ED 319 468 JC 900 318

AUTHOR Kendra, Lawrence M.; Clavner, Jerry B.
TITLE Coding Language vs. Authoring System: To Code or Author--That Is the Question.
PUB DATE Jun 90
NOTE 15p.; Paper presented at the National Education Computing Conference (Nashville, TN, J'une 25-27, 1990. Coding for computer programs (p.10-14) will not reproduce adequately.
PUB TYPE Speeches/Conference Papers (150) -- Viewpoints (120)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Authoring Aids (Programing); *Computer Assisted Instruction; *Computer Software Developme.t; *Teacher Developed Materials
IDENTIFIERS *Storyboard Programs

ABSTRACT

Drawing from experiences at Cuyahoga Community College, this paper offer guidance on the use of storyboard packages and authoring languages to develop interactive instructional materials. Introductory comments reveal that the greatest use of computers in courses other than computer and information sciences is still in drills, tutorials, writing composition, and simulations; and that there is little research-based data on the effectiveness of computers as a method of instruction or on the time, energy, and resources required to prepare to use computers in instruction. Next, the paper lists several requirements for interactive instructional materials, e.g., the lesson must extend from or reinforce classroom activities; a three-way, ongoing, systematized interaction must take place among the student, instructor, and material in the computer; the material available to the student must give clear fail-safe instructions, use a developmental/incremental process toward mastery of concepts, and draw forth questions that the student will bring to the instructor. Tips about storyboard lesson planning are presented next, suggesting that the approach is best suited to the visually oriented, that it often requires considerable tinkering and time, and that it tempts the instructor to include too many sub-menus. Next, authoring systems, which are designed for instruction based on lecturing-demonstrating-exampling, are discussed and five drawbacks are identified. Attachments include outlines of the curriculum and lesson development processes, and a basic language program for an employment theory lesson. (WJT)

CODING LANGUAGE VS. AUTHORING SYSTEM
To Code or Author - That is the Question

National Education Computing Conference
Nashville, Tennessee
June 25 - 27, 1990

Lawrence M. Kendra, Ph. D.
Economics, Computer Science, and General Business
Cuyahoga Community College

and

Jerry B. Clavner, Ph. D.
Sociology, Social Science, and Anthropology
Cuyahoga Community College

We need computers to halp students learn.
The new principles are clear.
Cybernetics is/are here. The future can be demonstrated.
The visual grooves the audio.
Concepts, processes, sequencing, branching. So where's the loop for the teacher.

In all the discussions about computers in the classroom very little attention has been directed to whether it assists in making teaching for the instructor more efficient or effective. (If computers are so good, why can't they go to our committee meetings for us.)

Several things stand out in the literature that indicate a general lack of knowledge in this area, i.e.:

Good teachers (whatever they are) feel that computers make them better teachers (however that is measured).

Teachers who go from no usage to some usage of computers view the computer as a useful teaching tool.

Most teachers using computers in the classroom use them as audio-visual aids.

Rarely are computers used interactively in the teaching/learning process (by students and teachers together) even in WAN and LAN configurations.

Rarely are they used to assist in model or paradigm development, or even to develop complex computational operations.

Our investigation at our college indicates that the greatest use of computers in and outside the classroom in courses other than computer and information sciences is still in drill, tutorial, writing composition, and simulation. We're afraid that this may be the case even in computer courses also. The use of the computer as a high tech student guide or lab manual is as fraught with the problems those early primitive learning tools had, and more (we know; we wrote one)!

Analyses of the effectiveness and efficiency of computers in the classroom as a method of instruction or adjunct to instruction still remain mostly anecdotal. It may be that colleges and universities give material support to the preparation of materials for teaching computers, but do not support the "write ups", but we doubt it! The development or modification of packages by teaching faculty is a time consuming task in and of itself. Note, introducing computers into one's teaching involves the stress of changing behaviors and dealing with both old beliefs and new ideas that can only be validated after significant successes.

It is also interesting in light of the technology and the programming capabilities that very little has been written on time and energy and resources (read: money) involved in assessing different configurations of preparing for teaching with computers as opposed to preparing in other ways. This section of the paper addresses some of the issues involved in studying this matter. The complete research will await the time, energy, and money needed to complete the project; right now we are too busy teaching.

Our framework here is to set out a paradigm by which teaching faculty can intelligently make some choices as to how they are going to use the computer. We feel it is necessary to stipulate that the decision whether to use a computer no longer optional. Peers and even some administrations nave convinced themselves that computers are essential components to "proper" and appropriate teaching. Many of these same administrators though seem to think that verbal support is sufficient.

Our report is based on our personal experience in using the computers to: generate individualized (essay and multiple guess tests), having students answer and proof their materials on the computer in a variety of disciplines, games and simulations, assisting in setting up a program in Art Graphics, writing a student interactive study guide with all the programs, and hours of debugging.

This report grew out of an attempt to prepare truly interactive material for students in Economics, Social Science, and Business courses to help them understand the core concepts and processes of micro and macro Economics. We tried several computer languages, several story board software packages, and different authoring systems. All of these tools take time to learn and use. Not just the time in front of the CRT, but the time in arranging the material and lessons in a manner that reflects cybernetic principles. All involve the same learning time, and there is no transfer of learning from one system to another; they design it that way!

It should be understood at the outset that simply transferring a lesson/unit/course to a computer is disastrous in terms of the educational process. It compares to the talking heads in some of the telecourses we have all seen. The rationale for any computer based instruction must be that the student learns best when he/she does whatever it is we say the processes of the field(s) are. Beware though, it isn't simply providing student-computer interaction that makes for learning that is actually augmented by the computer. The lesson must extend from or reinforce the classroom activities. The student needs to be in a three way interaction among him/herself, the instructor, and the material in/on the computer. This interaction must be ongoing and systemized.

The material available to the student through the computer must, in addition to giving clear fail safe instructions, set parameters within paradigms that encourage mastery of the concepts through a developmental process; it cannot be, at least for the beginning student, an unlimited "field" that confuses and confounds. Incremental advances should spur substantive questioning of breadth and depth that the student will take to the instructor individually, even by E Mail, but preferably in the classroom situation, where others can learn also. This questioning can then be utilized in a Lancasterian manner to encourage synergy, that when it occurs is worth almost any effort an instructor could expend. Which brings us full circle.

The processes and advantages for the student of both the storyboard and authoring systems are discussed elsewhere. Here, we wish to rehearse the cognitive and psychomoter, and we would argue, the affective components of the inputing processes to determine the best fit for instructor and method of material preparation.

Storyboard lesson planning is best done by and for those who are visually oriented. To roughly compare it to a set of cartoon panels does not do it an injustice, but is really quite a good metaphor. In the making of a movie or commercial, storyboards are constructed (and revised) to insure the development and flow of whatever is being produced. The storyboard approach often requires considerable tinkering, but has the distinct advantage of allowing the director/producer to start and restart anywhere at the beginning, middle, or end. Some authoring packages are almost impossible to edit.

Whole units can be substituted without disrupting the flow if done very carefully. Storyboard construction takes a long time even with user friendly/menu driven programs and experience doing one or more sets does not make the next set any easier if the material is discrepant. Storyboard programs are slightly easier to use for developing sub-menu directions and materials. We have not found, though, any significant difference in student inclination to review material that is insufficiently learned (with self-tests included). In fact, students appear frustrated with too many alternatives which the storyboard producer/director often feels compelled to insert, simply because it can be done!

Authoring systems are designed for "lecturing-demonstrating-exampling" instruction. Using core concepts for the basis of discrete lessons, the author writes much as he/she would if lesson plans were being constructed as part of a specific unit. The lesson is an independent entity, but builds on previous learned material. Authoring systems are suited for those instructors who have a comprehensive understanding of programming and can utilize mathematical constructs and transform them into verbal linear progressions.

The drawbacks are:

1. These type of instructors are often unwilling to "let go" and let the machine do the instruction. They feel they have to be around to do the teaching and answer the questions that the student has about the "whys." immediately.

2. Authoring systems are often designed to give information and do not require the students to progress in a manner that causes the instructor to receive the feedback and reinforcement. We are not casting dispersions. In fact, part of the exercise programs that we developed were inadvertently designed so that the instructor was giving high grades, not because students learned anything, but because they were clever, caught on easily, or had mathematical talent.

3. Authoring systems require that the student (and the author) do a great deal of work on his/her own in preparation for learning with the computer. There is much that can be inserted into the programs with an authoring system, but as much of it would turn out to be redundant, it is best left out. The instructor who relies on an authoring system to do all the work simply makes the machine as boring as the instructor who lectures exclusively and wonders why students don't come regularly. Like the afternoon soaps, more left to the imagination, the more viewers pay regular attention.

4. While program refinement and correction, particularly to allow for different parameters, is easy to do with authoring systems, it should be noted that storyboards are better suited for more extensive changes.

5. Directly related, errors of content that are not linked to programming are more difficulty to spot; it is assumed that the student did something incorrectly.

Note that nowhere in the discussion have we addressed the issue of IBM vs. Apple [trademarks acknowledged]. The question of machinery appears to be more a question of style or perceived need for bells and whistles. Those interested in this type of endeavor should understand machine limitations and be concerned with "the lesson"; not marrying a machine to a system, program, or package. Several points though should be made. Small screens on CRT's may impede. "Compatibles" may not be so compatible. Signs and symbols specific to certain disciplines may be easier to access on an Apple.

Curriculum Development Process

- Write, test and implement levels 1, 2 and 3 performance objective.

- Select objective statements which will be used as basis of lesson.

  - Level 1 – To be able to interpret a given Employmemt Theory table with respect to any specified economic tendency.

  - Level 2 – Given an Employment Theory table, identify the respective conditions and recommend specific strategies for reaching (continuing) equilibrium.

  - Level 3 – Given an Employment Theory table, write an essay in which you explain a specific economic condition and recommend a strategy to achieve and/or maintain an equilibrium.

In conclusion, the instructor who wishes to enter into the development of interactive learning programs need not reinvent the wheel. There are tools out there and they can actually assist the committed teacher in creating teaching/learning materials that are not only beneficial to the student, but stimulating and fulfilling to the instructor. In turn, this process can help the teaching/learning process itself.

Lesson Development Process

Authoring Systems

Storyboard the lesson

- Pedagogy
  - feedback required
  - learning assumptions
  - intervention requirements
  - classroom considerations
  - variety of learning modes

- Lesson Operation
  - bugs and breaks (free from)
  - error handling (messages)
  - directions (clear and correct)
  - documentation
    * teacher helpful
    * student helpful
  - bells and whistles (?)
  - graphic display effective

Programming Languages

➡ Flowchart the program

- Pedagogy
  - feedback required
  - learning assumptions
  - intervention requirements
  - classroom considerations
  - variety of learning modes

Program Operation (Modules)

- alpha/beta testing
- error handling routines
- write direction text
- write documentation (manual(s))
  * for teacher
  * for student
- determine sound and graphic needs
- select appropriate graphics mode

Lesson Development Process

Authoring Systems

➡ Storyboard the lesson

- Student Outcomes
  - Ease of lesson use
  - adaptability to system
  - enhancement of learning
  - enjoyability
  - learning measurement
  - unintended learning results
  - how does lesson compare with other available software
  - available support

Programming Languages

➡ Flowchart the program

- Student Outcomes
  - Ease of lesson use
  - adaptability to system
  - enhancement of learning
  - enjoyability
  - learning measurement
  - unintended learning results
  - how does lesson compare with other available software
  - available support

Exhibit 1. Basic Language Coding of Employment Theory Lesson.

```
  CLS:SCREEN 0:KEY OFF:C=0
  COLOR 3:PRINT "       Here is an example of how the Employment Theory model will work."
  COLOR 2:PRINT :PRINT "                                AGGREGATE EQUILIBRIUM TABLE"
  PRINT "-------------------------------------------------------------------------------"
  PRINT "Aggregate Business  Net  Disposable  House  House  Business Govrnment Aggregate"
  PRINT " Supply    Saving  Taxes   Income   consump saving  Invest   Spending   Demand"
  PRINT "--------- -------- ----- ---------- ------- ------ -------- --------- ---------"
  PRINT " $  100     $  2    $ 1   $   97    $ 102   $ -5   $   4    $   4     $  110  "
   PRINT "   120         4      2       114      114      0       6        5        125  "
  PRINT "    140         6      3       131      126      5       8        6        140  "
  PRINT "    160         8      4       148      138     10      10        7        155  "
  PRINT "    180        10      5       165      150     15      12        8        170  "
  PRINT "    200        12      6       182      162     20      14        9        185  "
  COLOR 3:LOCATE 22,1:PRINT "          $ 138 - $ 126     12                    1         1"
  PRINT "  MPC = --------------- = ---- = 70.59%     k = ---------- = ------ = 3.4"
  PRINT "          $ 148 - $ 131     17                    1 - .706     .294";
  IF C=1 THEN 11 ELSE FOR TD = 1 TO 5000:NEXT TD
  LOCATE 15.15:COLOR 7:PRINT "Type the letter of your choice followed by ";
  CHR$(34);CHR$(17);CHR$(196);CHR$(217);CHR$(34)
1 PRINT "                           B    to build new system"
2 PRINT "                           C    to make contractionary changes"
  PRINT "                           E    to make expansiotary changes"
3 PRINT "                           Q    to quit"
4 INPUT "                  What's your choice ";C$
  IF C$="q" OR C$="Q" THEN COLOR 7:LOAD"a:menu.bas",R
5 LOCATE 14,1:FOR LN=1 TO 8:
  PRINT "                                                                       ":NEXT LN
6 IF C$="b" OR C$="B" THEN 11
7 IF C$="c" OR C$="C" THEN C=1:GOTO 40
8 IF C$="e" OR C$="E" THEN C=1:GOTO 48
9 GOTO 27
10 COLOR 5:LOCATE 2,1:PRINT "Let's assume that we want to contract to an equilibrium of 120.":GOSUB 56
11 COLOR 4:LOCATE 17,1:PRINT "FIRST STAGE -> We could raise taxes and encourage business saving:":GOSUB 56
12 LOCATE 9,1:PRINT "    120         5      4       111      112     -1":LOCATE 9,75:PRINT "123"
13 COLOR 7:LOCATE 14, 1:PRINT "Our original disequilibrium position was:":
   PRINT "    120         4      2       114      114      0       6        5        125  ":COLOR 4:GOSUB 56
14 COLOR 6:LOCATE 18,1:
   PRINT "SECOND STAGE -> We could raise consumer interest and discourage debt spending:":GOSUB 56
15 LOCATE 9,40:PRINT "110      1":LOCATE 9,75:PRINT "121":GOSUB 56
16 COLOR 9:LOCATE 19,1:PRINT "THIRD STAGE  -> We could lower government spending:":GOSUB 56
17 LOCATE 9,66:PRINT "4":LOCATE 9,75:PRINT "120":GOSUB 56:GOTO 27
18 COLOR 5:LOCATE 2,1:PRINT "Let's assume that we want to expand to an equilibrium of 160.  "
19 COLOR 4:LOCATE 17,1:PRINT "FIRST STAGE -> We could discourage business saving and raise taxes.":GOSUB 56
20 LOCATE 11,1:PRINT "    160         6      3       151      140     11":LOCATE 11,75:PRINT "157":
   FOR TD= 1 TO 1000:NEXT TD
21 COLOR 7:LOCATE 14, 1:PRINT "Our original disequilibrium position was:":
   PRINT "    160         8      4       148      138     10      10        7        155  ":
22 COLOR 4:GOSUB 56 52 COLOR 6:LOCATE 18,1:PRINT "SECOND STAGE -> We could lower interest on household
   saving:":GOSUB 56
23 LOCATE 11,40:PRINT "142      9":LOCATE 11,75:PRINT "159":GOSUB 56
24 COLOR 9:LOCATE 19,1:PRINT "THIRD STAGE  -> We could encourage businesses to invest in new capital:":GOSUB 56
25 LOCATE 11,56:PRINT "11":LOCATE 11,75:PRINT "160":GOSUB 56:GOTO 27
26 LOCATE 25,39:PRINT CHR$(34)CHR$(17)CHR$(196)CHR$(217)CHR$(34);
         KEY$:IF LEN(A$)=0 THEN 57
         E 25,39·PRINT "        ";:RETURN
```

Exhibit 2. Pascal Language Coding of Employment Theory Lesson.

```
PROGRAM EmploymentTheory;
USES Dos,Crt;
VAR
   R,key,Get : Char;
   C, L      : Integer;
LABEL 1;
Procedure BottomArrow;
     BEGIN
          REPEAT
                GotoXY(25,25);
                Write('  Press '#34#17#196#217#34' for program MENU.')
          UNTIL KeyPressed;
          GotoXY(25,25);Write('                              ');
     END;
Procedure Stage;
     BEGIN
          GotoXY(25,25);Write('  Press '#34#17#196#217#34' for next stage .        ');
          REPEAT
                IF Keypressed THEN
                   BEGIN
                        key := readkey;
                   END;
          UNTIL key in [#13];
          GotoXY(25,25);Write('                              ');
     END;
Procedure Cleanup;
     BEGIN
          L := 14;REPEAT
                       GotoXY( 1,L);Write('
                       '); inc(L);
                  UNTIL L = 20;
     END;
Procedure ScreenOutput;
     BEGIN
          ClrScr;TextColor(15);GotoXY(1,1);
          Write('     Here is an example of how the Employment Theory model will work');
          GotoXY(1,3);TextColor(10);Write('                              AGGREGATE EQUILIBRIUM TABLE'):
          GotoXY(1,4);Write('-------------------------------------------------- --------------------------');
          GotoXY(1,5);Write('Aggregate Business Net Disposable House House Business Govrnment Aggregate');
          GotoXY(1,6);Write(' Supply    Saving  Taxes  Income   Consump Saving Invest  Spending  Demand ');
          GotoXY(1,7);Write('--------- -------- ----- ---------- ------- ----- -------- ------  - ---------');
          GotoXY(1,8);Write(' $  100    $  2    $ 1  $   97     $ 102  $ -5   $   4   $   4    $  110');
          GotoXY(1,9);Write('    120       4      2     114       114    0       6        5        125'),
          GotoXY(1,10);Write('   140       6      3     131       126    5       8        6        140');
          GotoXY(1,11);Write('   160       8      4     148       138   10      10        7        155');
          GotoXY(1,12);Write('   180      10      5     165       150   15      12        8        170');
          GotoXY(1,13);Write(    200      12      6     182       162   20      14        9        155');
          TextColor(15);GotoXY(1,22);Write('       $ 138 - $ 126     12                         1           1');
          GotoXY(1,23);Write('  MPC = --------------- = ---- = 70.58%     k = --------- = ------ = 3.4');
          GotoXY(1,24);Write('        $ 148 - $ 131     17                  1 - .706     .294');
          IF C = 0 THEN BottomArrow;
     END;
Procedure Choice;
     BEGIN
          key := ' ';C := 0;TextColor(6);
          GotoXY(10,15);Write('Press the letter of your choice (no '#34#17#196#217#34') is required');
          GotoXY(17,16);Write('B   to Begin a new session.');
```

Exhibit 2. Pascal Language Coding of Employment Theory Lesson.

```
          GotoXY(17,17);Write('C   to review CONTRACTIONARY changes.');
          GotoXY(17,18);Write('E   to review EXPANSIONARY changes.');
          GotoXY(17,19);Write('Q   to end (QUIT) program.');
          GotoXY(14,20);Write('What'#39's your choice ?');
          REPEAT
               IF keypressed THEN
                  BEGIN
                       key := readkey;
                       key := upcase(key);
                  END;
          UNTIL key in ['B','C','E','Q'];
          CleanUp;
     END;
 procedure Expansion;
 BEGIN
     Key := ' ';GotoXY(1,2);Textcolor(14);
     Write('         Let'#39's assume that we want to expand to an equilibrium 160');
     GotoXY( 1,17);Write(' FIRST STAGE => We could discourage business saving and lower taxes:');
     GotoXY(16,11);Write('6       3       151        140      11');
     GotoXY(75,11);Write('157');DELAY(2500);
     TextColor(15);GotoXY( 1,14);Write('Our original condition was:');
     GotoXY(1,15);Write('    160          8      4        148         133      10       10          7          155');Stage;
     key := ' ';Textcolor(4);
     GotoXY( 1,18);Write('SECOND STAGE => We could lower interest on household saving:');
     GotoXY(40,11);Write('142       9');GotoXY(75,11);Write('159');Stage;
     key := ' ';TextColor(9);GotoXY( 1,19);Write(' THIRD STAGE => We could encourage business investment:');
     GotoXY(56,11);Write('11');GotoXY(75,11);Write('160');BottomArrow;CleanUp;
 ND;
 procedure Contraction;
 BEGIN
     Key := ' ';GotoXY(1,2);Textcolor(14);
     Write('         Let'#39's assume that we want to contract to an equilibrium 120');
     GotoXY( 1,17);Write(' FIRST STAGE => We could raise taxes and encourage business saving:');
     GotoXY(16, 9);Write('5       4       111        112     -1');
     GotoXY(75, 9);Write('123');DELAY(2500);
     TextColor(15);GotoXY( 1,14);Write('Our original condition was:');
     GotoXY(1,15);Write('    120          4      2        114         114       0        6          5          125');Stage;
     key := ' ';Textcolor(4);
     GotoXY( 1,18);Write('SECOND STAGE => We could raise consumer interest and discourage debt spending');
     GotoXY(40, 9);Write('110       1');GotoXY(75, 9);Write('121');Stage;
     key := ' ';TextColor(9);GotoXY( 1,19);Write(' THIRD STAGE => We could lower government spending:');
     GotoXY(66, 9);Write('4');GotoXY(75, 9);Write('120');BottomArrow;CleanUp;
 ND;
BEGIN
     C:=0;
   : ScreenOutput;
     IF C=1 THEN Expansion;
     IF C=2 THEN Contraction;
     Choice;
     CASE key OF
             'B' : BEGIN;C:=0;GOTO 1;END;
             'C' : BEGIN;C:=2;GOTO 1;END;
             'E' : BEGIN;C:=1;GOTO 1;END;
     END;
```

Exhibit 3. C Language Coding of Employment Theory Lesson.

```
LESSON

EmploymentTheory
'
: i , a1;
ar w;
ple()

     for (i = 0; i <= 25; ++i) puts(" ");
     puts("     Here is an example of how the Employment Theory model will work");
     if (a1 == 1)puts("        Let s assume that we want to expand to an equilibrium 160");
     if (a1 == 2)puts("        Let s assume that we want to contract to an equilibrium 120");
     if (a1 == 0) puts(" ");
     puts("                        AGGREGETE EQUILIBRIUM TABLE");
     puts("------------------------------------------------------------------");
     puts("Aggregate Business  Net  Disposable  House  House  Business Govrnment Aggregate"),
     puts(" Supply    Saving  Taxes   Income   Consump Saving  Invest   Spending   Demand ");
     puts("--------- -------- ----- ---------- ------- ------ -------- --------- ---------");
     puts(" $  100    $   2   $ 1   $   97     $ 102  $ -5   $   4    $   4     $  110");puts(" ");
     if (a1 != 2)puts("    120        4      2       114        114     0       6          5          125");
     if (a1 == 2)puts("    120        5      4       111        110     1       6          4          120");
     puts(" ");
     puts("   140       6     3      131       126     5       8        6       140");puts(" ");
     if (a1 != 1)puts("    160        8      4       148        138    10      10          7          155");
     if (a1 == 1)puts("    160        6      3       151        142     9      11          7          160");
     puts(" ");
     if (a1 == 0)puts("    180       10      5       165        150    15      12          8          170");puts(" ");
     if (a1 == 0)puts("    200       12      6       182        162    20      14          9          185"):
     if (a1 == 1)puts(" FIRST STAGE => We could lower taxes and discourage business saving:");
     if (a1 == 1)puts("SECOND STAGE => We could lower interest on household saving:");
     if (a1 == 1)puts(" THIRD STAGE => We could encourage business investment:");
     if (a1 == 2)puts(" FIRST STAGE => We could raise taxes and encourage business saving:");
     if (a1 == 2)puts("SECOND STAGE => We could lower interest on household saving:");
     if (a1 == 2)puts(" THIRD STAGE => We could lower government spending:");
     if (a1 == 0) for (i = 0; i <= 2; ++i) puts(" ");
     if (a1 == 1 || a1 == 2)puts("Our original condition was:");
     if (a1 == 1)puts("    160        8      4       148        138    10      10          7          155"):
     if (a1 == 2)puts("    120`       4      2       114        114     0       6          5          125");
     puts("          $ 138 - $ 126     12                        1          1");
     puts("   MPC = --------------- = ---- = 70.58%     k = ---------- = ------ = 3.4");
     puts("          $ 148 - $ 131     17                    1 - .706     .294");
     printf("   Press then letter M and press carriage return for program MENU."):
            scanf("%1s", &w);
          . if (w != 'M');
     for (i = 0; i <= 25; ++i) puts(" ");
     }
ain()

     while ( w != 'Q') {
            for (i = 0; i <= 10; ++i)puts(" ");
            puts("   Press the letter of your choice followed by a carriage return :");puts(" ");
            puts("              B   to Begin a new session.");puts(" ");
```

Exhibit 3. C Language Coding of Employment Theory Lesson.

```
puts("          C   to review CONTRACTIONARY changes.");puts(" ");
puts("          E   to review EXPANSIONARY changes.");puts(" ");
puts("          Q   to end (QUIT) program.");puts(" ");puts(" ");
for (l = 0; l <= 5; ++l) puts(" ");
printf("    What s your choice ?");
        if (w != 'B' || w != 'C' || w != 'E' || w != 'Q')
                scanf("%1s" , &w);
        if ( w == 'B' )  al = 0;
        if ( w == 'C' )  al = 2;
        if ( w == 'E' )  al = 1;
        if ( w == 'Q' )  exit (0);
table(); }
```