")

If DebugMode = TRUE

Fcolor dark
at 2300
write <<s,EraseLine>>
Fcolor GREEN
at 2300
write <<s,Debug1>>
For i:=1,Rcount
Do
ClrMesg
Fcolor

Fcolor Dark
at 2300
write <<s,EraseLine>>
Fcolor GREEN
at 2300
write <<s,Debug1>>
For i:=1,Rcount
Do
ClrMesg
Fcolor

PromptLineColor

SetMesg(GREEN,RightFB i )

SetMesg(Magenta,"No Wrong Answers Specified- Press RETURN.")

PAUSE

Endif
For i:=1,Wcount
Do
ClrMesg
Fcolor

PromptLineColor

SetMesg(Magenta,WrongFB i )

EndFor
Do ClrMesg
at 2300
write <<s,EraseLine>>
Fcolor YELLOW
at 2300
write <<s,Debug3>>
Fcolor Dark
PAUSE
At 2300
write <<s,EraseLine>>
JUDGE STOP

Else
Do ClrMesg
Fcolor PromptLineColor
unit retunit
assign jumpptomn=true
Do ClrMesg
Fcolor CurrentColor
LOC FULL
RETURN
Unit SetQLoc

;*******************************************************************
; This unit set query parameters.
;*******************************************************************
Define UnderScore :STRING
PLock :INTEGER
Temp :INTEGER

LOOP QueryState <> 0
  assign LineFacet := SUBSTR(TextLine,1,QueryState-1)
  Fcolor CurrentColor
  at ScreenCoor
  write <<s.LineFacet>>
  assign Qcount := Qcount + 1
  assign PromptCoor.Qcount := WHERE + 1
  assign PLock := INSTRING(TextLine,"*")
  if PLock = 0
    assign Error := "MISSING SYMBOL (\*) IN QUERY LINE= ".
    Do RptError

at ScreenLoc + 1
write <<s,UnderLine>>
Fcolor GREEN
at ScreenLoc + 1
write <<s,RightAns 1 >>
Do SetMesg(GREEN,RightFB 1 )
JUDGE STOP

EndIf
EndIf
Endif

Else
If (UPPER(Qmode)="PICTURES")
  slide CurrentPic
  mode replace
  Branch 0=O,$QHEADER
Else
  Do ClrMesg
  Do SetMesg(RED,CatchAll)
  BRANCH 0=O,$QHEADER
EndIf
else
at baseline+13
write (press RETURN to continue)
judge STOP

EndIf
EndIf

at ScreenLoc + 1
write <<s,UnderLine>>
Fcolor GREEN
at ScreenLoc + 1
write <<s,RightAns 1 >>
JUDGE STOP

If Attempt > MaxTries
  Do ClrMesg
  Do SetMesg(PromptLineColor,GiveAnsFB)
  at ScreenLoc + 1
  write <<s,UnderLine>>
  at ScreenLoc + 1
  write <<s,RightAns 1 >>
  JUDGE STOP

(UPPER(Qmode)="PICTURES")

at ScreenLoc + 1
write <<s,UnderLine>>
Fcolor GREEN
at ScreenLoc + 1
write <<s,RightAns 1 >>
JUDGE STOP

If (Attempt=1) AND

/* End of program. */
**EndIf**

**assign** PromptLine Qcount := SUBSTR(TextLine,QueryState+1,PLock-QueryState-1)

**If**

**assign** Temp := LEN(PromptLine Qcount)

**assign** UnderScore := ""

**For** I := 1,Temp

**assign** UnderScore := UnderScore + "_"

**EndFor**

**Fcolor** PromptColor

**at** PromptCoor Qcount

**write** <<s,PromptLine Qcount>>

**assign** ScreenCoor := Where + 1

**assign** PromptLine Qcount := UnderScore

**Else**

**Fcolor** PromptLineColor

**at** PromptCoor Qcount + 1

**assign** ScreenCoor := Where + 1

**EndIf**

**assign** TextLine := SUBSTR(TextLine,PLock+1,LEN(TextLine)-PLock)

**assign** QueryState := INSTRING(TextLine,">")

**assign** ColorState := INSTRING(TextLine," ")

**If** (QueryState = 0) AND (ColorState = 0)

**Fcolor** CurrentColor

**at** ScreenCoor

**write** <<s,TextLine>>

**assign** TextLine := ""

**Else**

**If** ((ColorState > QueryState) AND (QueryState = 0)) OR ((ColorState < QueryState) AND (ColorState > 0))

**do** SetColor

**EndIf**

EndLoop

Unit TestTime(One,Two)

·********************************************************************

These units (TESTIME, PRESSRET, IO_PAUSE, PAUSEMSG, & START) are units not used in this present program but which may be added to display the time, date, and the campanille... PF4EXIT stops the lesson.

·********************************************************************

define space:string

one : integer
two : integer

**assign** space := Date(space)

at One

**write** <<s,space>>

**assign** space := TIME

at Two

**write** <<s,space>>

Unit IO_PAUSE

define Oldcolor:integer

**assign** Oldcolor := Fcolor

Size 1

Fcolor Dark

When string,Char(27)+"0S",PF4EXIT

When string,"Quit",PF4Exit

When string,"stop",PF4Exit
Input 2300
Do CLRBUF
Fcolor Oldcolor

UNIT PressRet
size 1
charset "camp.fnt"
ERASE 2300;2483
Fcolor white
at 2327
Write 001234402356270ut2073s501qr300
charset "standard"
Do IO_PAUSE
Mode Complement

UNIT PauseMsg
define tempx,tempy,oldcolor:integer
assign tempx := wherex
assign tempy := wherey
assign oldcolor := fcolor
charset "camp.fnt"
write 001234402356270ut20ot2300
charset "standard"
Do IO_PAUSE
mode replace
at tempx,tempy
fcolor dark
Write fcolor oldcolor
Mode Overlay
at tempx,tempy

UNIT Start
Fcolor 7
Do TestTime(235,335)
Size 2,3
Mode Replace
Italics -27
at 234,218
write One moment please.
charset "camp.fnt"
Charset "Standard"
Italics 0
Size 1
Do TestTime(235,335)
Do SetUp("KRO")
Do TestTime(235,335)

Unit PF4Exit
erase
stop
unit prep
define inc : INTEGER
mode overlay
mode normal
mode fixed
fcolor white
Unit MenuKey(Selector,Template)

;******************************************************************************
;       Unit to display menu and route students.
;******************************************************************************

Define More :BOOLEAN
xcoor :INTEGER
Selector:INTEGER
Template :STRING
Error :STRING
ycoor :INTEGER
top :INTEGER
bottom :INTEGER
KEYPRESS:INTEGER,FUNCTION
CHAPTER 10 :STRING
key :integer
i :integer
storefun:real

;******************************************************************************
; Topics for the menu.
;******************************************************************************

assign chapter 1 := "1. Introduction" $$$ TITLES HERE......
assign chapter 2 := "2. Two-word verbs -- hints and helps"
assign chapter 3 := "3. A family of two-word verbs"
assign chapter 4 := "4. Exercises on two-word verbs"
assign CHAPTER 5 := "5. Quit all work"

;******************************************************************************
; Sets appearance of the menu.
;******************************************************************************

assign xcoor := 220
assign ycoor := 43
assign top := 43
assign bottom := 143
MPLOT "M"
SIZE 1,2
fcolor white
mode replace
for i:=1,5
  at 220,43+(i-1)*25
  write <<s,Chapter i>>
  if INSTRING(Template,STRING(i))=0
    .
    .
    at 200,43+(i-1)*25
    .
    .
    fcolor red
  endif
  .
  .
  at 220,43+(i-1)*25
  .
  .
  fcolor white
endfor
DO SETUP("KP1KRO")
assign more := true
fcolor green
at 220,43+(selector-1)*25
write <<s,chapter Selector >>
at 175,210
write Press
do keychar("uparrow")
do
  keychar("downarrow")
at 320,210
write to move to your choice.
at 175,240
write Press
Fcolor GREEN
do
  keychar("return")
at where
write to choose the green item.
at 175,270
write The red
fcolor red
at 254,270
write *
fcolor green
at 265,270
write means the item is completed.

;******************************************************************
; Loop to process input from menu selections. Error messages
; are displayed if an inappropriate key is pressed.
;******************************************************************
loop more and (error="")
  assign Ycoor:=43+(Selector-1)*25
  assign key := KEYPRESS
test key
  value 13 $$ return key was pressed
  assign more := false $$ up arrow was pressed
  value 201
  Fcolor white
  at xcoor,ycoor
  write <<s,Chapter Selector >>
  if ycoor = top
    assign ycoor := bottom
  else
    assign ycoor := ycoor - 25
  endif
  assign Selector := (((ycoor-43)/25)+1)
  assign xcoor := 220
  fcolor green
  at xcoor,ycoor
  write <<s,Chapter Selector >>
  value 202 $$ down arrow was pressed
  Fcolor white
  at xcoor,ycoor
  write <<s,chapter Selector >>
  if ycoor = bottom
    assign ycoor := top
  else
    assign ycoor := ycoor + 25
  endif
  assign Selector := (((ycoor-43)/25)+1)
  assign xcoor := 220
  fcolor green
  at xcoor,ycoor
  write <<s,Chapter (((ycoor-43)/25)+1)>>
  value 203..204 $$ left or right arrow pressed
  fcolor dark
  at 2300
  write <<s,char(7)>>
erase 2300;2483
  Fcolor RED
  at 2300
  write Move UP and DOWN with UP or DOWN ARROW KEYS...
end loop

DO PF4exit
assign Error:="DONE"
at 2300
write <<s,char(7)>>
erase 2300;2483
Fcolor RED
at 2300
write Press RETURN to do an item. Use UP/DOWN ARROWS to move.
endtest

doit

Do SETUP("KPOKR1")
Fcolor CurrentColor
italics 0
mode normal
mode replace
Test Selector

value 1
open "intro.dat",5,read $$
do MainLoop
close 5
if jumptoman
  erase
  assign storefun:=rewind(5)
  assign numright:=0
  assign numok:=0
  return
endif
erase
assign diagtest:=true
assign practice:=false
assign practice2:=false
OPEN "exer1.dat",5,READ
DO MAINLOOP
CLOSE 5
if jumptoman
  erase
  assign storefun:=rewind(5)
  assign numright:=0
  assign numok:=0
  return
endif
erase
fcolor blue
rorigin 0,0
rbox 0,0;767,473:3
fcolor magenta
at 505
write Your score is <<s,numright>> correct out of 15.
If you had 10 or more correct, you may not need
to continue this lesson. Choose item
number 5 on the menu to quit the lesson.
If you got less than 10 correct, you may need
to practice more. Choose items 2,3 or 4.
at 2005
write press
do keychar("return")
at where+001
write to continue
pause
erase
WHEN DONE WITH THIS TOPIC IN THE MENU MARK IT WITH A STAR
THIS IS DONE BELOW IN THE NEXT LINE
assign Template:=REPLACE(Template,String(Selector),"*")
erase
assign numright:=0

VALUE 2
open "explain.dat",5,read
do MainLoop
close 5
if jumptoman
  . erase
  . assign storefun:=rewind(5)
  . assign numright:=0
  . assign numok:=0
  . return
dendif
assign Template:=REPLACE(Template,String(Selector),"*")
erase

VALUE 3
erase
open "rebintro.dat",5,read
do MainLoop
close 5
if jumptoman
  . erase
  . assign storefun:=rewind(5)
  . assign numright:=0
  . assign numok:=0
  . return
dendif
slide "rebus.pic"
pause
erase
open "stratin.dat",5,read
do MainLoop
close 5
if jumptoman
  . erase
  . assign storefun:=rewind(5)
  . assign numright:=0
  . assign numok:=0
  . return
dendif
open "strategy.dat",5,read
do MainLoop
close 5
if jumptoman
  . erase
  . assign storefun:=rewind(5)
  . assign numright:=0
  . assign numok:=0
  . return
dendif
erase
open "getready.dat",5,read
do MainLoop
close 5
if jumptoman
  . erase
.. assign storefun:=rewind(5)
.. assign numright:=0
.. assign numok:=0
.. return
   endif
slide  "tfr.pic"
pause
erase
slide  "tout.pic"
pause
erase
slide  "tout2.pic"
pause
erase
slide  "tdown.pic"
pause
erase
slide  "tdwncurt.pic"
pause
erase
slide  "deskup.pic"
pause
erase
slide  "tup.pic"
pause
erase
slide  "cheryl.pic"
pause
erase
slide  0,260;765,479
slide  "boxsave.pic"
pause
erase
slide  69,330;430,400
slide  "boxmess.pic"
pause
erase
slide  "offob.pic"
pause
erase
slide  "toffbig.pic"
pause
erase
slide  "planeoff.pic"
pause
erase
slide  "taft.pic"
pause
erase
slide  "tbk.pic"
PAUSE
erase
slide  "ton.pic"
pause
erase
slide  "tovet.pic"
pause
erase
assign Template:=Replace(Template,STRING(Selector),"*")
value  4
assign diagtest:=false
assign practice2:=true
open  "printro.dat",5,read
do  MainLoop
close  5
if jumptoman
.. erase
.. assign storefun:=rewind(5)
.. assign numright:=0
endif
open "temp.dat",5,read
do MainLoop
close 5
if jumptoman
  erase
  assign storefun:=rewind(5)
  assign numright:=0
  assign numok:=0
  return
endif
ERASE
fcolor blue
href
rorigin 0,0
rbox 0,0;767,473:3
fcolor magenta
at 505
write You have <<s,numok>> so far out of 10 possible. If you scored less than eight in this section, it might be good to review the earlier lesson. If you want to return to the menu, at the next > ______, press.

tcolor green
at where+001
do keychar("PF2")
at where
write .
fcolor magenta
at where+001
write To continue with the practice, press
tcolor green
at where+001
do keychar("return")
at where
write .
fcolor magenta
pause
ERASE
open "temp2.dat",5,read
do MainLoop
close 5
if jumptoman
  erase
  assign storefun:=rewind(5)
  assign numright:=0
  assign numok:=0
  return
endif
ERASE
fcolor blue
href
rorigin 0,0
rbox 0,0;767,473:3
fcolor magenta
at 505
write You have <<s,numok>> total in the practice section. There were 32 possible. If you scored less than 24 of the total, it might be valuable to review the material. To return to the menu, press

tcolor green
at where+001
do keychar("return")
at where
write
pause erase
assign Template:=REPLACE(Template,STRING(Selector),"*")
assign numok:=0
value 5 $$$***************QUIT ALL WORK
erase slide "quits2.pic"
ERASE
do PF4EXIT
ASSIGN ERROR:="DONE"
endtest
ASSIGN SELECTOR:=SELECTOR+1
if Selector>5
  assign Selector:=1
endif
RETURN
lesson select
;
; Define LineType :STRING $$ First Char in FileLine-
controls logic.
;   TextLine :STRING $$ Line of text read from
file.
;
Rcount :INTEGER $$ Number of right answers
read from file.
;
RightAns5 :STRING $$ array of possible right
ans for l question.
;
RFBcount :INTEGER $$ Number of right
feedbacks.
;
RightFB5 :STRING $$ Feedback corresponding to
each right answer.
;
Wcount :INTEGER $$ Number of wrong answers
read from file.
;
WrongAns5 :STRING $$ Array of possible wrong
ans from l question.
;
WFcount :INTEGER $$ Number of wrong feedbacks
read from file.
;
WrongFB5 :STRING $$ Feedback corresponding to
each wrong answer.
;
NewValue :INTEGER $$ Number read from file.
;
Spacing :INTEGER $$ Number of lines to space
between text on the screen.
;
MaxTries :INTEGER $$ Number of attempts before
student is given answer.
;
StartLine :INTEGER $$ Text Coor for Starting
postion of text.
;
Margin : INTEGER
indent from left screen boundary.
* BaseLine : INTEGER
on screen.
* Temp : INTEGER
number read from screen.
* FileLine : INTEGER
file- used for error pointer.
* Error : STRING
msg to display.
* ScreenCoor : INTEGER
writing on screen.
* PromptLine 5 : STRING
replotting for wrong answers.
* QueryState : INTEGER
if "x" is inline.
* ColorState : INTEGER

Qcount : INTEGER
a student response.
* LineFacet : STRING
textline.
* PromptCoor 5 : INTEGER
pending queries.
* CatchAll : STRING
unspecified wrong answer.
* Quest : INTEGER
query in a line.
* MatchFound : BOOLEAN
found.
* SkipQueSym : BOOLEAN
*
I : INTEGER
EraseLine : STRING
erasing a message up to 80 characters.
* HueTable : INTEGER
and converted to integers thru this table.
* PromptColor : INTEGER
student response color.
* PromptLineColor : INTEGER
following the promptColor symbol.
* CurrentColor : INTEGER
written in this color.

$$ Number of spaces to
$$ Y-componet of next line
$$ Temporary variable for
$$ Count of line read from
$$ Variable to pass error
$$ Current position for
$$ Array of prompt lines for
$$ Position/flag indicating
$$ Number of queries pending
$$ Unique component of
$$ Position of prompt for
$$ Feedback to catch any
$$ Looping index for each
$$ Set to true if answer is
$$ String of blanks for
$$ Table of HueTable read in
$$ Color of prompt and
$$ Color of prompt line
$$ Color of text to be
try

mode replace
assign Debug1 := " *** INSTRUCTOR/DEBUG MODE : PRESS RETURN FOR RIGHT ANSWERS ***"
assign Debug2 := " *** INSTRUCTOR/DEBUG MODE : PRESS RETURN FOR WRONG ANSWERS ***"
assign Debug3 := " *** INSTRUCTOR/DEBUG MODE : CATCHALL FOR UNSPECIFIED ANSWERS (NOW DISPLAYED) ***"
assign InstructorMode := TRUE
assign DebugMode := FALSE
assign GiveAnsFB := "The Correct answer is given above. Press RETURN to continue."
assign Spacing := 100
assign MaxTries := 3
assign Startline := 04
assign Margin := 4
assign EraseLine := "n"
assign RightFB1 := "OK, Press RETURN to continue..."
assign WrongFB1 := "No, Try again."
assign HueTable "BLACK" := 0
assign HueTable "DARK" := 0
assign HueTable "BLUE" := 1
assign HueTable "RED" := 2
assign HueTable "MAGENTA" := 3
assign HueTable "PURPLE" := 3
assign HueTable "VIOLET" := 3
assign HueTable "GREEN" := 4
assign HueTable "CYAN" := 5
assign HueTable "YELLOW" := 6
assign HueTable "WHITE" := 7
open "info.dat",3,WRITE 0=0
loop 0=0
  . mode replace
  . assign SkipQueSym := FALSE
  . assign Rcount := 0
  . assign RFBcount := 0
  . assign Wcount := 0
  . assign WFcount := 0
  . assign Qcount := 0
  . assign Quest := 1
  . assign FileLine := 0
  . prompt ">
  . assign Qcount := 0
  . assign BaseLine := 00
  . assign PromptLineColor := 6
  . assign PromptColor := 2
  . assign ScreenCoor := StartLine
  . assign CurrentColor := 7
  . do showpage
Unit Showpage fcolor Green erase 1200;1384 at 1215 write Enter italics -27 write filename.filetype italics 0 fcolor red prompt "=>" Query right

assign filename := response open filename,5,read if IResult <> 1

assign error := "NO SUCH FILE OR ERROR IN FILENAME"

Do RptError

EndIf

EndQ fcolor white prompt ">

do MainLoop include xDriver.dal include SelectJ.dal include SelectQ.dal include SelectC.dal include SelectU.dal endLesson