



# ATP164

## First Step in AutoLISP Programming

Segment 3

**Date:** January 22, 2007

**Instructor:** Mohmed Zuber Shaikh

**Level:** Beginning

**Category:** AutoLISP

**Web:** [www.AUGI.com](http://www.AUGI.com)

**Reuse of any or all material contained within this document for commercial purposes, without the express written consent of AUGI, Inc. or its authorized agents is expressly prohibited.**

© Copyright 2004 Autodesk User Group International, Inc. All rights reserved.

## Introduction

AutoLISP can handle files for Reading, Writing and Appending. This feature is useful for writing a program where input for program is stored in files. The Writing mechanism is useful for recording Design OR Bill of Material. The Program Users list etc. can be created by using the Appending feature of program. In short you can write an AutoLISP program capable of ----

- Reading data from file
- Creating Drawing File
- Creating Design file
- Creating Bill of Material

## File Handling in AutoLISP

### 6.1 STREAMS

Files could be considered as streams of data. AutoLISP can read from external files and write to them. You may store drawing parameters in an ASCII file on your hard disk which can be imported in LISP programme to generate the drawing. There is a direct interface available with 'C' language and Visual Basic.

AutoLISP allows files to be opened in only three modes. They are "r" for reading , "w" for writing and "a" for appending.

**"r" Read**                      To open a file for reading. If the file does not exist a nil will be returned.

**"w" Write**                      To open a file for writing. If the file doesn't exist then file is created and opened for writing and if it exists the file is overwritten.

**"a" Append**                      To open a file to append. If the file exists then writing starts from where it last stopped. If the file doesn't exist then a file is created and opened.

### 6.2 OPEN

A file in AutoLISP is opened by using a function called open. The open function requests the operating system to allocate a slot to a file. Your program can open many files at a time; you need a separate file handler for each file. Your program can open fixed name file or the file name as specified by the user.

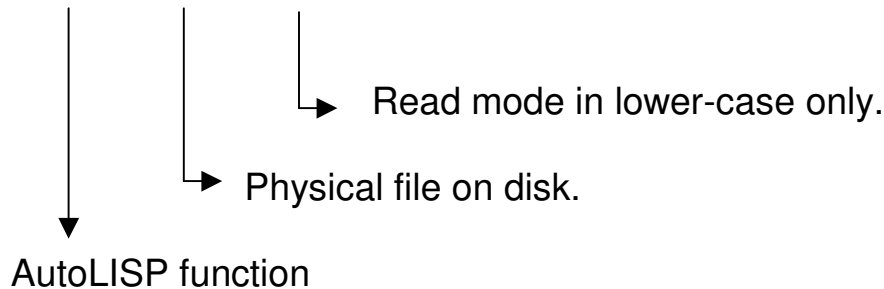
**Reuse of any or all material contained within this document for commercial purposes, without the express written consent of AUGI, Inc. or its authorized agents is expressly prohibited.**

© Copyright 2004 Autodesk User Group International, Inc. All rights reserved.

Syntax:

```
(open <filename> <mode>)
```

```
(open "filename.ext" "r")
```



The open function returns a file descriptor to be used by the other I/O functions. The file descriptor must be assigned to a symbol that uses the setq function.

```
(setq fh (open <filename> <mode>))
```

File Descriptor stored in variable fh

The targeted file is opened for operation or something goes wrong. The function findfile allows user programs to search the AutoCAD library path for a specified file.

Syntax:

```
(findfile <filename>)
```

The findfile function searches for a file supplied to it in the current directory, if it is not found, it will search in the path. Findfile is very important function to ascertain the existence of particular file on disk. An example below shows how the error check can be placed in programs handling files. The *Alert* function displays an alert dialog box with message passed as string.

```
(Setq nm (getstring "\n ENTER NAME OF DATA FILE : "))  
(graphscr)  
(if (= (findfile nm) nil)  
    (alert "File not found.")  
)
```

## 6.3 READ-LINE

The first call to read-line returned a string that was the first line in the file. Read-line takes one parameter and returns a string.

Syntax :

```
(setq nm (read-line fh))
```

Read-line function, with the file descriptor as its parameter, continues to reads the lines until read-line function returns a nil. The read-line could also be used without any parameter. If the read-line is called without a parameter, then it reads from the keyboard buffer. It stops at the moment, when the user presses a return.

Example:

ABC.DAT

ABC.LSP

ASCII file

AutoLISP Programme

```
(setq fh (open "ABC.DAT" "r"))
```

ABC LTD.

```
(setq cnm (read-line fh))
```

MY

```
(setq ln2 (read-line fh))
```

001

```
(setq ln3 (read-line fh))
```

FIRST

002

TRIAL

003

WITH

004

STREAM

```
(setq ln10 (read-line fh))
```

```
(close fh)
```

Create the ASCII file ABC.DAT as shown above. Read the content using AutoLISP programme and display the same on Graphics screen.

**Reuse of any or all material contained within this document for commercial purposes, without the express written consent of AUGI, Inc. or its authorized agents is expressly prohibited.**

© Copyright 2004 Autodesk User Group International, Inc. All rights reserved.



## 6.4 CLOSE

The opened file must be closed. The function **close** does this job.

Syntax:

(close <filedesc>)

The read-line function returns always a string, even the numeric data like 001 in the above example would be a string for operation in AutoLISP programme. We have to convert the string to an integer or real value using Atoi and Atof function for further mathematical operation.

## 6.5 Write-line

Now, so far we were talking about "how to read" now lets learn "how to write".

Syntax :

(write-line <string> [ <file-desc> ])

This function writes <string> to the screen or to the open file described by <file-desc>. It returns <string> quoted in the normal manner, but omits the quotes when writing to the file.

Example:

Don't forget to close the file. AutoLISP writes the string in an area called the file buffer, and the contents of the buffer are flushed, only when 512 bytes go in to the buffer Or when the file is closed.



## 6.6 Sample program for Drawing of Ground Profile and Road top

This sample program accepts site levels of ground and draws it as sequence of lines. The Road profile is created as single top-most formation line. The code of this program will pave a path for your lengthy program writing in the future. The sample site data are recorded in ASCII file CSr.txt as below. You can write the same data in Notepad using same sequence as shown below (one data per line). This file will be actually an input file for your main AutoLISP program CSRoad.lsp.

### CSr.txt

```
1200
7.50
16.0
10
0
12.5
2.5
13.56
3.5
13.2
5.5
14.5
2.25
12.78
4.0
13.05
5.5
11.80
```

DON'T WRITE ANYTHING BEYOND 11.80. THE DATA 11.80 IS THE last line of file CSr.txt. Save it.

### THE MEANING OF EACH DATA IS EXPLORED AS BELOW.

The First data 1200 is Chainage of Cross Section  
7.50 is Formation Width of Road  
16.0 is Formation Level of Road  
10 is Datum Level for drawing CS  
0 is CS center Chainage  
12.5 is Ground Level at 0 chainage  
2.5 is distance on right side for second ground level  
13.56 Ground level at 2.5 meter distance  
3.5 Distance Right Side  
13.2 Ground Level at 3.5 distance  
5.5 Distance Right Side  
14.5 Ground Level at 5.5 distance  
2.25 Distance Left Side  
12.78 Ground Level at 2.25 mt  
4.0 Distance Left Side  
13.05 Ground Level at 4.0 mt

**Reuse of any or all material contained within this document for commercial purposes, without the express written consent of AUGI, Inc. or its authorized agents is expressly prohibited.**

© Copyright 2004 Autodesk User Group International, Inc. All rights reserved.



5.5 Distance Left Side  
11.80 Ground Level at 5.5 mt

## WRITING OF ACTUAL PROGRAM

You may write the following code in CSroad.lsp file using Notepad or Visual LISP editor. The figures appears beside the code are my attempt to link code with actual graphics result. This way you can clearly understand the code and how to write it for particular result.

**;CSroad.lsp**  
**;Program for cross section of ground & road.**

```
(command "-style" "zz" "romans" "" "" "" "" "" "")
```

```
(setq flnm (getfiled "Pick inputfile for Cross Section of Road" "/program files/Autocad 2006/" "" 0))
```

;As alternative to above line, instead of calling dialogue box you can directly use following code for ;name of input file. The input file must be in the search path of AutoCAD

```
;(setq flnm (getstring "\n Enter name of input file : "))  
;(if (= (findfile flnm) nil)  
; (PROGN  
; (alert "File not found.")  
; (setq flnm (getstring "Try again with correct file name:"))  
; )  
;)
```

```
(setq fh (open flnm "r"))
```

```
(setq ch (read-line fh))  
(setq tr (atof(read-line fh)))  
(setq fl (ATOF (read-line fh)))  
(setq datum (ATOi (read-line fh)))  
(setq ddummy datum)  
(setq ch0 (ATOF (read-line fh)))  
(setq gl0 (ATOF(read-line fh)))  
(setq rch1 (ATOF(read-line fh)))  
(setq rgl1 (ATOF(read-line fh)))  
(setq rch2 (ATOF(read-line fh)))  
(setq rgl2 (ATOF(read-line fh)))  
(setq rch3 (ATOF(read-line fh)))  
(setq rgl3 (ATOF(read-line fh)))  
(setq lch1 (ATOF(read-line fh)))  
(setq lgl1 (ATOF(read-line fh)))  
(setq lch2 (ATOF(read-line fh)))  
(setq lgl2 (ATOF(read-line fh)))  
(setq lch3 (ATOF(read-line fh)))  
(setq lgl3 (ATOF(read-line fh)))
```

```
(close fh)
```

**Reuse of any or all material contained within this document for commercial purposes, without the express written consent of AUGI, Inc. or its authorized agents is expressly prohibited.**

© Copyright 2004 Autodesk User Group International, Inc. All rights reserved.

```
(command "units" 2 2 "" "" "" "")
(command "limits" "(0 0) '(40 30))
(command "zoom" "all")
```

```
(command "pline" '(0 0) "w" "0.1" "" '(40 0) '(40 30) '(0 30) "c")
```

```
(setq stpt (list 25 3.0))
(setq p2 (list 6 3.0))
(setq p3 (list 37 3.0))
```

```
(command "line" p2 p3 "")
```

```
(command "array" "l" "" "r" 6 "" 1.5)
```

```
(setq dp (list 6 15))
(setq dp1 (list (- (car dp) 1.5) (cadr dp)))
(setq dp2 (list 37 15))
```

```
(setq dp3 (list 6 25))
```

```
(setq fw2 (/ tr 2.0))
```

```
(setq tp1 (list 1.5 3.6))
```

```
(command "text" tp1 0.20 "" "DISTANCE IN METERS.")
```

```
(setq tp2 (list 1.5 (+ (cadr tp1) 1.5)))
```

```
(command "text" tp2 0.20 "" "GROUND LEVALES.")
```

```
(setq tp3 (list 1.5 (+ (cadr tp2) 1.5)))
```

```
(command "text" tp3 0.20 "" "FORMATION LEVALES.")
```

```
(setq tp4 (list 1.5 (+ (cadr tp3) 1.5)))
```

```
(command "text" tp4 0.20 "" "HEIGHT OF BANK.")
```

```
(setq tp5 (list 1.5 (+ (cadr tp4) 1.5)))
```

```
(command "text" tp5 0.20 "" "DEPTH OF CUTTING.")
```

```
(setq tp6 (list 1.5 (+ (cadr tp5) 5.40)))
```

```
(command "text" tp6 0.3 "" "DATUM")
```

```
(setvar "pdmode" 4)
```

```
(setvar "pdsz" 0.2)
```

```
(SETQ MPT (LIST (+ (CAR P2) 15) (CADR P2)))
```

```
(SETQ MPT1 (LIST (CAR MPT) (+ (CADR MPT) 0.3)))
```

```
(COMMAND "TEXT" MPT1 0.2 90 (RTOS CH0 2 2))
```

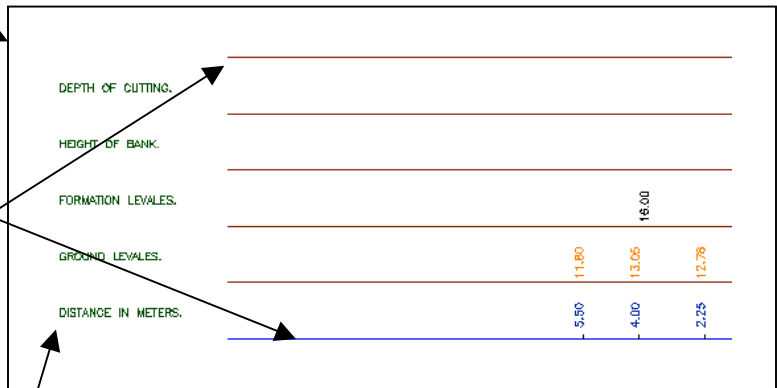
```
(setq rpt1 (list (+ (car mpt1) rch1) (cadr mpt1)))
```

```
(command "text" rpt1 0.2 90 (rtos rch1 2 2))
```

```
(setq rpt2 (list (+ (car mpt1) rch2) (cadr mpt1)))
```

```
(command "text" rpt2 0.2 90 (rtos rch2 2 2))
```

```
(setq rpt3 (list (+ (car mpt1) rch3) (cadr mpt1)))
```



Reuse of any or all material contained within this document for commercial purposes, without the express written consent of AUGI, Inc. or its authorized agents is expressly prohibited.

© Copyright 2004 Autodesk User Group International, Inc. All rights reserved.



```
(command "text" rpt3 0.2 90 (rtos rch3 2 2))
```

```
(setq lpt1 (list (- (car mpt1) lch1) (cadr mpt1)))
```

```
(command "text" lpt1 0.2 90 (rtos lch1 2 2))
```

```
(setq lpt2 (list (- (car mpt1) lch2) (cadr mpt1)))
```

```
(command "text" lpt2 0.2 90 (rtos lch2 2 2))
```

```
(setq lpt3 (list (- (car mpt1) lch3) (cadr mpt1)))
```

```
(command "text" lpt3 0.2 90 (rtos lch3 2 2))
```

```
(setq mpt11 (list (car mpt1) (- (cadr mpt1) 0.3)))
```

```
(command "point" mpt11)
```

```
(setq rpt11 (list (car rpt1) (- (cadr rpt1) 0.3)))
```

```
(command "point" rpt11)
```

```
(setq rpt22 (list (car rpt2) (- (cadr rpt2) 0.3)))
```

```
(command "point" rpt22)
```

```
(setq rpt33 (list (car rpt3) (- (cadr rpt3) 0.3)))
```

```
(command "point" rpt33)
```

```
(setq lpt11 (list (car lpt1) (- (cadr lpt1) 0.3)))
```

```
(command "point" lpt11)
```

```
(setq lpt22 (list (car lpt2) (- (cadr lpt2) 0.3)))
```

```
(command "point" lpt22)
```

```
(setq lpt33 (list (car lpt3) (- (cadr lpt3) 0.3)))
```

```
(command "point" lpt33)
```

```
(SETQ MPT2 (LIST (CAR MPT) (+ (CADR MPT) 1.6)))
```

```
(COMMAND "TEXT" MPT2 0.2 90 (RTOS gl0 2 2))
```

```
(setq glrpt1 (list (+ (car mpt2) rch1) (cadr mpt2)))
```

```
(command "text" glrpt1 0.2 90 (rtos rgl1 2 2))
```

```
(setq glrpt2 (list (+ (car mpt2) rch2) (cadr mpt2)))
```

```
(command "text" glrpt2 0.2 90 (rtos rgl2 2 2))
```

```
(setq glrpt3 (list (+ (car mpt2) rch3) (cadr mpt2)))
```

```
(command "text" glrpt3 0.2 90 (rtos rgl3 2 2))
```

```
(setq gllpt1 (list (- (car mpt2) lch1) (cadr mpt2)))
```

```
(command "text" gllpt1 0.2 90 (rtos lgl1 2 2))
```

```
(setq gllpt2 (list (- (car mpt2) lch2) (cadr mpt2)))
```

```
(command "text" gllpt2 0.2 90 (rtos lgl2 2 2))
```

```
(setq gllpt3 (list (- (car mpt2) lch3) (cadr mpt2)))
```

```
(command "text" gllpt3 0.2 90 (rtos lgl3 2 2))
```

```
(SETQ MPT3 (LIST (CAR MPT) (+ (CADR MPT) 3.1)))
```

```
(COMMAND "TEXT" MPT3 0.2 90 (RTOS fl 2 2))
```

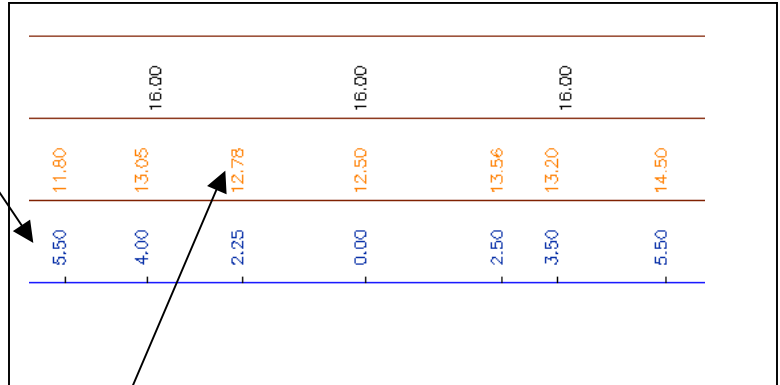
```
(setq flrpt1 (list (+ (car mpt3) fw2) (cadr mpt3)))
```

```
(command "text" flrpt1 0.2 90 (rtos fl 2 2))
```

```
(setq fllpt1 (list (- (car mpt3) fw2) (cadr mpt3)))
```

```
(command "text" fllpt1 0.2 90 (rtos fl 2 2))
```

```
(command "text" dp1 0.3 0 (itoa datum))
```



Reuse of any or all material contained within this document for commercial purposes, without the express written consent of AUGI, Inc. or its authorized agents is expressly prohibited.

© Copyright 2004 Autodesk User Group International, Inc. All rights reserved.

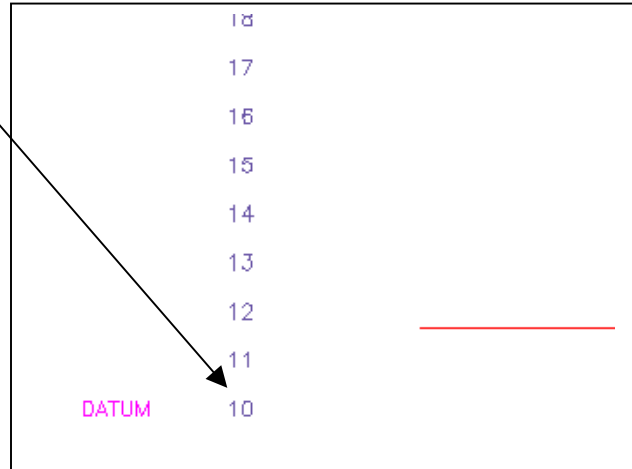
```
(setq cnt 1)
(while (<= cnt 10)
  (command "_color" cnt)
  (setq dp1 (list (car dp1) (+ (cadr dp1) 1)))
  (setq datum (+ datum 1))
  (command "text" dp1 0.3 0 (itoa datum))

  (setq cnt (+ cnt 1))
)
(setq glp0 (- gl0 ddummy))
(setq gll0 (list (car mpt) (+ (cadr dp) glp0)))

(setq glpr1 (- rgl1 ddummy))
(setq gll1 (list (car rpt1) (+ (cadr dp) glpr1)))

(setq glpr2 (- rgl2 ddummy))
(setq gll2 (list (car rpt2) (+ (cadr dp) glpr2)))

(setq glpr3 (- rgl3 ddummy))
(setq gll3 (list (car rpt3) (+ (cadr dp) glpr3)))
```

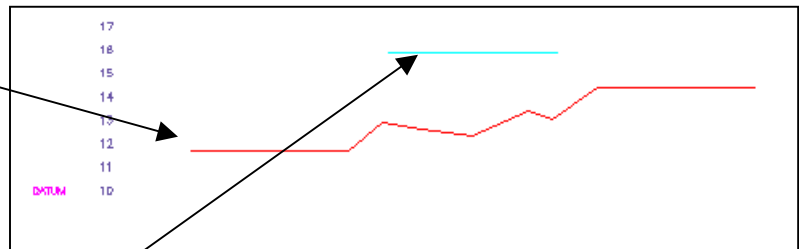


```
(setq glpl1 (- lgl1 ddummy))
(setq gll4 (list (car lpt1) (+ (cadr dp) glpl1)))

(setq glpl2 (- lgl2 ddummy))
(setq gll5 (list (car lpt2) (+ (cadr dp) glpl2)))
```

```
(setq glpl3 (- lgl3 ddummy))
(setq gll6 (list (car lpt3) (+ (cadr dp) glpl3)))
```

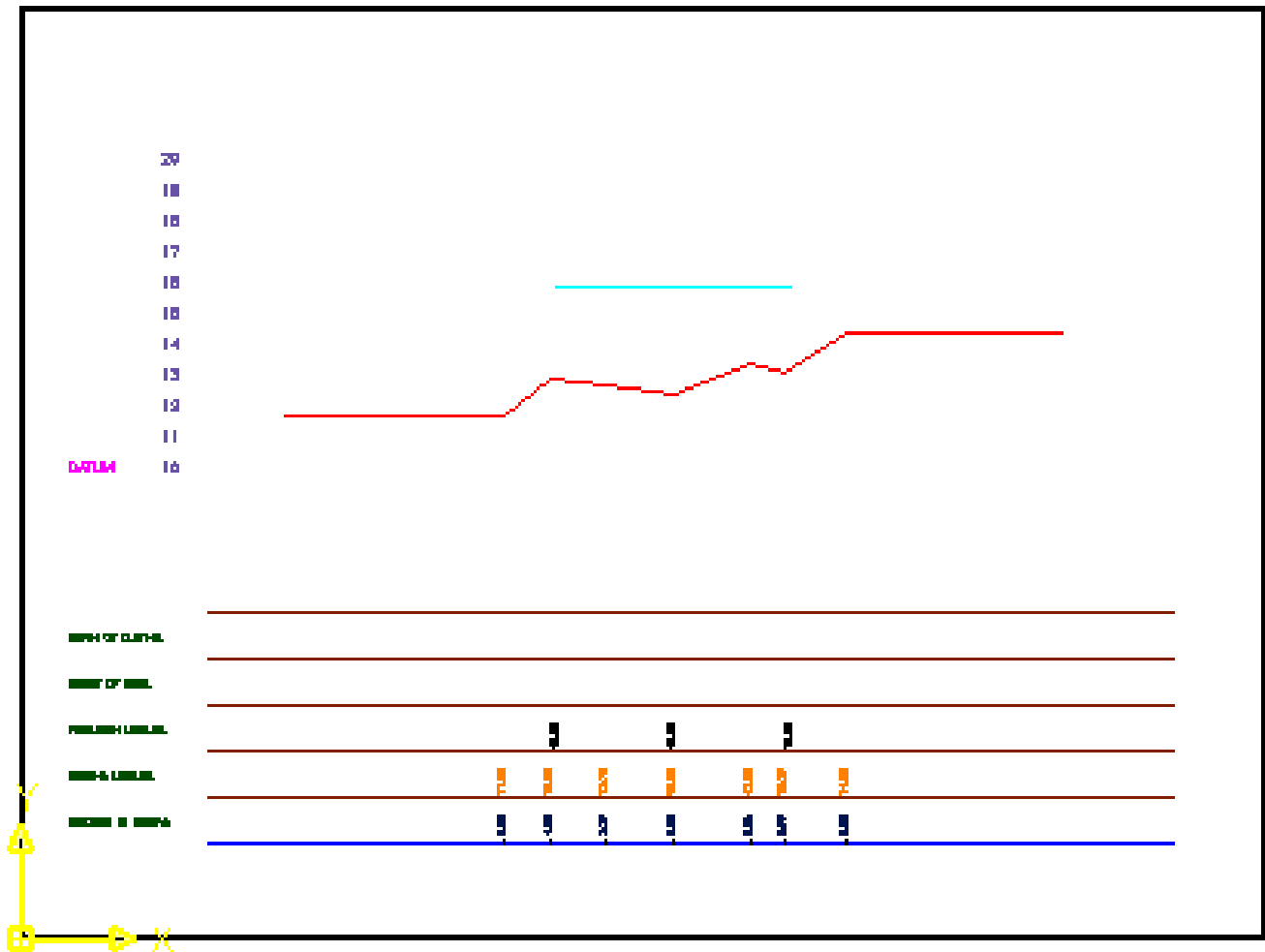
```
(command "line" gll0 gll1 gll2 gll3 "")
(command "line" gll3 "@7<0" "")
(command "line" gll0 gll4 gll5 gll6 "")
(command "line" gll6 "@7<180" "")
(setq fl1 (- fl ddummy))
(setq fldp1 (list (car mpt) (+ (cadr dp) fl)))
(setq fldp2 (list (car flrpt1) (cadr fldp1)))
(setq fldp3 (list (car fllpt1) (cadr fldp1)))
```



```
(command "_color" 1)
(command "line" fldp2 fldp3 "")
```

Reuse of any or all material contained within this document for commercial purposes, without the express written consent of AUGI, Inc. or its authorized agents is expressly prohibited.

© Copyright 2004 Autodesk User Group International, Inc. All rights reserved.



*My Dear participants, I hope this ATP has answered many questions cropping up in your mind and has generated some more questions for AutoLISP. Use the Course Forum for your remaining questions. Or as my friends in the ATP program like to say:*

**Remember that this material is only a portion of the class; support is always available online in the private course forum. I encourage you to visit the course forum and ask any questions that you may have**

**Reuse of any or all material contained within this document for commercial purposes, without the express written consent of AUGI, Inc. or its authorized agents is expressly prohibited.**

© Copyright 2004 Autodesk User Group International, Inc. All rights reserved.



about this segment or simply join in the discussion. The ATP Mantra is: the only stupid question is the one you don't ask. *Thanks again for attending this course!*

*I know – When I am teaching, I am learning more.*

**Reuse of any or all material contained within this document for commercial purposes, without the express written consent of AUGI, Inc. or its authorized agents is expressly prohibited.**

© Copyright 2004 Autodesk User Group International, Inc. All rights reserved.