

CHAPTER 2

GETTING STARTED

- Keyboard
- Back to Normal
- Loading and Saving Programs
- PRINT and Calculations
- Precedence
- Combining Things

EDITING

No one is perfect, and the Commodore 64 takes that into account. A number of editing keys let you correct typing mistakes and move information around on the screen.

CRSR

There are two keys marked **CRSR** (CuRSor), one with up and down arrows **↑ CRSR ↓**, the other with left and right arrows **← CRSR →**. You can use these keys to move the cursor up and down or left and right. In the unshifted mode, the **CRSR** keys will let you move the cursor down and to the right. Using the **SHIFT** key and **CRSR** keys allows the cursor to be moved either up or to the left. The cursor keys have a special repeat feature that keeps the cursor moving until you release the key.

INST/DEL

If you hit the **INST/DEL** key, the cursor will move back a space, erasing (DEleting) the previous character you typed. If you're in the middle of a line, the character to the left is deleted and the characters to the right automatically move together to close up the space.

A **SHIFT**ed **INST/DEL** allows you to INSErT information on a line. For example, if you noticed a typing mistake in the beginning of a line—perhaps you left out part of a name—you could use the **← CRSR →** key to move back to the error and then hit **INST/DEL** to insert a space. Then just type in the missing letter.

CLR/HOME

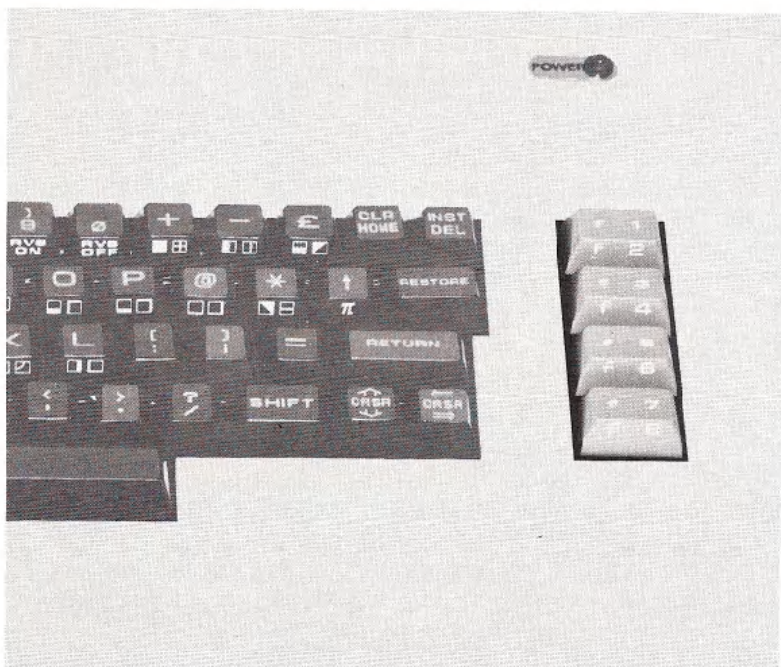
CLR/HOME positions the cursor at the "HOME" position of the screen, which is the upper left-hand corner. A shifted **CLR/HOME** will clear the screen and place the cursor in the home position.

RESTORE

RESTORE operates as the name implies. It restores the computer to the normal state it was in before you changed things with a program or some command. A lot more will be said on this in later chapters.

FUNCTION KEYS

The four function keys on the right side of the keyboard can be "programmed" to handle a variety of functions. They can be defined in many ways to handle repetitive tasks.



CTRL

The **CTRL** key, which stands for ConTRoL, allows you to set colors, and perform other specialized functions. You hold the **CTRL** key down while depressing another designated key to get a control function. You had an opportunity to try the **CTRL** key when you changed text colors to create different color bars during the setup procedure.

RUN/STOP

Normally, depressing the **RUN/STOP** key will stop the execution of a BASIC program. It signals the computer to STOP doing something. Using

the **RUN/STOP** key in the shifted mode will allow you to automatically load a program from tape.

C COMMODORE KEY

The Commodore key **C** performs a number of functions. First, it allows you to move between the text and graphic display modes.

When the computer is first turned on, it is in the Upper Case/Graphic mode, that is, everything you type is in upper case letters. As was mentioned, using the **SHIFT** key in this mode will display the graphic on the right side of the keys.

If you hold down the **C** key and **SHIFT** key, the display will change to upper and lower case. Now, if you hold down the **C** key and any other key with a graphic symbol, the graphic shown on the left side of the key will be displayed.

To get back into the upper case/graphic mode hold down the **C** key and **SHIFT** key again.

The second function of the **C** key is to allow you access to a second set of eight text colors. By holding down the **C** key and any of the number keys, any text now typed will be in the alternate color available from the key you depressed. Chapter 5 lists the text colors available from each key.

BACK TO NORMAL

Now that you've had a chance to look over the keyboard, let's explore some of the Commodore 64's many capabilities.

If you still have the color bars on the screen from adjusting your TV set, hold **SHIFT** and **CLR/HOME**. The screen should clear and the cursor will be positioned in the "home" spot (upper left-hand corner of the screen).

Now, simultaneously hold **C** and the **7** key. This sets the text color back to light blue. There is one more step needed to get everything back to normal. Hold **CTRL** and **0** (Zero not Oh!). This sets the display mode back to normal. If you remember, we turned REVERSE type on with the **CTRL 9** to create the color bars (the color bars were actually reversed spaces). If we were in the normal text mode during the color test, the cursor would have moved, but just left blank spaces.

TIP:

Now that you've done things the hard way, there is a simple way to reset the machine to the normal display. First press the **RUN/STOP** key and then press the **RESTORE** key. **RUN/STOP** must always be held down in order to use the **RESTORE** key function.

This will clear the screen and return everything to normal. If there is a program in the computer, it will be left untouched. This is a good sequence to remember, especially if you do a lot of programming.

If you wish to reset the machine as if it were turned off and then switched on again, type, SYS64759 and press **RETURN**. Be careful using this command! It will wipe out any program or information that is currently in the computer.

LOADING AND SAVING PROGRAMS

One of the most important features of the Commodore 64 is the ability to save and load programs to and from cassette tape or disk.

This capability allows you to save the programs you write for use at a later time, or purchase prewritten programs to use with the Commodore 64.

Make sure that either the disk drive or datasette unit is attached properly.

LOADING PREPACKAGED PROGRAMS

For those of you interested in using only prepackaged programs available on cartridges, cassette, or disk here's all you have to do:

1. CARTRIDGES: The Commodore 64 computer has a line of programs and games on cartridge. The programs offer a wide variety of business and personal applications and the games are just like real arcade games—not imitations. To load these games, first turn on your TV set. Next turn **OFF** your Commodore 64. **YOU MUST TURN OFF YOUR COMMODORE 64 BEFORE INSERTING OR REMOVING CARTRIDGES OR YOU MAY DAMAGE THE CARTRIDGE AND/OR YOUR COMMODORE 64!** Third insert the cartridge. Now turn your Commodore 64 on. Finally type the appropriate **START** key as is listed on the instruction sheet that comes with each game.

2. CASSETTES: Use your DATASSETTE recorder and the ordinary audio cas-


ettes that came as part of your prepackaged program. Make sure the tape is completely rewound to the beginning of the first side. Then, just type LOAD. The computer will answer with PRESS PLAY ON TAPE, so you respond by pressing play on your datasette machine. At this point the computer screen will go blank until the program is found. The computer will say FOUND (PROGRAM NAME) on the screen. Now you press down on the **⏪** KEY. This will actually load the program into the computer. If you want to stop the loading simply press the **RUN/STOP** key.

- 3. DISK:** Using your disk drive, carefully insert the preprogrammed disk so that the label on the disk is facing up and is closest to you. Look for a little notch on the disk (it might be covered with a little piece of tape). If you're inserting the disk properly the notch will be on the left side. Once the disk is inside close the protective gate by pushing down on the lever. Now type LOAD "PROGRAM NAME", **⏪** and hit the **RETURN** key. The disk will make noise and your screen will say:

```
SEARCHING FOR PROGRAM NAME  
LOADING
```

```
READY
```



When the READY comes on and the  is on, just type RUN, and your prepackaged software is ready to use.

LOADING PROGRAMS FROM TAPE

Loading a program back from tape or disk is just as simple. For tape, rewind the tape back to the beginning and type:

```
LOAD "PROGRAM NAME"
```

If you don't remember the program name, just type LOAD and the first program on the tape will be loaded into memory.

After you press **RETURN** the computer will respond with:

PRESS PLAY ON TAPE

After you depress the play key, the screen will blank, turning the border color of the screen as the computer searches for the program.

When the program is found, the screen will display:

FOUND PROGRAM NAME

To actually LOAD the program, depress the **C** key. To abandon the LOADING procedure, hit **RUN/STOP**. If you hit the Commodore key, the screen will again turn the border color while the program is LOADED. After the LOADING procedure is completed, the screen will return to the normal state and the READY prompt will reappear.

LOADING PROGRAMS FROM DISK

Loading a program from disk follows the same format. Type:

LOAD "PROGRAM NAME",8

The 8 is the code for the disk, so you're just letting the computer know that you want the program loaded from the disk.

After you hit **RETURN** the disk will start whirring and the display shows:

SEARCHING FOR PROGRAM NAME
LOADING

READY

NOTE:

When you load a new program into the computer's memory, any instructions that were in the computer previously will be erased. Make sure you save a program you're working on before loading a new one. Once a program has been loaded, you can RUN it, LIST it, or make changes and re-save the new version.

SAVING PROGRAMS ON TAPE

After entering a program, if you wish to save it on tape, type:

```
SAVE "PROGRAM NAME"
```

"PROGRAM NAME" can be up to 16 characters long. After you hit

RETURN the computer will respond with:

```
PRESS PLAY AND RECORD ON TAPE
```

Press both the record and play keys on the datasette. The screen will blank, turning the color of the border.

After the program is saved on tape, the READY prompt will reappear, indicating that you can start working on another program, or just turn off the computer for a while.

SAVING PROGRAMS ON DISK

Saving a program on disk is even simpler. Type:

```
SAVE "PROGRAM NAME",8
```

The 8 is the code for the disk, so you're just letting the computer know you want the program saved to disk.

After you press **RETURN** the disk will start to turn and the computer will respond with:

```
SAVING "PROGRAM NAME"  
OK  
READY  
█
```

PRINT AND CALCULATIONS

Now that you've gotten through a couple of the more difficult operations you need in order to keep the programs you like, lets start making some programs for you to save.

Try typing the following exactly as shown:

```
PRINT "COMMODORE 64"  
COMMODORE 64  
  
READY  
█
```

TYPE THIS LINE AND
HIT **RETURN**
COMPUTER TYPED

If you make a typing mistake, use the **INST/DEL** key to erase the character immediately to the left of the cursor. You can delete as many characters as necessary.

Let's see what went on in the example above. First, you instructed (commanded) the computer to PRINT whatever was inside the quote marks. By hitting **RETURN** you told the computer to do what you instructed and COMMODORE 64 was printed on the screen.

When you use the PRINT statement in this form, whatever is enclosed in quotes is printed exactly as you typed it.

If the computer responded with:

?SYNTAX ERROR

ask yourself if you made a mistake in typing, or forgot the quote marks.

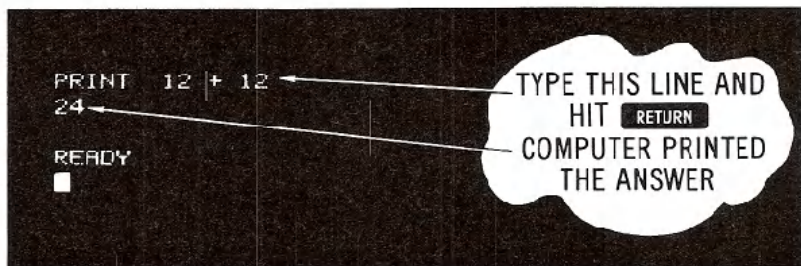
The computer is precise and expects instructions to be given in a specific form.

But don't get worried; just remember to enter things as we present them in the examples and you'll get along great with the Commodore 64.

Remember, you can't hurt the computer by typing on it, and the best way to learn BASIC is to try different things and see what happens.

PRINT is one of the most useful and powerful commands in the BASIC language. With it, you can display just about anything you wish, including graphics and results of computations.

For example, try the following. Clear the screen by holding down the **SHIFT** key and **CLR/HOME** key and type (be sure to use the '1' key for one, not a letter 'l'):



What you've discovered is that the Commodore 64 is a calculator in its basic form. The result of "24" was calculated and printed automatically. In fact, you can also perform subtraction, multiplication, division, exponentiation, and advanced math functions such as calculating square roots, etc. And you're not limited to a single calculation on a line, but more on that later.

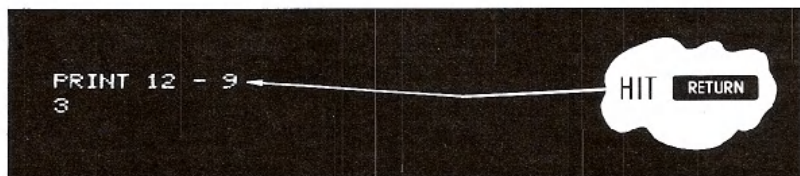
Note that in the above form, PRINT behaved differently from the first example. In this case, a value or result of a calculation is printed, rather than the exact message you entered because the quote marks were omitted.

ADDITION

The plus sign (+) signals addition: we instructed the computer to print the result of 12 added to 12. Other arithmetic operations take a similar form to addition. Remember to always hit **RETURN** after typing PRINT and the calculation.

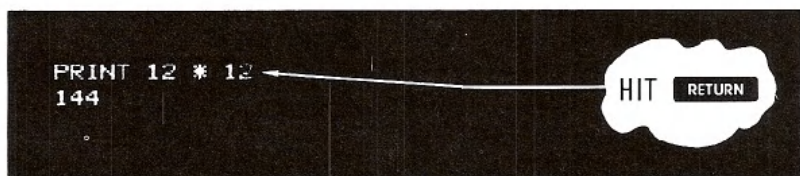
SUBTRACTION

To subtract, use the conventional minus (-) sign. Type:



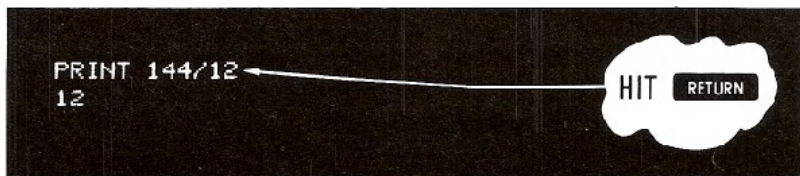
MULTIPLICATION

If you wanted to multiply 12 times 12, use the asterisk (*) to represent multiplication. You would type:



DIVISION

Division uses the familiar "/". For example, to divide 144 by 12, type:



EXPONENTIATION

In a like fashion, you can easily raise a number to a power (this is the same as multiplying a number by itself a specified number of times). The '^' (Up arrow) signifies exponentiation.

```
PRINT 12 ^ 5  
248832
```

This is the same as typing:

```
PRINT 12 * 12 * 12 * 12 * 12  
248832
```

TIP:

BASIC has a number of shortcut ways of doing things. One such way is abbreviating BASIC commands (or keywords). A ? can be used in place of PRINT, for example. As we go on you'll be presented with many commands; Appendix D shows the abbreviations for each and what will be displayed on the screen when you type the abbreviated form.

The last example brings up another important point: many calculations may be performed on the same line, and they can be of mixed types.

You could calculate this problem:

```
? 3 + 5 - 7 + 2  
3
```

THIS ?
REPLACES THE
WORD PRINT

Up to this point we've just used small numbers and simple examples. However, the Commodore 64 is capable of more complex calculations.

You could, for example, add a number of large figures together. Try this, but don't use any commas, or you'll get an error:

```
? 123.45 + 345.78 + 7895.687
8364.917
```

That looks fine, but now try this:

```
? 12123123.45 + 345.78 + 7895.687
12131364.9
```

If you took the time to add this up by hand, you would get a different result.

What's going on here? Even though the computer has a lot of power, there's a limit to the numbers it can handle. The Commodore 64 can work with numbers containing 10 digits. However when a number is printed, only nine digits are displayed.

So in our example, the result was "rounded" to fit in the proper range. The Commodore 64 rounds up when the next digit is five or more; it rounds down when the next digit is four or less.

Numbers between 0.01 and 999,999,999 are printed using standard notation. Numbers outside this range are printed using scientific notation.

Scientific notation is just a process of expressing a very large or small number as a power of 10.

If you type:

```
? 1230000000000000000
1.23E+17
```

This is the same as $1.23 * 10^{17}$ and is used just to keep things tidy. There is a limit to the numbers the computer can handle, even in scientific notation. These limits are:

Largest: $\pm 1.70141183E \mid 38$

Smallest (different from zero): $\pm 2.93873588-39$

PRECEDENCE

If you tried to perform some mixed calculations different from the examples we showed earlier, you might not have gotten the results that you expected. The reason is that the computer performs calculations in a certain order.

In this calculation:

$$20 + 8 / 2$$

you can't tell whether the answer should be 24 or 14 until you know in which order to perform the calculations. If you add 20 to 8 divided by 2 (or 4), then the result is 24. But, if you add 20 plus 8 and then divide by 2 the answer is 14. Try the example and see what result you get.

The reason you got 24 is because the Commodore 64 performs calculations left to right according to the following:

- First: - minus sign indicating negative numbers
- Second: \uparrow exponentiation, left to right
- Third: */ multiplication and divisions, left to right
- Fourth: +- addition and subtraction, left to right

Follow along according to the order of precedence, and you will see that in the above example the division was performed first and then the addition to get a result of 24.

Make up some problems of your own and see if you can follow along and predict the results according to the rules set down above.

There's also an easy way to alter the precedence process by using parentheses to set off which operations you want performed first.

For example, if you want to divide 35 by 5-plus-2 you type:

```
? 35 / 5 + 2
9
```

you will get 35 divided by 5 with 2 added to the answer, which is not what you intended at all. To get what you really wanted, try this:

```
? 35 / (5 + 2)
5
```

What happens now is that the computer evaluates what is contained in the parentheses first. If there are parentheses within parentheses, the innermost parentheses are evaluated first.

Where there are a number of parentheses on a line, such as:

```
? (12 + 9) * (6 + 1)
147
```

the computer evaluates them left to right. Here 21 would be multiplied by 7 for the result of 147.

COMBINING THINGS

Even though we've spent a lot of time in areas that might not seem very important, the details presented here will make more sense once you start to program, and will prove invaluable.

To give you an idea how things fit in place, consider the following: how could you combine the two types of print statements we've examined so far to print something more meaningful on the screen?

We know that by enclosing something within quote marks prints that information on the screen exactly as it was entered, and by using math operators, calculations can be performed. So why not combine the two types of PRINT statements like this:

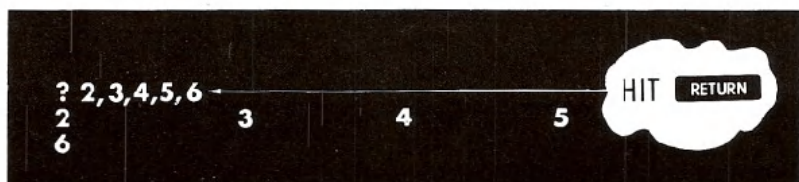
SEM. CO. ON MEANS NO SPACE.

```
? "5 * 9 = "; 5 * 9
5 * 9 = 45
```


Even though this might seem a bit redundant, what we've done is simply use both types of print statements together. The first part prints "5 * 9 =" exactly as it was typed. The second part does the actual work and prints the result, with the semicolon separating the message part of the statement from the actual calculation.

You can separate the parts of a mixed print statement with punctuation for various formats. Try a comma in place of the semicolon and see what happens.

For the curious, the semicolon causes the next part of the statement to be printed immediately after the previous part, without any spaces. The comma does something different. Even though it is an acceptable separator, it spaces things out more. If you type:



the numbers will be printed across the screen and down on to the next line.

The Commodore 64's display is organized into 4 areas of 10 columns each. The comma tabs each result into the next available area. Since we asked for more information to be printed than would fit on one line, (we tried to fit five 10-column areas on one line) the last item was moved down to the next line.

The basic difference between the comma and semicolon in formatting PRINT statements can be used to our advantage when creating more complex displays: it will allow us to create some sophisticated results very easily.