PET USERS' GROUP NEWSLETTER

N-O-T-E-S

Meetings: The East Bay PET Users' Group met 14 March at the Old
Shell Building in Emeryville. Neil Bussey chaired the meeting. An
engineer from Pan-Am discussed his PET test control applications and
Kim Rubin from U.C.Berkeley Physics Dept. elaborated on his customizing
the character set. Milt Lee started organizing the tape library and
taking contributions. There were 40 plus people in attendance with
a dozen PETs. The most notable activity at this meeting was the selec-
tion of a name. Thanks to Jacob Mishell, a young PET enthusiast who
now has a free six-issue subscription, SPHINX was adopted as the new
name for the East Bay Users' Group. SPHINX could stand for Society
of Pet Handlers INformation eXchange.

The South Bay PET Users' Group has also adopted a name: The PET
Users' Group or PUG. Also at the 30 March meeting, Bill Seiler had
a PET with a RAM character set and he was altering the 2001's charac-
ters on-line! John Terry, one of the local PET interfacing experts,
again demonstrated his PET-S100-TTY interface. There were so many
PETs, at least nineteen, that the Mercury Savings circuit servicing
the room faultered when nearly all the PETs were powered up. The
group had to rely on time-slicing a fewer number of computers to
guard against an over-loaded circuit. More than 75 Pet owners and
handlers were in attendance.

N-E-W-S-L-E-T-T-E-R   S-U-B-S-C-R-I-P-T-I-O-N

THIS IS THE LAST FREE PET NEWSLETTER. At the last SPHINX meeting,
members voted to offset the costs of reproduction and mailing by
personal subscription. The cost of one subscription is $3.00 for six
issues starting with the next issue. IF YOU HAVE AN "*" AFTER YOUR
NAME, YOU WILL RECEIVE THE NEXT SIX ISSUES. If you do NOT have an "*",
then you can order a subscription using the form below. Back issues
are now also available:

PET USERS' GROUP NEWSLETTER SUBSCRIPTION AND BACK ISSUE ORDER FORM

Please send me:

( ) Six-issue subscription to PET Newsletter $3.00
( ) Back Issue, Volume 0, Number 0   .75
( ) Back Issue, Volume 0, Number 1   .75

        Your Name _____
          Address _____
       City/State _____   ZIP _____


Send order to: PET Newsletter
                      Lawrence Hall of Science
                      Computer Project
                      University of California
                      Berkeley, CA   94720

T-A-B-L-E   O-F   C-O-N-T-E-N-T-S

P-U-T   A   H-A-N-D-L-E   O-N   Y-O-U-R   P-E-T
### by Alfred Neilson

Our PET travels around from school to home to school. It was dif-
ficult to carry about and get it through doors so we put a handle on
the side. We used a heavy-duty four-screw-hole handle and put it on
the light side (side opposite power transformer). We have used no
bracing at all and find the white cabinet will easily take the strain.
On the other side of the cabinet we placed three rubber feet, two thin
ones (half inch each) and one large one (one and one-half inches)
The only precaution is to avoid getting metal chips inside the cabinet
while drilling for the feet and handle. Use several layers of close
knit fabric to catch any and all chips. Do NOT use sheet metal screws
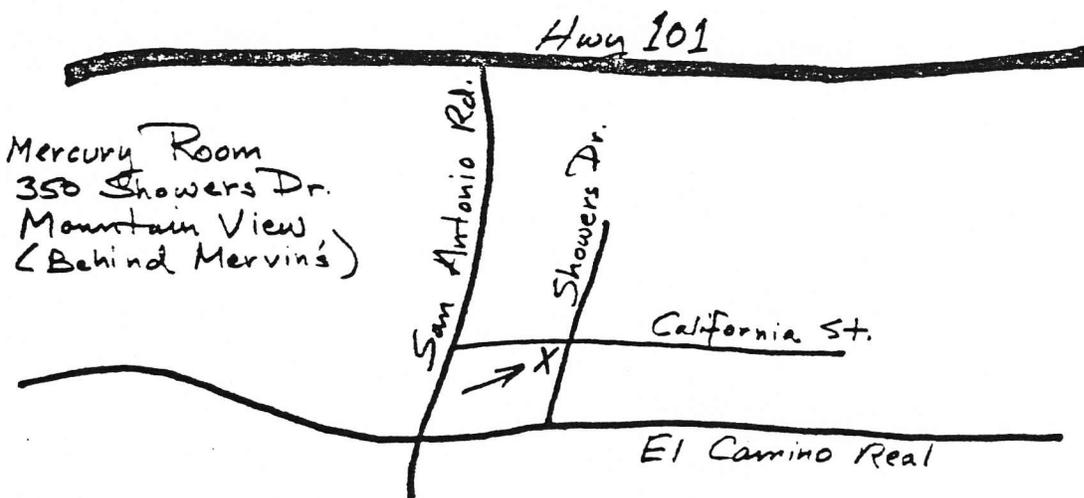for any of the attachments. Use nuts and bolts only.
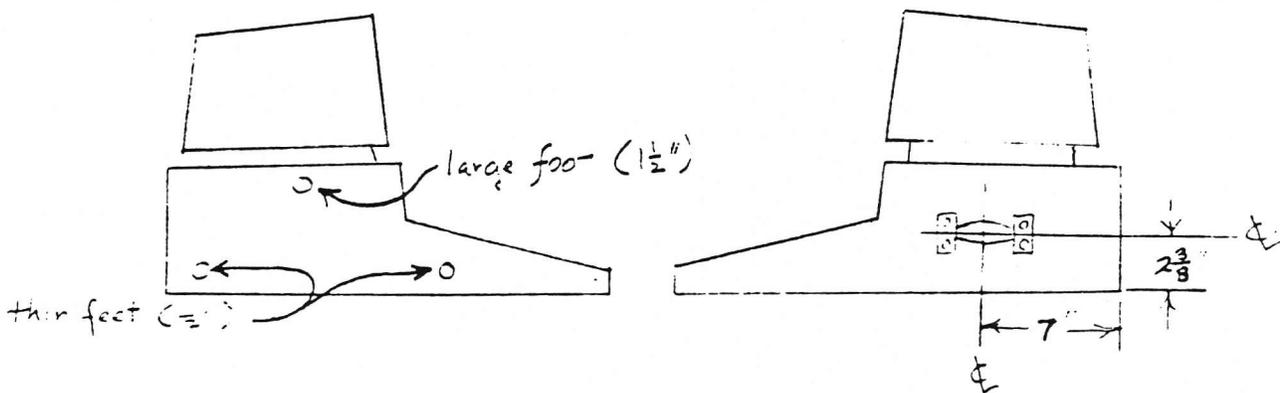
---> SPHINX
     7:00 PM   13 APRIL 1978
     TEACHER SHELTER, OAKLAND PUBLIC SCHOOL ADMINISTRATION BUILDING

Teacher Shelter
O.P.S.
314 10th
Oakland
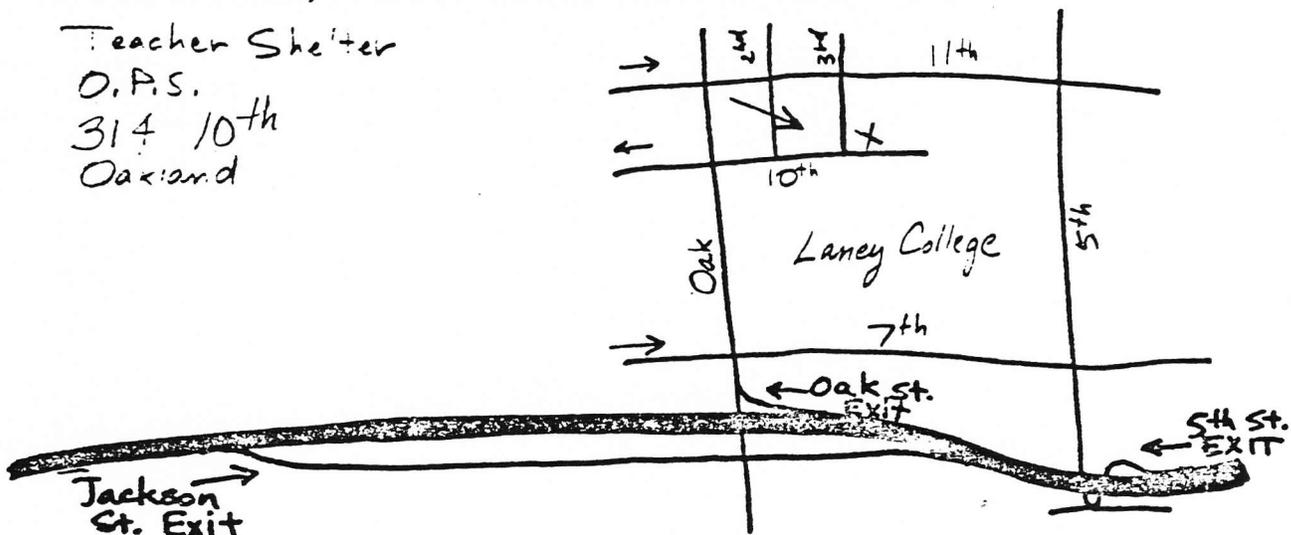
---> PUG (South Bay)
     7:00 PM   3 MAY 1978
   MERCURY SAVINGS, MOUNTAIN VIEW

Mercury Room
350 Showers Dr.
Mountain View
(Behind Mervin's)

A   G-O-L-D   P-E-T

Jack Rossum of San Jose retired recently from California Water
Service Company. Jack's peers consulted with his family about an
appropriate gift for his retirement and thirty five years of service.
One of his sons suggested a PET computer. When I called Jack last
week, he was programming his PET to do curve fitting and graphing
the raw data against the computed curve. Jack said he could provide
the code for his curve fitting and graphing and also the exact for-
mula for the normal distribution, instead of the usual iteration
method. If you are interested, see Mr. Rossum at the next meeting;
he will have the gold-colored PET.

Did you see a hardcopy PET printer at the Second West Coast Computer Faire? You did! Were you impressed? I was, but that was just proto-type number one. It had a five by seven dot matrix that was all right for the ASCII characters without lower case, yet was unable to render the special PET graphic characters. Take a PEEK below. This is proto-type number two. It has a seven by eight dot matrix and Commodore has been furiously working at the complete rendition of all its characters, both normal and reversed! The known features of this prototype set for July release are:

(1)   Will output at 60 characters per second.
(2)   Uses standard teletype paper.
(3)   Uses a 7x8 dot matrix.
(4)   Will cost $595.00
(5)   Will be released July 1978.
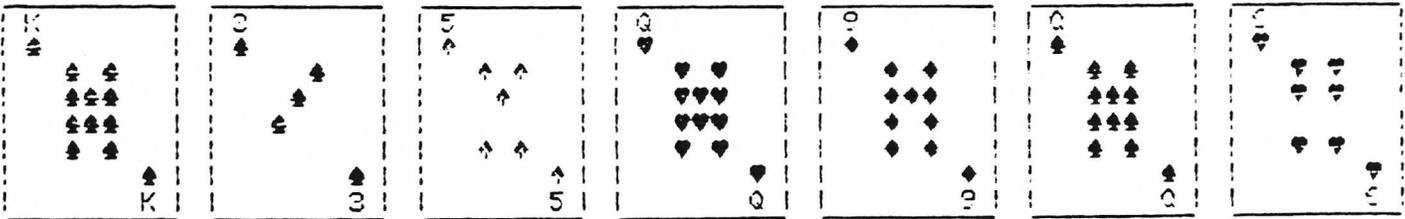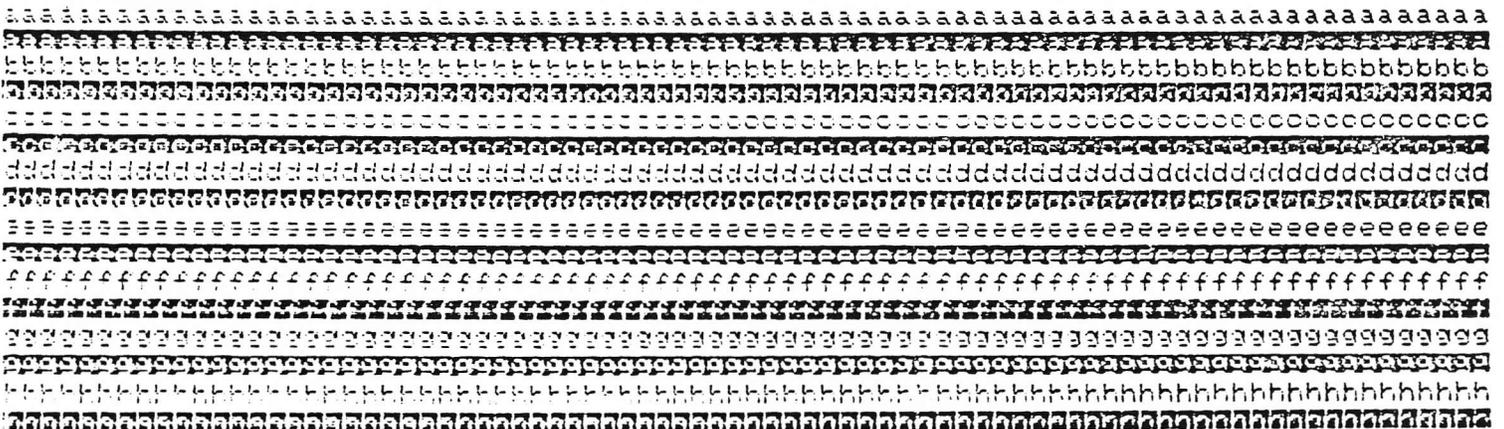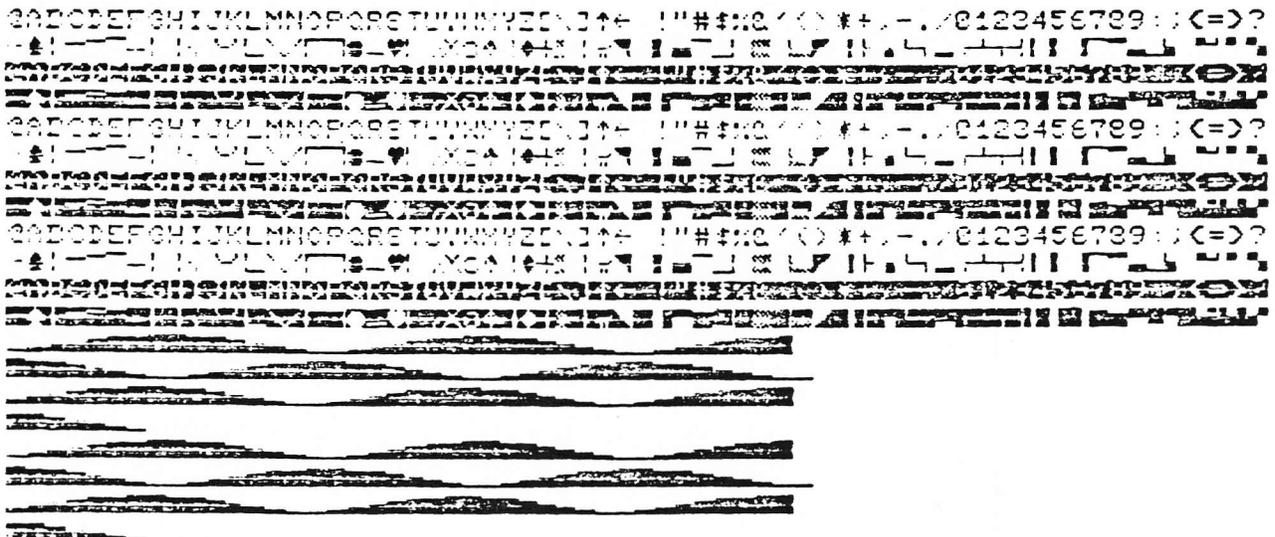
T-H-E A-U-T-O F-L-A-S-H C-U-R-S-O-R M-O-D
by Harry J. Saal

As fine as the PET is, there are a number of little details that could stand improving. One of my PET Peeves is the cursor flashing logic. It mainly flashes on and off at its own pace as I furiously move the cursor across the screen, and is invisible half the time I move it.

An improvement would be to have the cursor always flash "on" whenever a new keystroke is recognized. Now if we could modify ROM, the change would require the following code sequence just prior to the JMP located at $E73C. (This JMP exits from interrupt handling at those times a new key is recognized).

```
        LDX #1
        STX $225        ;force cursor flip next frame
        DEX
        STX $227        ;force inverse video cursor
```

So what can we do about it if we don't want to redo Commodore's ROM? By trapping interrupts ourselves [See also pages 10 to 12 of the last issue, Vol.0, No.1 for interrupt trapping], we can detect the situation with a bit more work. The following BASIC program demonstrates the principle in a rough way. It is left as an exercise to the reader to do it in 6502 code.

```
100 POKE 548,0          :REM ENABLE FLASHING CURSOR
110 GOTO 1800           :REM INITIALIZE OLD CURSOR LOCATION
120                      REM GET A CHARACTER AND ECHO IT
130                      REM TRY THIS FOR LETTERS AND SPACES
140                      REM CURSOR MOVES ARE NOT HANDLED CORRECTLY
150 GET C$:IF C$="" THEN 150
160 PRINT C$;

1000                     REM THIS IS THE "INTERRUPT CODE". CURSOR MOVE?
1100 FOR I=0 TO 2:IF A(I)<>PEEK(I+224) THEN 1500
1200 NEXT I
1300 GOTO 150
1400                     REM RESET CURSOR STATUS AS DESCRIBED
1500 POKE 549,1
1600 POKE 551,0
1700                     REM RECORD CURRENT CURSOR POSITION
1800 FOR I=0 TO 2:A(I)=PEEK(I+224):NEXT I
1900 GOTO 150
```
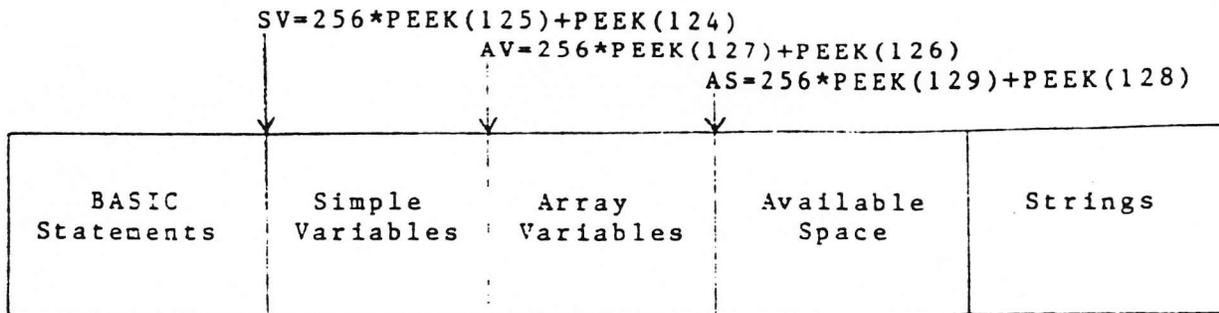
P-E-T B-U-L-L-E-T-I-N-S

Last issue we listed all the Bulletins Commodore was supplying to PET owners. Since then, Commodore has stopped producing these Bulletins and is now compiling them into one booklet for distribution starting at the beginning of next month. Along with the booklet will be TIM, the Machine Language Monitor, and a few unknown mini-programs. To receive this booklet/cassette you must have registered your PET through Commodore by sending in the PET COMPUTER LIMITED WARRANTY sheet supplied with your PET to:

    Sales Department
    Commodore
    901 California Avenue
    Palo Alto, CA 94304

P-E-T   V-A-R-I-A-B-L-E-S   E-X-P-O-S-E-D
by Pete Rowe

$$SV=256*PEEK(125)+PEEK(124)$$
$$AV=256*PEEK(127)+PEEK(126)$$
$$AS=256*PEEK(129)+PEEK(128)$$

BASIC Statements	Simple Variables	Array Variables	Available Space	Strings

In the first PET Newsletter, number 0, a similar chart was printed as the "Principal Pointers in PET RAM". Pointers AV (found in locations126 and 127) and AS (found is locations 128 and 129) seemed to point to the same location. And, infact, when PEEKing at AV and AS during the execution of a program, they did point to the same place when using only simple variables. But as soon as the program dimensioned a string or numeric array, the pointers parted company and contained different values. This prompted an exploration of how variables are stored in PET BASIC.

Each variable definition; whether string, numeric or array; uses seven bytes. the first two bytes determine the variable type:
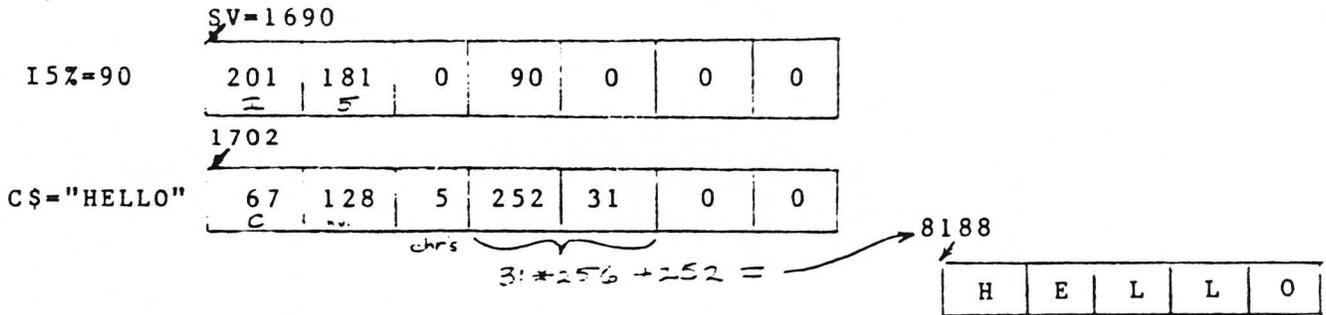
	byte 1	byte 2
INTEGER	first chr + 128	second chr +128 or 128
FLOATING	first chr	second chr or 0
STRING	first chr	second chr + 128 or 128

The remain five bytes are different for each SIMPLE VARIABLE type:

	byte 3	byte 4	byte 5	byte 6	byte 7
INTEGER	actual value 256 * HI	LO	0	0	0
FLOATING	actual value in binary floating point				
STRING	chr count	pointer LO	256 * HI	0	0

Note that for the two simple numeric variable types, the actual values are contained in the seven byte definition. While the simple string variable points to a location in high memory where the actual string characters are contained.

Example:

```
        SV=1690
        ↙
I5%=90  201 | 181 | 0 | 90 | 0 | 0 | 0
          I     5
        1702
        ↙
C$="HELLO" 67 | 128 | 5 | 252 | 31 | 0 | 0
           C    └ chrs ┘
                  31*256+252 = ──────→ 8188
                                       ↓
                                   | H | E | L | L | O |
```

The author has no encoding/decoding algorithm for the five floating point bytes. The algorithm is needed for the USR function and I hope to present some clues or actual code for the next Newsletter.

Arrays carry about the same format except that they begin at the location found in AV, a byte after the simple variables end. For vector arrays, the third thru seventh bytes in the definition are:

	byte 3	byte 4	byte 5	byte 6	byte 7
VECTOR ARRAYS	7+(size+1)* (dim)*A	0	1	0	size + 1

where A = 2 for integer, = 3 for string, or = 5 for floating

The difference between simple variables and arrays is that after the seven byte definition comes the actual data for numeric arrays and character counts/pointers for string arrays.

Here is a BASIC program to list all your variables:

```
1000    I=0:SV=0:AV=0:AS=0
1010    SV=256*PEEK(125)+PEEK(124)
1020    AV=256*PEEK(127)+PEEK(126)
1030    AS=256*PEEK(129)+PEEK(128)
1040    FOR I=SV TO AV-1 STEP 7
1050    GOSUB 1200:PRINT
1060    NEXT I
1070    REM --DECODE ARRAYS--
1080    I=AV
1090    IF I>=AS THEN RETURN
1100    GOSUB 1200:PRINT " ARRAY"
1110    I=PEEK(I+2)+I
1120    GOTO 1090
1200    IF PEEK(I)<128 THEN 1220
1210    PRINT CHR$(PEEK(I)-128);CHR$(PEEK(I+1)-128);"%";:RETURN
1220    IF PEEK(I+1)>127 THEN 1240
1230    PRINT CHR$(PEEK(I));CHR$(PEEK(I+1));:RETURN
1240    PRINT CHR$(PEEK(I));CHR$(PEEK(I+1)-128);"$";:RETURN
```

To use the above subroutine, simply define one or more variables and GOSUB 1000, all in immediate mode. Example:

Z3%=54:A$(5)="HELLO":RT=1E22:GOSUB 1000

(Challenges for the reader: 1. Why do we need line 1000?
    and 2. Can you devise a program to decode the bytes containing
    the actual data in both numeric and string variables, simple and
    array?)

P-E-T   C-L-A-S-S-E-S

As part of a summer program, the Computer Project at Lawrence Hall of
ience will offer five classes designed specifically for PETs. Classes
begin July 10th -- brochures will be available in early May and sign-ups
start soon after that. As a special service for the PET user groups we are
previewing class descriptions. If you are interested in signing up for a
class, call L.H.S. (415 642-5132) and ask to receive the summer brochure.
(There will be 12 PETs available for each class -- enrollments are limited.)

An Introduction To Computers for Teachers

An opportunity for classroom teachers and school administrators to learn
about computers and how they enrich the students educational experience.
Teachers will learn how to operate a "personal" computer. The L.H.S. timeshare
computer, and various small computers will be available for use and comparison.
Programming will be taught. This class will prepare teachers and administrators
to choose a computer that will best meet the needs of their students and present
some ideas and activities for classroom use.

Programming in BASIC

Students will learn to program a personal computer using the BASIC program-
ming language. Students will be grouped according to experience so as to allow
he instructors to provide individualized instruction. With the guidance of the
instructors, students will learn to develop their own computer programs.

Computer Lab with Personal Computers

Do you want guaranteed access to a PET "personal" computer? Here is an
opportunity to write your own programs, or play any games from our tape library.
A staff member will be available for consultation.

Advanced Uses of Personal Computers

This class is directed toward "owners and trainers" of the Commodore PET.
Through individual projects the class will explore extended capabilities of
personal computers. Particular emphasis will be placed on the use of graphics
and the PET's machine language.

Programming Without Numbers

Teach the computer to write your name; fill the screen with designs you
create and change. You will learn simple aspects of computer programming as
u work with a PET personal computer. Emphasis will be on non-mathematical
approaches to programming; if you are shy of numbers, you will feel comfortable
here.