STARTING:
1. INSERT LANGUAGE DISK
2. TURN ON APPLE *//e* or (CONTROL + ♁ + RESET)

APPLE® LOGO //

(PRODUCT of APPLE COMPUTER INC)

SCREEN EDITOR COMMANDS

EDIT *name*	Enter EDITOR
♁ ?	HELP menu (ESC = EXIT)
←	LEFT one character
→	RIGHT one character
↑	UP one line
↓	DOWN one line
♁←	LEFT one WORD
♁→	RIGHT one WORD
♁,	BEGINNING of LINE
♁.	END of LINE
♁↑	TOP of PAGE
♁↓	BOTTOM of PAGE
♁*n*	Move to rel position *n* (1-9)
DELETE	DELETE PREVIOUS character
CONTROL D	DELETE PREVIOUS character
CONTROL F	DELETE CHARACTER
CONTROL X	DELETE LINE
CONTROL Y	DELETE to END of LINE
CONTROL R	RECALL last deleted
CONTROL O	CREATE NEW LINE
♁A	EXIT EDITOR
♁ ESC	EXIT EDITOR (NOT saved)

SCREEN COMMANDS

FS	FULL SCREEN (or CONTROL L)
SS	SPLIT SCREEN (or CONTROL S)
TS	TEXT SCREEN (or CONTROL T)
CT	CLEAR TEXT SCREEN (homes cursor)
CS	CLEAR SCREEN (homes turtle)
CLEAN	CLEAR SCREEN
CURSOR	Outputs Cursor Position
SETCURSOR [*c l*]	Positions to *column* and *line*
SETWIDTH *n*	Set Screen width (*n* = **40** or **80**)
WIDTH	Outputs current screen width

PEN INFORMATION

BG	Background color
PEN	State of turtles pen
PC	Pen color
DOTP [*x y*]	TRUE if dot is at *x y*; else FALSE

PEN COMMANDS

PU	Pen UP
PD	Pen DOWN
DOT [*x y*]	OUTPUTS DOT at location *x,y*
FENCE	Keeps TURTLE WITHIN SCREEN
FILL	FILLS SHAPE with current color
PE	ERASES LINE
PX	Pen down reversing colors
SETBG *n*	Background color *n*
SETPC *n*	Pen color *n*
WINDOW	Makes turtle field unbounded
WRAP	Makes turtle field wrap around

TURTLE MOVEMENT

HT	HIDE TURTLE
ST	SHOW TURTLE
FD *n*	FORWARD *n* STEPS
BK *n*	BACKWARD *n* steps
RT *n*	RIGHT TURN *n* degrees
LT *n*	LEFT TURN *n* degrees
HOME	HOME turtle
SETX *n*	Move HORIZONTALLY to *n*
SETY *n*	Move VERTICALLY to *n*
SETH *n*	SET HEADING *n* degrees
XCOR	Turtle *x* coordinate
YCOR	Turtle *y* coordinate
HEADING	Turtle direction in degrees
CS	Erase screen and home cursor
SETPOS (*x y*)	Move to point *x y*
POS	Coordinates of turtle
SHOWNP	TRUE if not hidden, else FALSE
TOWARDS (*x y*)	Points turtle toward *x y*

WORKSPACE MANAGEMENT

NODES	OUTPUTS number of Free nodes
RECYCLE	Performs Garbage Collection
PO *name*	Prints the *named* procedures and variables
POALL	Prints out all procedures and variables
PON *name*	Prints out Name and Value of *named* variables
PONS	Prints Name and Value of every variable
POPS	Prints Out Procedures in workspace
POT *name*	Prints title of *named* procedures
POTS	Prints all title lines
ERALL	Erase all procedures, variables and properties
ER *name*	Erase the *named* procedures
ERN *name*	Erase the *named* variables
ERNS	Erase ALL variables
BURY *name*	Buries listed procedures
BURYALL	Buries all procedures and variables
BURYNAME *name*	Buries listed variables
UNBURY *name*	Unburies the listed procedures
UNBURYALL	Unburies all procedures and variables
UNBURYNAME *name*	Unburies the listed variables

NOTE: The BURY commands protect procedures and variables from commands ERALL, ERPS, POALL, POPS, POTS, and SAVE.

MISCELLANEOUS COMMANDS

.AUXDEPOSIT *loc value*	Stores *value* at memory location *loc*
.AUXEXAMINE *loc*	Returns value of memory at location *loc*
.BLOAD *pathname loc*	Loads binary file starting at location *loc*
.BSAVE *pathname loc n*	Saves memory starting at *loc n* bytes long
.CALL *loc*	Calls machine language subroutine at *loc*
.DEPOSIT *loc value*	Stores *values* at memory location *loc*
.EXAMINE *loc*	Returns *value* of memory at location *loc*
.SCRUNCH	Outputs ratio of Vertical step to Horizontal
.SETSCRUNCH *n*	Sets aspect ratio to *n*
.CONTENTS	Lists all objects LOGO knows about
.QUIT	Safe way to exit LOGO (closes all files)

NOTE: AUXDEPOSIT and AUXEXAMINE work in the auxiliary memory bank and not main memory like DEPOSIT and EXAMINE.

PROCEDURE COMMANDS

TO *name*	CREATE procedure
END	TERMINATE procedure
name	Name of file or variable
:*name*	Value of a variable
[]	List

NAMING

MAKE *name object*	Variable *name* = *object*
NAME *object name*	Variable *name* = *object*
THING *name*	Outputs the VALUE of its input
ED *name*	Invoke LOGO EDITOR (Procedures)
EDN *name*	Invoke LOGO EDITOR (Variables)
LOCAL *name*	Make variable LOCAL procedure
NAMEP *word*	TRUE if word exists; else FALSE

CONTROL

GO *word*	Go to label
OP	Causes OUTPUT to calling procedure
REPEAT *n* [...]	REPEATS *list... n* times
RUN *list*	EXECUTES a *list*
STOP	STOPS current procedure
WAIT *n*	WAIT for *n* 60ths of a second
CATCH *name proc*	Jump to Procedure *proc*
THROW *name*	RETURN to CATCH with *name*
ERROR	Outputs ERROR condition
LABEL *word*	Make *name* a LABEL

CONDITIONALS

IF *exp list1 list2*	If *exp* true execute *list1*; else execute list2
TEST	Test condition to be TRUE or FALSE
IFF	Execute if FALSE (after TEST)
IFT	Execute if TRUE (after TEST)

DEBUG COMMANDS

STEP *name*	Execute Procedure name line by line
TRACE *name*	Put Procedure *name* in Trace mode
UNSTEP *name*	Take Procedure *name* out of STEP mode
UNTRACE *name*	Take Procedure *name* out of TRACE mode
♁ ESC	STOPS whatever is running
CONTROL W	Interrupts whatever is running (any key resumes)
CONTROL Z	Cause running procedure to PAUSE
PAUSE	PAUSE between executions (same as CONTROL Z)
CO	CONTINUE after pause

GENERAL FILE MANAGEMENT

CATALOG	Prints names of files in current directory
CREATEDIR *pathname*	Creates subdirectory *pathname*
EDITFILE *pathname*	Loads file into edit buffer
ERASEFILE *pathname*	Erase file from disk
FILEP *pathname*	TRUE if file exists on disk
LOADHELP *pathname*	Loads file into main help screen memory
ONLINE	Outputs name of each disk in drives
POFILE *pathname*	Prints file onto screen
PREFIX	Outputs current PRODOS prefix
RENAME *pathname newpathname*	Changes name
SETPREFIX *prefix*	Set PRODOS prefix to *prefix*

MANAGING VARIOUS FILES

LOAD *pathname*	LOAD FILE into workspace
SAVE *pathname*	SAVES all UNBURIED PROCEDURES and VARIABLES
SAVEL *name pathname*	SAVES the NAMED PROCEDURES and all UNBURIED VARIABLES
LOADPIC *pathname*	LOAD PICTURE FILE
PRINTPIC *slotnbr*	PRINT GRAPHICS SCREEN to printer in listed *slotnbr*
SAVEPIC *pathname*	SAVE PICTURE FILE
DRIBBLE *pathname*	SEND CHARACTERS FROM SCREEN TO FILE
NODRIBBLE	Turn OFF DRIBBLE
ALLOPEN	LISTS ALL FILES and DEVICES that are OPEN
CLOSE *file*	CLOSES FILE
CLOSEALL	CLOSES ALL open FILES
FILELEN *pathname*	List LENGTH of FILE
OPEN *file*	OPENS FILE
READER	LISTS CURRENT OPEN READ FILE
READPOS	OUTPUT POSITION in current READ FILE
SETREAD *file*	SETS current READER to *file*
SETREADPOS *n*	SETS READ POSITION to *n* in current reader
SETWRITE *file*	SETS current WRITER to *file*
SETWRITEPOS *n*	SETS WRITE POSITION to *n* in current writer
WRITEPOS	OUTPUTS current WRITE POSITION
WRITER	LISTS CURRENT OPEN WRITE FILE

TO FORMAT DISK

1. Make sure LOGO II disk is in drive 1
2. Load "FORMAT"
3. Remove LOGO disk and insert blank disk
4. Enter *slot#* (6); RETURN
5. Enter *drive#* (1); RETURN
6. Enter *volume name* ; RETURN
7. ♁ ESC returns to Logo

PRINT DISK FILE TO PRINTER

TO DUMP : *file*
DRIBBLE 1
PROFILE : *file*
NODRIBBLE
END

NOTE: To use DUMP "file also prefix must be set to directory containing file (see "SETPREFIX")

APPLE® LOGO //

WORDS AND LIST OPERATIONS

=	COMPARES numbers, words, lists
BF *obj*	All BUT the FIRST characters or element
BL *obj*	All BUT the LAST characters or element
FIRST *obj*	FIRST character of word or element of list
FPUT *obj list*	FIRST INPUT followed by second input
LAST *obj*	LAST character or element
LIST *obj1 obj2 ...*	LIST of the inputs
LPUT *obj list*	LIST of second input followed by FIRST
SE *obj1 obj2 ...*	SENTENCE combines all inputs into a list
WORD *word1 word2 ...*	CONNECTS words
ITEM *n obj*	OUTPUTS *n*th element of *object*
MEMBER *obj1 obj2*	OUTPUTS part of *obj1* in which *obj2* is 1st element
PARSE *word*	OUTPUTS list obtained from parsing word
ASCII *character*	OUTPUTS ASCII code for *character*
BEFOREP *word1 word2*	TRUE if *word1* comes before *word2*
CHAR *n*	OUTPUTS character whose ascii code is *n*
COUNT *obj*	OUTPUTS number of elements in *obj*
EMPTYP *obj*	TRUE if *obj* is an empty word or list
EQUALP *obj1 obj2*	TRUE if object *obj1* and *obj2* are equal
LISTP *obj*	TRUE if *obj* is a list
MEMBERP *obj1 obj2*	TRUE if *obj1* is a member of *obj2*; else FALSE
NUMBERP *obj*	TRUE if *obj* is a number
WORDP *obj*	TRUE if *obj* is a word
LOWERCASE *word*	OUTPUTS *word* in lower case letters
UPPERCASE *word*	OUTPUTS *word* in uppercase letters

INPUT/OUTPUT

OP *object*	OUTPUT *object* from procedure
PR *object*	Prints *object* on screen
RC	Read characters from keyboard or file
BUTTONP *buttonnbr*	TRUE if button on paddle is down
PADDLE *paddlenbr*	OUTPUTS number between 0-255 (dial rotation)
KEYP	TRUE if character is waiting to be read
RCS *n*	OUTPUTS first *n* characters read
RL	Reads line from file and outputs as list
RW	Read word from file
SHOW *object*	Display *object* followed by carriage return
TYPE *object1 object2 ...*	Display objects without carriage return
TOOT *freq duration*	Generates a tone

PROPERTY COMMANDS

PPROP *name prop* [*obj*]	Property of *name* = *object*
PPS	Prints property list
PLIST *name*	Outputs property list of *name*
GPROP *name property*	Get property of *name*
ERPROPS	Erase all properties
REMPROP *name property*	Remove *property* from list of *name*

UNDER PROGRAM CONTROL

COPYDEF *name newname*	COPY DEFINITION
DEFINE *name list*	CREATE PROCEDURE *name* with *list*
DEFINEDP *word*	TRUE if *word* is NAME OF PROCEDURE
PRIMITIVEP *name*	TRUE if *name* is a LOGO COMMAND
TEXT *name*	OUTPUTS DEFINITION of *name* as list

TOOT FREQUENCY

Note	Frequency, by Octave							
B	62	123	247	494	988	1973	3946	
A#	58	117	233	466	932	1864	3743	
A	55	110	220	440	881	1761	3510	
G#	52	104	208	415	830	1663	3327	
G	49	98	196	392	784	1566	3142	
F#	46	92	185	370	740	1480	2959	
F	44	87	175	349	698	1398	2797	
E	41	82	165	330	659	1319	2637	
D#	39	78	156	311	622	1244	2495	4990
D	37	73	147	294	587	1176	2346	4713
C#	35	69	139	277	554	1109	2213	4426
C	33	65	131	262	523	1047	2095	4172

Middle C

LEROY'S CHEATSHEET®

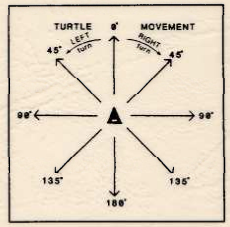
COLORS

0	BLACK
1	WHITE
2	GREEN
3	VIOLET
4	ORANGE
5	BLUE
6	BLACK (Background)

(for B/W TV)

ARITHMETIC OPERATORS

+	ADDITION
-	SUBTRACTION
*****	MULTIPLICATION
/	DIVISION
>	GREATER THAN
<	LESS THAN



ARITHMETIC OPERATIONS

COS *n*	COSINE of angle *n*	**SQRT** *n*	SQUARE ROOT OF *n*	**INTQUOTIENT** *x y*	INTEGER QUOTIENT of *x*/*y*
REMAINDER *f e*	REMAINDER after divide of *f*/*e*	**ARCTAN** *n*	Arc Tangent of *n*	**PRODUCT** *x y z ...*	PRODUCT of *x*y*z...*
RANDOM *n*	RANDOM NUMBER (less than *n*)	**DIFFERENCE** *x y*	OUTPUT = *x-y*	**QUOTIENT** *x y*	REAL QUOTIENT of *x*/*y*
ROUND *n*	Rounds REAL number	**FORM** *n f p*	Outputs *n* in *f* spaces with *p* dec pos	**RERANDOM**	Reproduce Previous RANDOM Sequence
SIN *n*	SINE of angle *n*	**INT** *n*	Converts REAL to INTEGER	**SUM** *x y z ...*	SUM of *x+y+z...*

LOGICAL OPERATIONS

AND *x y z ...*	TRUE if all inputs are TRUE; else FALSE
NOT *x*	TRUE if all inputs are FALSE; else FALSE
OR *x y z ...*	FALSE if all inputs are FALSE; else TRUE

CHEATSHEET PRODUCTS INC
P.O. BOX 111366 PGH., PA 15238
(412) 781-1551

LEROY'S CHEATSHEET® - Keyboard Overlay for the APPLE //e

LC1AE-LA1