TN #52C

EduCALC TECHNICAL NOTES

27953 CABOT ROAD LAGUNA NIGUEL, CA 92677

WRITING YOUR FIRST HP 48 PROGRAM

The HP 48 is unlike most other calculators. It uses a very different concept of integrating many different operations into one whole. This structure may be described as object orientation. Most other calculators deal with numbers, and some of the more recent designs add text to the numbers. In the HP 48 environment text and numbers are called objects.

OBJECTS, STACK, AND OPERATORS

Imagine that you are making the art work for the cover of a book. You may use a photograph as part of the layout. You may also use an artist drawing, or perhaps a type set title, etc. You use each of these parts, objects, as needed to get the job done. The HP 48 uses 31 object types. A real number or text object are two examples. Others are Binary Numbers, Complex Numbers, Arrays, Algeberaics, etc.

If you add two numbers on any calculator you must input the numbers and tell the machine to add - by the user interface (Arithmetic, Algebraic, Reverse Polish, or Command Line) of the machine. To add you press the "+" key. The "place" you store the numbers is called the stack, a term not used by some manufacturers to keep things as simple as possible.

In the addition example above there are three important concepts involved. The two numbers (objects, data), the "place" where they are stored (stack), and the addition (operation). When you use the HP 48 you will:

1. Place one or more objects on the stack;
2. Perform an operation, which results in;
3. A new stack of zero or more objects.

Each object on the stack occupies a level. Levels are numbered, the first object goes on level one. When a second object is added the first object is "pushed" onto the stack to level 2. This sequence shows in the display with the stack numbered 1: on the bottom, 2:, 3:, and 4: at the top. Memory is the only limit to the number of objects you may place on the stack. The stack is covered in detail in class handouts elsewhere in this book.

The concept of dealing with objects is very important. If you use the HP 48 you will add this most important word to your vocabulary. Below is a table of Object Types used by the HP 48.

Most objects are identified with a delimiter such as single, ', or double, ", quotes, curly braces, { }, etc. Most HP 48 users will seldom use object types beyond type 17. To determine an object type use DUP, TYPE.

TABLE 1 - HP 48 Object Types

Type	Object	Example	Remarks
0	Real number	7.3	No Delimiter
1	Complex number	(1.5,3.4)	Ordered pair
2	String	"Hello"	Text
3	Real array	[3 5 7]	Matrix
4	Complex array	[(2,4) (6,8)]	
5	List	{ "Hi" 7 [9 7] }	Any object in list
6	Global name	A or 'A'	Menu name
7	Local name	a	a -> in programs
8	Program	<< 10 Mod >>	Store with name
9	Algebraic	'C=A+B'	Equation writer
10	Binary integer	# 153Fh	Usually HEX
11	Graphics object	Graphic 131 x 64	GROB
12	Tagged object	Days: 13.5	Operation ignores tag
13	Unit object	153_cm^2	OK in equations
14	XLIB name	XLIB 1010 3	48 & Card use
15	Directory	DIR ... END	Works in system only
16	Library	Library 1010: . . .	Works in port only
17	Backup object	Backup ABCDIR	Stored in port only
18	Built-in function	LOG	
19	Built-in command	CLEAR	
20	Int. bin integer	<56h>	
21	Extended real No.	Long Real	
22	Extended cmplx No.	Long complex	
23	Linked array	Linked Array	
24	Character object	Character	
25	Code object	Code	Machine language
26	Library Data	Library data	Works in port only
27-30	External objects	External	System RPL

Beginners will find the HP 48 "keystroke busy" until they gain an over view of the keyboard, display, and menus.

HP 48 DISPLAY

The figure below, taken from the G series User's guide, illustrates the important aspects of the display.

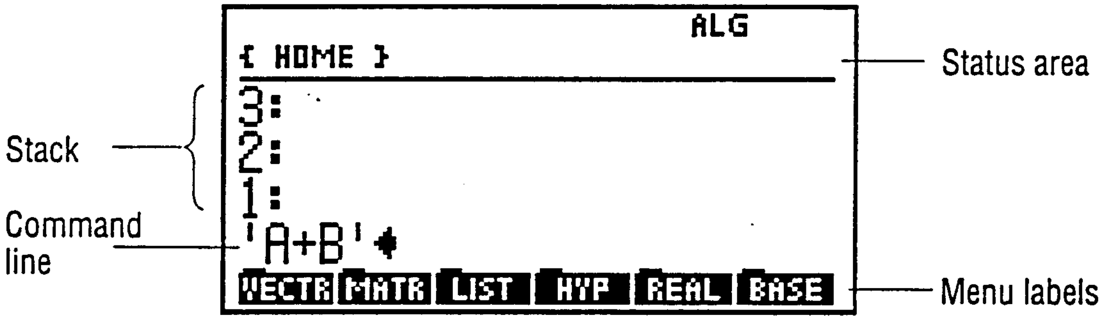

Figure 1 - Typical HP 48 Display with four major areas identified.

When you turn on your HP 48 the contents of the display will depend on what you have previously done. The display is preserved when the machine turns off, either by the 10 minute auto off, or pressing right shift, ON. The display is actually two parts. The dot matrix part shown in figure 1 and the annunicator part shown in figure 2.

The display uses three character sizes, 1, 2, & 3. Size one is one to five dots wide (kerning) and five dots high. Size two is five dots wide and seven dots high. Size three is five dots wide and nine dots high. The display is 131 dots across and 64 dots down. This allows eight lines of character size two or three and 22 characters across. Using a graphics object in the display it is possible to use size one characters and fit ten lines of about 30 characters each.

Status Area

The "normal" HP 48 display has a line across the top portion. The area above the line is the display Status area. The list, { HOME }, is always present in a "normal" display. This is the current directory path. This same list is returned with the PATH command.

The smallest size, 1, is used for the current directory path. Size one only displays in upper case letters. In Figure 1 the status area also shows that the machine is in Algebraic entry mode. The ALG in figure 1 is also size one. If the machine is in radian mode the RAD indicator will show in the upper left corner of the Status area.

The space at the end of the current directory path is used for the date and ticking clock display, if activated. Aside from the merged memory, ticking clock bug in REV M HP 48GX machines, it is still good practice to leave the ticking clock off. This will allow the full line to be used for the current directory path - an important consideration as your machine fills up with objects.

Stack Area

Stack levels are displayed in size three characters. Three or four levels will be shown depending on the presence of the Command line. Once you have pressed ENTER to complete the Command line the stack area will show four numbered levels. There is no indicator to tell you if there are objects in level five or above. The display shows four levels at a time - special programs have been written to show five or seven levels by over writing other areas that are normally in the display. You may get copies of these programs while in class.

Command Line

If you press a key when a command line is not present and you are in the normal HP 48 environment, the Command line becomes active. You will see a blinking cursor and you may enter numbers, text, etc. into the command line. Pressing ENTER terminates the command line and starts the HP 48 working on it to determine what to do. If an error occurs the machine will beep and position the cursor to approximately indicate where the problem may be. If you change your mind just press ON (CANCEL) to ignore and clear the command line.

Menu Labels

The bottom line in the display shows the menu presently selected. The default menue, MTH, is shown in figures 1 and 2. The white keys directly below the blue menu boxes are used to select the command or object desired. Pressing <─┐, MENU will store the object on level one into memory. THIS IS VERY DANGEROUS; PRACTICE STORING, CLEARING, AND RENAMMING OBJECTS SO YOU UNDERSTAND THIS CONCEPT. Pressing ┌─>, MENU recalls the object to the stack.

The space above the dot matrix area of the display is used for the Annunciators. See figure 2.

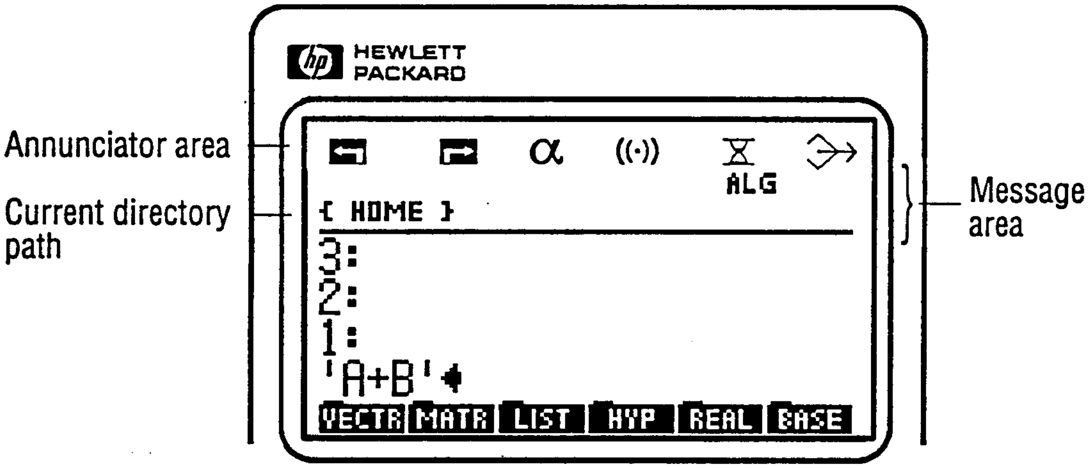

Figure 2 - Typical HP 48 Display showing HP 48 annunciators.

The annunciators are "turned on" to indicate the important conditions of the machine. The first (left to right) is the left shift key annunciator. This toggels on/off when you press the left shift key. Next is the right shift key pressed annunciator. Pressing one will cancel the other. Next is the alpha shift key pressed annunciator. Press the alpha shift key a second time will either lock alpha mode on or turn it off depending on the setting of flag -40. If your alpha key is set - as it comes out of the box, default mode - for double press lock set it for single key lock as follows: Press 40, +/-, SF, ENTER. We expect single alpha key lock in the class. If you have used an HP 41 in the past this is how the alpha key worked on the HP 41C/CV/CX machines.

The next annunciator, just right of center, indicates an appointment has come due or a low battery condition has been detected. This will remain on even if the calculator is turned off. Next is an hour glass symbol to tell you the calculator is busy - and in a very high battery drain condition. The last annunctiator indicates that the machine is transmitting data to an external device. See section one of the G Series User's Guide for additional details regarding the Display, especially the table on the top of page 1-3.

WRITING THE PROGRAM

Suppose you wish to compute the area of a circle assuming that the radius is on the stack. You want the name AREA to be in the menu and pressing it will compute the area and tag the answer Area: nnn.

The program will calculate the area based on: Area = π*r^2.

where: π is PI (3.14159265359).
r is the circle radius.
^2 indicates squaring level one object.

Here is how you enter this program object into the calculator. The key strokes are for G series machines with S series keys given if different.

1. Turn on the calculator. Clear the stack with <┐, DEL. On the HP 48 S series press <┐, <= (row 5 column 5).

2. Press VAR to show your menu instead of one of the machines menus.

3. Press <┐, - to start the command line with <<>>. This is what makes the entry a program object.

4. The first step of the program is to square the number assumed to be on the stack. Press <┐, √X (row 4, column 4).

5. Next we need π. Press <┐, SPC. This places π on level one and the radius squared on level two.

6. Now we multiply with *. Press X. The answer is on level one at this time. The last step is to TAG it for a nice display.

7. Press ┌>, - Alpha, see "" in the command line with the alpha annunciator turned on. The TAG command requires a text object to TAG the level two object. Since we want "Area" we press A. We want the next three letters to be lower case so we lock the alpha mode to lower case with <┐, alpha. Now type rea. Finish with alpha off by pressing the alpha key.

8. Press right cursor to move over the double quotes. Press SPC.

9. The last command in the program is ->TAG. This may be typed or it may be found in PRG, TYPE. On S series machines PRG, OBJ. This finishes the program. It should look like:

   ```
   Area:    << SQ π * "Area" ->TAG >>
   ```

   If this correct press ENTER.

10. The program object is now on level one. To store the program we need a name. Press ' (tic, row 3 column 1), alpha, AREA, ENTER. We now have an object to store on level two, and a named object on level one. This meets the requirements of STO. Press STO (row 3, column 2). The stack is cleared - the HP 48 "consumes" its arguments - and AREA will appear in the left most menu box. Press VAR if needed.

11. Test the program with 1, AREA. This will give the value of π. Try it. Do you get π? Yes? Try 2, AREA. Is this the right answer? Do you want a number? OK, let's "debug" the program and make sure that we get a numerical answer, not a symbolic answer.

12. Recall the program to the stack with: ┌─>, AREA. Next we want to add a command. Press down cursor. This adds the blinking cursor over the open french quote program delimiter. Press the right cursor nime times to move it over the Quote in Area. We wish to insert a command here. Press <─┐, EVAL to enter the ->NUM command. This converts the symbolic π into a numberic π. The edited program now looks like:

```
AREA    << SQ π * ->NUM "Area" ->TAG >>
             31.5 bytes  # B8C0h
```

    It may be difficult to see it all and you may have to press SKIP-> to move to the right. Press ENTER.

13. Now we will store the new program on level one into the menu AREA. This is one of the most dangerous key stroke combinations the new user can use because you may wipe out something you wanted to keep. Press <─┐, menu AREA. This consumes the object as it is stored. Now test the program as before. Run AREA with 2 as an input. Do you get Area: 12.5663706144?

<u>HERE ARE THE BASIC STEPS</u>

1. Start by pressing left shift minus to start the command line with two french quotes << >>. This is a program.

2. Key the objects you want in the program.

3. Press ENTER when you have entered a few objects or the complete program and are ready to test. This places the program object on level one of the stack.

4. Press TIC alpha and the letters that make up the name of the program. Press ENTER when finished. Now you have the program object on level two and a name object on level one.

5. Press STO. The arguments on levels one and two will disappear. The name will appear in the lower left menu. Pressing this key will run the program.

This process should be practiced several times until you can key in the program in just a few seconds. Think about each step and don't be afraid to experiment a little. This type of program is not fancy, but this process will allow you to start writing programs immediately.

Key in the program, test and RUN. Is there an alternative to ->NUM? What roll do system flags -2 and -3 play here? Try alternate methods of entering each object. There are many ways to do what you want. The method you use will depend on the circumstances at the time. It takes time to get familiar with the machine. How long is the program? How many objects are in the program? What is a check sum? How long does it take to run? Could ENTER * be used instead of SQ?