AN INVITATION TO PROGRAMMING™ 2:
WRITING PROGRAMS ONE AND TWO

Model CX4106
Use with
ATARI® 400™ or ATARI 800™
PERSONAL COMPUTER SYSTEMS
# CONTENTS

1 INTRODUCTION 1

2 HOW TO USE THIS COURSE 3
   ATARI Components Required 3
   General Instructions for Loading the Cassette Tape 3
   Preparation for the Practice Sessions 5

3 SUMMARY OF LESSONS:
   WRITING PROGRAMS ONE 7

4 DESCRIPTION OF FRAMES FOR
   WRITING PROGRAMS ONE 9
   Lesson One: Keyboard 9
   Lesson Two: Printing on the Television Screen 9
   Lesson Three: Characteristics of a Program 11
   Lesson Four: Names and Numbers in Memory 15
   Lesson Five: Computer Logic 19
   Lesson Six: Library Function 23
   Lesson Seven: FOR/NEXT Loops 25
   Final Quiz: Writing Programs One 28

5 SUMMARY OF LESSONS:
   WRITING PROGRAMS TWO 29

6 DESCRIPTION OF FRAMES FOR
   WRITING PROGRAMS TWO 31
   Lesson One: Screen Formatting Instructions 31
   Lesson Two: READ/DATA 33
   Lesson Three: Arrays 35
   Lesson Four: PEEK and POKE 38
   Lesson Five: ATASCII Code and Instructions 40
   Lesson Six: String Handling 43
   Lesson Seven: Multiple Statements 46
   Lesson Eight: Subroutines 48
   Final Quiz: Writing Programs Two 51
AN INVITATION TO PROGRAMMING™ 2 covers beginning through advanced BASIC programming techniques. This course provides a two-part format with an interactive style of presentation, allowing you to experience immediate use of your computer. Completion of WRITING PROGRAMS ONE supplies you with rudimentary skills which can be applied in extending your knowledge of BASIC through actual "hands-on" application in WRITING PROGRAMS TWO.

The course has the following components:

- Cassette for Writing Programs One containing both the computer program and audio material
- Cassette for Writing Programs Two containing both the computer program and audio material
- This workbook containing both sections of the course to be used as a study guide and reference source
HOW TO USE THIS COURSE

ATARI COMPONENTS REQUIRED

- ATARI® 400™ or ATARI 800™ Personal Computer System with:
  - 8K (minimum) Random Access Memory (RAM)
  - ATARI BASIC Computing Language Cartridge
- ATARI 410™ Program Recorder
- Writing Programs One Program Cassette
  Writing Programs Two Program Cassette

GENERAL INSTRUCTIONS FOR LOADING THE CASSETTE TAPE

Refer to Figure 1.

1. Connect the ATARI 400 or ATARI 800 Personal Computer System to your television set and to a wall outlet as instructed in the Operator’s Manual.

2. Make sure that the ATARI 410 Program Recorder is properly connected to the computer console and to a wall outlet (see your ATARI 410 Program Recorder Operator’s Manual for further details, if necessary).

   Note: If you have “daisy chained” (connected in series) other ATARI peripherals to your computer console and do not wish to disconnect them, connect your ATARI 410 Program Recorder to the I/O CONNECTOR of the last unit in the chain.

3. Make sure that at least 8K of RAM is installed in your ATARI Personal Computer System. See the Operator’s Manual for Memory Module™ loading instructions.

   Note: If an ATARI Disk Drive is connected to the computer console, the Disk Operating System (DOS) and system software use some of the available Random Access Memory (RAM). The amount of RAM used varies with the version of DOS you are using. Take this overhead into account when calculating the amount of RAM required to run a program.

4. Open the cartridge door and insert the ATARI BASIC cartridge into the cartridge slot. Use the LEFT CARTRIDGE slot on the ATARI 800 Personal Computer System. Close the cartridge door.

5. Turn on your television set. (You may have to adjust the volume on your television set, since the voice from the audio track comes from the television speaker.)
6. Turn on your ATARI Personal Computer System. The POWER switch is on the right side of the computer console.

7. If all equipment is properly connected and turned on, your television screen should display the READY prompt, with the white cursor just below.

8. Press STOP/EJECT on your Program Recorder to open the cassette door.

9. Load Lesson One by holding the cassette tape so that the label (Side 1) is up and the tape leader is facing you.

10. Slide the cassette into the cassette holder and close the door.

11. If necessary, press REWIND to rewind the tape to the beginning. (Set the counter on the Program Recorder to 000.)

12. Type CLOAD on the computer keyboard and press RETURN. The computer will "beep" once to remind you to press PLAY on the Program Recorder.

13. Press PLAY and the RETURN key to start the tape. Through the window in the Program Recorder, note that the tape is turning. The beeps and other sounds you hear coming from the television speaker tell you that Lesson One is being loaded into computer memory.

   **Note:** If you have problems loading either of the course programs and you have peripherals in addition to the Program Recorder attached to the computer console, try disconnecting the other peripherals and connecting the Program Recorder directly to the computer console to isolate any problem. If problems persist, consult the ATARI 410 Program Recorder Operator's Manual.

14. When the television screen displays the READY prompt, the first program encountered on the cassette, Lesson One, has been loaded into the computer. Make sure that the PLAY button on the Program Recorder is still down (and remains down). On the chart provided, make a note of the number on the Program Recorder's counter. Enter this number in the START AUDIO column for Lesson One. If you want to run Lesson One again, simply rewind the tape to this number. As you proceed with the course, similarly keep track of the START AUDIO numbers of the other lessons.

15. Type RUN and press RETURN to start Lesson One. From now on, follow the instructions given in the lesson. At the end of Lesson One, make a note of the Program Recorder counter number. Enter this number in the END AUDIO column for Lesson One.

16. **DO NOT** rewind the cassette tape. When you are ready to begin Lesson Two, follow the same procedure as you did to load Lesson One (begin with Step 12). Follow this procedure for all the lessons in the course and for the Final Quiz.

   To go to a specific lesson in the course: Completely REWIND the tape. Set the counter at 000. Push FAST FORWARD on the Program Recorder. Advance the tape until you reach the START LOADING number that you entered on the chart.

   To repeat a lesson just completed: REWIND the tape to the START AUDIO number on your chart.
Memory in a computer system is divided into two components, ROM and RAM:

**ROM.** Read-Only Memory contains programs permanently stored in your computer by the manufacturer. This information is essential for computer operation. The ROM programs are called read-only because they can only be read, and they cannot be altered by use of the keyboard.

**RAM.** Random Access Memory temporarily stores programs and data in your computer. You can enter information directly into RAM from the keyboard, which allows you to create new programs, or bring stored data into your RAM workspace from diskette, cassette, or cartridge.

Turning off the computer console clears all the information stored in RAM. To write new data without turning your console off and on, use the BASIC instruction, NEW followed by [RETURN]. A NEW command clears any information in RAM.

A CLOAD command brings a stored program from the cassette into RAM. To clear out this program and prepare for the practice sessions, you MUST enter the NEW command.

The following statement is repeated at the beginning of each practice session because of its importance:

**Note:** To clear RAM computer memory, type NEW and press [RETURN] before EVERY practice session and between each example program.
SUMMARY OF LESSONS: WRITING PROGRAMS ONE

Lesson One: Keyboard
Reviews the ATARI computer console keyboard.

Lesson Two: Printing on the Television Screen
Covers the PRINT command for words or math in the direct operating mode.
Includes the rules for procedural precedence in evaluating mathematical expressions.

Lesson Three: Characteristics of a Program
Explains BASIC program formatting and simple revisions. Gives initial exposure to the deferred operating mode by explaining a program RUN.

Lesson Four: Names and Numbers in Memory
Explains the manipulation of numeric and string variables.

Lesson Five: Computer Logic
Shows how to use IF/THEN and GOTO statements to allow for logical branching in the structure of a program.

Lesson Six: Library Function
Explains intrinsic functions available on the computer to get generation of random numbers, absolute or integer values, as well as square root computation.

Lesson Seven: FOR/NEXT Loops
Covers repetitive action using FOR/NEXT loops, the STEP instruction, and nesting of FOR/NEXT loops.

Final Quiz: Writing Programs One
A comprehensive review of the material covered in the course.

<table>
<thead>
<tr>
<th>WRITING PROGRAMS ONE</th>
<th>LESSON</th>
<th>CASSETTE RECORDER COUNTER VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>START LOADING</td>
</tr>
<tr>
<td>SIDE 1</td>
<td>1</td>
<td>000</td>
</tr>
<tr>
<td>SIDE 2</td>
<td>5</td>
<td>000</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FINAL TEST</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2  Program Recorder Counter Numbers for Lessons in Writing Programs One

Summary of Lessons: Writing Programs One 7
LESSON ONE: KEYBOARD

The first lesson is a quick review of the ATARI keyboard. It includes an explanation of the following important keys:

- Backspace key (DELETE BACK SPACE)
- Capital letter/lower case key (CAPS LOCK)
- Quotation mark key (")
- ATARI symbol key (for reverse letters)
- Break, control, escape keys (BREAK, CTRL, ESC)
- Math keys (+, -, *, *)

When you are ready, load Lesson Two on the cassette by typing CLOAD and pressing [RETURN] twice. After the READY message appears on the screen, type RUN and press [RETURN] to start the lesson.

LESSON TWO: PRINTING ON THE TELEVISION SCREEN

Frame 1: Summary of Lesson Two

1. PRINT causes a display on the screen of whatever text or numeric characters you have placed within quotes.

2. PRINT is always in capital letters. Material you put inside the quotes can be upper or lower case, alpha or numeric characters.

3. PRINT used without quotes performs mathematical functions according to the following rules:

   The computer performs the math computation by doing:

   - Multiplication and division inside parentheses first.
   - Addition and subtraction inside parentheses next.
   - Multiplication and division outside parentheses next.
   - Addition and subtraction last.

On each step, the computer moves from left to right across the expression.
Example: \((4/2+1) \times 3 + (2+2) - 12\)

Step 1: \((2+1) \times 3 + (2+2) - 12\)
Step 2: \(3 \times 3 + 4 - 12\)
Step 3: \(9 + 4 - 12\)
Step 4: 1

Computer prints: 1

Frame 2: Practice Session for Lesson Two

Note: To clear RAM computer memory, type NEW and press RETURN before EVERY practice session and between each example program.

Enter each line on the computer by typing the command. Press RETURN to see the result. Try to determine beforehand what will be displayed on the screen and see if you’re right.

PRINT “HELLO”
PRINT “MY NAME IS GEORGE”
PRINT “OH”
PRINT “Oh boy!”
print “OOPS”
PRINT “OOPS”
PRINT “@#$%&”
PRINT “THIS IS RIGHT”

Try this math. Enter each command and press RETURN.

PRINT “3−3”
PRINT “7*7”
PRINT 4*4
PRINT 10/2 + 4 − 5
PRINT (3*6)/(3*3)
PRINT 3*6/3*3
PRINT 25+(6/2-2)*8
PRINT 200/(3+2)*2-60
PRINT (6-2)*2*5
Now use the computer to solve these problems:

- Add three numbers and PRINT the result (pick the numbers).
- Compute the average of five numbers. (Add them and divide by 5. Remember the operational precedence of mathematical expressions when you enter this exercise into the computer. Placement of parentheses is emphasized in this case.)

When you are ready, load Lesson Three on the cassette by typing CLOAD and pressing RETURN twice. After the READY message appears on the screen, type RUN and press RETURN to start the lesson.

**LESSON THREE:**
CHARACTERISTICS OF A PROGRAM

**Frame 3: Entering a Program**

```
READY
10 PRINT "THIS IS THE BEGINNING"
20 PRINT "THIS IS THE MIDDLE"
30 PRINT "THE END"
RUN
THIS IS THE BEGINNING
THIS IS THE MIDDLE
THE END
READY
```

**Frame 4: Example of Executing Consecutive Line Numbers**

```
READY
20 PRINT "THIS IS THE MIDDLE"
10 PRINT "THE BEGINNING"
10 PRINT "THE END"
RUN
THIS IS THE BEGINNING
THIS IS THE MIDDLE
THE END
READY
```
Frame 5: Insertion of Program Line

READY
10 PRINT "WE DON'T"
20 PRINT "MAKE MISTAKES"
10 PRINT "USUALLY"
LIST
10 PRINT "WE DON'T"
20 PRINT "DO NOT"
20 PRINT "MAKE MISTAKES"

Frame 6: Deletion of Program Line

READY
10 PRINT "DON'T"
20 PRINT "GO"
10 PRINT "AWAY"
LIST
20 PRINT "GO"
30 PRINT "AWAY"
READY

Frame 7: Use of END Command

READY
15 PRINT "THE SUM OF 2 AND 2 IS"
20 PRINT 2+2
30 PRINT "THE SUM OF 5+10 IS"
40 PRINT 5+10
50 END
60 PRINT "HOW MUCH IS 3+4?"
RUN
THE SUM OF 2 AND 2 IS
4
THE SUM OF 5+10 IS
15
READY
Frame 8: Use of STOP Command

Frame 9: Summary of Lesson Three

1. A program is a series of logical steps for the computer. Each step gets a unique line number.

2. RUN tells the computer to execute the lines in consecutive numerical order.

3. LIST tells the computer to display all the program lines in consecutive numerical order.

4. NEW erases the program from computer memory.

5. END or STOP statements cause the computer to terminate program execution at the line number of the command.

6. To edit your program
   - Retype the line.
   - Add a new line with a line number that is numerically between other lines (25 placed between line numbers 20 and 30), or
   - Remove a line by typing its number and pressing [RETURN].

Frame 10: Practice Session for Lesson Three

Note: To clear RAM computer memory, type NEW and press [RETURN] before EVERY practice session and between each example program.

Enter and run each of these programs.
Use the core program above. Add the following additional program lines one at a time to the basic program. After you add each line, run the program to see the effect of the change.
When you are ready, load Lesson Four on the cassette by typing **CLOAD** and pressing **RETURN** twice. After the READY message appears on the screen, type **RUN** and press **RETURN** to start the lesson.
Frame 13

```
10 PRINT "WHAT'S THE PERIMETER OF A SQUARE WITH A SIDE OF 5?"
20 LET SIDE=5
30 LET PERIMETER=SIDE*4
40 PRINT PERIMETER
```

Frame 14

```
10 PRINT "HOW OLD ARE YOU?"
20 INPUT AGE
30 LET REAL=AGE/10
40 PRINT "I BET YOU'RE REALLY"
50 PRINT REAL
```

Frame 15

```
10 DIM WHO$[20], WHATS[20], WHERE[20]
20 LET WHO$="OLD MAN"
30 LET WHATS="LIVED"
40 LET WHERE="HOME"
50 PRINT "THERE WAS AN ";
60 PRINT WHO$;
70 PRINT WHATS;
80 PRINT WHERE;
90 PRINT "I M A ");
100 PRINT WHERE
```

Frame 16: Summary of Lesson Four

1. Variables identify memory locations in which the computer stores specific alpha or numeric characters, for example, the result of a computation or an assigned value.
2. Variable types determine the kind of labels required.
   • Numeric variables reference numbers ONLY.
   • String variables store any assigned sequence of alpha or numeric characters or their combination.

Each item of information is given a name, which will be used throughout the program to refer to that particular item. This name is called an identifier.

3. Specific rules for variable labels include:
   • Assign identifiers like A, B, C3, DE49 to numeric variables.
   • Assign identifiers like A$, B2$, ERROR$ to string variables.
   • DO NOT use reserved BASIC words as labels, such as: FOR, NEXT, DIM, LET, PRINT.

4. Commands for variable operation include:
   • LET stores numbers or letters into a variable.
     LET X=100
     LET X$= "SHIRLEY"
   • DIM signifies a maximum amount of characters allowed into a variable.
     DIM A$(10)
   • INPUT commands the computer to wait for entry of the number or word to be stored in the numeric or string variable.
     INPUT X
     INPUT X$ (remember to DIM X$ first)

Frame 17: Practice Session for Lesson Four

Note: To clear RAM computer memory, type NEW and press RETURN before EVERY practice session and between each example program.

Run each of these sample programs.
Add these lines to make an expanded program.

```
32 PRINT "SIDE=";
34 PRINT SIDE
36 PRINT "VOLUME=";
```

THEN change Line 20 to find the volume of other cubes.

The following program converts centimeters to inches.

```
10 INPUT CENT
20 LET INCH=0.394CENT
30 PRINT "CENTIMETERS="
40 PRINT CENT
50 PRINT "INCHES="
60 PRINT INCH
```

Now change Line 10 and run the program several times, typing in a different number each time.

```
10 INPUT CENT
```

Add this line and RUN AGAIN.

```
5 PRINT "HOW MANY CENTIMETERS"
```
LESSON FIVE: COMPUTER LOGIC

Type NEW and try this program.

```
REMAIN NAMES(20), MONTHS(20)
20 PRINT "WHO ARE YOU?"
30 INPUT NAME$
40 PRINT "WHAT MONTH WERE YOU BORN?"
50 INPUT MONTH$
60 PRINT "HELLO ":
70 PRINT NAME$
80 PRINT "THE NICEST PEOPLE ARE BORN IN"
90 PRINT MONTH$
```

When you are ready, load Lesson Five on the cassette. You must turn over the cassette, Writing Programs One, to Side 2. Refer to General Instructions for Loading the Cassette Tape. Remember to reset the Program Recorder counter to 000.

Frame 18

```
REMAIN "WHAT IS 3+3?"
20 INPUT ANSWER
30 INPUT RESPONSE
40 IF RESPONSE=ANSWER THEN PRINT "CORRECT"
50 END
RUN "WHAT IS 3+3?"
```

Frame 19

```
REMAIN "HOW OLD ARE YOU?"
20 INPUT AGE
30 IF AGE<=18 THEN PRINT "YOU CAN'T VOTE"
40 IF AGE>18 THEN PRINT "YOU CAN VOTE"
```
Frame 20

```
10 PRINT "WHOA ARE YOU"
20 INPUT NAMES
30 IF NAMES="GEORGE" THEN PRINT "HELLO"
40 IF NAMES="GEORGE" THEN PRINT "WHO?"
```

Frame 21

```
10 PRINT "HELLO"
20 GOTO 50
30 PRINT "GOODBYE"
40 END
50 PRINT "SO LONG!"
```

Frame 22

```
10 PRINT "WHAT'S IT?"
20 INPUT ANSWER
30 IF ANSWER<>2 THEN GOTO 60
40 PRINT "SUPER"
50 PRINT "SMART"
55 END
60 PRINT "WRONG, TRY AGAIN"
70 GOTO 10
```
Frame 23

Frame 24: Summary of Lesson Five

1. **IF/THEN** program statements provide for conditional logic sequencing. If a stipulated condition holds true, the computer performs the instruction present in the **THEN** command. If the condition proves false, the computer ignores the **THEN** portion of the statement and goes to the next program line.

2. Logical Operators are:
   - `=` means "equals"
   - `<>` means "not equal"
   - `>` means "greater than"
   - `<` means "less than"

   An **IF/THEN** statement comparison between two string variables will only hold true if the strings prove equal, character for character.

3. **GOTO** provides for an unconditional logic sequence. The computer moves to the specified line and continues processing.

4. Use the **BREAK** key to end computer execution of an infinite loop.

Frame 25: Practice Session for Lesson Five

**Note:** To clear RAM computer memory, type **NEW** and press **RETURN** before EVERY practice session and between each example program.
Change line 40 in the above program to GOTO 5. Why doesn't this work?

Run the following program and make three intentional mistakes to see how the loop ends.

```plaintext
READY
1 LET TRIES=1
2 DIM WORDS(20)
3 PRINT "WHAT'S A SYNONYM OF SMALL?"
4 INPUT WORDS
5 IF WORDS="LITTLE" THEN GOTO 00
6 IF TRIES=1 THEN GOTO 00
7 PRINT "WRONG. TRY ANOTHER WORD"
8 LET TRIES=TRIES+1
9 GOTO 50
10 PRINT "RIGHT.
11 PRINT ""LITTLE" IS THE WORD I MEANT"
```
LESSON SIX: LIBRARY FUNCTION

When you are ready, load Lesson Six on the cassette by typing CLOAD and pressing RETURN twice. After the READY message appears on the screen, type RUN and press RETURN to start the lesson.

Frame 26

```
10 PRINT "WHAT'S THE TEMPERATURE?"
20 INPUT X
30 IF X < -1 THEN GOTO 70
40 PRINT ABS(X)
50 PRINT "BELOW ZERO"
60 STOP
70 PRINT "HOW WARM!"
```

Frame 27

**LOG(X) = Logarithm**
**SIN(X) = Sine**
**COS(X) = Cosine**
**SGN(X) = Sign—plus or minus**

Frame 28: Summary of Lesson Six

1. Library functions perform their intended calculation based upon previously established internal programming. Therefore, the use of these functions allows you to "return" a number in the form you need for further computation in a calculation or storage in a variable.

2. Common library functions include:
   - **INT** rounds a number to its next lower whole number.
   - **ABS** gives you the value of a number without the + or - sign.
   - **RND** used with (0) gives a random number between 0 and 1.
   - **SQR** gives you the square root of a number.
   - **FRE** used with (0) tells how much free memory is available.
Frame 29: Practice Session for Lesson Six

Note: To clear RAM computer memory, type NEW and press RETURN before EVERY practice session and between each example program.

Here's a program that tells whether a number is odd or even.

```
10 PRINT "TYPE A WHOLE NUMBER:"
20 INPUT W
30 IF W/2 INT(W/2) THEN PRINT "EVEN"
40 IF W/2 INT(W/2) THEN PRINT "ODD"
50 END
```

Try this program with different inputs. Then BREAK it.

```
60 DIM NAMES(20)
10 PRINT "WHO ARE YOU?"
20 INPUT NAMES
30 X=RND(3)
40 Y=INT(Y)
50 IF Y=0 THEN PRINT "WHAT KIND OF NAME IS THIS?"
60 IF Y=1 THEN PRINT "WHAT IS YOUR NAME?"
70 IF Y=2 THEN PRINT "NICE NAME!"
75 PRINT
80 PRINT
90 GOTO 10
```

When you are ready, load Lesson Seven on the cassette by typing CLOAD and pressing RETURN twice. After the READY message appears on the screen, type RUN and press RETURN to start the lesson.
LESSON SEVEN: FOR/NEXT LOOPS

Frame 30

```
10 FOR J=1 TO 10
20 PRINT "HERE'S A NUMBER:";
30 PRINT J
40 PRINT "SQUARE OF THIS NUMBER IS:";
50 PRINT J**2
60 NEXT J
```

Program continues until loops are completed.

Frame 31

```
10 FOR NUMBER=5 TO 50 STEP 5
20 PRINT NUMBER
30 NEXT NUMBER
```

Frame 32

```
10 FOR X=10 TO 0 STEP -1
20 PRINT X
30 NEXT X
40 PRINT "BLAST-OFF!!!"
```
Frame 33

```
10 PRINT "SIMPLE RESERVATIONS SYSTEM"
15 PRINT "THIS PROGRAM WILL ALLOW YOU
to see"
15 PRINT "WHAT SEAT CAN YOU SELL NEXT:
"
20 FOR R=1 TO 5
30 PRINT "ROW"
35 PRINT R
40 FOR S=1 TO 10
50 PRINT "SEAT":
55 PRINT S
60 PRINT "TYPE 1 WHEN THIS SEAT IS SOLD"
0
70 INPUT S
80 NEXT S
90 NEXT R
100 PRINT "ALL SOLD OUT"
```

Frame 34: Summary of Lesson Seven

A FOR/NEXT counting loop repeats a program segment by executing those statements contained within the loop a given number of times.

Set a dummy variable X at 1 and end the loop after a specified number of increments.

```
10 FOR X=1 TO 10
    ...
50 NEXT X
```

In the program above, line statement 50 tells the computer to return to the FOR statement line and process back down to line 50. Each time the computer encounters this program line, X increments by 1 and returns to the beginning of the FOR/NEXT loop until all program lines contained between the FOR/NEXT statements are executed. The "loop" process continues until the desired number of times is reached. At that point, execution of the program advances to the next consecutive numeric line statement.

Frame 35: Practice Session for Lesson Seven

**Note:** To clear RAM computer memory, type NEW and press RETURN before EVERY practice session and between each example program.

Run the following program.
Change line 10 to get more loops.

The following program generates ten math problems.

```
READY
10 FOR PROBLEM=1 TO 10
20 X=INT(10*RND(0))
30 Y=INT(10*RND(0))
40 PRINT "WHAT IS" X , "+" , Y ;
45 PRINT "= ";
50 INPUT ANSWER;
60 IF ANSWER<Y THEN PRINT "RIGHT"
70 IF ANSWER<X THEN PRINT "WRONG"
80 NEXT PROBLEM
```

Change the program so you get problems with larger numbers.

The next program prints a line of dots.

```
READY
10 FOR X=1 TO 10
20 PRINT "*";
30 NEXT X
```
After running the program, add these lines to make a “nested” loop and run it again.

15 FOR Y=1 TO X
25 NEXT Y
26 PRINT

Then change this line.

10 FOR X=10 TO 1 STEP -1

Add these lines.

11 Z=37-X
12 FOR A=1 TO Z
13 PRINT " ";
14 NEXT A

Change the program to invert the triangle but leave it at the right.

CONGRATULATIONS! GO ON TO THE FINAL QUIZ ON THE CASSETTE.

Take the Final Quiz for Writing Programs One. The program will automatically calculate your scores after you complete the quiz. Load the Final Quiz in the same manner in which you loaded the other lessons. Proceed to the next section of this workbook by loading your cassette, Writing Programs Two, using the procedure outlined in General Instructions for Loading the Cassette Tape.
SUMMARY OF LESSONS: WRITING PROGRAMS TWO

Lesson One: Screen Formatting Instructions
Shows method for clearing screen and demonstrates how to position the cursor and print in columns.

Lesson Two: READ/DATA
Shows how to assign values to different variables through the use of READ and DATA statements.

Lesson Three: Arrays
Shows how to set up single-dimension arrays and how to use them in a program.

Lesson Four: PEEK and POKE
Explains computer memory and how to access it directly by means of PEEK and POKE statements.

Lesson Five: ATASCII Code and Instructions
Explains ASCII code as it relates to the ATARI Personal Computer System, and how to use the CHR$ and ASC functions.

Lesson Six: String Handling
Demonstrates how to examine the contents of string variables and how to use parts of the contents in logic statements.

Lesson Seven: Multiple Statements
Shows how to write lines with more than one statement. Explains logic flow when multiple statement lines are used with IF/THEN statements.

Lesson Eight: Subroutines
Explains the concept of subroutines and demonstrates the use of GOSUB and RETURN.

Final Quiz: Writing Programs Two
A review of the material covered in Writing Programs Two. No voice track is used with this part of the program.
<table>
<thead>
<tr>
<th>WRITING PROGRAMS TWO</th>
<th>LESSON</th>
<th>CASSETTE RECORDER COUNTER VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>START LOADING</td>
</tr>
<tr>
<td>SIDE 1</td>
<td>1</td>
<td>000</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>SIDE 2</td>
<td>5</td>
<td>000</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>FINAL TEST</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3  Program Recorder Counter Numbers for Lessons in Writing Programs Two
LESSON ONE: SCREEN FORMATTING INSTRUCTIONS

Frame 1: Summary of Lesson One

1. **CURSOR**: Wherever the cursor appears on the screen, the computer will accept input and display the character received from the keyboard.

2. **GRAPHICS 0**: Clears the screen of text and sets the background color to blue. The cursor will automatically move to the top and extreme left of the screen, commonly called the "home" position.

3. **SEMICOLON**: Use of the semicolon in a PRINT statement keeps the output from moving down to the next line when text and numbers are printed on the screen.

   Example:
   
   ```
   10 DIM NAME$(20)
   20 NAME$ = "JOE"
   30 PRINT "MY NAME IS "; NAME$;
   
   MY NAME IS JOE.
   ```

4. **COMMA**: Working in a fashion similar to the semicolon, the comma places the printed text or numbers into zones on the screen.

   Example:
   
   ```
   PRINT "ONE","TWO","THREE"
   PRINT 1,2,3
   
   ONE    TWO    THREE
   1      2      3
   ```

Frame 2: Practice Session for Lesson One

**Note**: To clear RAM computer memory, type NEW and press **RETURN** before EVERY practice session and between each example program.
The following program illustrates the use of the comma and the semicolon.

```
READY  
5 GRAPHICS 0
6 PRINT
10 PRINT "LINE"
20 PRINT "SPACE"
30 PRINT
40 PRINT "LINE"
50 PRINT "SPACE"
60 PRINT
70 PRINT "LINE", "SPACE"
80 END
```

Now run this program:

```
READY  
5 GRAPHICS 0
6 PRINT
10 PRINT "TYPE IN TWO NUMBERS"
20 PRINT "FIRST NUMBER =";
25 INPUT A
30 PRINT "SECON NUMBER =";
35 INPUT B
40 LET C = A + B
50 PRINT "A + B = " 
60 PRINT A + B
70 PRINT "A/B = " 
75 PRINT A / B
80 GOTO 10
```

This program illustrates the use of the comma to make columns.

```
READY  
5 GRAPHICS 0
10 PRINT
20 PRINT "NUMBER", "SQUARE", "SQ. ROOT"
25 PRINT
30 FOR NUMBER = 1 TO 10
35 LET SQUARE = NUMBER
40 LET ROOT = SQRT(NUMBER)
50 PRINT NUMBER, SQUARE, ROOT
60 NEXT NUMBER
70 NEXT NUMBER
80 END
```
The next program illustrates how text can be strung together with semicolons. To view parts of the program LIST as follows:

LIST 10,50

This will list all lines between 10 and 50.

When you are ready, load Lesson Two on the cassette by typing CLOAD and pressing RETURN twice. After the READY message appears on the screen, type RUN and press RETURN to start the lesson.

LESSON TWO: READ/DATA

Frame 3

The READ and DATA statements contained in the following program illustrate the method of execution used by the computer. When a READ statement is encountered, the first DATA statement is scanned by the computer. The assigned variable receives the initial value contained in the DATA statement. This value will remain constant until a new READ instruction is encountered.
Frame 4

```
READY
5: REM READ ... DATA WITH WORDS
10 DIM WORDS(D)
20 READ WORDS
30 PRINT WORDS
40 GOTO 20
1000 DO 16 UP, DOWN, IN, OUT
```

Frame 5: Practice Session for Lesson Two

**Note:** To clear RAM computer memory, type `NEW` and press **RETURN** before EVERY practice session and between each example program.

```plaintext
10 READ A
20 PRINT A
30 GOTO 10
40 DATA 5, 6, 7, 8
```

Run the above program and then add the following line to it.

```plaintext
25 IF A=8 THEN RESTORE
```

Add the following lines:

```plaintext
5 LET B=0
26 LET B=B+1
27 IF B=8 THEN STOP
```

Add the following line.

```plaintext
6 DATA 1, 2, 3, 4
```

Add the following lines.

```plaintext
10 READ X, Y
20 PRINT X, Y
```

And delete 35, 36, and 37.
Run the following program:

```
10 DIM ARRAY(10)
20 FOR X = 1 TO 10
30 LET ARRAY(X) = 25.2
40 NEXT X
```

When you are ready, load Lesson Three on the cassette by typing **CLOAD** and pressing **RETURN** twice. After the READY message appears on the screen, type **RUN** and press **RETURN** to start the lesson.

**Frame 6: Summary of Lesson Three**

1. Arrays are sets of variables with the same name. Each array variable has the name (in capitals) followed by a number in parentheses. If only one number is assigned, the array is one dimensional. Visualize the elements in the array being placed in one long column. For maximum speed and efficiency of operation, dimension your arrays into rows and columns. The first number within the parentheses indicates the assigned row, and the second number indicates the assigned column in a two-dimensional array. Carry this concept into n-dimensions by simply adding rows and columns.

   Examples of one-dimensional arrays:
   - A(10), VAR(5) or VAR(X) or ARRAY(NUM)

   Examples of two-dimensional arrays:
   - A(10,50), VAR(5,5) or VAR(X,Y) or ARRAY(ROW,COLUMN)

2. Arrays must first be set up in **ATARI BASIC** with a **DIM** (dimension) statement.

   ```
   DIM VAR(10)
   ```

3. Values can be placed into and removed from array variables with simple **FOR/NEXT** loops.

   ```
   10 DIM ARRAY(10)
   20 FOR X = 1 TO 10
   30 LET ARRAY(X) = 25.2
   40 NEXT X
   ```
4. The LET instruction in the above program, Statement 30, places a value in the array variable. When READ/DATA statements and INPUT statements are used to place values into array variables, the computer must use these instructions indirectly. For example,

```
READ N
LET A(X) = N, etc.
```

Frame 7: Practice Session for Lesson Three

Note: To clear RAM computer memory, type NEW and press RETURN before every practice session and between each example program.

Run the following program:

```
10 DIM A(20)
20 FOR X = 1 TO 20
30 PRINT A(X)
40 NEXT X
```

When setting up an array it is necessary to store values in each variable in the array, as in the following example. Run the next program, and then check what is in each variable with a PRINT instruction, such as PRINT A(3).

```
20 DIM A(10)
25 PRINT "WE MAKE THE ARRAY"
30 FOR X = 1 TO 20
40 LET A(X) = INT(RAND(1))
50 NEXT X
```
Add the following lines to the previous program and run it again:

```
REM 4+Y
20 GRAPHICS 0
30 PRINT "JOGGING CALENDER"
40 PRINT "SHORTENED TO 7 DAYS"
50 PRINT
60 DIM DISTANCE(7)
70 FOR DAY=1 TO 7
80 PRINT "HOW MANY MILES DID YOU JOG ON DAY ";DAY
90 INPUT HLES
100 DISTANCE(DAY)=HLES
110 NEXT DAY
120 PRINT "RESULTS FOR WHAT DAY?"
130 PRINT "ON THAT DAY YOU WENT " ;DISTANCE(DAY) ;" MILES"
140 GOTO 80
```

Run this sample program:

```
REM 4+Y
20 GRAPHICS 0
30 PRINT "THIS IS THE RESERVATION SYSTEM"
40 PRINT "FLY-BY-NIGHT AIRWAYS"
50 PRINT "YOU CAN RESERVE FLIGHTS 1-7."
60 FOR M=1 TO 7
70 LET FLIGHT(M)=4
80 NEXT M
90 PRINT "WHAT FLIGHT (1 THRU 7) DO YOU WANT?"
100 INPUT N
110 IF Flight(M)>=FLIGHT(M) THEN PRINT "SORRY, ALL FLIGHTS TAKEN; TRY ANOTHER."
120 IF FLIGHT(M)>0 THEN FLIGHT(M)=FLIGHT(M)-1
130 IF FLIGHT(M)=0 THEN PRINT "ISSUE TICKET FOR FLIGHT ";N
140 GOTO 90
```

Run this sample program:
When you are ready, load Lesson Four on the cassette by typing CLOAD and pressing RETURN twice. After the READY message appears on the screen, type RUN and press RETURN to start the lesson.

LESSON FOUR: PEEK and POKE

Frame 8: Computer Memory

The computer’s raw memory is like a set of post office boxes, where each box contains just one number ranging from 0 to 255. Each box or location is given an address.

An 8K computer has about 8000 different memory locations that can be used for program storage.

Each number in memory can serve to represent a number, a letter, or a computer instruction. For example, the number 76 could stand for the letter L, or it could act like a GOTO instruction for the computer’s machine language. What the number represents exactly depends on where it is in memory.

Frame 9: POKE

POKE 54018, 52 Memory address is 54018
Instruction to turn on Program Recorder is 52.

POKE 54018, 60 Instruction to turn off Program Recorder is 60.

Frame 10: PEEK

PRINT PEEK (800) prints the number stored in memory location 800.

X = PEEK(800) places the number stored in memory location 800 into the variable X.

Frame 11: Practice Session for Lesson Four

Note: To clear RAM computer memory, type NEW and press RETURN before EVERY practice session and between each example program.

Try this program to see the numbers stored in different memory locations:
Use **POKE** instructions with caution. No damage results to the computer physically, but you can crash your program! Try this **POKE** to see a small example of what we mean:

**POKE 755,4**

To correct the “damage,” type in **POKE 755,2**.

Memory location 20 is used by the computer to count the number of frames the computer puts on your television set. The number in 20 starts at 0 and 1 is added every 1/60th of a second. When the number hits 255, the contents start again at 0.

You can POKE 20 and put the number 0 in it to start. It will then count 1/60ths of a second, as in the following program:

```
10 READY
20 PRINT "START COUNTING"
30 PRINT
40 POKE 20,0
50 PRINT "NOW FOR 4 SECONDS OF MUSIC...
60 POKE 54010,60
70 END
```

The next program turns on your Program Recorder and allows it to run for 4 seconds. The Program Recorder turns off automatically.

```
3 DIM A$(12)
4 PRINT "PUT IN MUSIC CASSETTE OF YOUR CHOICE"
5 PRINT "AND PRESS 'RETURN'
6 PRINT "(DON'T USE THE TEACHING CASSETTE)"
7 INPUT A$(12)
8 POKE 54010,52
9 PRINT "NOW FOR 4 SECONDS OF MUSIC...
10 PRINT "START COUNTING"
11 COUNT=PEEK(20)/60
12 IF COUNT<4 THEN GOTO 10
13 PRINT "4 SECONDS ARE UP"
14 POKE 54010,60
15 END
```

When you are ready, load Lesson Five on the cassette. You must turn over the cassette, Writing Programs Two, to Side 2. Refer to General Instructions for Loading the Cassette Tape. Remember to reset the Program Recorder counter to 000.
Frame 12: Some ATASCII Characters and Their Code Numbers

<table>
<thead>
<tr>
<th>ASCII CODE NUMBER</th>
<th>CHARACTER</th>
<th>ASCII CODE NUMBER</th>
<th>CHARACTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>(space)</td>
<td>81</td>
<td>Q</td>
</tr>
<tr>
<td>33</td>
<td>!</td>
<td>82</td>
<td>R</td>
</tr>
<tr>
<td>34</td>
<td>&quot;</td>
<td>83</td>
<td>S</td>
</tr>
<tr>
<td>35</td>
<td>#</td>
<td>84</td>
<td>T</td>
</tr>
<tr>
<td>36</td>
<td>$</td>
<td>85</td>
<td>U</td>
</tr>
<tr>
<td>37</td>
<td>%</td>
<td>86</td>
<td>V</td>
</tr>
<tr>
<td>38</td>
<td>&amp;</td>
<td>87</td>
<td>W</td>
</tr>
<tr>
<td>39</td>
<td>'</td>
<td>88</td>
<td>X</td>
</tr>
<tr>
<td>40</td>
<td>(</td>
<td>89</td>
<td>Y</td>
</tr>
<tr>
<td>41</td>
<td>)</td>
<td>90</td>
<td>Z</td>
</tr>
<tr>
<td>42</td>
<td>*</td>
<td>91</td>
<td>[</td>
</tr>
<tr>
<td>43</td>
<td>+</td>
<td>92</td>
<td>\</td>
</tr>
<tr>
<td>44</td>
<td>,</td>
<td>93</td>
<td>]</td>
</tr>
<tr>
<td>45</td>
<td>-</td>
<td>94</td>
<td>a</td>
</tr>
<tr>
<td>46</td>
<td>.</td>
<td>95</td>
<td>b</td>
</tr>
<tr>
<td>47</td>
<td>/</td>
<td>96</td>
<td>c</td>
</tr>
<tr>
<td>48</td>
<td>0</td>
<td>97</td>
<td>d</td>
</tr>
<tr>
<td>49</td>
<td>1</td>
<td>98</td>
<td>e</td>
</tr>
<tr>
<td>50</td>
<td>2</td>
<td>99</td>
<td>f</td>
</tr>
<tr>
<td>51</td>
<td>3</td>
<td>100</td>
<td>g</td>
</tr>
<tr>
<td>52</td>
<td>4</td>
<td>101</td>
<td>h</td>
</tr>
<tr>
<td>53</td>
<td>5</td>
<td>102</td>
<td>i</td>
</tr>
<tr>
<td>54</td>
<td>6</td>
<td>103</td>
<td>j</td>
</tr>
<tr>
<td>55</td>
<td>7</td>
<td>104</td>
<td>k</td>
</tr>
<tr>
<td>56</td>
<td>8</td>
<td>105</td>
<td>l</td>
</tr>
<tr>
<td>57</td>
<td>9</td>
<td>106</td>
<td>m</td>
</tr>
<tr>
<td>58</td>
<td>:</td>
<td>107</td>
<td>n</td>
</tr>
<tr>
<td>59</td>
<td>;</td>
<td>108</td>
<td>o</td>
</tr>
<tr>
<td>60</td>
<td>&lt;</td>
<td>109</td>
<td>p</td>
</tr>
<tr>
<td>61</td>
<td>=</td>
<td>110</td>
<td>q</td>
</tr>
<tr>
<td>62</td>
<td>&gt;</td>
<td>111</td>
<td>r</td>
</tr>
<tr>
<td>63</td>
<td>?</td>
<td>112</td>
<td>s</td>
</tr>
<tr>
<td>64</td>
<td>@</td>
<td>113</td>
<td>t</td>
</tr>
<tr>
<td>65</td>
<td>A</td>
<td>114</td>
<td>u</td>
</tr>
<tr>
<td>66</td>
<td>B</td>
<td>115</td>
<td>v</td>
</tr>
<tr>
<td>67</td>
<td>C</td>
<td>116</td>
<td>w</td>
</tr>
<tr>
<td>68</td>
<td>D</td>
<td>117</td>
<td>x</td>
</tr>
<tr>
<td>69</td>
<td>E</td>
<td>118</td>
<td>y</td>
</tr>
<tr>
<td>70</td>
<td>F</td>
<td>119</td>
<td>z</td>
</tr>
<tr>
<td>71</td>
<td>G</td>
<td>120</td>
<td></td>
</tr>
<tr>
<td>72</td>
<td>H</td>
<td>121</td>
<td></td>
</tr>
<tr>
<td>73</td>
<td>I</td>
<td>122</td>
<td></td>
</tr>
<tr>
<td>74</td>
<td>J</td>
<td>123</td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>K</td>
<td>124</td>
<td></td>
</tr>
<tr>
<td>76</td>
<td>L</td>
<td>125</td>
<td></td>
</tr>
<tr>
<td>77</td>
<td>M</td>
<td></td>
<td></td>
</tr>
<tr>
<td>78</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>79</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>P</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(This is a sample of the 255 possible characters of ATASCII)
Frame 13: Interesting ATASCII Characters

PRINT CHR$(125)  Clears television screen
PRINT CHR$(253)  Produces a buzzing sound
PRINT CHR$(28)   Moves cursor up one line (without erasing anything)
PRINT CHR$(29)   Moves cursor down one line
PRINT CHR$(30)   Moves cursor over one space to the left
PRINT CHR$(31)   Moves cursor over one space to the right

Frame 14: Summary of ASC and CHR$

CHR$(N) will produce the character of the ATASCII number.
ASC(L$) will produce the ATASCII number of the character stored in L$.
PRINT CHR$(65) will print the letter A from the ATASCII code number 65.
PRINT ASC("A") will print the ATASCII number 65.

Frame 15: Practice Session for Lesson Five

Note: To clear RAM computer memory, type NEW and press RETURN before every practice session and between each example program.

The following program will print the whole range of ATASCII characters. Note what happens when CHR$(125) is printed.

```
10 FOR NUMBER=0 TO 255
20 PRINT CHR$(NUMBER)
30 NEXT NUMBER
```
Add the following lines to keep the screen from being erased.

```
6 PRINT "ATASCII CHARACTERS"
20 IF NUMBER(125) THEN PRINT CHR$(125)
```

This program will show you the ATASCII code for certain letters and symbols.
(Don't use any "invisible" characters.)

```
6 GRAPHICS 0
10 DIM LETTER$(125)
20 PRINT "TYPE IN CHARACTER OR SYMBOL"
50 INPUT LETTERS
40 PRINT "THE ATASCII CODE FOR "+LETTERS
50 PRINT ASC(LETTERS)
60 GOTO 20
```

Press the "BREAK" key to end program execution. Add this line to the above program and run it again.

```
50 PRINT CHR$(253)
```
LESSON SIX:
STRING HANDLING

Frame 16

The following instruction will tell the computer to look at the string stored in WORD$ starting with the fifth character:

\[ \text{WORD$(5)} \]

We can print or store the result in another string variable:

\[ \text{PRINT WORD$(5)} \quad \text{PART$} = \text{WORD$(5)} \]

Any number can be used in the parentheses just so long as it does not exceed the length of the string stored in the variable.

Example:

\[ \text{STR$} = \text{"A SHORT PHRASE"} \]
\[ \text{P$} = \text{STR$(8)} \]
\[ \text{PRINT P$} \]
\[ \text{PHRASE} \]

Frame 17

The following statement will tell the computer to look at the string stored in WORD$ between two numbers:

\[ \text{WORD$(5,8)} \]

When you are ready, load Lesson Six on the cassette by typing CLOAD and pressing [RETURN] twice. After the READY message appears on the screen, type RUN and press [RETURN] to start the lesson.
PRINT WORD$(5,8) will now print the string from the fifth to the eighth character.

Example:

\[
\begin{align*}
\text{STR$} &= \text{"A SHORT PHRASE"} \\
\text{P$} &= \text{STR$(3,7)$} \\
\text{PRINT P$} \\
\text{SHORT}
\end{align*}
\]

Frame 18

The following program demonstrates how to get one character from a string that has been input into a string variable. In this case the program looks at the first character only.

```
10 DIM ANSWERS$(5)
20 GRAPHICS 0
30 PRINT
40 PRINT
50 PRINT "DO YOU LIKE ICE CREAM?"
60 INPUT ANSWERS
70 IF ANSWERS$(1,1): = "Y" THEN PRINT "SO"
80 ELSE
90 IF ANSWERS$(1,1): = "N" THEN PRINT "HOW"
100 GOMON"
110 END
```

Frame 19: Practice Session for Lesson Six

Note: To clear RAM computer memory, type NEW and press RETURN before every practice session and between each example program.

Enter the following instructions as direct commands:

```
DIM WORD$(40)
LET WORD$ = "A SHORT PHRASE"
```

Now experiment with printing out pieces of the words in WORD$ by using the following kinds of instructions:

```
PRINT WORD$(3) or PRINT WORD$(5,6) etc.
```
After trying this a few times, run the following program:

```
= READY
5 DIM WORDS(40)
10 GRAPHICS 6
20 PRINT
30 PRINT "TYPE IN WORD"
40 INPUT WORDS
50 PRINT WORDS(""); PRINT "CONTAINS "; LEN(WORDS)"
60 PRINT
70 GOTO 20
```

Now add the following lines. (Watch the quotation marks and the semicolons!)

```
= READY
70 FOR N=1 TO LEN(WORDS)
80 PRINT "WORDS(";N;")"=";WORD$(N)
90 NEXT N
20 PRINT
```

Try this program:

```
= READY
5 DIM WORDS(40)
10 GRAPHICS 6
20 PRINT
30 PRINT "TYPE IN WORD"
40 INPUT WORDS
60 PRINT "HERE IS THE WORD VERTICALLY-
```

```
The next program is an analysis program. It looks at a sentence for specific words:

```
10 GRAPHICS 0
20 PRINT
30 PRINT "THIS IS A VERY SHORT PROGRAM"
40 END
```

When you are ready, load Lesson Seven on the cassette by typing LOAD and pressing RETURN twice. After the READY message appears on the screen, type RUN and press RETURN to start the lesson.

### Frame 20

Separate statements can be placed on the same line in a BASIC program if they are separated with a colon (:\n
```
10 GRAPHICS 0:
20 PRINT: PRINT "THIS IS A VERY SHORT PROGRAM": END
```

### Frame 21

The IF/THEN condition determines whether statements that follow on the same line will be executed.

```
10 INPUT A
20 IF A=0 THEN PRINT "TRY AGAIN": GOTO 10
30 PRINT "THAT'S ALL": END
```

In the above program, the computer will reach the GOTO 10 statement only when A=0. If A > 0, then the computer ignores everything after the IF/THEN statement on the same line.
Frame 22: REM (or Remark) Statement

The REM statement is a label only. The computer will ignore it. Here the REM statement has been added to the program used previously:

```
10 REM THE NEXT STATEMENT WILL BE IGNORED: GRAPHICS 0
```

The computer will also ignore any statement that follows a REM statement on the same line.

```
10 REM THE NEXT STATEMENT WILL BE IGNORED: GRAPHICS 0
```

Be very careful with the following in multistatement lines:

```
REM
DATA
IF/THEN
GOTO
```

It is safest to have each DATA statement on its own line with no other statements.

Frame 23: Practice Session for Lesson Seven

Note: To clear RAM computer memory, type NEW and press RETURN before EVERY practice session and between each example program.
The following program appeared in an earlier lesson. We have added line 110 to the program so that you can see how much memory the program used. Run the program, and check how much memory you have left.

Now rewrite the above program using multiple statement lines, and see how much memory space the new version occupies.

Rewrite the program below using multiple statement lines. See if you can get rid of two of the IF/THEN statements.

When you are ready, load Lesson Eight on the cassette by typing CLOAD and pressing RETURN twice. After the READY message appears on the screen, type RUN and press RETURN to start the lesson.

**LESSON EIGHT: SUBROUTINES**

**Frame 24: Purpose of a Subroutine**

In this program, we use the same routine several times, so we moved it to the end. Now it's called a subroutine.
10 ...
20 ...
30 IF A=1 THEN PRINT "YES"
40 ...
50 ...
60 IF A=1 THEN PRINT "YES"
70 ...
80 END

1000 IF A=1 THEN PRINT "YES" Subroutine written just once.

Frame 25: GOSUB and RETURN Statements

When the computer reaches the GOSUB statement in Line 40, it goes to Line 1000.

Here is how the RETURN statement works:

10 ...
20 ...
30 ...
40 GOSUB 1000
45 ...
50 ...
55 ...
60 ...
70 ...
80 END

1000 PRINT
1010 PRINT "TYPE 0 AND PRESS THE RETURN KEY"
1020 INPUT A
1030 PRINT

1040 RETURN
Frame 26: Question Example

```
10 REM SIMPLE SUBROUTINE PROGRAM
20 PRINT "FIRST TIME THRU SUBROUTINE:";
30 GOSUB 1000
40 PRINT "SECOND TIME THRU:";
50 GOSUB 1000
60 PRINT "THIRD TIME THRU:";
70 GOSUB 1000
80 STOP
1000 PRINT 1;2;3;4;5
1010 PRINT
1020 RETURN
```

Frame 27: Practice Session for Lesson Eight

Note: To clear RAM computer memory, type NEW and press [RETURN] before every practice session and between each example program.

The following program has three subroutines. Run it and see what happens.

```
10 REM SHORT PROGRAM WITH SUBROUTINES
20 GOSUB 100
30 GOSUB 200
40 GOSUB 100
50 GOSUB 200
60 GOSUB 100
70 END
100 PRINT "*********";
200 PRINT "*********";
300 PRINT "*********"
```

The following frame is another version of the program above. However, the GOSUB statements use variables instead of numbers.
The following program illustrates many of the points taught in the last two lessons.

CONGRATULATIONS! GO ON TO THE FINAL QUIZ ON THE CASSETTE.
You should be ready now for the Final Quiz for Writing Programs Two. Your score will automatically appear on the screen at the conclusion of the quiz. To load the Final Quiz, use the same procedure you used to load the other lessons.

Frame 28: Program to Use With Final Quiz

```
10 REM CALENDAR INFORMATION
20 DIM MOS(12), DAS(12)
30 FOR X=1 TO 12
40 SUB 200
50 NEXT X
60 FOR I=1 TO 15
70 REM PRINT OUT CALENDAR SUBROUTINE
80 READ MOS, D.AB
90 D.A=I
100 GRAPHIC: 0
110 PRINT "MONTH ""TOTAL DAYS"
120 PRINT MOS, DD, DD
130 NEXT I
140 IF I=38 THEN A=4:B=3
150 NEXT I
160 PRINT "TOTAL"
170 IF I=40 THEN A=4:B=3
180 NEXT I
190 PRINT "TOTAL"
200 PRINT "TOTAL"
210 PRINT "TOTAL"
220 PRINT "TOTAL"
230 PRINT "TOTAL"
240 PRINT "TOTAL"
250 PRINT "TOTAL"
260 PRINT "TOTAL"
270 PRINT "TOTAL"
280 PRINT "TOTAL"
290 PRINT "TOTAL"
300 PRINT "TOTAL"
310 PRINT "TOTAL"
320 PRINT "TOTAL"
330 PRINT "TOTAL"
340 PRINT "TOTAL"
350 PRINT "TOTAL"
360 PRINT "TOTAL"
370 PRINT "TOTAL"
380 PRINT "TOTAL"
390 PRINT "TOTAL"
400 PRINT "TOTAL"
410 PRINT "TOTAL"
420 PRINT "TOTAL"
430 PRINT "TOTAL"
440 PRINT "TOTAL"
450 PRINT "TOTAL"
460 PRINT "TOTAL"
470 PRINT "TOTAL"
480 PRINT "TOTAL"
490 PRINT "TOTAL"
500 PRINT "TOTAL"
510 PRINT "TOTAL"
520 PRINT "TOTAL"
530 PRINT "TOTAL"
540 PRINT "TOTAL"
550 PRINT "TOTAL"
560 PRINT "TOTAL"
570 PRINT "TOTAL"
580 PRINT "TOTAL"
590 PRINT "TOTAL"
600 PRINT "TOTAL"
610 PRINT "TOTAL"
620 PRINT "TOTAL"
630 PRINT "TOTAL"
640 PRINT "TOTAL"
650 PRINT "TOTAL"
660 PRINT "TOTAL"
670 PRINT "TOTAL"
680 PRINT "TOTAL"
690 PRINT "TOTAL"
700 PRINT "TOTAL"
710 PRINT "TOTAL"
720 PRINT "TOTAL"
730 PRINT "TOTAL"
740 PRINT "TOTAL"
750 PRINT "TOTAL"
760 PRINT "TOTAL"
770 PRINT "TOTAL"
780 PRINT "TOTAL"
790 PRINT "TOTAL"
800 PRINT "TOTAL"
810 PRINT "TOTAL"
820 PRINT "TOTAL"
830 PRINT "TOTAL"
840 PRINT "TOTAL"
850 PRINT "TOTAL"
860 PRINT "TOTAL"
870 PRINT "TOTAL"
880 PRINT "TOTAL"
890 PRINT "TOTAL"
900 PRINT "TOTAL"
910 PRINT "TOTAL"
920 PRINT "TOTAL"
930 PRINT "TOTAL"
940 PRINT "TOTAL"
950 PRINT "TOTAL"
960 PRINT "TOTAL"
970 PRINT "TOTAL"
980 PRINT "TOTAL"
990 PRINT "TOTAL"
```

Now that you have successfully completed this course, you have acquired the necessary skills for AN INVITATION TO PROGRAMMING™ 3. Course content includes BASIC programming of sound and graphics.
LIMITED 90 DAY WARRANTY
ON ATARI® PERSONAL COMPUTER PRODUCTS

ATARI, INC. ("ATARI") warrants to the original consumer purchaser that this ATARI Personal Computer Product (not including computer programs) shall be free from any defects in material or workmanship for a period of 90 days from the date of purchase. If any such defect is discovered within the warranty period, ATARI's sole obligation will be to repair or replace, at its election, the Computer Product free of charge on receipt of the unit (charges prepaid, if mailed or shipped) with proof of date of purchase satisfactory to ATARI at any authorized ATARI Service Center. For the location of an authorized ATARI Service Center nearest you, call toll-free:

In California (800) 672-1430
Continental U.S. (800) 538-8547
or write to: Atari, Inc.
Customer Service Department
1340 Bordeaux Drive
Sunnyvale, CA 94086

YOU MUST RETURN DEFECTIVE COMPUTER PRODUCTS TO AN AUTHORIZED ATARI SERVICE CENTER FOR IN-WARRANTY REPAIR.

This warranty shall not apply if the Computer Product: (i) has been misused or shows signs of excessive wear, (ii) has been damaged by being used with any products not supplied by ATARI, or (iii) has been damaged by being serviced or modified by anyone other than an authorized ATARI Service Center.

ANY APPLICABLE IMPLIED WARRANTIES, INCLUDING WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE HEREBY LIMITED TO NINETY DAYS FROM THE DATE OF PURCHASE. CONSEQUENTIAL OR INCIDENTAL DAMAGES RESULTING FROM A BREACH OF ANY APPLICABLE EXPRESS OR IMPLIED WARRANTIES ARE HEREBY EXCLUDED. Some states do not allow limitations on how long an implied warranty lasts or do not allow the exclusion or limitation of incidental or consequential damages, so the above limitations or exclusions may not apply to you.

This warranty gives you specific legal rights and you may also have other rights which vary from state to state.

DISCLAIMER OF WARRANTY
ON ATARI COMPUTER PROGRAMS

All ATARI computer programs are distributed on an "as is" basis without warranty of any kind. The entire risk as to the quality and performance of such programs is with the purchaser. Should the programs prove defective following their purchase, the purchaser and not the manufacturer, distributor, or retailer assumes the entire cost of all necessary servicing or repair.

ATARI shall have no liability or responsibility to a purchaser, customer, or any other person or entity with respect to any liability, loss, or damage caused directly or indirectly by computer programs sold by ATARI. This disclaimer includes but is not limited to any interruption of service, loss of business or anticipatory profits or consequential damages resulting from the use or operation of such computer programs.

REPAIR SERVICE

If your ATARI Personal Computer Product requires repair other than under warranty, please contact your local authorized ATARI Service Center for repair information.

IMPORTANT: If you ship your ATARI Personal Computer Product, package it securely and ship it, charges prepaid and insured, by parcel post or United Parcel Service.