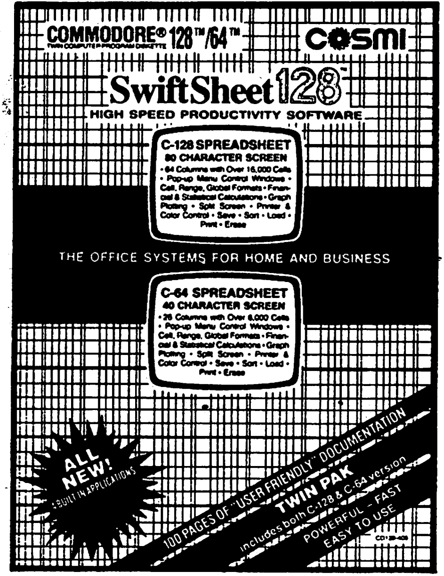
COMMODORE® 128™/64™

CD128-405


COMMODORE® 128™/64™
TWIN COMPUTER PROGRAM DISKETTE
COSMI
SwiftSheet 128™
HIGH SPEED PRODUCTIVITY SOFTWARE
C-128 SPREADSHEET
80 CHARACTER SCREEN
• 64 Columns with Over 16,000 Cells • Pop-up Menu Control Windows • Cell, Range, Global Formats • Financial & Statistical Calculations • Graph Plotting • Split Screen • Printer & Color Control • Save • Sort • Load • Print • Erase
THE OFFICE SYSTEMS FOR HOME AND BUSINESS
C-64 SPREADSHEET
40 CHARACTER SCREEN
• 26 Columns with Over 6,000 Cells • Pop-up Menu Control Windows • Cell, Range, Global Formats • Financial & Statistical Calculations • Graph Plotting • Split Screen • Printer & Color Control • Save • Sort • Load • Print • Erase
ALL NEW!
BUILT IN APPLICATIONS
100 PAGES OF "USER FRIENDLY" DOCUMENTATION
TWIN PAK
includes both C-128 & C-64 version
POWERFUL – FAST
EASY TO USE
CD128-405

COSMI

Swift Sheet 128™

CONTAINS 2 SEPARATE DISKETTE PROGRAMS

Side 1 C-128, 80 Character Screen – 64 Column Spreadsheet.
Side 2 C-64, 40 Character Screen – 26 Column Spreadsheet.

GAMES WORTH PLAYING

PREFACE

Cosmi's SwiftCalc is a personal tool for calculating all kinds of problems. Easy to use, it is practical for very simple calculations to the most complex. You can use it up to figure sales taxes, mortgage payments, expense reports, budgets, business plans, cost estimates, virtually any math you can set down on a piece of paper.

That's what SwiftCalc is: an electronic sheet of paper on your video screen that remembers how to do your calculations. It's automatic: once you've told it how to figure sales tax, for example, all you need to do is give it a new price, then sit back and watch it do the rest. More complicated work, like an annual budget broken down by month, is made easy. (We've already designed a personal budget for you-included on the diskette.) You can go ahead and enter your income and expenses now, then analyze your budget by changing your entries. Quickly find out how much you would have left if your income increases $100.00 in September, your heating bills rise, but you are able to cut down on auto expenses.

SwiftCalc has the features you should expect in a full-featured spreadsheet program: math, statistics, table lookup, and present value functions, printer controls, plus two very desirable enhancements. The first: its "pop-up menus." The second: programmed operation.

SwiftCalc's pop-up menus make operation very easy. All your choices are laid out in front of you on the screen, saving **your** memory, and SwiftCalc catches your errors, saving you from making damaging mistakes.

Its programmed operation gives you special ability to solve problems. You can make a spreadsheet that tells a user when and what to enter, and even automatically print the results when the proper entries have been made or the proper result reached.

The first part of this manual is designed to tutor you in the use of SwiftCalc. The second part is your reference guide, which summarizes SwiftCalc's features.

Enjoy.

TABLE OF ILLUSTRATIONS

CONTENTS

PART 2 REFERENCE TO SWIFTCALC

Appendices

Index

INTRODUCTION

Swiftcalc replaces pencil, pad of paper and the modern calculator with its electronic worksheet. The worksheet, (or, when filled out, the spreadsheet, since it is all spread out in front of you), is a table of entries organized into rows and columns. Each intersection of a row and column in the table is called a cell, and into each cell a text label, a number, or a formula can be entered. When using the worksheet, the formula is not displayed in the cell--it calculates automatically and displays its **result** in the cell. When viewing the worksheet, then, only labels, your entries and the results are displayed. The whole worksheet, or any part of it, can be printed, giving the best features of pen, pad and printing calculator with much more efficiency and control.

The process of creating and using a spreadsheet frees you from the use of mundane tools. Before SwiftCalc, recalculating meant tedium, with careful review of lengthy calculator tapes or computer printouts, or repetitious scrawling of numbers on reams paper.

With SwiftCalc, recalculation is immediate and easy. Since the formulas express the mathematical relationships between the cells, changing an entry instantaneously produces the correct result in all the related cells. When in programmed operation, a calculated answer automatically replaces the original entry, making recalculation automatic. You have many ways to ask "What if?" and observe the changes.

For example, what if:

I make a small down payment and stretch the payments?

My expenses decrease 5%?

Collections of receivables improve?

I compare cost-of-sales to increased sales forecasts?

Production costs decrease?

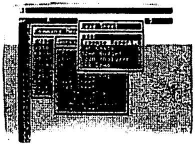
My budget provides for more ready cash?

The ability to immediately see and compare the results of your questions allow fine-tuned analysis of plans, forecasts and technical computations to a level until now impractical. Besides the considerable improvement over "hand calculation" methods, you may find that SwiftCalc offers an improvement over existing programmed solutions programmed in BASIC or other languages.

Use of SwiftCalc's cells is very flexible. You can move, insert, erase columns, rows, cells, and ranges(blocks) of cells as required: save or load them from disk: print them. All computation, disk and print operations are programmable, and formulas and programs may be printed. You can set each cell, a range of cells, or the whole spread sheet to the decimal accuracy required for your results, and widen columns as necessary.

Its pop-up menus simplify the learning, and are a powerful aid when using SwiftCalc. You will soon be getting results. Included on the diskettes are four applications for your immediate personal use: auto expenses, monthly expense report, 12-month personal budget, auto payment calculation, and a demonstration interest-calculation (The IRR Demo) program. These applications are provided for your convenience and as a learning tool: as with any spreadsheet created by someone else, review them carefully to insure that their that formulas and procedures produce results to your satisfaction.

PICTURE OF SCREEN
SHOWING
APPLICATIONS MENU



Applications Menu

How to Use This Manual

Though knowledge of its fundamentals will make SwiftCalc immediately useful, its value to you increases considerably as you learn its extensive abilities. Practice and experiment--you can't hurt the computer or SwiftCalc--and you'll be pleased at your progress.

This manual is constructed to involve you by gradually, increasing your knowledge and skill by contructing an example spreadsheet. A review of the table of contents shows the learning sequence to be followed through **Part 1, Using SwiftCalc**.

Part 2, Reference to SwiftCalc, is your technical reference manual, tabulated for ready reference. The pop-up menus and error-trapping greatly reduce need to refer to the manual for operating SwiftCalc, but reference to the use of its special Functions will likely be frequent and necessary as you build more powerful spreadsheets

The last page of the manual is a **Quick-reference Guide**, provided for your convenience.

GETTING STARTED

To prepare to use SwiftCalc:

What you need:

1. A Commodore 64 or C 128 computer.
2. A Commodore 1541 or 1571 disk drive.
3. A video monitor or television set and appropriate Commodore connectors. **NOTE!** The Commodore C 128 requires connection to the RGBI port, and provides an 80-character wide display.
4. The SwiftCalc program disk.
5. Some blank disks.
6. (Optional), a printer:
   a. Commodore printer connected to serial port.
   b. Parallel printer, interfaced through serial port.
   c. Parallel printer interfaced through user port with a user-port to parallel cable.

How to start SwiftCalc

Please refer to your equipment manuals as necessary.

To load the Commodore 64 version.

1. Connect and power up your computer, monitor, disk drive and printer.
2. Install the SwiftCalc disk in the drive.
3. When the **READY** prompt is displayed, enter
   LOAD"*",8
   Press the **RETURN** key.
4. When the **READY** prompt reappears, type
   RUN
   and press **RETURN**.
5. When the Quick-reference screen displays, press
   f5 (Function key 5), called the **GO** key,
   and the worksheet appears.

To load the C128 version.

1. Connect your computer, monitor, disk drive and printer. Do **not** power up your computer, but **do** power up the disk drive and other peripherals.
2. Install the SwiftCalc disk in the drive.
3. Power up the C 128 computer.
4. The program automatically loads from the disk into the computer.
5. When the Quick-reference screen displays, press **any key**, and the worksheet appears.

All versions:

7. Before you can proceed, the second line down from the top of the screen, called the **Command Line**, prompts:

   Centronics Printer, Yes or No?

   If your printer is connected to the computer's serial port, either directly or through an interface, answer **No**.

   If your printer is Centronics paralle, connected by a cable from the user port to the printer, answer Yes.

   The cursor, or "cell pointer" is in positiom **A1** on the worksheet: that is, in **Column A, Row 1**.

8. If you are not going to load one of the applications on the SwiftCalc disk, remove the program disk from the drive and return it to its envelope.

You are ready to work with SwiftCalc. Before turning to Part 1, please familiarize yourself with the keyboard, described on the following pages and in Appendix E. The Quick-reference Chart.

THE KEYBOARD

SwiftCalc controls your computer's keyboard, and the keys perform only as SwiftCalc directs.

The Function keys.

When a pop-up menu is called, the cursor is positioned on the menu and controlled with the function keys.

f1 Two modes:

1. On a menu: Jumps the cursor to the EXIT command, always the topmost menu selection.
2. During formula entry and range selection: **Pointer mode,** when the cursor names the cells used as terms in formulas.

f3 Moves the cursor one line up the menu.

f5 f5 is called the **GO** key, and pressing it makes the menu selection.

1. On the worksheet: when the cursor is on the worksheet, (or spreadsheet), pressing GO calls the **Command Menu**.
2. On a menu: when pressed while the cursor is over a menu item, that menu item is selected and executed.
3. ERROR condition: when interrupted by an error condition, pressing GO clears the error condition and returns to the worksheet.

f7 Moves the cursor one line down the menu.

f1 is SHIFTed to get f2, f3 SHIFTed to get f4, f5 to get f6, and f7 to get f8.

f2 Causes the spreadsheet to recalculate immediately. In **programmed** mode, it calls the Program Menu.

f4 Jumps the cursor 20 lines up the page.

f6 Exits from the menu to the work sheet.

f8 Jumps the cursor 20 lines down the page.

Cursor Movement keys

The screen scrolls smoothly or jumps according to the cursor action.

Action	Key (with explanation)
Up	**CRSR** Up Arrow or **f3**
Up 20 lines	**f4** (SHIFTed f3)
Down	**CRSR** Down Arrow or **f7**
Down 20 lines	**f8** (SHIFTed f7)
Right	**CRSR** Right Arrow
Left	**CRSR** Left Arrow
Upper left of screen	**HOME** (First press of key)
To Cell A1	**HOME** (Second press of key)
GOTO CELL	See **Command Menu**, Section 1.
SWAP OVER	See **Split-Screen Windows**, Section 5.

Execute Entry keys

Execute menu selection	**f5**, the **GO** key.
Terminate entry	**RETURN**
Stop execution	**STOP**
Enter text	**A** thru Z, and [ ] < ? ! $ % & ' **a** thru z, and : ; , / =
Force entry of text	" followed by any character. ↑ centers text in cell. > forces text to justify right.
Enter numbers	**0** thru 9, + -
Enter formulas	+ (add), - (negative or subtract), * (multiply), / (divide), (open parenthesis requires close parenthesis), and @ calls SwiftCalc's mathematical functions.
Enter program	# followed by the program name must be entered in Column Z. See Section 8, **Programming SwiftCalc**.

Edit Operations

Editing takes place on the screen's Command Line.

Edit Cell	← places contents of cell on the Command Line.
Cursor right	**CRSR** Right Arrow
Cursor left	**CRSR** Left Arrow
Delete character	**DEL**
Insert character	**INST**, or just enter text: SwiftCalc auto-inserts.
Terminate edit	**RETURN**
Delete cell	See **Range Commands**, Section 4.
Delete range	See **Range Commands**.
Delete row or column	See Section 7, **Additional Commands**.
Insert row or column	See Section 7.
Clear Worksheet	See Section 7.

In the next section, **Fundamentals**, you will begin operating SwiftCalc. Please refer to the keyboard listings on these pages as often as necessary.

C 128 additional key functions

The C 128's keyboard is different than the Commodore 64's. The upper row of the keyboard has four more cursor keys: UP, DOWN, LEFT, and RIGHT. For convenience, they operate like the function keys.

The function keys are arranged differently, but perform the same functions.

C128 Key:	Additional function:
CONTROL UP	20 lines up
" DOWN	20 lines down
" RIGHT	5 cells right
" LEFT	5 cells left
ESCape	Calls the **Select Program Menu**.
RETURM	Also the **GO** key, like f5.
SHIFT RETURN	Also **EXITs** a menu, like f6.

Section 1 -- FUNDAMENTALS

THE SCREEN

The top three lines of the screen display are permanent and display the Status Line, the Command Line, and the column-label bar. Below them, a portion of the worksheet is displayed.

The Status Line

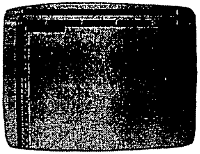
The Status line is the first line on the screen. The cell cursor's coordinates are displayed on this line, followed immediately by the contents of the indicated cell.

The Command Line

Data are entered on this line, then executed upon pressing the RETURN key.

The Label Bars

The column bar labelled A,B,C... and the row bar, labelled 1,2,... are permanent. Where the bars meet are two characters that signify the re-calculation mode, ar, for example, means automatic by row.



PICTURE OF BLANK SCREEN
WITH CURSOR ON A1

The Worksheet Screen

Managing the Cursor

The cursor is used in three ways in SwiftCalc. First, when you are moving it on the worksheet, it is the **cell cursor**, and covers a single cell. Each cell is one character high and as many characters wide as the column it is in.

On the Command Line, information is entered and edited one character at a time, so the cursor is a **character cursor** when on the command line.

On a Menu, the cursor marks the menu selection, and then is the **selection cursor**.

Before reading the next section, you may wish to review the previous section, **Getting Started**, for the cursor movement, execution, and edit commands.

Windows and Scrolling

The screen is a "window" to only part of your worksheet's columns and rows. By pressing the CRSR Down Arrow key, you may move your cursor down to the 21st row and see the row numbers change as you scroll vertically down the sheet. Then press the CRSR Right Arrow key: the cursor will move to the right. When it reaches the rightmost column, watch the column numbers change as you scroll horizontally.

You are scrolling the window. Please note that the Status Line always displays the current cell coordinate of the cursor, whereever you scroll.

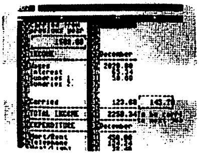
Slowly now, press the HOME key twice, and note where you are after the first press--in the cell at the upper right corner of the workksheet. After the second press, you're back in cell A1.

It is often important to compare sections of a spreadsheet that are many columns or rows apart. With SwiftCalc's **Split-Screen** option, you may split the worksheet vertically on any column and horizontally on any row, and scroll the splitted sections separately to match and view any two sections of the sheet.

You can permit the splitted sections to scroll independently of one another in both the horizontal and vertical directions, or may lock them together. When horizontally split and locked, column alignment is locked: free scrolling is allowed in the vertical direction, but the columns scroll, locked together, when the cursor moves horizontally.

When vertically split and locked, horizontal scrolling is free, but row alignment is locked.

PICTURE OF VERTICALLY
SPLIT SCREEN
WITH DATA



A Split Screen

Pop-up Menu operation

All of SwiftCalc's commands are available to you through its Pop-up Menus, which may be called and exited at will. Certain menu selections will ask you to make entries on the command line, but control is always returned to the most recently used menu, or directly to the worksheet.

Menus are controlled with the **GO** key. Instructions, will read, for example, "Press the GO key," or "Select" a menu item.

Here are the fundamental rules for running the menus:

* Use the special function keys to move up or down the menu, and press GO to make your selection.

* The top selection in each menu is EXIT. Moving the cursor to EXIT, then pressing GO will return you the last menu used--in the case of the Command Menu, to the worksheet.

* Some menu operations return you directly to the previous menu or the worksheet.

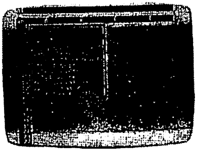
* A menu may be exited at any time by pressing SHIFTed GO, which returns you to the worksheet.

* The STOP key halts any operation and returns you to the worksheet.

* ERROR CONDITIONS halt all operations, and an error message is displayed on the Status Line. Pressing f6 releases the error condition. (Error **Messages** are described in Section 13.)

The Command Menu

PICTURE OF
COMMAND MENU SCREEN



The Command Menu

Press the GO key. The main menu, the **Command Menu** pops up. All the other menus are accessed through the Command Menu.

GOTO cell

The selection cursor is on the first menu item after EXIT.

Press GO.

The words **Goto cell** appear on the Command Line, followed by the character cursor, which is prompting you to make an entry.

Type in Q200 and press RETURN, or GO.

Note that SwiftCalc has immediately moved to the location named, and the row/column labels and Status Line verify it. Any cell on the sheet may be accessed in this manner, or scrolled to.

Entry and Edit

When SwiftCalc sees a text character, it intelligently realizes that all the following characters on the line are to be interpreted as text. Number characters, coordinate values (like A1 or Q200), and formula characters automatically trigger formula entry.

To force non-text characters to be entered as text, the first entry on the line must be a double-quote ", vertical arrow key ↑, or the greater-than character >.

Type your name into the keyboard, using upper and lower case and your middle name in full. As you type, the characters will appear on the Command Line. When you are finished, press RETURN to enter your name into the cell.

If you have made a mistake, and have not yet completed your entry with RETURN, move the character cursor back over your name with the CRSR Left Arrow key, then edit as follows:

To delete a character, move the cursor over the character and press the DEL key.

To insert a character, move the cursor over the character that must make way for your insertion, and merely type in the new character.

To replace a character, first delete, then insert.

Cell Entry Termination

Cell entry may be terminated with the RETURN key, or any cursor movement key but the Left Cursor Arrow key.

If you wish to edit the cell, you have two methods. First, place the cell cursor on the cell, then:

1. Follow the entry sequence above as though the cell had no existing contents: they will be overwritten.

2. Press the **Edit Cell** Key, ←, found at the upper left corner of the keyboard. This recalls the contents of the cell to the Command Line. Then, use the editing techniques listed above.

Press GO when halted by an ERROR MESSAGE.

Number entry

Enter a number, with or without a decimal, and press RETURN when done. For a negative number, precede the number with the - sign. To edit a number, follow the same editing procedures as for text, described above.

Formula entry

A formula is composed of terms consisting of numbers, cell coordinates, and special mathematical functions connected by the math operators +, -, /, and * The formula is interpreted upon pressing RETURN, displaying the result of the formula in the cell. These are formula entry rules:

* If entry starts with a number it can be extended into a a formula merely by adding terms with the arithmetic operators, + - * /.

* To start the formula with a cell coordinate, enter + or - before entering the alpha value that begins the coordinate.

  Not A1+@SQRT(Q200)
  But +A1+@SQRT(Q200)

  If the coordinate is typed in without an arithmetic operator, SwiftCalc interprets the alpha character as forcing text entry.

* Start the formula with an @ function.

  @SQRT(Q200)+A1

SwiftCalc Math

SwiftCalc interprets mathematical formulas from left to right, a term at a time, but not algebraically. (See Section 3 for a more detailed explanation of **SwiftCalc Hierarchy**.) Therefore, it is important to use parentheses to manage the calculation sequence of a cell formula. When RETURN is pressed to enter a formula, SwiftCalc counts the pairs of open and closed parentheses, and prints the ERROR MESSAGE "Uneven brackets" when it discovers an incomplete pair.

The next section, **Building a Spreadsheet**, provides practice, and more detailed instructions for commanding the worksheet and entering data.

SECTION 2 -- BUILDING A SPREADSHEET

So far, you have learned how to start SwiftCalc, and its fundamental concepts of operation.

You have also begun learning the process of entering data into the worksheet in the form of text, numbers and formulas. To oversimplify, these are all the techniques necessary to build a spreadsheet. Just as a living organism is built a cell at a time, so is the dynamic mathematical organism of the spreadsheet.

In this chapter, we will start with the fundamental techniques, then use powerful **Range Commands** and the special **@ Functions** to accelerate the growth of this mathematical organism.

So that you will have an example for comparison, we will construct the Car Costs Report provided on the SwiftCalc disk. As you will do when constructing spreadsheets in the future, we will improve it as we see fit. When you are finished, you will be able to tailor it to your special requirements and have not only a convenient calculating tool, but an auto expense record keeper.

Many references can be found in bookstores showing different applications on spreadsheets. Spreadsheet programs operate on similar principles, so most of the references can be useful. As SwiftCalc's unique abilities are presented to you, numerous applications, from simple to complex, will come to mind. You'll find them within the grasp of SwiftCalc.

The Worksheet Grid

Load SwiftCalc according to the instructions in **Getting Started**.

The row and column numbers serve as coordinates to locate and label the worksheet's cells. Depending on the version, columns extend from **A** to **Z** or from **A** to **BL**, rows from **1** to **254**. When a spreadsheet is printed, only the information on the sheet prints: the row and column numbers don't.

If you were to keep a formal record of your annual auto expenses, you might set a sheet like this to record and sum expenses.

```
                               CAR COSTS
Last odometer reading    :
                           January February....December  Year Total
------------------------ ------- --------    --------  ----------
Price or loan payment    :
Cost of repairs          :
Maintenance              :
Fuel Cost                :
Fuel Quantity (Gallons)  :
Odometer reading         :
Other Expenses           :
------------------------ ------- --------    --------  ----------
        Miles per Gallon:
------------------------ ------- --------    --------  ----------
                  TOTAL:
======================== ======= ========....========  ==========
```

This spreadsheet looks almost exactly like what you will enter into SwiftCalc's worksheet, with a major difference. Invisible, but most definitely there, will be formulas at each cell at the intersection of a month column and the Miles Per Gallon row and TOTAL row. When data is entered into the spreadsheet, results will automatically be produced as the formulas direct.

New data can be entered, and the results immediately known. Formulas or data can be changed at any time, with like results.

Since legible results are as important as accurate results, the format commands are designed to make the spreadsheet easy to read. Dollars and cents as 23333.4500 is harder to read than $23,333.45. As a matter of fact, this is the **default** display mode of SwiftCalc on startup, with two decimal places and commas every three places in front of the decimal. You have to change it if you want different decimal precision, or no commas.

Saving your work

If interrupted while building your spreadsheet, you can save the work-in-progress. Jump to Section 6, **Managing Disk Files**.

Entering row labels

To begin, press the HOME key: this will put the ciursor over cell A1. With the CRSR Down Arrow, move to cell **A6**. (We're leaving space above this row so we can imitate the spreadsheet format laid out above.)

Type in the following keystrokes. Please note that CRSR Down Arrow produces the same result as RETURN, and that to make a narrower column of labels, the labels are shortened somewhat.

Type the word,	**then Press:**
Payment	CRSR Down Arrow
Repairs	CRSR Down Arrow
Maintenance	CRSR Down Arrow
Fuel Cost	CRSR Down Arrow
Fuel Qty(GAL)	CRSR Down Arrow
Odometer reading	CRSR Down Arrow
Other expenses	RETURN

The cell cursor should remain in cell A12 over the words Other expenses.

If an error was made, please review **Entry and Edit** in the previous section for editing and error recovery procedures.

Entering separators

Next, enter the dashed line in cell A13 by first moving to A13 and typing the double quote ". Remember, the dash is a minus sign, if we don't put in ", ↑ or > before it, SwiftCalc will think you're entering a number or formula. Type in a dashed line equal in length to your longest line of text.

Proceed with the following labels as shown separated by dashed horizontal lines as in the example.

In cell A14, enter **>MPG**
In cell A16, enter **>TOTAL**

The alignment isn't right, but you'll fix it in a moment.

Column Widths

The text is wider than the column, and when we later enter numbers into column B your row labels will visually interfere with them. Calculations will be cor'rect, but your spreadsheet may be unreadable.

To widen the column:

Your action	**Command Line**
Press GO for the Command Menu	
Select Column-Width	Use <=CRSR=> to adjust width
Press CRSR Left or Right	Use <=CRSR=> to adjust width
When the column width is satisfactory, press RETURN.	

If you wish to escape from the operation, or have taken a wrong turn somewhere, press the EXIT key or the STOP key, or move to EXIT menu selection and press GO. All these operations will return you to the worksheet.

You don't have to put in the column of colons as shown in the example, but they look good, don't they? You can edit them onto your existing labels, but here's a better way.

> Move the cursor to column B and execute Column-Width again. Make the column just one or two characters wide and enter the colons as text, using RETURN or the CRSR Down Arrow.

Entering column labels

If you chose not to put in the column of colons, move the cursor to cell **B4**, if you did, move to cell **C4**. From here on in it will be assumed you entered the colons in column B, and references will be to Column C.

Entering column labels is the same as row label entry, but move to the right with the CRSR Right Arrow key, rather than down.

Type the word,	**then Press**
January	CRSR Right Arrow
February	CRSR Right Arrow
March	CRSR Right Arrow
April	CRSR Right Arrow
May	CRSR Right Arrow
June	CRSR Right Arrow
July	CRSR Right Arrow
August	CRSR Right Arrow
September	CRSR Right Arrow
October	CRSR Right Arrow
November	CRSR Right Arrow
December	RETURN

The cell cursor should be over the word December.

Now, to center the Month titles in their columns for each column:

Your action	**Command Line**
Move cursor over January text	
Press the cell edit key	January

Press ↑ (center text)	January
Press CRSR Right Arrow	February
Repeat the edit sequence.	February, etc.

Entry shorthand

Hereafter, the flexibility and speed of entry provided by the cursor UP, DOWN, and Right Arrow keys will not be emphasized. Please assume:

> **Entries** are terminated with **RETURN**
> **Selections** are made by pressing **GO**.

Use of RETURN and GO will be stated only for purposes of clarity.

Copying cells

Typing in the bar lines everywhere you want them is tedious and unnecessary. Any single cell can be copied to any other cell or area of cells.

Follow carefully this sequence of instructions.

Your Action	**Command Line**
Move to cell A5	
Press GO	
Select Range Commands	
Select Range Copy	Range to copy from A5
Move to A13	Range to copy from A13
Press GO	Range to copy to A5
Press GO	

The cursor should now be in cell A5 over the dashed line you just copied into it. Following the procedure below, copy A5 into the range C5 through N5.

Select Range Copy	Range to copy from A5
Press GO	Range to copy to A5

Move to C5	Range to copy to C5
Press the comma (,) key	Range to copy to C5,
Type in N5, or move to N5	Range to copy to C5,N5
Press GO	

Now, copy the range C5 through N5 into C13 through N13:

Move to C5	
Select Range Copy	Range to copy from C5
Press the , key	Range to copy from C5,
Enter N5	Range to copy from C5,N5
Press GO	Range to copy to C5
Move to C13	Range to copy to C13
Press GO	

You should now have a line across all the columns in row 13. You can copy this the line of dashes into the other rows from either row 5 or row 13. If, for some reason, you want to format the dashes differently, edit one cell and proliferate it to your heart's desire.

Later, you will copy (or replicate) formulas with this powerful technique, and eliminate the opportunities for error that would be caused if you had to enter each of them individuall.

Entering Numbers

Normally, when building a spreadsheet, the formulas are entered next. Instead, you will enter some numbers in the January column. Then you can verify the workings of the first formula when you do enter it.

Cell	Enter
C6	555
C7	125.33
C8	1.20
C9	64.80
C10	58
C11	18334
C12	130.01

You may wish to reduce the Fuel to one decimal place and the odometer reading to 0 decimal places. Wait until later, when you can reformat a whole range of values. If the word ERROR appears in any cell, widening the column should make it disappear and display the value.

Your "Other" expense is those sheepskin seat covers you've always wanted.

Again, the editing procedures are the same as before, and alas! You've found a missing repair bill of $123.46!

> Move to C7 and press the edit key.
> Just type in **123.46+** and watch the Command Line.
> Press RETURN; the result of the addition is in C7.

You've just entered your first formula.

Section 3 -- ENTERING FORMULAS

The first formula on the Car Costs spreadsheet will calculate the MPG. If you haven't already done so, please enter the text, **Last odometer reading**, in cell A2, reserving cell C2 for entry of the reading, then enter **17000** in C2. Now you're ready for the MPG formula.

The formula: Subtract the last odometer reading in C2 from the current reading in C11. Divide this difference by Fuel Quantity(GAL) in cell C10.

Algebraically: (Current odometer - Last Odometer)/Fuel Qty

In cell terms: (C11-C2)/C10

SwiftCalc hierarchy

SwiftCalc honors the priority of parentheses, but does not follow the rules of algebraic hierarchy. It evaluates the terms from left to right. Since SwiftCalc honors parentheses, it will calculate the difference between the parentheses, then divide by C10. However, this formula will also produce the correct result:

In SwiftCalc: +C11-C2/C10

How it works: Entering the plus sign forces formula entry, rather than interpreting the C as text. C11 is entered, followed by the minus operator. In algebraic hierarchy, the entire term would be evaluated, the division calculated first and then the subtraction. In SwiftCalc hierarchy, the difference is immediately calculated. The next operator and term execute in sequence, and the difference is divided by the contents of C10. As you add operators and terms, each executes as they are entered right to left, unless segregated by parentheses. Within parentheses, SwiftCalc hierarchy rules.

TIP: Analyze your formulas.
When in doubt, use parentheses.

Enter the formula:

Your action	**Command Line**
Move to cell C14	
Type in +C11-C2/C10	+C11-C2/C10
Press RETURN	
The result appears in C14!	

The Cell Equation

Look at the **Status Line**. After the cell coordinate C14, the formula displays, just as you entered it. Yet in the cell, the result displays! This ability is what makes SwiftCalc such a convenient and powerful calculation tool. A cell that contains a formula always **displays the result** of the formula.
To test the dynamic value of this,

Your action
Move to C11
Enter a different number

Your MPG changes with the new entry!

The SwiftCalc equation: **Cell = Formula**
The number displayed = The formula that produced it.

Re-calculation

In certain rare situations, it is possible to get an incorrect result from a reasonable-looking formula. You can now test your formula by pressing the Re-calculation key to force the spreadsheet to re-calculate. If values change, re-calculate until no changes occur.

Later, when building or using complex spreadsheets, this possibility--that a reasonable-looking formula may not produce correct results--should be eliminated. But, an important fundamental for your use of spreadsheet programs is understanding how it can happen. A second important fundamental is this tip:

TIP: Do not use any spreadsheet, from whatever source, without first testing it. The recommended way to do this is to test the sheet with values that will produce known results.

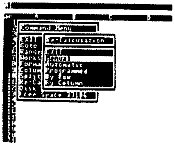
It will be well worth your time to study the following, which is a scenario describing how hidden errors can happen.

First, how the spreadsheet calculates and re-calculates:

At the intersection of the column and row label bars, in the upper left corner, are the two letters **ar**, which signify that SwiftCalc's calculation mode is set to automatically calculate by row.

From the Command Menu, select **Re-calculation**.

PICTURE OF SCREEN
SHOWING
RECALCULATION MENU



Re-calculation Menu

Exercise selections from this menu to see how the various combinations are displayed. You can mix manual/automatic with row/column, but programmed mode, explained in detail in Section 8, **Programming SwiftCalc**, is fixed in **pc**, programmed by column.

Definition	Explanation
Automatic	Upon entry of RETURN, SwiftCalc passes at least once through the whole worksheet and executes what it finds in each cell.
Manual	Produces the same result as automatic, but is calculation is triggered by pressing the Recalculation key.
By Row	The path SwiftCalc follows while passing through the spreadsheet is one row at a time from column A to the rightmost column, starting with row 1 and proceeding to row 254.
By Column	The path is one column at a time from row 1 to row 254, from column A proceeding to the rightmost column.
Programmed	Programs are written in the rightmost column. This option displays a menu of programs when the Recalculation key or the ESCape key is pressed. Then, selection of the program from this Program Menu runs it.

The cause of the calculation errors:

1. The result in C14 is produced from values in cells C10 and C11 according to the formula +C11-C2/C10.

2. Suppose that a formula in other cell, say D15, uses the term C14.

3. Further suppose that the value in C11 is not entered, but is calculated from a formula using D15 as a term. This vicious circle, in which C10 changes C14 which changes D15 which changes C10 which changes C14 and so on, is called a **circular reference**.

Iteration

A circular reference is not a hypothetical, or rare, occurence. Many spreadsheet applications use iterations, which depend on circular references. The **IRR Demo** program on your SwiftCalc disk, (describeed in Appendix C), does. In the IRR Demo, an estimated interest rate is entered, then the future cash flows over a periods of time and a present-day target cash value are entered.

Using the estimated interest rate, the values of the cash flows are calculated to the present day and subtracted from the target amount, producing a difference.

A SwiftCalc @ function tests this difference. If it is not 0, the amount of the difference is used to adjust the estimated interest rate. Recalculation occurs automatically until the difference reduces to 0: then the @ function test causes calculation to stop. The interest rate that produced the difference of 0 is the correct IRR, which means Internal Rate of Return. The IRR is the interest rate that will produce a break-even from a cash outlay and periodic cash returns.

---

5. **To test a spreadsheet** for this condition after either a manual or automatic calculation, press the Re-calculation key and note if any cells change value.

6. Methods to correct this condition:

   a. Press the Re-calculation key until no change occurs.

b. Redesign the spreadsheet to eliminate the cause.

c. Determine by test whether row or column sequencing of calculations eliminates it.

d. Add the necessary @ function tests to provide iteration.

A REMINDER

YOU WILL FIND that this possibility of getting incorrect results is overemphasized here, but is a fundamental principle for your working knowledge of spreadsheets that provides insurance against bad decisions. For example: a correct, checked out, stock analysis spreadsheet will help you make a good decision, because you have confidence in its results.

Erasing

The contents of a cell can be edited away, but editing is much less convenient than erasing its contents. To do this, and also prepare for the next exercise, **Pointing**, erase the formula in cell C14 as follows:

Your action	**Command Line**
Move to C14	
Press GO	
Select Range Commands	
Select Range Erase	Range to blank C14
Press GO	

The cell C14 is blank. For verification, check the Status Line: the formula is gone. If you wish to learn more about ranges now, see Section 4, **Range Commands**, for description of ranges and use of the Range Commands Menu.

Pointing

Pointing is a very efficient formula-building technique. Observe the Command Line while rebuilding the formula in C14 using this powerful technique, implemented with **Pointer** key.

Your action	Command Line
Move to C14	
Enter +	+
Press Pointer	+C14
Move to C11	+C11
Press -	+C11-
Press Pointer	+C11-C14
Move to C2	+C11-C2
Press /	**+C11-C2/C14**
Move to C10	+C11-C2/C10
Press RETURN	

The result again!

You are free at anytime to enter cel! coordinates by keying in or Pointing. The techniques can be intermixed: a coordinate can be edited out with the DELete key and re-entered either way. If a formula is too complex for editing, erase and rewrite. Practice as you learn, learn as you practice.

The @SUM Function

@ Functions are formula terms that do mathematical, logic and other special tasks which greatly enhance the power and flexibility of SwiftCalc. For a summary of the functions available to you, please scan Section 12 of this manual: **LIST OF FUNCTIONS**. Section 9, **Using Functions**, covers them in detail.

The next formula will compute the **Total Monthly Cost** in the January column, where we have our other numbers. One way to do this: Place the cursor in C16, enter + and Point each cell from C6 through C12, making the formula +C6+C7+C8+C9+C10+C11+C12. But there's a better way, using the @SUM function:

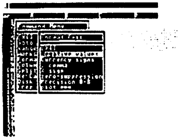
Your action	Command Line
Move to C16	
Enter @	@
Type in sum(.	@SUM(
Press Pointer	@SUM(C16
Move to C6	@SUM(C6
Press ,	@SUM(C6,C16
Move to C9	@SUM(C6,C9
Press)	@SUM(C6,C9)
Press +	@SUM(C6,C9)+
Press Pointer	@SUM(C6,C9)+C16
Move to C12	@SUM(C6,C9)+C12
Press RETURN	

The sum displays in C16, and should be 1,000.00.

Formatting Cells

Not all the values on the spreadsheet are in the most acceptable format. Dollars and cents figures like Payment, Repairs, Maintenance, Fuel Cost, and Other Expenses are OK, since the default setting on startup is two decimal places, with commas. You might like to add the dollar sign to these figures, then reduce Fuel Qty(GAL), Odometer reading and MPG to one decimal place, their customary accuracy.

PICTURE OF SCREEN
SHOWING
FORMAT CELL MENU

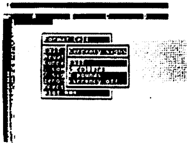


Format Cell Menu

Your action

Move to C6
Select Format Cell
Select Currency Signs
Select $ dollars

PICTURE OF SCREEN
SHOWING
CURRENCY SIGNS MENU



Currency Signs Menu

Please note that the $ symbol **only** has been added to display, with no effect on decimal accuracy or computation. The % format is also only a symbol, with no effect on computations.

Before continuing, EXIT to worksheet. you may then repeat the action above for each cell in which you would like to see the dollar sign displayed.

The General Format Rule

The general rule for formatting SwiftCalc is this: **Set the worksheet format before entering data, then format cells and ranges after data has been entered.**

To change Fuel Qty, Odometer readings and MPG to one decimal place:

Your action	Command Line
Move to C2	
Press GO	
Select Format Cell	
Select Precision 0-8	Number of places (0-8)
Enter 1	Number of places (0-8) 1
Press GO	
EXIT to worksheet	

Repeat the procedure for cells C10, C11 and C14.

How SwiftCalc rounds numbers

SwiftCalc is unique in the way it works with rounded numbers. First, as would be expected, the numbers are calculated and rounded to the precision set for that particular cell. The default startup condition rounds each answer to two decimal places. But, whatever the decimal precision, the program behaves according to the following example.

At a precision of two:			
	0.005	rounds to	0.01
	0.004	rounds to	0.00
	9.995	rounds to	10.00
	9.994	rounds to	9.99

The uniqueness of SwiftCalc is in the improved way it calculates with rounded results. A source of inaccuracy when working with numbers on a computer is the difference between the number displayed and the number actually stored inside the computer, and which of them is used when calculating. Described below and in the table on the next page are two situations used by spreadsheets when rounding.

Situation #1

In this situation, regardless of the decimal setting of the displayed cell, all values are computed with the full significance of the computer, and the number is rounded only on the display, but not rounded in the computer's emory. Calculating with numbers of more significant places than the display is sometimes desirable for scientific calculations, but the rounded **result** is the preferred value for business and statistics.

Situation #2
Some spreadsheets display the numbers unrounded, and compute with the full internal precision. This also produces incorrect results, as shown below.

The Sum	**Situation #1**	**Situation#2**	**SWIFT**
12.36641 etc.	12.37	12.36	12.37
25.25650 etc.	25.26	25.25	25.26
--------	-----	-----	-----
37.62291	37.62	37.62	37.63

The first column of the example, summed at full precision, is correct and unrounded.

Situation #1 produces a result 0.01 too low, since it is not adding the numbers displayed, but actually producing The Sum shown in the first column. In Situation #2 the result appears to be 0.01 too high for the numbers displayed!

In SwiftCalc, the addition is actually performed with the numbers which have already been rounded, correct to the penny, or mil, or digit, depending on the precision you require.

Now that you know that SwiftCalc calculates with the results rounded as displayed, **when you change the precision of any cell, re-calculate by pressing the Re-calculation key**.

Formula or value?

Your formulas need not be buried forever inside SwiftCalc, available to you only for an occasional peek on the Status Line. You can display them. To do this, scroll the column in which the formulas reside toward the left hand edge of the screen, (so you can see an especially long formula in its entirety).

Your action

Press GO
Select Worksheet Commands
Select Formula or Value?

The formulas display. You may scroll the screen and operate just like when it displayed only values.

This Formula or Value? selection **toggles** the display, in that it works like a push button switch. To return the display to the display of values, repeat the above sequence. But for the moment, wait until the opportunity to print the formulas, shown in the next section, **Range Commands**.

Section 4 -- RANGE COMMANDS

Defining ranges

A range is any area of a sheet described with cell coordinates.

It may be:
- a single cell.
- a group of cells in a row.
- a group of cells in a column.
- a group of cells more then one row high by more than one column wide.

Examples of range descriptions:

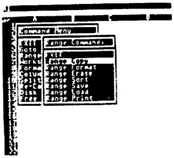
The range:	**The range description:**
C2	The cell C2
A6,A19	A group of cells in column A that includes cells A6 through A19. In the Car Costs spreadsheet you have been building with this manual, it contains the row labels.
C4,N4	A group of cells in row 4 that includes cells C4 through N4. In Car Costs, it contains the month column labels.
A1,C19	A rectangular area three columns wide by 19 rows high: the label column through the January column in Car Costs.
A1,N19	A rectangular area 14 columns wide by 19 rows high, containing the entire Car Costs spreadsheet.

"Named" range   A range of any dimension may be called by a special Command Line prompt: Named Ranges are described in more detail on Page 32.

Printing

Printing is executed from the Range Command Menu.

PICTURE OF SCREEN
SHOWING
RANGE COMMAND MENU



Range Commands Menu

Serial or Parallel printer?

Please refer to **Getting Started**--before you could use the worksheet, the Command line contained the query, **Centronics printer, Yes or No**. To answer, press Y or N and press the RETURN key.

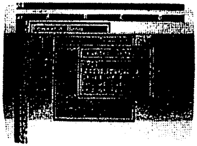
Choice:	**What it means:**
No	Your printer, **regardless what type,** is connected to the computer through its **serial port**.

Yes	Your printer is parallel, interfaced to the Commodore **user** port by a cable.

Printer setup

As a matter of convenience and good practice, if a spreadsheet is printed on 8 1/2" by 11" paper and laid out so you don't have to turn the paper sideways, it is easier to read in a binder--as a record or as a page in a report. For wide spreadsheets, one solution is to buy a 13 inch wide printer, print the spreadsheet across the page, then fold the right-hand edge of the page in. Then it can be folded open and read in the normal page orientation when bound. If you are limited by an 8 1/2" wide printer, you may find it desirable to lay out your spreadsheets on the screen so it prints on an 8 1/2" by 11" page, or with Range Print, print your spreadsheet in segments, so they can be cut and pasted on to a page. It's your choice. SwiftCalc gives you control over the printer's behavior. You can use printers of different widths according to your needs. If you are limited to 8 1/2" wide paper, you can change the printer's character pitch (Most competitive printers offer the option) and print more characters across the page.

PICTURE OF SCREEN
SHOWING
PRINTER SET-UP MENU



Printer Set-up Menu

EXAMPLE PROBLEM: I must print a 14-column spreadsheet that is 132 characters wide, and I can only print on 8 1/2" wide paper. The sheet is 20 rows high, which poses no problem. My printer manual shows that the ASCII codes listed below will set up the printer. The decimal equivalents are the values used by SwiftCalc.

Printer Action	ASCII Codes	Decimal Equivalent
Carriage Return	CR	13
Line Feed	LF	10
Form Feed	FF	12
17.1 Pitch	GS	29
8 lines per inch	ESCape 8	27 56

Proceed as follows:

Your action	Command Line
Press GO	
Select Worksheet Commands	
Select Printer Setup	
Select Page Width	Page Width 80
(8"x17.1char/" = 136.8)	
Edit out 80, Enter 136	Page width 136
Press GO	
Select Start of Print	Start of print
Edit in 29,27,56	Start of print 29,27,56
Press GO	
Select End of Line	End of Line 13
Edit in 13,10	End of Line 13,10
Press GO	
Select End of Print	End of print 13,13
Edit in 13,12	End of print 13,12
Press GO	
EXIT to Command Menu	
Select Range Commands	
Select Range Print	Range to print
Enter A1,N20	Range to Print A1,N20
Press GO	

Setup parameters saved!

SwiftCalc's Printer Setup Menu parameters: Page Width setting, Start-of-Print codes, End-of-Line codes, and End-of-Print (End of Page) codes are saved with your spreadsheet. Once the sheet is set up properly, you don't have to do it again each time you print.

At the end of the last section,in Formula or Value?, you were asked to keep the formula on the screen. To print them without changing SwiftCalc's default setup parameters:

Your action	**Command Line**
Move to C2	
Select Range Print	Range to print C2
Enter ,	Range to print C2,C2
Move to C20	Range to print C2,C20
(Check your printer.)	
Press GO	

Select Formula or Value? to display values again.

Named ranges

A shorthand method for calling ranges in answer to a Command Line prompt is to use **Named Ranges**. On the spreadsheet you have been building, entering the name of the month would print the named month's column. To name a range, one additional instruction must be entered in any blank spreadsheet cell.

Your action	**Command Line**
Move to C25	
Enter **[january] C2,C20**	**[JANUARY]C2,C20**
Press GO	
Select Range Print	Range to Print
(Check your printer.)	
Enter [Ja	Range to print [Ja
(The complete name, or	

any beginning portion
of the name sufficient
to identify it, may be
entered: [J, [Ja, [Jan,
etc., to[January are OK.)
Press GO

Entering February D2,D20 in D25, and repeating this procedure in each column for each month allows you to print any month when prompted without concerning yourself with the actual range values. A whole spreadsheet can be printed by merely entering [s when [sheet] A1,C19 is in any cell!

Because calculation and re-calculation scan the entire worksheet, the name ranging cell can be anywhere on the sheet. Any size spreadsheet can be named, and you can place as many different, related or unrelated, spreadsheets on the worksheet as capacity allows.

You can respond to **any range prompt** with a Named range. Review the capabilities listed on the Range Command Menu: usefulness is only limited by imagination.

Copying formulas

Hold on to your hat. Building the spreadsheet is speeding up.

First, let's solve a little problem. The MPG formula for January depends upon the beginning odometer reading up in C2 for to calculate mileage. To calculate February's MPG correctly, the difference between D11 and C11 must be calculated. A value in the second row, especially only in C2, is useless in months February through December.

Your action	**Command Line**
Move to D14	
Enter +D11-C11/D10	+D11-C11/D10
Press GO	

If you're unhappy that the word ERROR is displayed, it's caused by a divide by 0 in D10, and will go away when data is entered in that cell. Now, to copy the formula

Your action	**Command Line**
Move to D14	
Select Range copy	Range to copy from D14
Press GO	Range to copy to D14
Move to E14	Range to copy to E14
Enter ,	Range to copy to E14,D14
Move to N14	Range to copy to E14,N14
Press GO	

Now copy **C16** to the range D16,N16.

Relative reference

Place your cursor in each cell from D14 through N14 and observe the formulas on the Status Line. Notice that the cell terms have changed relative to the column they are in! When copying spreadsheet cells, SwiftCalc remembers the relative positions of the columns and rows, and adjusts the formulas to maintain that relative difference.

Fixed Reference

There are occasions when you wish to keep a cell term constant in a formula, while others change relatively. To **fix** any cell reference, precede the term with the $ sign, like this: $C2. To demonstrate this fixed reference, add Cost to date in row 18 and Cost/Mile in row 19. Cost per mile will print monthly, but show the cost to date divided by the mileage to date.

Your action	**Command Line**
Move to A18	
Enter >Cost to date	Cost to date
Press GO	
Move to C18	

```
Enter +                  +
Press Pointer            +C18
Move to C16              +C16
Press GO
```

The formula in C18 is +C16, which can be read as 0+C16. The result is in C18 is the duplicate of the value in C16.

Another way to duplicate the value in a cell is with the **@ASSIGN** function. It assigns the value designated by the first term to the cell designated by its second term. In the @ASSIGN function, C16,C18 does **not** mean the range C16 through C18.

```
Move to A25
Enter @ASSIGN(C16,C18)   @ASSIGN(C16,C18)
Press GO
```

To accumulate the monthly costs requires a different formula in D18.

```
Move to D18
Enter +C18+D16           +C18+D16
Press GO
```

Now copy it from D18 to E18,N18.

Calculate cost per mile using fixed reference:

```
Move to C20
Enter +C18/(C11-$C2)     +C18/(C11-$C2)
Press GO
```

As you copy the formula above from C20 to D20 through N20, C2 will be fixed and not change. The parentheses in the formula are necessary so the difference will be calculated before the division occurs. Otherwise, C18 would be divided by C11 and then C2 would subtract from that answer.

To verify that the cells were entered and copied correctly, scan them and observe the Status Line.

TIP: Any cell term may be made a constant by fixed reference.

Now the Cost to date and the Cost/Mile will be correct to the last month data is entered.

Flexibility of Range Copy

The simplified manner notation SwiftCalc uses makes raange operations flexible and powerful, which is indicated by the examples of Range Copy below.

Type of copy	From -- To	Description
Row to row	A9,C9 -- A10	Copies one row.
	A9,C9 -- A12,A14	Copies one row to three rows.
Column to column	A1,A9 -- B6	Copies one column.
	A1,A9 -- B6,C6	Copies one column to three columns.
Cell to cell(s)	A5 -- C6	One cell to one cell.
	A5 -- A8,C8	Copies one cell to three cells.
Area to Area	A5,D8 -- G10	Copies area A5,D8 to area G10,J13.

Range Format

Enter data into the area C6,N12, taking care not to omit Fuel cost, Fuel Qty(GAL) and Odometer readings. Cost entries can be entered as you wish. If Payment is regular and monthly, copy it across its row.

Though your results display efficiently and accurately, the format of the Fuel Qty(GAL), Odometer reading, and MPG rows are incorrect. Make it right with Range Format.

Your action	**Command Line**
Move to C10	
Select Range Format	Range to format C10
Enter ,N10	Range to format C10,N10
Press GO	

Repeat this operation in ranges C11,N11 and C14,N14. Then prettify your spreadsheet with separator lines, etc., as you wish.

Range Erase

Range Erase merely blanks the range you select. To test it, blank the data areas for October, November and December.

Your action	**Command Line**
Move to L6	
Select Range Erase	Range to erase L6
Enter ,N12	Range to erase L6,N12
Press GO	

Done. But now you are at the right end of the sheet, want to enter new values, and can't see the spreadsheet row labels. This is a nuisance easily eliminated in the next section, **Split-Screen Windows.**

Range Sort

Range Sort rows data in order according their alphanumeric rank. Sorting is done along a column, causing the order of rows of data to change. The exercise below shows you how to set up a sort by following these three rules.

* Sort on any column in a range: the entire row within the range will be sorted.

* The column on which the sort is being performed must contain the same data type.

* All cells within the range must contain numbers or text: no formulas are allowed.

Enter the "before & after" table below, do the operations following the able.

Before Range Sort

	A	B	C
1	Sugar	60.00	Smith
2	Cream	40.00	Green
3	Milk	70.00	Brown

After Range Sort

	A	B	C
1	Cream	40.00	Green
2	Milk	70.00	Brown
3	Sugar	60.00	Smith

Your action	**Screen action or Command Line**
Move to A1	
Select Range Sort	Range to sort A1
Enter ,	Range to sort A1,
Enter C3	Range to sort A1,C3
Press GO	Column to sort A1
Press GO	

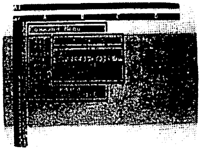
Re-sort the range by repeating the above procedure, but select the other columns. The cursor can be in any row in the column.

Section 5 -- SPLIT-SCREEN WINDOWS

Windows

In Section 1, **Fundamentals**, the fact that the screen was a window into part of the worksheet, and that scrolling the window eventually presented the whole sheet--but only in parts. **Split Screen** provides the convenience of viewing together parts of the spreadsheet that are normally far apart. It splits the screen into two windows that operate independently of one another. Each can be scrolled independently, so you have the freedom to compare any two parts of a spreadsheet, no matter how big it is or what its shape is. If it is necessary to keep rows or columns aligned, you may choose the **synchronization option**.

PICTURE OF SCREEN
WITH
SPLIT SCREEN MENU



Split Screen Menu

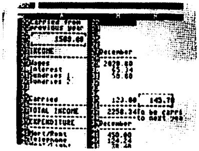
Synchronization & Swapping

To try synchronization, use the sheet you have been building, or load Car Costs from the SwiftCalc program disk.

Your action	**Command Line**
Move to Column C (any cell)	

Select Split Screen	
Select Vertical Split	Select position for split
Press GO	Synchronized movement? Enter Yes or No
Enter Y, Press GO	
EXIT to screen	

PICTURE OF SCREEN
WITH
SCREEN SPLIT



A	H	
Carried from previous year		
1500.00		
INCOME	December	
Wages	2020.00	
Interest	55.34	
Sundries 1	50.00	
Sundries 2		
Carried	123.00	143.7
TOTAL INCOME	2258.34	To be carried to next yea
EXPENDITURE	December	
Mort/Rent	450.00	
Telephone	134.24	
[illegible]	38.20	

A screen split

Your cursor should be in the right half. If not, Select **Swap Over**. Swap Over moves the cursor from one side of the split to the other, so that you may choose which split to scroll.

Your action	**Screen Action**
Scroll up or down	Synchronization forces the rows to scroll together vertically. (Horizontal split and synchronization forces the columns to scroll together horizontally.)
Scroll right to bring the December column into view.	The left-hand screen, which remains fixed, while the right-hand screen scrolls.

If the left-hand screen's labels are misaligned, Swap Over, scroll them into view and Swap Over again.

To test unsynchronized scrolling, select Close Split, split again, and answer No to synchronization.

To test horizontal split:

Move to row six
Select Horizontal Split
Yes to synchronization.
Exit to screen

Split screen provides the ability to visually compare any two worksheet areas.

Section 6 -- MANAGING DISK FILES

Disk files are controlled from three menus:

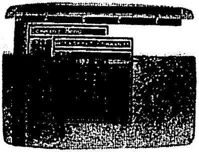
- Spreadsheets are saved and loaded, and the spreadsheet directory called, from the Worksheet Commands menu.
- Ranges are saved and loaded, and the "Ranges saved" directory called, from the Range Commands menu.
- The Disk Commands menu provides:

  Validation--economizes use of the disk surface.

  Delete file--Produces aa complete SwiftCalc file directory, from which the file to be deleted is selected. This menu can also be EXITed.

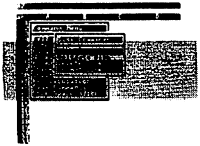
  Format Disk--Formats a new disk for the recording of information. Disks for other applications may be formatted.

PICTURE OF SCREEN
WITH
WORKSHEET COMMANDS



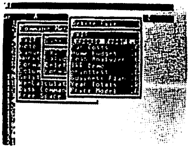
Worksheet Commands

PICTURE OF SCREEN
WITH
DISK COMMANDS



Disk Commands

PICTURE OF SCREEN
WITH
DELETE FILE MENU



Delete File Menu

Saving and Loading spreadsheets

To save your spreadsheets, a formatted disk must be used! Please follow the steps below and use a blank, new disk or an old disk that holds information you no longer need. If you have disks already formatted, skip to **To save new work**, below.

To **format** a disk:

Your action	**Screen action or Command Line**
Check the disk drive. Is it connected and powered up? Is a blank disk installed in the drive?	
Select Command Menu	
Select Disk Commands	
Select Format Disk	Disk name
Enter a name for the disk in this format: Name, XX where the name is alphanumeric characters of your choice and the XX stands for two digits.	
Example: My Sheets,01	Disk name My Sheets,01
Press RETURN	

It takes a few minutes to format the disk. With a formatted disk in the drive, you are ready to manage your spreadsheet files.

To **save new work**:

Select Command Menu	
Select Worksheet Commands	
Select Save Sheet	Replace Sheet? Enter Yes or No

Enter N	
Press RETURN	Worksheet Name (14 char. max)
Enter a name	
Press RETURN	

When the disk light active light goes out, the file is saved.

To **recall a spreadsheet**:

Select Load Sheet	Spreadsheet directory displays Select file, by name. Please note that the directory suffix for spreadsheets is **.G**.

The spreadsheet displays on the screen when loading is complete.

To **replace the disk file**:

Select Save Sheet	Replace Sheet? Enter Yes or No
Enter Y	
Press Return	Spreadsheet directory displays
Select file	

To **delete a file**:

Select Disk Commands	
Select Delete File	A complete directory of SwiftCalc files displays.
Select file	Are you sure? Enter Yes or No
Decide and Enter Y or N	

The Delete File Directory may be EXITed.

To **validate a disk**, select Validate Disk.

Saving and Loading ranges

Saving and loading ranges allows a portion of a spreadsheet to be saved to disk, and then loaded in to another spreadsheet. It is an overlay process, in that cells in a saved range, say C6,C12 will replace the contents of these same cells in the spreadsheet currently displayed.

Save Range is used to save selected data and formulas from a spreadsheet.

Load Range is used to enter or replace selected data or formulas in a spreadsheet

Procedures for saving and loading ranges are identical to those for saving and loading spreadsheets, except that the Save Range and Load Range commands are on the Range Command Menu. Note the following points, however.

* Column width and printer setup information are not saved with a range as they are with a spreadsheet.

* The Command Line queries are the same for ranges and spreadsheets.

* The range file suffix is **.R**.

* The Range Load directory displays range files exclusively.

* The Load Sheet directory displays spreadsheet files exclusively.

* The Delete File directory displays both range and sheet files.

* Sheet loading overwrites the entire display.

* Range loading overlays the entire display, but only matches and overwrites its specified cell range.

Section 7 -- ADDITIONAL COMMANDS

This section covers commands displayed on previously shown menus, but not explained. This section:

- Compares and lists the Format Commands.
- Shows SwiftCalc's start-up default format.
- Demonstrates plotting and Row/Column Delete/Insert.
- Shows how to change screen colors.
- Explains the **Clear Sheet** command.

Free Space

The Free Space command, on the Command Menu, displays the remaining unused worksheet memory, expressed in numbers of characters. When this number reaches zero, you have no more space for your spreadsheet. On startup, you have space for over 33,000 letters and digits. This number may vary slightly according to your system set-up.

Format commands

Three levels of format control are available:

Format	**Where found**	**Description**
Cell	Command Menu	Formats a single cell.
Range	Range Commands	Formats range specified, after range prompt.
Global	Worksheet Commands	Formats entire sheet: normally used to set up the blank worksheet, before entering data.

All of the format options are available at each level, with one exception: plotting is not available from Global Format.

The format menus are below. The first item on the actual menu, EXIT, is omitted.

	Usage notes
Negative Values	
-ve sign	Displays the - sign before a negative number.
( )	Encloses negative numbers in parentheses. A trailing space is added to positive numbers in place of the right parentheses, for alignment of decimal points.
CR/DR	Adds the letters dr after negative numbers, and cr after positive numbers.
Currency signs	
$	Precedes the number with the dollar sign.
	Precedes the number with the English pound sign.
currency off	No currency signs are displayed.
, comma	
on	A comma is inserted every three digits left of the decimal point. Example: 1,999,453.22
off	No commas are inserted. Example: 1999453.22
% sign	
on	Performs no calculations: includes the per cent symbol after the number displayed. Example: 3.3%
off	Per cent sign is not included.

Zero-suppression

show — The cell's zero value (or result) is displayed.
Example: 0.00

blank — Zero values are displayed as a blank cell.

Precision 0-8 — Displays the number of decimal places according to your entry in response to the Command Line prompt, and automatically rounds the number displayed.
Examples: After the prompt Precision 0-8, with response 2 **RETURN**:

```
0.005 rounds to  0.01
0.004 rounds to  0.00
9.995 rounds to 10.00
9.994 rounds to  9.99
```

Plot ***

on — Plotting on converts the number to a row of asterisks. The length of the row of asterisks is proportional to the size of the number, and is independent of the cell (column) width. Negative numbers are treated as 0.

off — The numeric values display.

Demonstration of Plot ***

The advantage of plotting, as always, is to produce a quickly-read visual scalar comparison of related numbers.

To demonstrate how plotting works, use the Car Costs worksheet you have been building.

Your action	**Screen action or Command Line**
Move to N6	
Select Range Commands	
Select Range Format	Range to format N6
Press ,	Range to format N6,N6
Move to N20	Range to format N6,N20
Select Plot ***	Plot *** menu displays
Select plotting on	Numeric values display as line of dots.

This is intentionally not a good example of plotting--the mileage number is too large--erase that one value with Range Erase and observe what happens. Print N6,N20 with Range Print. **Experiment with the plot display**, until you get a feel for SwiftCalc plotting. Often, creating a second column with the values **scaled** by multiplying or dividing by a factor, will show the relationships you desire.

Default format

The default global format is:

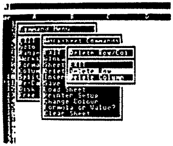
Format	**State**
Negative values	-ve sign on
Currency signs	off
, comma	on
% sign	off
Zero-suppression	blank
Precision 0-8	2
Plot ***	off

Insert or Delete rows or Columns

Inserting and deleting are separate menu selections on the Worksheet Commands menu.

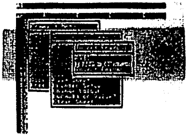
You can either place your cell cursor in the the selected row or column and make your selection, or, when the Command Line prompt appears, SwiftCalc is automatically in point mode. Just move your to the selected row or column and press RETURN.

PICTURE OF SCREEN
SHOWING
DELETE ROW/COLUMN MENU



Delete Row/Column Menu

PICTURE OF SCREEN
SHOWING
INSERT ROW COLUMN MENU



Insert Row/Column Menu

When a row is deleted, the entire row and its contents are eliminated from the worksheet and the contents of all the rows below it move up one row. The relative formula values in all other rows, including the rows that moved, that reference the moved rows are decremented by one to maintain the correct cell reference.

When a column is deleted, the entire column and its contents are eliminated from the worksheet and the contents of all the columns to the right of it move one column left. The relative formula values in all other columns, including the columns that moved, that reference those columns are decremented by one to maintain the correct cell references.

When a row is inserted, the contents of the row beneath the cursor and all the rows below it move down one row. The relative formula values in all other rows, including the rows that moved, that reference those rows are incremented by one to maintain the correct cell reference. A blank row appears under the cursor.

When a column is inserted, the contents of the column beneath the cursor and the columns to the right of it move one column right. The relative formula values in all other columns, including the columns that moved, that reference those columns are incremented by one to maintain the correct cell references. A blank column appears under the cursor.

NOTE! Range Erase merely blanks the cells in the range without moving any rows or columns.

To move rows or columns

SwiftCalc does not have a move column or move row command: To move a row, first insert a blank row (or rows), making space for your move, then Range Copy the rows you want to move into the blankd space you've made. Use Range Erase or Delete Row to eliminate the copy source when you wish to. Use the same technique for copying columns.

Change color

The colors available to you may not appear on your CRT exactly as they are numbered and named below. There are number of causes for this: the tube itself, its age, whether its monitor- or TV-connected, black and white TV instead of color, interference, etc. You can select color combinations that look good, even if in shades of gray.

1	White	9	Brown
2	Red	10	Light Red
3	Cyan	11	Light Gray
4	Purple	12	Medium Gray
5	Green	13	Light Green
6	Blue	14	Light Blue
7	Yellow	15	Dark Gray
8	Orange	16	Black

The default color set is displayed with the prompt upon selecting **Change Color**. It consists of a string of eight numbers separated by commas.

Each of the eight positions controls the color of a different part of the display.

Position	Display	Default number	Color
1	Command Menu	2	Red
2	1st overlaid menu	5	Green
3	2nd overlaid menu	6	Blue
4	3rd overlaid menu	12	Gray
5	Border & Status Line	12	Gray
6	Worksheet characters	16	Black
7	Screen border	1	White
8	Background	1	White

The prompt: Enter color codes: 2,5,6,12,12,16,1,1

Please write your system's best color combination below.

Position 1 2 3 4 5 6 7 8

----- ----- ----- ----- ----- ----- ----- -----

Clear Sheet

When you choose to erase your whole worksheet by selecting **Clear Sheet**, SwiftCalc asks you:

Really clear the Sheet? Yes or No

STOP and think: Have I saved the spreadsheet to disk?

Does the disk file need to be updated?

Should I print it before I destroy it?

And answer accordingly by entering Y for Yes or N for No and press RETURN.

Section 8 -- PROGRAMMING SWIFTCALC

SwiftCalc's sequence of execution--by row or by column--is controlled by the re-calculation menu, but **altered** by use of special @ functions and programming.

@ function Sequence Control

The @ functions that control sequence are:

General Form	Specific Form	Operation
@GOTO(cell)	@GOTO(Z254)	Jumps to the cell, from where the calculation sequence is continued.
@IFGOTO(test, true cell, false cell)	@IFGOTO(A1=6,A6,A20)	If the logical result of **test** is true, calculation sequence will jump to the **true cell**. If not true, sequencing jumps to the **false cell**.

For explanation of the @ functions, please see Section 9, **Using Functions**. For a summary, see Section 12, **List of Functions**.

Programming SwiftCalc

With programs, you can set up an automatic sequence of operations that not only include computations, but also prompted entries of text and values, @ function sequence control, and automatic printing, saving and loading of ranges.

Once a program is written, it is selected from the recalculation menu.

To write a program:

- First, select manual re-calculation.
- Move to any cell in column Z to enter a program.
- Give the program a name--up to 20 characters long.
- Type the name into the cell in column Z, preceding the name with the **#** sign, for example, **#SUM**.
- Enter the program instructions in the rows below the name. (See the example program below.)
- End the program with **@QUIT**.

Set up the worksheet as follows. On the C 128, replace all coordinate references **Z** with **BL**.

ROWS		**COLUMNS**		
	A	. . . .	Y	Z
1	1			#SUM
2	2			@SUM(A1,A3)
3	3			@ASSIGN(Z2,A4)
4	6			@QUIT
5				
6				#LOAD
7			Filename?	@TEXT(Y7,Y8)
8			(Car Costs)	@LOAD(Y8)
9				@PRINT(A1,N20)
10				@QUIT

The **bold** values are spreadsheet entries that result from program operation.

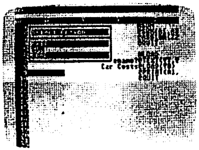
The program #SUM sums the range A1,A3 and then assigns the result produced by cell Z2 to cell A4.

The #LOAD program's @TEXT function places the contents of cell Y7 on the Command Line as a prompt, then places the response, Car Costs, in cell Y8. The @LOAD function loads the named file. @PRINT prints the range A1,N20.

To run a program:

Your action	**Screen action or Command Line**
Select Re-calculation	
Select Programmed	
EXIT to worksheet	
Press Recalculation key	Select Option menu displays.

SCREEN DISPLAY
OF
SELECT OPTION MENU



Select Program Menu

Select SUM
Press GO

Inspecting A4 discloses the value 6. The program **#LOAD** is exercised by the same procedure.

Section 9 -- USING FUNCTIONS

SwiftCalc contains powerful and unique functions for mathematics, statistics, program sequence control, data entry and program management of disk files and printing.

Function syntax

The general syntax of a function is:

@FUNCTION(Parameters within parentheses)

Function parameters are described below, and detailed description of the function is found on the page indicated. The @AND and @OR functions are described through use in examples.

Page 57
@ASSIGN(exp,r)
@COUNT(r1,r2)
@GOTO(r)

Page 60
@NUMBER(rt,r)
@PRINT(r1,r2)
@PV(p$,p%,pt)
@QUIT
@SAVE(rt,r1,r2)
@SORT(rc,r1,r2)

Page 58
@IF(p?,p+,p-)
@IFGOTO(p?,r+,r-)
@LOAD(rt)

Page 61
@SUM(r1,r2)
@SQRT(exp)
@STD(r1,r2)
@TEXT(rt,r)
@VAR(r1,r2)

Page 59
@LOOKUP(rr,r1,r2,rc,r)
@MAX(r1,r2)
@MEAN(r1,r2)
@MIN(r1,r2)

Parameter definitions

r1,r2 = range
r = target cell
r+ = target cell if true
r- = target cell if false
rt = cell containing text
rc = cell in reference column
rr = lookup reference cell

expression = number, cell or formula
p? = test expression
p+ = result if true
p- = result if false
p$ = dollar amount
p% = interest rate, from 0 to 1
pt = time period for @PV

The Functions

The examples used to define the functions demonstrate the comprehensive forms the parameters can take. Since it is easier to show with examples, the function definitions will use the spreadsheet below.

	A	B	C	D	E	F	G
1	7	8	9				Costs
2	4	5	6				
3	1	2	3				
4							

FUNCTION: **DOES:**

@ASSIGN(exp,r)
This function assigns the content of one cell or the results of a calculation to another.

```
@ASSIGN(A1,D6)     assigns the value 7 to D6.
@ASSIGN(A1*2,D8)   assigns the value 14 to D8.
@ASSIGN(G1,H1)     assigns the text Costs to H1.
@ASSIGN(5.45,H2)   assigns the value 5.45 to H2.
```

@COUNT(r1,r2)
Counts the cells containing numbers in the range, excluding zero-value and text cells.

```
@COUNT(A1,D3) gives a result of 9
@COUNT(B1,B4) gives a result of 3
```

@GOTO(r)
Causes calculation sequence to jump to the referenced cell.

@GOTO(Z254) directs calculation to the last cell, causing immediate exit from the calculation sequence.

@IF(p?,p+,p-)

Produces either the true expression p+ or the false expression **p-** according to the truth of the statement **p?**.

p? is a statement using logical operators. The logical operators are:

=	Equal to
<	Less than
<= or =<	Less than or equal to
>	Greater than
>= or =>	Greater than or equal to
<> or ><	Not equal to

Multiple tests can be made in the same statement using @AND and @OR.

Statement:	**Result provided:**	
@IF(B2=5,1,2)	true result	1
@IF(B2<5,1,2)	false result	2
@IF(A1>5@ANDC3=3,A3,B2)	true result	1
@IF(A1=5@ORB2=8,A3,B2	true result	1
@IF(A1=5@ORA1=60@ANDB1=8,@SQRT(@SUM(A1,A10)),+A1*(60+5))	false result	455

@IFGOTO(p?,r+,r-)

Jumps sequence of calculation to the true cell **r+** or the false cell **r-** according to the truth of the statement p? Logical operation is the same as @IF.

Statement:	**Jumps to:**	
@IFGOTO(A1=60@ORB1=10,A6,A20	false cell	A20
@IFGOTO(A2<=4@ANDC3<=3,A6,A20)	true cell	A6

@LOAD(rt)

Loads a range of cells previously saved by the Range Save command under the (text) name located in the cell **rt**.

@LOAD(G1) loads a range file named **Costs**.

@LOOKUP(rr,r1,r2,rc,r)

Searches the range **r1,r2** for the value contained in cell **rr**. The contents of the cell at the intersection of the row in which the value is found and the column that conttains cell **rc** is assigned to cell **r**.

	A	B	C
1	PC		IBM
2	Commodore	C64	100
3	IBM	PC	30
4	Acorn	Electron	500
5			
6	@LOOKUP(A1,B2,B4,A2,C1)		

The function in cell **A6** searches the range **B2,B4** for the value in cell **A1**, which is the text PC. Found in row 3, the colum designator A6 identifies the text IBM, which is assigned to cell **C1**.

@LOOKUP(A1,B2,B4,C4,C1) will assign 30 into C1.

@MAX(r1,r2)

Produces the largest value in the range.

@MAX(A1,C3) gives a result of 9
@MAX(B1,B3) gives a result of 8

@MEAN(r1,r2)

Produces the average of all cells in the range, excluding zero-value and text cells.

@MEAN(A1,D3) gives a result of 5 (45/9)
@MEAN(A1,A3) gives a result of 4 (12/3)

@MIN(r1,r2)

Produces the smallest value in the range.

@MIN(A1,C3) gives a result of 1
@MIN(A1,C1) gives a result of 7

@NUMBER(rt,r)

The contents of the text cell **rt** display as a Command Line prompt. The number entered in response to the prompt is assigned to cell **r**.

@NUMBER(G1,G2) displays the prompt **Costs** on the command line. The number entered in response is placed in cell G2.

@PRINT(r1,r2)

Prints the specified range.

@PRINT(A1,B3) prints the block including upper left corner A1 and lower right corner B3.

@PV(p$,p%,pt)

Returns the present value of the amount **p$** at the percentage **p%** over the period **pt**.

p$ can be a cell name or value.
p% can be a cell name or value between 0 and 1.
pt can be cell name or value between 0 and 255.

@PV(1000,0.12,5) gives the value $67.43
@PV(1000,0.12,A2) gives the value $35.52

@QUIT

Stops re-calculation and is used to end a program: programs are written in the rightmost column of the worksheet.

@SAVE(rt,r1,r2)

Saves the range of cells **r1** to **r2** under the (text) name contained in cell **rt**.

@SAVE(G1,A1,B3) saves the range A1,B3 under the name **Costs**.

@SORT(rc,r1,r2)

Sorts the range **r1,r2** according to the numbers or text in the column containing the cell **rc**. Operates under the same parameters as the command Range Sort.

@SORT(B1,A1,C3) starting in row 1 and proceeding through row 3, sorts the range according to the values in column B.

@SUM(r1,r2)

Adds the values in the range of cells.

```
@SUM(A1,C3) gives a result of 45
@SUM(B2,B3) gives a result of  7
```

@SQRT(exp2)

Produces the square root of the expression.

```
@SQRT(A1*C1/B3) gives a result of  5.61
5+@SQRT(C1)+3   gives a result of 11
```

@STD(r1,r2)

Calculates the standard deviation of the range, excluding zero-value and text cells.

```
@STD(A1,D3) gives a result of  2.58
@STD(A1,A3) gives a result of  2.44
```

@TEXT(rt,r)

The contents of the text cell **rt** display as a Command Line prompt. The text entered in response to the prompt is assigned to cell **r**.

@TEXT(G1,G2) Displays the prompt **Costs** on the command line. The text entered in response is placed in cell G2.

@VAR(r1,r2)

Calculates the variance of the range, excluding zero-value and text cells.

```
@VAR(A1,D3) gives a result of  6.67
@VAR(A1,A3) gives a result of  6.00
```

Section 10 -- OPERATING ELEMENTS

CONTENTS

The SwiftCalc Spreadsheet

The worksheet is a grid of cells arranged in intersecting rows and columns. The sheet is 26 or 64 columns wide by 254 rows high. An area one column wide by one row high is called a cell. Each cell is one character high and can be from one to 22 characters wide.

The area of the worksheet where information is entered into cells is called the spreadsheet.

Each cell may contain a text label, a numeric value, or a formula that calculates a numeric result. Text labels are displayed as is, numeric values displayed according to specified formats, and formulas display their result as a formatted numeric value.

Normally, when viewing the spreadsheet, only text and values are seen by the operator. While a cell may contain a formula. only its numeric result is displayed. A special option allows the formulas to display instead of their result.

The text label can exceed the size of the cell, overprinting the blank cells to the right, but being overprinted by cells to the right that contain text or values. Labels shorter than the width of the cell are normally left-justified, but can be centered or right-justified in the cell.

Numeric values are right-justified in the cell, and displayed with precision from zero to eight decimal places. Numeric values can also be changed to a bar-chart plot.

The formulas are created from SwiftCalc's mathematical functions, numbers, and terms that are references to the values in other cells.

That the value displayed in one cell is dependent upon the values in other cells is the key feature of the spreadsheet. When the independent value changes, the dependent value changes also. This process is called re-calculation, and occurs automatically upon each entry of information into the spreadsheet, or may be made to "manually" re-calculate on a keystroke.

Whether automatic or manual, the sequence of re-calculation sequence is automatically followed by SwiftCalc from cell to cell, and may be directed to proceed from top to bottom of each column before proceeding to the next column, or from left to right in each row before proceeding to the next row.

Occasionally, a spreadsheet is written such that a formula references a cell that contains a formula which back-references the cell of the first formula. This back-reference is called a **circular reference**. The result: a circular reference means that one re-calculation may leave inaccurate vaues in the back-referenced cells. Built into SwiftCalc are decisional functions that make the re-calculation process automatic until the references balance--that is, the cell values no longer change on re-calculation.

A portion of the worksheet is displayed on the screen of the CRT. The display is virtually a "window" into the contents of the worksheet. The window may be scrolled to view different parts of the sheet. The window may also be split into two windows, each of which can be scrolled separately, so different areas of the sheet can be juxtaposed.

Across the top of the window, on the third line from the top of the screen, are the letters identifying the columns, down the left hand edge are numbers identifying the rows. As the window is scrolled, the column and row identifiers change to show which area of the sheet you are viewing.

Scrolling is accomplished by moving the cell cursor.

In the worksheet window is a screen object called the cell cursor, which marks the active cell.

The active cell is changed by moving the cell cursor with the computer's cursor keys. When the cell cursor reaches the edge of the window, it scrolls the screen in the direction of movement.

The active cell receives the entry of information into the sheet. The coordinates of the active cell, by column and row, are displayed on the first line, called the Status Line, of the screen along with the contents of the cell. While the cell displays the result of the formula, the Status Line displays the formula resident in the active cell. The term "active cell" only refers to the fact that it is selected by the cursor for input, and does not refer to any calculation characteristics or the program mode of SwiftCalc.

Entry is displayed on the third line of the screen, called the Command Line, and is completed by pressing a key to terminate the entry operation. The contents of a cell may be recalled to the third line, edited, and returned to the sheet.

The computer's HOME and CLeaR keys may be used to jump the window up and down quickly, the cursor may be caused to jump 20 rows at a time in addition to cell by cell movement to columns and rows, and a command is provided to jump immediately to any cell.

Entering Commands

SwiftCalc is controlled by Pop-up menus, that list the program's commands. When a key (the GO key) is pressed, they appear on the screen with the list of commands for the task selected. The cursor is used to select the command.

When the worksheet is displayed, it is re-calculating or waiting for you to enter information or a command.

Most of the keyboard's keys are reserved for text, number and formula entry. Keys reserved for commands are:

STOP, which ends any operation, manual re-calculation.
RETURN entry termination.
Cursor movement.
Menu exit.
The Pointer.
Select.

Pressing the Select key, called GO, makes the main **Command Menu**,pop-up. Commands selected on this menu either execute immediately, or pop-up a subsidiary menu. Selection is made with the GO key, regardless of the menu, including the EXIT command displayed on each menu in addition to the convenient exit key.

EXIT returns you to the previous menu, all the way to the main Command Menu, or directly to the worksheet. Some commands produce a text prompt on the Command Line, which must be answered before proceeding.

To select a command:

1. Move the cursor to the desired active cell.

2. Press the GO key, which causes the Command Menu to pop up.

2. Move the cursor up or down to the command of choice and press GO.

3. If a subsidiary menu pops up, repeat step 3.

4. If a prompt appears on the Command Line: answer the prompt and terminate with GO or the RETURN key.

   a. If another prompt appears, repeat step 4.

   b. If a subsidiary menu appears, repeat step 3.

5. Depending upon the command,
   a. Return to the worksheet is automatic, or

   b. Select EXIT command or use the exit key to return to the spreadsheet.

Responses to prompts.

Responses to prompts require entry of yes or no answers, entry new file names, cell coordinates, or entry of decimal precision.

When existing disk files are loaded, the prompt is a pop-up menu displaying the names of the files.

Otherwise, responses are entered from the keyboard as alphanumeric characters, or when a cell coordinate entry is required, it may be either entered as characters or **Pointed**.

How to Point:

1. When prompted for a single cell coordinate, press the Pointer key and move the cursor over the cell desired: the cell coordinate appears on the Command Line. Terminate entry.

2. The command is completed and exit is automatic,

or

3. A second coordinate is needed to satisfy the prompt. Press the comma key and either enter the cell coordinate or repeat step 1.

Editing

The contents of any cell are edited in two ways:

1. Make it the active cell and enter as though the the cell were blank. (Editing of the line is described in the step below.) When entry is completed, terminate entry.

2. Make it the active cell and press the Edit Cell key. The contents of the active cell, whether they are text label, number, or formula, appear on the Command Line. The cursor keys are used to move to the correct character position, and characters inserted or deleted as necessry. On the Command Line, **SwiftCalc auto-inserts**. For example, to correct Expnse to Expense, move the cursor over the letter n and type the letter e: the letters nse conveniently move aside for the letter e. The cursor can be anywhere on the Command Line to terminate editing.

Cell Contents

Cells are either blank or contain information, which may be text, numbers, or formulas.

Cell Contents	Description & Use
Number	The digits 0-9, the plus (+) and the minus (-) sign automatically force number entry.
Label	Alphanumeric characters other than the numeric keys, the arithmetic operatorsand the @ key, (which calls the functions), force text entry. Text entry is always forced when the first character on the line is the double quote ("), the right-justification key (>), or the centering key (↑). By editing one of these characters into the cell, a number or formula can be instantly converted to a text label.
Formula	Mathematical terms composed of numbers and arithmetic operators, extended by cell coordinates, logical operators, and functions. Entry is forced by entry of numeric characters or functions, or by extension of a number by adding operators and other mathematical terms. Used to express the mathematical relationships of the spreadsheet's cells.

Numbers

Numbers may be integers or decimal fractions with format attributes added.

Examples of formats with attributes:

```
        100.001
        $99.95
         13.5%
    $11,234.23
     11,234.23
999,999,999.99999
```

The numbers are calculated to 14 digit accuracy and to eight decimal places of precision, rounded up on half and down on less than one half of the place beyond the precision setting.

The smallest displayable number is:

0.00000001

The largest displayable number is:

999,999,999,999.99

When spreadsheet mathematics produces a number a number with more characters than the column width, the word ERROR is displayed, which is removed by widening the column or correcting the mathematical condition that produces the error.

Mathematical operators are:

Operator	Meaning
+	Addition
-	Subtraction, also use in front of a number or term to denote negative.
*	Multiplication
/	Division
()	Parentheses force completion of calculation within.

This manual's Section 3 describes the SwiftCalc's right-to-left mathematical hierarchy, with honored parentheses.

Section 9, **Using Functions**, describes SwiftCalc functions for math, statistics, and financial calculations, and data management.

Logical operators are described below and examples of the logical functions AND and OR are demonstrated in Section 9.

Logical operators

Some SwiftCalc functions evaluate the result of a logical test, then take different action according to the true or false result of the test. Besides the AND and OR functions, the following comparison operators can be used:

Operator	**Meaning**
<	Less than
<= or =<	Less than or equal to
=	Equal to
>	Greater than
>= or=>	Greater than or equal to
<> or ><	Not equal to

The functions @IFGOTO and @IF, explained in Section 9, test the results of logic statements.

Text

A cell may contain up to XX characters of text, which, when larger than the column width, writes over the blank cells to the right of the text cell. If cells to the right are not blank, their contents display, and writeover does not occur.

Text characters are recognized automatically as labels. The keyboard characters not used as commands are recognized as text, as summarized in the table under **Formulas**, above.

Examples of text, with reason:

Text	**Reason**
Smith	Starts with alpha character.
automatic	Starts with alpha character.
↑June	Starts with column-centering character
↑144/12	Starts with column-centering character.
>1985 Totals	Starts with column-right-justification character.
"0.003	Starts with force-text character.
"A5	Starts with force-text character: prevents SwiftCalc form interpreting it as a cell coordinate.

Cell coordinates

References to cells are by coordinates decribing first the column, then the row, of the cell. A cell in column H of the 26 columns labelled A through Z, and in row 80 of the 254 rows labelled 1 through 254 is described by the coordinates H80.

If the value displayed in H80 is the sum of the values in cells C22 and N55, then when H80 is the active cell, the Status Line displays:

```
          H80    +C22+N55
```

which may be interpreted as:

```
          H80 = +C22+N55
```

Regardless whether the contents of C2 and N55 are entered numbers or formulas, H80 displays the resulting sum of the values displayed in those cells.

A group of cells in a column, row, or area of the sheet is described by naming the highest or leftmost cell, entering a comma, then naming the lowest or rightmost cell.

Description	Meaning
A5	The single cell A5.
A5,A10	The range of cells in column A including row 5 through row 10.
A5,N5	The range of cells in row 5 including column A through column N.
A5,N10	Includes the range of cells in the area whose upper left corner is cell A5 and lower left corner is N10.

The **comma** (,) is the **range operator** since it specifies a range of cells. Please note that in certain functions, described in Section 9, **Using Functions, the comma is also used as a parameter separator.**

Please see Section 4, Range Commands for additional detail, including the ability to Name a range of cells.

Cell references

Creating a spreadsheet often necessitates the movement of rows and columns, or the duplication of formulas from one location on the spreadsheet to one or many others.

Relative cell reference

Two examples:

1. On a budget spreadsheet, another expense row is added. The formula in the cell at the bottom of the column of expenses totals the range of cells that include all expenses--the range **C10,C20**. With SwiftCalc's commands the new expense row is inserted between rows C15 and C16, moving the last expense row to C21. When we view the formula in the cell where the total is calculated, the range has changed to **C10,C21**! This automatic change in the referenced cell is necessary to produce the correct result relative to the changed size of the range.

2. On the same spreadsheet, the first total using the range C10,C21 is in the January column. Each of the remaining 11 months can use the same formula, but with the range corrected to produce the total for its own column: February's range is D10,D21; March's E20,E21 and so on. Using SwiftCalc's range copying commands, the formula is copied from Column A to the next 11 columns. On viewing the February's formula, it diplays the range D10,D21--and this change also occurs in all the other columns. When copying formulas, the **relative position** of the referenced cells is maintained. It's as though the references to other cells were cast rigidly such that when the formula moves, the references move rigidly with the formula.

Fixed cell reference

On the same budget spreadsheet, it's decided to make one cell the reference cell for the cost-per-mileage value (in dollars per mile). Then, when each month's mileage total is entered, the month's mileage is calculated and multiplied against the cost per mileage cell to produce an expense figure. When the formula is copied to each month's column, the mileage differences must change relatively as in the above example, but the cost-per-mileage cell reference must remain fixed. If the cost-per-mileage cell is A5, the formula would look something like this when copied:

Column	**Formula**
D	(D11-C11)*$A5
E	(E11-D11)*$A5
F	(F11-E11)*$A5
....	....
N	(N11-M11)*$A5

The dollar sign ($) when prefixing a cell coordinate (not as a numeric format symbol) **fixes the reference** so it will not change when copied.

File handling

SwiftCalc implements the comprehensive file handling features of the computer system for its data files. Please refer to Section 6, **Managing Disk Files** to learn how to prepare disks for use, how to save your work, and recall (load) it for later use.

Error-handling

System errors are displayed on the comand line, and are relieved by pressing GO The summary of these errors Status Line messages and their meaning is described in Section 13, **ERROR MESSAGES**.

Calculation or format errors are flagged by the word ERROR displayed in the offending cell. Correction of the error eliminates it.

Section 11 -- OUTLINE OF MENUS

Press GO to select. Commands that exit directly to the sheet are noted. To stop, press f6 or press the STOP key. EXIT returns you to the previous menu.

```
Command Menu                  Prompt                          EXIT
   EXIT
   Goto Cell                  Go to Cell                      To sheet
   Range Commands
      EXIT
      Range Copy              Range to Copy From
                              Range to Copy To
      Range format            Range to Format
         EXIT
         Negative values
            EXIT
            -ve sign
            () parentheses
            cr/dr notation
         Currency signs
            EXIT
            $ dollars
              pounds
            currency off
         , comma
            EXIT
            , comma on
            , comma off
         % sign
            EXIT
            % sign on
            % sign off
         zero-suppression
            EXIT
            show zeros
            blank zeros
         Precision 0-8        Number of Decimal places (0-8)
         Plot ***
            EXIT
            plotting on
            plotting off
      Range Erase             Range to Blank                  To sheet
      Range Sort              Range to Sort
                              Column to Sort                  To sheet
      Range Save              Range to Save:
                              Replace Sheet? Enter Yes or No:
      (If Yes)
         EXIT
         Disk directory                                       To sheet
      (If No)                 Worksheet name:(14 char. max)   To sheet
      Range Load
         EXIT
         Disk directory                                       To sheet
      Range Print             Range to Print                  To sheet
```

```
Worksheet commands
   EXIT
   Global Format
      (Format menu)
   Sheet Column-Width       Use <=CRSR=> to adjust            To Sheet
   Delete Row/Column
      EXIT
      Delete Row            Delete Row
      Delete Column         Delete Column
   Insert Row/Column
      EXIT
      Insert Row            Insert Row
      Insert Column         Insert Column
   Save Sheet               Replace Sheet? Enter Yes or No:
   (If Yes)
      EXIT
      Disk directory                                          To sheet
   (If No)                  Worksheet name:(14 char. max)     To sheet
   Load Sheet
      EXIT
      Disk directory                                          To sheet
   Printer Setup
      EXIT
      Page Width 80         Enter Page Width
      Start of Print        Enter Printer Codes:
      End of Line           Enter Printer Codes:
      End of Print          Enter Printer Codes:
   Change color             Enter color codes:                To sheet
   Formula or value?                                          To sheet
   Clear Sheet              Really clear the sheet?           To sheet
Format Cell
   (Format menu)
Column width                Use <=CRSR=> to adjust            To sheet
Split Screen
   EXIT
   Vertical Split           Select position & press GO
                            Syncronized movement?
   Horizontal Split         Select position & press GO
                            Syncronized movement?

   Swap Over
   Close Split
Re-Calculation
   EXIT
   Manual
   Automatic
   Programmed
   By Row
   By Column
Disk Commands
   EXIT
   Validate disk
   Delete file
      EXIT
      Disk directory                                          To sheet
   Format Disk              Disk Name                         To sheet
Free space 34114                                              To sheet
```

Section 12 -- LIST OF FUNCTIONS

r1,r2	= range	expression	= number, cell or formula
r	= target cell	p?	= test expression
r+	= target cell if true	p+	= result if true
r-	= target cell if false	p-	= result if false
rt	= cell containing text	p$	= dollar amount
rc	= cell in reference column	p%	= interest rate, from 0 to 1
rr	= lookup reference cell	pt	= time period for present value

@AND(exp), @OR(exp)	performs the logical AND and OR.
@ASSIGN(exp,r)	assigns the expression to cell r.
@COUNT(r1,r2)	counts the cells in the range containing numbers.
@GOTO(r)	directs calculation sequence to cell r.
@IF(p?,p+,p-)	tests, then produces the true or false result.
@IFGOTO(p?,r+,r-)	tests, then directs sequence to true or false cell.
@LOAD(rt)	loads range file whose name is in cell rt.
@LOOKUP(rr,r1,r2,rc,r)	places in cell r the value in range r1,r2 picked by row of rr and column of rc.
@MAX(r1,r2)	produces the maximum number in the range.
@MEAN(r1,r2)	calculates the mean value of a range.
@MIN(r1,r2)	produces the minimum number in the range.
@NUMBER(rt,r)	Numeric response to the prompt text in rt is put in r.
@PRINT(r1,r2)	prints the range.
@PV(p$,p%,pt)	present value of p$ at rate p% for period pt.
@QUIT	stops calculation sequence.
@SAVE(rt,r1,r2)	saves range r1,r2 to filename contained in cell rt.
@SORT(rc,r1,r2)	sorts range r1,r2 on the column that cell rc is in.
@SQRT(exp)	calculates the square root of the expression.
@STD(r1,r2)	calculates the standard deviation of the range.
@SUM(r1,r2)	calculates the sum of the numbers in the range.
@TEXT(rt,r)	Text response to the prompt text in rt is put in r.
@VAR(r1,r2)	calculates the variance of the range.

Section 13 -- ERROR MESSAGES

Error messages appear on the Status Line, program progress is halted, and the cursor marks the offending error. Use the GO or EXIT key to release the error. Edit to correct errors in syntax.

ERROR	Description
Range Error	A correct range description cannot be extracted from the information given.
Uneven brackets ()	An incomplete pair of brackets was entered.
Unsure of meaning	Program doesn't understand your instructions.
Cells are A1 to X only	A nonexistent cell was addressed.
Unknown function	@ is followed by an incorrect function.
Number too large	The number exceeds 999,999,999,999.
Too many decimal points	Only one decimal point per number allowed.
GOTo can't jump to itself	@GOTO must jump to a different cell.
IFGOTO can only go to a cell	An attempt was made to jump to a number or expression.
Number or cell address only	Only a number or cell reference is allowed.
ASSIGN to a different cell	Use a different cell reference for assignment.
Numbers and commas only	The command will only accept numbers separated by commas.
Numbers between 1-255 only	An attempt was made to enter a number outside this range.
Only 15 codes allowed	The command will only allow 15 numbers to be entered.
PV format PV(amt,%,duration)	An error was detected in the use of the PV function.
Prompt with a text cell	An attempt was made to use a prompt cell without text in it.
The sheet is full!	The sheet is full! Erase unused data from your spreadsheet, if possible.
Backgr'd color must be unique	An attempt was made to change background color to match an existing color selection.
Error in Math	A syntax error was made in the expression.

APPENDIX A

HINTS

Save work-in-progress. A power failure or other interruption can destroy valuable time.

Duplicate your files on a backup diskette.

To rename a disk file, save under a new name: delete the old.

Look to simplify your spreadsheets, rather than expand them.

Verify the results of a spreadsheet, whatever the source.

Try to organize your spreadsheets to be read in normal orientation on an 8 1/2" by 11" sheet. It reads much better when included in a report. Many low-cost printers offer compressed character sizes, allowing up to 136 characters across 8" width.

The SwiftCalc system is also a record keeping system. Format a new disk, for example, then store the Expense report on it. Each month, as you call the blank report and enter data into it, store it by the month's name: January, February, etc. At the end of the year, you can correct and sum the entire year.

More than one spreadsheet at a time can reside on the worksheet.

Range Save and Load provide the ability to pass data between spreadsheets.

Think about combining related functions on the worksheet. Expenses, trial balance, Cost of sales, Income Statement and Balance Sheet can calculate simultaneously, making them easier to balance. The fact that labels can be longer than the cells and powerful editing features allow addition of extraordinary items.

Analyze your data with comparison tables, or print the sheet each time you vary the data to make a permanent record for comparison.

A technique for saving computed cell values with Range Save: with @ASSIGN, assign the values to empty cells, then Range Save those cells. At the end of the year, a program to extract those ranges and sum them automates the task of annual totals.

When editing a cell, delete, then type, SwiftCalc auto-inserts.

Don't trouble yourself with trapped errors, but read the message, press GO to relieve the condition, and edit to correct it.

Save keystrokes on each spreadsheet by using Named Ranges to print the spreadsheet.

It saves time to enter numeric data into the spreadsheet while in manual re-calculation mode. When finished, press f2 to calculate. Another way to save time: at the lower right hand corner of your spreadsheet area, @GOTO Z254 (or BL254), and skip the automatic scanning of all the cells in between.

Copy cells when you can. It's more efficient.

Complex equations can be broken up, with intermediate values placed in cells you plan not to print.

Mathematical functions not available in SwiftCalc can be derived either by formula, iterated, or extracted from tables with @LOOKUP.

Iterative solutions can be reached without programming: see the IRR Demo.

Position the cell cursor before selecting the command that prompts for it. It's faster and less confusing.

Get in the habit of using GO instead of RETURN. The few instances where only RETURN is required, GO doen't respond.

Take time to determine the best color combination for your display and record the code sequence in a convenient place.

With SwiftCalc, a parallel interfaced printer requires only a cable between the printer and the Commodore's user port, not a complex interface box.

APPENDIX B

SWIFTCALC GLOSSARY

alignment The vertical alignment of decimal points in a column.

algebraic See hierarchy

alphanumeric Number and letter characters used as text labels.

area A rectangular-shaped grid of cells--a range.

ASCII Acronym for All-Purpose Symbolics Code for Information Interchange, which represents alphanumeric characters and non-printing machine communications codes. Refer to your computer and printer manualsfor ASCII tables, when needed.

cell A location on the work sheet defined by column--row coordinates that is one character high and from one to 20 characters wide.

character A single-digit number or a letter.

column A worksheet area one cell wide and 254 rows high.

command A program operator.

coordinate Description of a cell location by the intersection of its column and row: .i.e., cell Z254.

CRT Cathode Ray Tube: the picture tube in your TV or monitor.

cursor The blinking or non-blinking movable screen object that marks the active cell or character position.

data Useful text and numeric information manipulated by the computer system.

decimal equivalents: Decimal number codes that are specified to be equivalent to alphanumaric characters. Usually, decimal equivalent to ASCII.

default parameters: The value of the program's changeable parameters on start-up.

delete Removal of cells from the spreadsheet.

directory A listing of files, usually from a peripheral storage device, i.e. disk directory.

disk The magnetic storage medium used by the disk drive.

display The images placed on the CRT screen by the computer system.

edit As used, the process of modifying the display to correct data.

electronic worksheet: As is SwiftCalc, where columnar pad operations of entry, calculations, corrections and adjustments, display and retention of results are automated.

erase The removal of data from cells.

error-trapping Feature of well-written programs that anticipates operator errors and provides for recovery without loss of program and data.

expression A symbolic mathematical form. In SwiftCalc, a numeric value, function, cell coordinate, or formula.

file Stored, catalogued, information.

format The capacity and attributes of the CRT display or printed output that are defined by the program. For example, column width or display dimensions, decimal display, symbols included, etc.

formula Mathematical expressions and operators combined to calculate desired results.

function An operator whose result is determined by variable parameters.

global	An operator that affects or uses the entire worksheet.
glossary	A vocabulary of specialized terms with accompanying definitions.
grid	The rectangular pattern of cells that comprise the worksheet.
hierarchy	Mathematical hierarchy: the rules of order sequencing calculations. **Algebraic hierarchy**: Transcendental functions are calculated first, then in order: exponentials, multiplication and division, last addition and subtraction, with parentheses and bracket precedence honored. **SwiftCalc hierarchy**: Proceeds from left to right, and honors parentheses.
interface	The means of connecting the components off an operating system.
iteration	Repeated sequence.
jump	On the worksheet, departure from sequence by GOTO Cell, @IFGOTO, @GOTO.
label	Text characters (not numeric values) in a cell.
load	Move data or information from storage to use.
menu	A list of choices.
mode	System or program behavior, dependent upon commands.
numeric	On the worksheet, displayed values.
operator	A symbol that represents a mathematical, or program, operation.
overlay	When loaded, ranges overlay like a transparency with printing on it: you can see through the overlay, but not the printing. The specified range prints over (overwrites) that range on an existing spreadsheet.
overwrite	When spreadsheets are loaded, the worksheet grid is entirely replaced.
parallel	As used, a printer interface that conveys the eight bits of information that comprise a character to the printer on eight lines simultaneously.
parameter	A variable or arbitrary constant whose value determines the form of an expression or function.
plot	Graphic representation of numeric values: in SwiftCalc, a bar of asterisks.
pointing	The technique provided by SwiftCalc that allows you to enter a cell coordinate by moving to it, rather than typing it in.
Pop-up Menu	SwiftCalc's handy presentation of program operator's, (or commands). Sometimes also called pull-down menus.
precision	The number of places to the right of the decimal point.
program	A coherent sequence of operators organized to do a particular function.
round	Increasing or eliminating one place beyond the selected precision to produce a more significant results. SwiftCalc's rounds ou on 0.5 or more of the precision, down on less than 0.5.
row	A worksheet area one character high and 26 cells wide.
save	Move information from the computer to storage media.
scalar	A number on a scale.
scroll	Move the window by manipulating the cursor.
screen	The viewable image surface of the CRT.
select	The act of executing a menu command.
separators	Lines composed of labels (like hyphens) that separate areas of the spreadsheet for easier reading and interpretation.

sequence A series of commands.

serial As used, a printer interface that conveys the eight bits of information that comprise a character to the printer single file on one wire.

set-up Operations necessary to prepare the system to handle programs and data.

sort Ordering data according to specified parameters.

spreadsheet The area of the worksheet defined by data.

status As used, the state of the cursor in cell coordinates and the contents of the cell.

suffix As used, the disk file notation attached to the file name to indicate whether it is a range (.R), or spreadsheet (.G).

synchronization Coordination of the movement of split windows so that they c an move relative to each other in one axis direction, but not the other: i.e., vertical axis vs. horizontal axis.

swap(over) Moving the cell cursor from one split window to the other.

terminate To end, or execute an operation.

text Alphanumeric characters used exclusively as labels or separators.

toggles Alternating states produced by the same repeated operation: each time Formula or Value? is selected it moves to the other state, from display of cell values to display of cell formulas, and vice versa.

validate Reorganization of disk files for more efficient use of the disk memory surface.

value An assigned or calculated numerical quantity.

window The portion of the worksheet displayed.

worksheet The entire grid of cells, 26 rows by 254 columns, available for the entry of data. Usually used to describe a lined pad of paper organized in rows and columns. See **electronic worksheet**.

zero-suppression: The option to display no characters--a blanked cell--in which the value is zero.

APPENDIX C

Solving Iterative Problems -- THE IRR DEMO

This example shows the Net Present Value function (@PV) used to calculate the Internal Rate of Return (IRR). Before you enter the sheet below, be sure **Manual** and By row re-calculation are set. Press the STOP key if you accidentally start calculation.

```
           A              B              C              D
 1 @NUMBER(B1,C1) Starting IRR?                 @ASSIGN(C1,C3)
 2                              @ASSIGN(C3+0.005,C3)
 3                >Present IRR=
 4 >End of year   >Enter Amounts >Present Value
 5 >------------- >------------- >-------------
 6 1              500            @PV(B6,$C3,A6)
 7 2              300            @PV(B7,$C3,A7)
 8 3              400            @PV(B8,$C3,A8)
 9 4              150            @PV(B9,$C3,A9)
10 5              150            @PV(B10,$C3,A10)
11
12
13
14                >PV of Inflows @SUM(C6,C13)
15                >Enter Outlay   1000
16                >Net PV        +C14-C15
17
18 @IFGOTO(C16<=0.0,Z254,A2)
```

How it works
First, enter the cash outlay and periodic cash inflows, then press the Re-calculation key. In response to the prompt **Starting IRR?**, enter a number between 0 and .99 (suggested value = .15) and press RETURN. The program cycles through the calculations until Net PV in cell C16 is equal to or less than 0.

When the Re-calculation key is pressed, calculation starts in cell A1 and proceeds across row 1, then row 2, and so on, until re-directed by the @IFGOTO function. The function in A1 places the text prompt in A2 on the Command Line, then puts the responsein cell C1. The @ASSIGN in D1 copies the number you entered from C1 into C3. Proceeding to row 3, the @ASSIGN function in cell C2 adds 0.005 to the value in C3 and puts the result back in C3. Each cell in each row is calculated in turn. The @IFGOTO function in C18 directs the calculation sequence to cell A2 when its test is false, and to Z254 when true,thereby exiting the calculations

In each loop through the calculation, the Present IRR is displayed. On exit it is the correct value, since it is the rate that produced a present value of the cash flows equal to the initial outlay.

Please note that the programmed mode is not required to cause automatic iteration. When the Re-calculation key is pressed in manual mode, the sheet passes through one calculation cycle, exiting iin the last, lower right hand cell. The @IFGOTO test keeps the calculation sequencing "alive" until the exit conditions are met, then directs the sequence to a natural exit from the sheet.

APPENDIX D

SCREEN COLORS

SwiftCalc's screen colors and their numeric codes are:

White	1
Red	2
Cyan	3
Purple	4
Green	5
Blue	6
Yellow	7
Orange	8
Brown	9
Light Red	10
Light Gray	11
Medium Gray	12
Light Green	13
Light Blue	14
Dark Gray	15
Black	16

Alphabetically:

Black	16
Blue	6
Blue, Light	14
Brown	9
Cyan	3
Gray, Dark	15
Gray, Light	11
Gray, Medium	12
Green	5
Green, Light	13
Orange	8
Purple	4
Red	2
Red, Light	10
White	1
Yellow	7

APPENDIX E

QUICK REFERENCE CHART

Commodore 64 (& C 128)

EDIT KEYS		SHIFTed keys		Name a program
f1	Point to cell	f2	Re-calculate(or ESC)	***NAME** in column Z
f3	UP 1 line	f4	UP 20 lines*	(or BL)
f5	GO(& RETURN)	f6	EXIT(& SHIFT RETURN)	Name a range
f7	DOWN 1 line	f8	DOWN 20 lines*	**[NAME]r1,r2**

TEXT CELL FORMAT

← re-edit cell   > right   ↑center   " Text   + expression

FUNCTIONS

@AND(exp)	@ASSIGN(exp,r)	@COUNT(r1,r2)	@GOTO(r)
@IF(p?,p+,p-)	@IFGOTO(p?,r+,r-)	@LOAD(r)	@LOOKUP(rr,r1,r2,rc,r)
@MAX(r1,r2)	@MEAN(r1,r2)	@MIN(r1,r2)	@NUMBER(rt,r)
@OR(exp)	@PRINT(r1,r2)	@PV(p$,p%,pt)	3QUIT
@SAVE(rt,r1,r2)	@SORT(rc,r1,r2)	@SQRT(exp)	@STD(r1,r2)
@SUM(r1,r2)	@TEXT(rt,r)	@VAR(r1,r2)	

exp (mathematical expression) = number, cell or formula

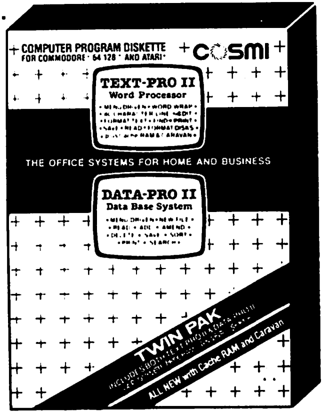
r =	target cell	p? =	test expression
r+ =	target cell if true	p+ =	result if true
r- =	target cell if false	p- =	result if false
r1,r2 =	range	p$ =	dollar amount
rt =	cell containing text	p% =	interest amount, from 0 to 1
rc =	cell in reference column	pt =	time period for present value
rr =	table lookup reference cell		

*(Pressing CONTROL & C128 cursor keys in the top row.
Cursor right or left jumps five cells, up or down jumps 20 lines)

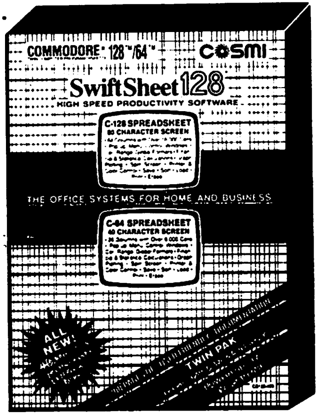
INDEX TO SWIFTCALC

Office Systems for Home and Business

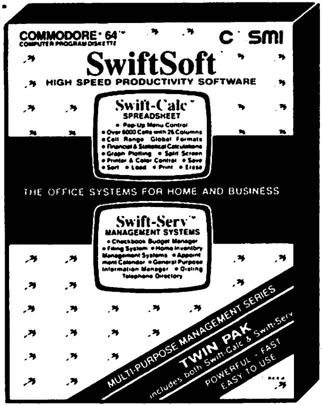
A.


COMPUTER PROGRAM DISKETTE
FOR COMMODORE 64 128 AND ATARI
COSMI
TEXT-PRO II
Word Processor
THE OFFICE SYSTEMS FOR HOME AND BUSINESS
DATA-PRO II
Data Base System
TWIN PAK
ALL NEW with Cache RAM and Caravan

B.



C.


COMMODORE 64
COMPUTER PROGRAM DISKETTE
COSMI
SwiftSoft
HIGH SPEED PRODUCTIVITY SOFTWARE
Swift-Calc
SPREADSHEET
• Pop-Up Menu Control
• Over 6000 Cells with 26 Columns
• Cell Range Global Formats
• Financial & Statistical Calculations
• Graph Plotting • Split Screen
• Printer & Color Control • Save
• Sort • Load • Print • Erase
THE OFFICE SYSTEMS FOR HOME AND BUSINESS
Swift-Serv
MANAGEMENT SYSTEMS
• Checkbook Budget Manager
• Filing System • Home Inventory
Management Systems • Appointment Calendar • General Purpose
Information Manager • Dialing
Telephone Directory
MULTI-PURPOSE MANAGEMENT SERIES
TWIN PAK
includes both Swift-Calc & Swift-Serv
POWERFUL • FAST
EASY TO USE

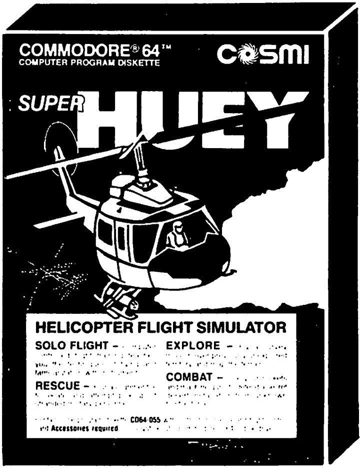
A. **TEXTPRO II/DATAPRO II for Commodore® 64/128™ and Atari.®** All **NEW** version of COSMI'S best selling word processor and data base system "twin pak" . . . Includes exclusive NEW features Cache RAM and CARAVAN.

B. **SWIFT SHEET 128 for Commodore® 128.™** All new expanded 64 column version of COSMI'S award winning SwiftCalc spreadsheet program designed specifically for the Commodore 128 computer. Extensive "user-friendly" documentation.

C. **SWIFT-CALC™ SPREADSHEET.** Easy to use program will complete calculations, whether simple or complex. Ideal for budgets and forecasts, expense reports, business plans, P & L and balance sheets, cash flow analysis, job quotations, statistical analysis, interest calculations . . . and more.

SWIFT-SERV™ MANAGEMENT SYSTEMS Comprehensive package of home management software. Performs most functions you expect from a home computer on any printer. Front end loader and five server modules.

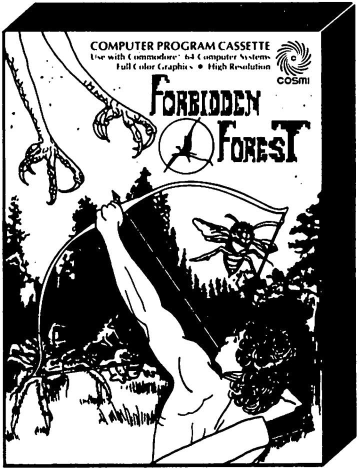
Check out these other Paul Norman 'Super Hit' Computer Programs!



Super Huey is a new, experimental high-performance helicopter utilizing the latest in electronic control systems and stabilization, and YOU are going to learn how to fly it! Space age controls and instruments, along with an in flight computer, teach you the intricacies of rotary wing aviation. Test your skill in the sky with challenges of solo flight, exploration, rescue and combat. Study the instructions carefully . . . you're going to be tested!

cosmi

where the action is!

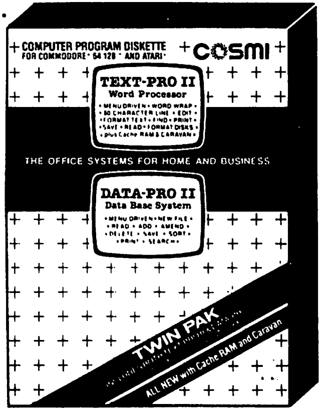

COMPUTER PROGRAM CASSETTE
Use with Commodore 64 Computer Systems
Full Color Graphics • High Resolution
cosmi
FORBIDDEN FOREST

Zounds! And gadzooks! You were just out to do a little target practice with your bow and arrow when you lost your way. Now the moon is coming out and it's getting darker; the forest is getting more ominous and there are some strange rustling noises coming from the bushes. Egad! You have mistakenly wandered into the "Forbidden Forest!" Only your skill as an archer can protect you now. Here they come! Giant spiders, an enormous bumble bee, huge leaping frogs, a fire breathing dragon, a phantom protected by killer skeletons, an 80 foot snake and, finally, the ghost demogorgon who only appears during flashes of lightning that herald the coming storm. Move quickly, aim accurately, destroy the monsters and you just may escape from the "Forbidden Forest."

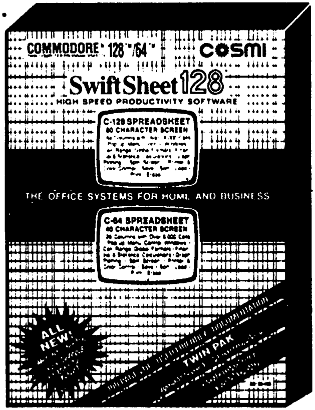
F: 11 (B)

Office Systems for Home and Business

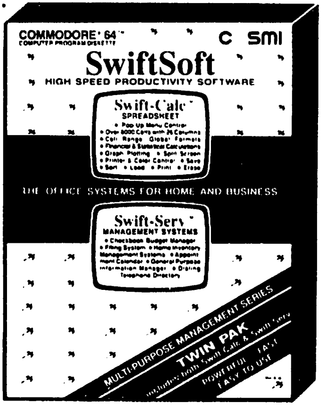
A.


COMPUTER PROGRAM DISKETTE
FOR COMMODORE 64 128 AND ATARI
COSMI
TEXT-PRO II
Word Processor
MENU DRIVEN • WORD WRAP • 80 CHARACTER LINE • EDIT • FORMAT TEXT • FIND • PRINT • SAVE • READ • FORMAT DISKS • plus Cache RAM & CARAVAN
THE OFFICE SYSTEMS FOR HOME AND BUSINESS
DATA-PRO II
Data Base System
MENU DRIVEN • NEW FILE • READ • ADD • AMEND • DELETE • SAVE • SORT • PRINT • SEARCH
TWIN PAK
ALL NEW with Cache RAM and Caravan

B.



C.


COMMODORE 64
COMPUTER PROGRAM DISKETTE
COSMI
SwiftSoft
HIGH SPEED PRODUCTIVITY SOFTWARE
Swift-Calc
SPREADSHEET
• Pop Up Menu Control • Over 8000 Cells with 26 Columns • Cell Range • Global Formats • Financial & Statistical Calculations • Graph Plotting • Split Screen • Printer & Color Control • Save • Sort • Load • Print • Erase
THE OFFICE SYSTEMS FOR HOME AND BUSINESS
Swift-Serv
MANAGEMENT SYSTEMS
• Checkbook Budget Manager • Filing System • Home Inventory Management Systems • Appointment Calendar • General Purpose Information Manager • Dialing Telephone Directory
MULTI-PURPOSE MANAGEMENT SERIES
TWIN PAK
includes both Swift-Calc & Swift-Serv
POWERFUL FAST
EASY TO USE

A. **TEXTPRO II/DATAPRO II for Commodore® 64/128™ and Atari.®** All **NEW** version of COSMI'S best selling word processor and data base system "twin pak" . . . Includes exclusive NEW features Cache RAM and CARAVAN.

B. **SWIFT SHEET 128 for Commodore® 128.™** All new expanded 64 column version of COSMI'S award winning SwiftCalc spreadsheet program designed specifically for the Commodore 128 computer. Extensive "user-friendly" documentation.

C. **SWIFT-CALC™ SPREADSHEET.** Easy to use program will complete calculations, whether simple or complex. Ideal for budgets and forecasts, expense reports, business plans, P & L and balance sheets, cash flow analysis, job quotations, statistical analysis, interest calculations . . . and more.

SWIFT-SERV™ MANAGEMENT SYSTEMS Comprehensive package of home management software. Performs most functions you expect from a home computer on any printer. Front end loader and five server modules.

SUPER HUEY
II
HELICOPTER
FLIGHT
SIMULATOR
Solo Flight • Combat • Explore • Rescue
COSMI
WHERE THE ACTION IS!

Have you tried these other exciting Cosmi Programs?

COMMODORE® 64/128™
COMPUTER PROGRAM DISKETTE
COSMI
BEYOND THE FORBIDDEN FOREST
OMNI DIMENSION 4-D
Contains one program diskette CD64-227 with instruction manual
Accessories required: 1) Joystick 2) Commodore® disk drive

BEYOND THE FORBIDDEN FOREST in Omni Dimension 4D for Commodore® 64.™ America's favorite programmer, Paul Norman, has exceeded his best selling title "Forbidden Forest" with the development of Omni Dimension 4D and the all New continuing saga of the intrepid archer who finally has a chance to go "Beyond The Forbidden Forest" to the cave beneath the castle of the Demogorgen.

COSMI

where the action is!

COMMODORE® 64™
CASSETTE AND DISKETTE
COSMI
CURB
Richard Petty's
TALLADEGA
BELL
43
RACE AGAINST King Richard
AND 18 TOP PRO DRIVERS
Contains one program diskette and one program cassette CDC64-043 with instruction manual
Accessories required: 1) Joystick 2) Cassette recorder or disk drive

Take to the super speedway as you race against KING RICHARD PETTY and 18 top pro drivers. Test your skill in qualifying for a pole position and then let'er rip in the real life 3-D main event, where you lock horns with some of the "good ole boys!" Use strategy to plan your pit stops, draft leading cars to pick up time and save fuel, and be prepared for yellow caution flags. Make the right decisions, keep the pedal to the metal, have a little "racing luck" and you might beat Richard to the checkered flag. Just remember . . . it's not going to be easy, the KING doesn't like to lose!

cosmi

415 North Figueroa Street, Wilmington, CA 90744

GAMES WORTH PLAYING