



Errata to the Applesoft Tutorial

Introduction

The information in this errata is a supplement to The Applesoft Tutorial. Before you use the manual, it would be helpful to go through it and mark the pages listed below. Then, as you do the Tutorial exercises, you will know when to refer to this document to get more information. You will be especially glad you did this when you run into some program errors that occur in the Tutorial!

The Tutorial pages affected by this document are: 20, 23-24, 29, 37, 59, 60, 63, 65, 69, 71, 78-79, 85-90, 93, 95, 96, 105, 109-110, 130.

Chapter 2

Page 20

Playing LITTLE BRICK OUT

This page and others in the Tutorial talk about game controllers (paddles), and some of the programs use them. If you don't have game controllers, you can still learn to use the concepts presented by reading those sections. Try it!

If you want some game controllers, contact your dealer. The game controllers are available as Apple Product Number A2M0007.

Pages 23-24

A First Look At The PRINT Statement

The Apple can do five, not six, elementary arithmetic operations: "+" (Addition), "-" (Subtraction), "*" (Multiplication), "/" (Division), and "^" (Exponentiation).

Page 29

Putting Color On The Screen

If you do not have your Apple hooked to a color TV, don't worry! The examples will show up as various shades of gray, with dot and line contrasts, on your black and white screen.

This document suggests some program changes that will provide better black and white contrast when you are using the "color" examples. The changes are described in the section on subroutines, pages 85-90.

Page 37

Pigeonholes And More Calculator Abilities

When you use the variable name GAMEPOINTS in the example at the top of the page your Apple will tell you there is a SYNTAX ERROR. The reason for this message is explained in the text of the Tutorial. (Can you find the reserved word contained in GAMEPOINTS?) You might want to substitute GAMEPTS or GAMESCORE so you won't continue to get this error message.

Chapter 3

Page 59

The IF Statement

As you may have discovered, the program at the bottom of the page really counts from 1 to 9. If you want it to count from 1 to 10, change line 230 to

```
230 IF N <= 10 THEN GOTO 210
```

Page 60

Saving Programs On Diskette

Before you can SAVE a program using a disk drive, you must first initialize a blank diskette. If you don't have an initialized diskette, simply follow these step-by-step instructions and then return to page 60 in the Tutorial. If you already know how to use INIT, you can skip this section.

1. Place your DOS 3.3 Master Diskette in Drive 1.
2. Boot the diskette using either the "warm start" boot (PR#6 RETURN) or the "cold start" boot (turning the computer off and on again).
3. When the blinking cursor appears on the screen and the whirring stops, put a new diskette in Drive 1. (If you have two disk drives, put the new diskette in Drive 2, and add ",D2" to each of the commands below, as shown in parentheses. Note: type only the comma and the D2, not the quotation marks.)
4. Type INIT HELLO and press RETURN. (If you have a second drive, you'll type INIT HELLO,D2 and then press RETURN.)
5. The disk drive will run for a while. When the diskette is initialized, the blinking cursor will reappear, and the red light on the disk drive will go out. That's all there is to it!

Initializing is a process used to prepare a diskette to store information. Once you have initialized your diskette, it will contain a "greeting" program called HELLO. To make sure this process has taken place, and to look at the contents of your diskette at any time, type CATALOG and press RETURN. (If you have two drives, type CATALOG,D2). The only program name you'll see will be HELLO; you'll also see the amount of space that this program takes up on the diskette.

There are other ways to initialize diskettes. You don't need to learn them right now, but if you're curious, you can read pages 13 and 14 in The DOS Manual.

Now, return to page 60 and follow the instructions for SAVEing a program. (If you want to save a program on the diskette in Drive 2, type ",D2" after the name; when you see the red light and hear the whirring of the second drive, you'll know you have your program where you want it.)

Page 63

More Graphics Programs

In the program at the bottom of the page, line 270 should read

```
270 X = X / 7
```

Page 65

FOR/NEXT Loops

At mid-page, line 3220 should read

```
3020 COLOR = N
```

Page 69

PRINTs Charming

In the program line at the bottom of the page, include a blank space in the quotes:

```
120 PRINT "THE STRIKES AND BALLS ARE "; STRIKES;  
    " "; BALLS
```

The semicolon in the program line tells the computer to "pack" the output -- to leave no spaces between the words or numbers. A comma tells the computer to print words (or numbers) in columns, as you will see when you do the examples.

Page 71

PRINTs Charming

You will find that lines 650 and 660 in the program at the top of the page give crossed loops. To fix this, change the lines to

```
650 NEXT Y  
660 NEXT X
```

Chapter 4

Pages 78-79

Off The Walls

Some of the lines that have been added to or changed from the program that bounced the ball between two walls are printed in green when they should be black: 420, 620, 640, 700, 740, 800, 820. Line 890 is the same as line 880 in the earlier version on page 74.

Near the bottom of page 79, the GR in BACKGROUND is a reserved word, so it will be separated from the surrounding letters when the computer prints it. To avoid this, substitute BACKDROP or FIELD.

Page 85

Simulating A Pair Of Dice

Line 220 should read

```
220 COLOR = INT (16 * RND (1))
```

Pages 85-90

Subroutines

If you have a black and white monitor and want better contrast in the horse drawing program examples on these pages, you could substitute

```
COLOR = 2 : REM DARK BLUE
```

for all the COLOR = 7 (light blue) lines. Green (number 12) can be replaced with dark green, number 4, for better black and white effects.

Page 93

High-Resolution Graphics

All the green print zeros are missing the slashes that distinguish them from the letter O, and should be typed as

```
H PLOT 130, 100  
H PLOT 0,0 TO 279,0
```

Page 95

High-Resolution Graphics

The program will give incorrect values for the Y variable unless you install some time delay. Add this line:

```
1005 FOR J = 1 TO 10: NEXT J
```

Page 96

High-Resolution Graphics

Both of the program lines listed below are REMarks, and therefore don't affect the running of the program. But to write them correctly, change to:

```
360 FOR S = 0 TO 1: REM 2 LINES, FROM Y AND Y + 1  
400 REM DRAW LINE THROUGH "CENTER" TO OPPOSITE SIDE
```

Chapter 5

Page 105

Concatenation Got Your Tongue?

Line 120 should read

```
120 WHOLE$ = HALF$ + " " + OTHERHALF$
```

To improve the looks of this program, add

```
105 PRINT
```

Pages 109-110

Introducing Arrays

Change the two lines in the program at the bottom of page 109 as follows:

```
310 REM IF NOT, TRY AGAIN
340 TEMP = GLASS(WINE) : GLASS(WINE) = GLASS(MILK) :
    GLASS(MILK) = TEMP
```

In the explanatory paragraph on page 110, correct the last four sentences of the first paragraph, and the first sentence in the second paragraph, as follows:

Line 290 sets variable MILK to random integers from 1 to 8. Then line 320 makes sure that the value of WINE is not equal to the value of MILK at any given time. The contents of variables GLASS(WINE) and GLASS(MILK) are switched in line 340. Finally the array is printed with lines 370 through 390.

The switching that occurs in line 340 can be thought of like this.

Appendix C

Page 130

Summary of Edit Features

The symbols for clearing the entire screen are incorrect. To clear the screen, press

```
ESC SHIFT P
```

If you want to know more about inserting or deleting text, and fixing program lines, Appendix B: Program Editing in The Applesoft Reference Manual, pages 110-114, is very helpful.