

Chapter

4

The Arithmetic Functions

In this chapter, you will learn how to use the following AutoLISP functions to World Class standards:

- **Practicing AutoLISP Functions in AutoCAD**
- **The Twenty Arithmetic Functions**
 1. **Adding**
 2. **Subtracting**
 3. **Multiplying**
 4. **Dividing**
 5. **Exponents**
 6. **Square Roots**
 7. **Absolute Value**
 8. **Return the Maximum Number**
 9. **Return the Minimum Number**
 10. **1+**
 11. **1-**
 12. **Greatest Common Denominator**
 13. **Sine**
 14. **Cosine**
 15. **Arctangent**
 16. **Fix**
 17. **Float**
 18. **Remainder**
 19. **Zero Proof**
 20. **Minus Proof**

Practicing AutoLISP Functions in AutoCAD

Of the 26 arithmetic functions, twenty are plausible to end up in a program where one would need the math to calculate or extract a number to draw any architectural or engineering project. AutoLISP functions are the easiest to practice than any other programming tools in all the categories of coding languages. This is possible by typing the expression at the command line in AutoCAD, so you can practice each argument independently, without worrying whether some other portion of the instructions is causing an error. By practicing every function three, four or even ten times, the new programmer can learn very efficiently and actually enjoy the education process by receiving quick feedback that their ideas are correct.

An example of the ease to learning AutoLISP is doing this simple exercise. Launch AutoCAD and at the command line type an open parenthesis, then a plus sign +, a single space, a three, a single space, a two, and finally a closed parenthesis. (See figure 4.1) This is a simple method to practice typing the structure of the code and get a result. To check your work press the Enter key, and a five will appear on a new line below the AutoLISP expression. Error will occur in the line code if there is not a space between the adding function, the plus sign and the first argument, the three, like `(+3`. An error, bad function note will follow the pressing of Enter. No problem since there is a simple fix to this difficulty, just retype the code on the command with a space between the function and the first argument. The same goes for the space between the first and second argument `(+ 32)`. There needs to be a space between the three and the two, and the code will run flawlessly.

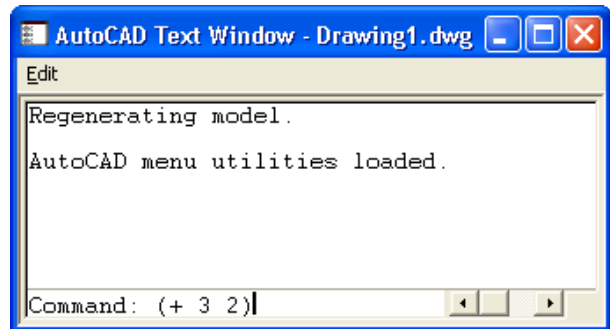


Figure 4.1 – `(+ 3 2)` on the Command line

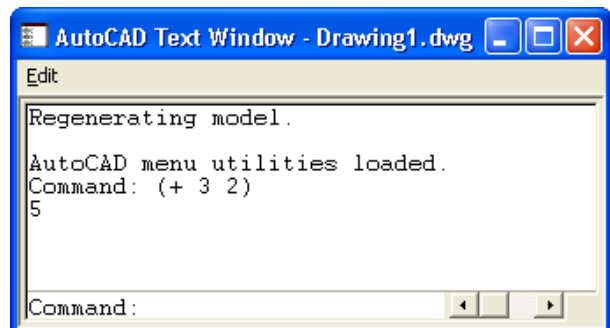


Figure 4.2 – The Answer to the Expression

Whenever a programmer desires, numbers can be substituted with variables that will represent an assigned value. This can be best demonstrated by another uncomplicated illustration using the AutoCAD Command line to demonstrate how to code an expression. In this case, we will use the `setq` function to assign a numeric value to the variable `a` using the following expression which we will type here in black and white and not in color as we see the AutoLISP editor.

`(setq a 3)`

The variable `a` is the integer 3 and unlike other programming languages there is not a dim statement declaring the type of variable. The `a` variable represents the integer 3 and needs to be treated as an integer throughout the program. You can check your variable setting by typing `!a` at the Command line and the return on the new line will be 3.

Next type:

(setq b 2)

The variable **b** is assigned the number **2** for this iteration as you can see in figure 4.3. We need to able practice these different techniques to build our confidence when entering any AutoLISP phrase.

Finally type:

(+ a b)

Again the answer **5** is returned. (See figure 4.4) We have the same answer as we saw before, but using the same addition function, we have an expanded capability in writing the AutoLISP programming code. Just because we use simple numbers in the arithmetic practice set in this chapter, we can substitute the variables **a** and **b** or make up our own to develop expertise.

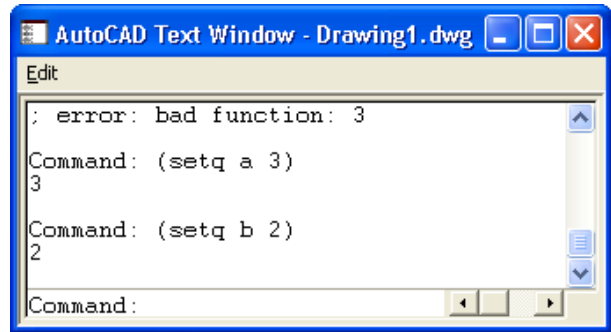


Figure 4.3 – Setting variables a and b

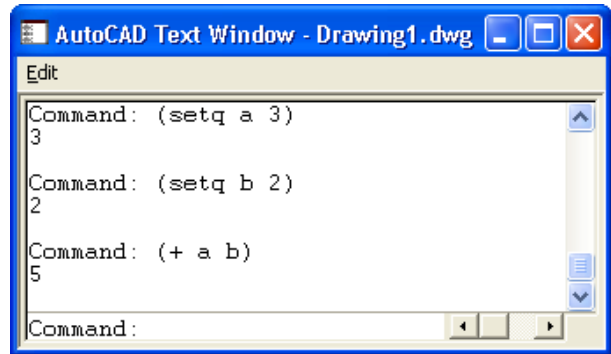


Figure 4.4 – Adding two variables

The Twenty Arithmetic Functions

The first arithmetic function we will address is one of the most common, the adding function which is displayed by the icon **+**. The addition function allows us to add two or more numbers. The values of the numbers can be whole like 1,2,3... or decimals, positive or negative. Remember we can have more than two numbers like (+ 2 3 7 4).

Function	Name	Description
+	Adding	The addition function will add two or more numbers
Examples		
Using integers	(+ 4 3)	Answers 7
Using decimals	(+ 5.5 3.2)	Answers 8.7
Using negatives	(+ 154 -123)	Answers 31

The subtraction function is similar to the adding function, but all of the numbers following the first argument are subtracted from the first number. Again there is not a limit to the number of arguments following the subtraction icon **-**. An easy error to make is mistaking the negative sign attached to the front of a number constructing a negative value with the minus sign in the subtraction function. Be aware of these two symbols are their meaning to prevent a troubleshooting dilemma.

Function	Name	Description
-	Subtracting	The subtraction function will subtract two or more numbers
Examples		
Using integers	(- 4 3)	Answers 1
Using decimals	(- 5.5 3.2)	Answers 2.3
Using negatives	(- 154 -123)	Answers 277

We can use the multiplying function to multiply two or more numbers together using the asterisk icon. After practicing the simple examples in the multiplying table, we can be experimenting with compound expressions on the AutoCAD Command line like:

$$(* (+ 4 3) (- 7 3))$$

After computing $4 + 3 = 7$ and $7 - 3 = 4$, the computer calculates $7 * 4 = 28$. Very neat.

Function	Name	Description
*	Multiplying	The multiplication function will multiply two or more numbers
Examples		
Using integers	(* 4 3)	Answers 12
Using decimals	(* 5.5 3.2)	Answers 17.6
Using negatives	(* 154 -123)	Answers -18942

As the first three arithmetic functions were very similar in their handling of integers, decimals and negatives, the division function will cause some problems when a whole number is divided by another whole number. The first argument is divided by the second and if there is a third, the process will continue until there are no more arguments, but the danger in this function exists when the whole numbers are used. On the AutoCAD Command line, type the expression (/ 4 2) and the return is 2, and there is no problem, but type in (/ 3 2) and the answer will be 1. When the only arguments in the expression are whole numbers, the answer will return as a whole number with the decimal remainder left off, so the (/ 3 2) is 1 instead of 1.5. Some programmers make the error thinking the AutoLISP code will round the number up if the remainder is 5 or greater, but not so, the decimal component is just removed. To solve this difficulty, always use at least one number with a decimal ending like (/ 3.0 2) and the answer will be 1.5.

Function	Name	Description
/	Dividing	The division function will divide the first number by the second number
Examples		
Using integers	(/ 4 3)	Answers 1
Using decimals	(/ 5.5 3.2)	Answers 1.71875
Using negatives	(/ 154 -123.0)	Answers -1.25203

The exponent function is used when applying formulas like the area of a circle, which is

$$A = \pi r^2$$

Setting the radius of the circle to 3 type:

(setq r 3.0)

The value of π in AutoLISP is already assigned with syntax, **pi**. Remember we can type **!pi** to check any variable, so try check the value of **pi**. Now to find the area of the radius 3 circle, enter the code shown.

(setq a (* pi (expt r 2)))

Many formulas representing architectural or engineering tools and find the area of a surface or volumes of a solid will have exponents, so this function comes in handy periodically.

Function	Name	Description
expt	Exponent	The exponent function will raise the first number to the power of the second number
Examples		
Using integers	(expt 4 3)	Answers 64
Using decimals	(expt 5.5 0.2)	Answers 1.40628
Using negatives	(expt -2.0 4)	Answers 16.0

The square root function will return a decimal number representing the side of a square area. This function only expects a single argument after the function **sqrt**. Remember no negative numbers are permissible with the square root function. A complex expression that would be common place would be finding the hypotenuse of a triangle using the variables **a** and **b** that are already defined:

(setq c (sqrt (+ (expt a 2)(expt b 2))))

Function	Name	Description
sqrt	Square Root	The square root function will find the root of the square represented a number
Examples		
Using integers	(sqrt 4)	Answers 2.0
Using decimals	(sqrt 5.5)	Answers 2.34521
Using negatives	(sqrt -154)	Negatives Not Allowed

The absolute value function, one more function that will only allow a single case to follow the syntax **abs** will return with a positive number. This function is useful in formula writing when deciphering positive distances in AutoCAD. Later in this book, we will discover that the AutoCAD program uses a starting point and ending point to explain the construction of a line. When subtracting the starting point x-value and ending point x-value depending on how the designer drew the line, our answer can be a negative number. Placing the absolute value function in front of any answer will result in a positive number.

We typed in **(setq sp (getpoint))** and picked any point on the right side of the graphical display, to

create a starting point. Then we typed **(setq ep (getpoint))** and picked any place on the left side of the graphical display. Our numbers can be different from yours as shown in figure 4.5. Next we enter the compound AutoLISP expression.

(setq x_distance (abs (- (car sp) (car ep))))

Remember from chapter 2 and 3 that the **car** function will return the x value of a x,y,z coordinate. If there was not an absolute value function in front of the subtraction expression, the answer would be negative. Try the expression with and without the **abs**, but remove the open parenthesis in front of the **abs** and one close parenthesis at the end of the expression.

(setq x_distance (- (car sp) (car ep)))

The absolute value function is a specialized tool to always return a positive value.

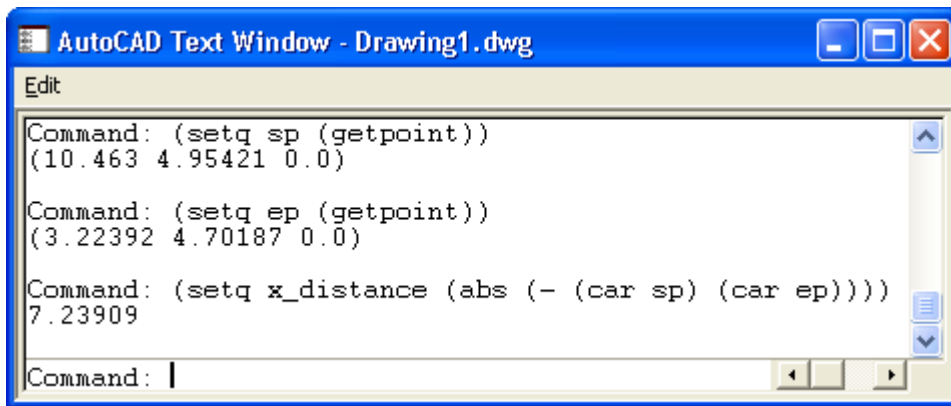


Figure 4.5 – Using the Absolute Value Function

Function	Name	Description
abs	Absolute Value	The absolute value function will return the positive value of a number
Examples		
Using integers	(abs 4)	Answers 4
Using decimals	(abs 5.5)	Answers 5.5
Using negatives	(abs -154)	Answers 154

The maximum and minimum functions will return the largest or the smallest value from a group of arguments following the function **max** or **min**. There are many times when a programmer has a set of values and needs to choose the largest or smallest of them. When that occurrence happens the function is placed in front of the group and the answer will be simple.

(setq bignumber (max a b)) or **(setq smallnumber (min a b))**

We already have the variables **a** and **b** defined, so test out the code.

Function	Name	Description
max	Maximum	The maximum function will return the largest value from a group of numbers
Examples		
Using integers	(max 4 3 2 1 0)	Answer 4
Using decimals	(max 5.5 3.7 2.2 1.0 0.75)	Answers 5.5
Using negatives	(max -154 -223 -687)	Answers -154

Function	Name	Description
min	Minimum	The minimum function will return the smallest value from a group of numbers
Examples		
Using integers	(min 4 3 2 1 0)	Answer 0
Using decimals	(min 5.5 3.7 2.2 1.0 0.75)	Answers 0.75
Using negatives	(min -154 -223 -687)	Answers -687

With the min and max functions being a matched set in programming, so are the increment and one minus functions. But first we need to discuss the properties of a programming loop. The easiest technique we have to demonstrate this concept is to take a small group of students and form a circle. We identify the beginning of the circle to be you and we give you the value of 0 and you say out loud “zero”. You hold a marker and pass that marker around clockwise until a student hands you the same maker back. Now in order to make a basic programming loop function properly, we will add one to your value, so you announce “one” and pass the marker around the group again. The next time you receive the marker you have figured out the game by now, and state “two”. To allow this to become common place in your programming skill set, typically while we are discussing the writing of the 1+ expression, we allow you to continue until the entire group is tiring of passing the marker in a circle and hearing you count by ones, but finally someone will ask how to stop the loop. That will be another discussion with another function, but remember where you heard about loops first, using the increment function. Okay, we can stop counting, now.

The increment function is commonly used to increase the counter in a programming loop by the value of one. We will use this function in the ninth program in this text, the Motion program.

Function	Name	Description
1+	Increment	The increment function will add one to number
Examples		
Using integers	(1+ 4)	Answer 5
Using decimals	(1+ 5.5)	Answers 6.5
Using negatives	(1+ -154)	Answers -153

Some programmers are so confident that they count down to zero instead of counting up to a set

value in the **while** loop. Go ahead and practice the one minus function.

Function	Name	Description
1-	One Minus	The one minus function will subtract one from number
Examples		
Using integers	(1- 4)	Answer 3
Using decimals	(1- 5.5)	Answers 4.5
Using negatives	(1- -154)	Answers -155

The function **gcd** defining the greatest common denominator which will return the largest value representing a common multiple of two numbers. Decimals and negative numbers are not allowed with this function.

Function	Name	Description
gcd	Greatest Common Denominator	The greatest common denominator function will return the largest number that is a common multiple of two numbers
Examples		
Using integers	(gcd 4 8)	Answer 4
Using decimals	(gcd 5.5)	Decimals Not allowed
Using negatives	(gcd -154 -2)	Negatives Not allowed

The sin function is a key tool in find the length of the side opposite the angle and in the case of the standard AutoCAD drawing system using the Cartesian coordinate system, the sine of the angle times the hypotenuse of the triangle or length of the line will represent the delta-y of the line being examined. In our example, we will use the distance function to get the hypotenuse of the line determined by **sp** and **ep**. The angle function will return the angle of the line in radians of the points **sp** and **ep**. Use this function at the AutoCAD command line to experiment with the **sin** function.

```
(setq delta_y (* (distance sp ep)(sin (angle sp ep))))
```

Function	Name	Description
sin	Sine	The sine function will return the length of the side opposite the angle of a right sided triangle when the hypotenuse is 1
Examples		
Using integers	(sin 2)	Answer 0.909297
Using decimals	(sin 0.5)	Answer 0.479426
Using negatives	(sin -0.2)	Answer -0.198669

The **cos** function is likewise a tool in find the length of the side adjacent to the angle and in the case of the standard AutoCAD drawing system using the Cartesian coordinate system, the cosine of the angle times the hypotenuse of the triangle or length of the line will represent the delta-x of the line being examined. In another example, we will again use the distance function to get the hypotenuse of the line determined by **sp** and **ep**. The angle function will return the angle of the line in radians of the points **sp** and **ep**. Use this function at the AutoCAD command line to experiment with the **cos** function.

(setq delta_x (* (distance sp ep)(cos (angle sp ep))))

Function	Name	Description
cos	Cosine	The cosine function will return the length of the side adjacent to the angle of a right sided triangle when the hypotenuse is 1
Examples		
Using integers	(cos 2)	Answer -0.416147
Using decimals	(cos 0.5)	Answer 0.877583
Using negatives	(cos -0.2)	Answer 0.980067

The arctangent function will return the angle in radians when the first number represents the opposite side of a right triangle and the second number symbolizes the adjacent side of the same right triangle. With the **sin**, **cos** and **atan** functions, they use radians as the system of angular measurement, which we usually never convert, since the other functions will use those radian angles to compute linear measurements of a triangle.

Function	Name	Description
atan	Arctangent	The arctangent function will return an angle in radians where the first number is the opposite side and the second number is the adjacent side of a right triangle
Examples		
Using integers	(atan 4 8)	Answer 0.463648
Using decimals	(atan 5.5 3.7)	Answer 0.97861
Using negatives	(atan -154 -2)	Answer -1.58378

The fix function is in our list of number modifiers for arithmetic functions, but we will visit the tool again in the conversion function list. This contrivance will take a real number and remove the decimal places leaving a whole number. There is not any rounding, but the numbers are just removed to the right of the decimal place. This coding method will be useful when computing an array command in AutoCAD, where we want a whole number response. Draw a small circle on your graphical display, and type the following multifaceted expression at the AutoCAD Command line.

(command "array" "1" "" "r" (fix (/ 20.3 2)) (fix (* 4.2 2.3)) 1.0 0.8)

The fix function returned **10** rows for **(fix (/ 20.3 2))** and **9** columns for **(fix (* 4.2 2.3))**. This tool is very helpful for any programmer.

Function	Name	Description
fix	Fix	The fix function will return a whole number of a value by removing the number's decimal places
Examples		
Using integers	(fix 4)	Answer 4
Using decimals	(fix 5.5)	Answer 5
Using negatives	(fix -154.75)	Answer -154

The float function will change a whole number into a real number by adding decimals to the number such as the integer 4 when written as **(float 4)** returns as **4.0000**. Remember that you cannot divide accurately using a whole number in AutoLISP, so sometimes we have to change the integer to a real number using the float function.

Function	Name	Description
float	Float	The float function will change a whole number into a real number adding decimals to the number
Examples		
Using integers	(float 4)	Answer 4.0000
Using decimals	(float 5.5)	Answer 5.5000
Using negatives	(float -154)	Answer -154.0000

The remainder function will return a number displaying the remainder after the second number is divided into the first number. This is a useful tool when determining whether a counter number even or odd. If you write the following expression to set the counter.

(setq counter 4)

And the even and odd checker to determine the state of the counter.

(setq even_or_odd (rem counter 2))

The variable **even_or_odd** will be a **0** for even or a **1** for odd.

Function	Name	Description
rem	Remainder	The remainder function will return a number displaying the remainder after the second number is divided into the first number
Examples		
Using integers	(rem 4 2)	Answer 0
Using decimals	(rem 5.5 2.0)	Answer 1.5
Using negatives	(rem -154 -2)	Answer 0

The zero proof function will return T for true when the number is zero which will help us not to divide by zero in a division expression.

Function	Name	Description
zerop	Zero Proof	The zero proof function will return T for true when the number is zero
Examples		
Using integers	(zerop 0)	Answer T
Using decimals	(zerop 5.5)	Answer nil
Using negatives	(zerop -154)	Answer nil

The minus proof function will return T for true when the number is a negative. This is not a common expression, but can work with a while loop.

Function	Name	Description
minusp	Minus Proof	The minus proof function will return T for true when the number is a negative
Examples		
Using integers	(minusp 4)	Answer nil
Using decimals	(minusp 5.5)	Answer nil
Using negatives	(minusp -154)	Answer T

Already we have used 32 functions in 4 chapters and 8 unique programs. We are learning how to experiment with single and compound expressions. At the end of the training, most of us will complete 50 running programs and have a great working knowledge of over one hundred programming tools.

Continue to Appendix B and solve some mathematical problems.

