

## Introduction to Visual AutoLISP

---

**In this chapter, you will learn the following to World Class standards:**

- 1. Engineering Programming, a Third and Fourth Step**
- 2. What is Construction Code**
- 3. The Boxcircle Program**
- 4. The Anglemaker Program**
- 5. Practicing the Construction Code Technique**
- 6. Other Chapters in this Textbook**
- 7. Exploring the 100 Programming Functions**
- 8. Object Selection and Rip Code**
- 9. Programs That Do It All**
- 10. Companies Using Custom Engineering Programs**
- 11. The Future of Engineering Programming**

## Engineering Programming, a Third and Fourth Steps

---

The third and fourth steps in the long process of understanding Computer Aided Design (CAD) begins with the understanding of automating the drawing process and therefore allowing the designer and their customer the luxury of viewing and examining more than a single design. When technicians in a department are manually drawing on their computers, we often see that they only have time for one look at an engineering or architectural problem. Companies spend great amounts of time physically making simple changes to their base design, where the previous set of drawings are modified for today's project. Adapting a drawing template is fine for current jobs, but in the last thirty years, the pace in industry has quickened, adding more custom processes and requiring a greater level of detail in manufacturing and construction documentation. We notice supervisors that do not have the confidence to expand into new business arenas since their drawing models do not work for the next application. At other organizations, we can view areas in the drafting standards or the design procedure where basic principles are not followed, and therefore the worker in the field is confused by the print release. All of these tribulations evolve from corporate systems that rely on multiple layers of capability in the trained Computer Aided Drafter with a single level of checking and approving, and not placing intelligence from previous designs into the department's software to assist in the drawing and checking process.

In a way, CAD software has aided every company in previous years to provide every aspect in the release package to the contractor or shop manager, but an area that is still largely under utilized by almost every major engineering firm is automated software packages. These routines are designed to create the entire drawing set or even a single sheet from a list of data or a series of answered questions. The reason that we do not see companies providing automated drafting and design services is largely due to the lack of knowledge and experience in this discipline coming out of CAD training institutes, junior colleges and universities. Also, if we select a textbook from the few available on the subject, we are confronted with the same problem in training other programming languages, which is a lack of any uniform standard, and where every program is a re-invention process. Just this single concept is foreign to the architectural and engineering branch of learning, where previous design principles and procedures are applied in nearly every situation. So therefore, the modern A & E firm will confidently use accelerated tools provided by third party software suppliers to assist in the drawing process, such as adding a window or door, but not likely to develop their own drawing and design software to design a building or a house. Such is the dilemma for modern business, having managers fear the next two steps resulting in drawing automation and their desire to have a capability to see more done in less time without errors.

**What has been the success of the World Class CAD training method?** One, we treat engineering programming just like any other science in the architectural and engineering family of disciplines, by developing types of programming templates and strategies that are 100% infallible in their ability to accomplish a task or solve or problem. Two, we provide training that allows the engineering programmer to quickly achieve a solution to any 2D orthographic presentation by drawing, labeling, describing variables and writing their code in a standard form in an hour from start to finish. Three, the training evolves to permit the

advanced CAD programmer to rip the information from an existing set of drawings, and record the data for other uses such as a bill of material, creating an estimate or changing parameters in the drawing.

**What is the goal of this textbook?** Without a doubt, our routines will be drawing entire details, orthographic views, borders, title blocks, bill of materials and three dimensional models, so by the end of the text, we will be able to make entire drawings from scratch. We have the experience, the confidence and the training techniques to change the section, department or company where our professionals apply the knowledge in these chapters. What we do with the time saved from automating the Computer Aided Design process, we have future textbooks to guide our readers to better product or project design.



**Figure 8.1 – A World Class Programmer**

## **Construction Code**

---

The first type of code formation is called “Construction Code”, since every 2D component and later with more training; any 3D entity can be drawn using this simple and repeatable process. The process is broken down to starting the program, inputting data or simply asking the program user a series of questions, followed by a math computation section, then point assignments, drawing the entire drawing or detail and finally ending the program. Let’s see those steps again:

<b>Step 1</b>	<b>Start the program</b>
<b>Step 2</b>	<b>Drawing setup</b>
<b>Step 3</b>	<b>User input</b>
<b>Step 4</b>	<b>Do the math</b>
<b>Step 5</b>	<b>Point assignments</b>
<b>Step 6</b>	<b>Lets draw</b>
<b>Step 7</b>	<b>End the program</b>

Yes that is the process, and for an engineering programmer, the first handful to the fiftieth program, the only variance in the procedure is the length of the code, mostly based upon the number of vertexes in the programming model.

Other programs in the chapters may be modified from the basic Construction code pattern since we may not need to do any math. A number of routines will have drawing sections that will search for information similar to the user input step, so the program can continue drawing automatically without prompting a user to select an object. We will do plenty of useful code that will be benefit our future programming endeavors.

## The Boxcircle Program

The first problem that the World Class CAD Visual AutoLISP Training Method utilizes is the “Boxcircle” program. This exercise, like the 2D Rectangular Problem in World Class CAD, Fundamentals of 2D Drawing was created in 1996 and has been trained to hundreds of college students, who today are working as architectural and mechanical designers, engineers, and high school and college instructors throughout the American Midwest. The Boxcircle problem will automatically draw four lines and a circle in seconds after answering two questions. This is a very simple arrangement of syntax following the seven Construction Code steps, start the program, drawing setup, ask questions, do the math, point assignments, draw and end the program.

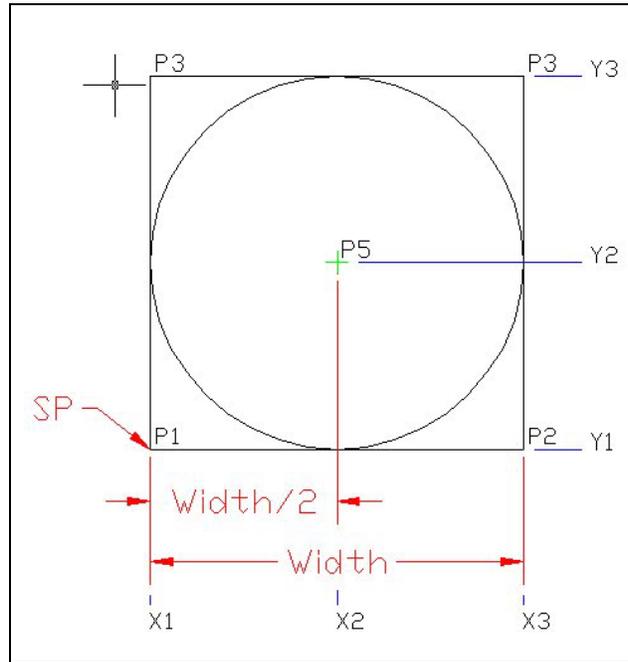


Figure 8.2 – The Boxcircle Problem

## The Anglemaker Program

The second problem will be the Anglemaker (1996). We increase the number of points to address in the second routine to seven from the five intersections in the Boxcircle routine. We will also add the Arc tool to the list of AutoCAD commands that we are able to utilize. The Anglemaker routine will have six questions. We will prompt the user for the starting point **sp**, the length of **leg1** and **leg2**, the thickness **thk1** and **thk2**, and the radius **rad**.

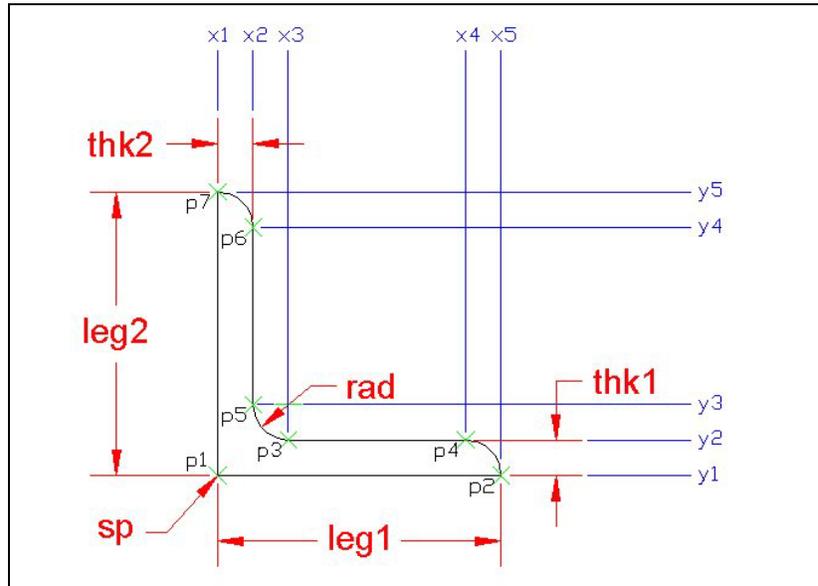


Figure 8.3 – The Anglemaker Problem

The third through eight problems are a series of programs that will increase your level of

expertise in developing Construction Code. In this process, through repetition we will learn the basic math functions, how to manipulate a three number list (x, y and z), how to draw lines, circles and arcs, and few basics in code syntax and troubleshooting.

## Practicing the Construction Code Technique

---

After the first two routines, we will present ourselves with six practice problems as shown in the table below. These six simple and common orthographic views will help us practice the skills we learn in the first two sets of coded instructions. The only difference between the first program and the eighth is the number of junctions.

Problem	Name
1	Boxcircle
2	Anglemaker
3	Channel
4	Bathtub
5	Window
6	Door
7	I-beam
8	Bearing Plate

**Figure 8.4 – The First Eight Problems**

Another factor will be the World Class CAD Challenge. Every exercise will dare us to increase our efficiency when sketching, coding and running the new AutoLISP programs. As in all of the other World Class textbooks, the Visual AutoLISP Training Method will inspire us to achieve finished code in record times and therefore give us new skills to complete orthographic views quicker. Worded below is a typical World Class CAD Challenge that we find with each drill.

**\* World Class CAD Challenge \* - Create a New AutoCAD file and sketch the entire problem on the proper layers, using proper dimensions and finally placing the points and x and y grid on the drawing. Save your AutoCAD sketch. Open the Visual AutoLISP editor and code the problem using the Construction coding method. Save the code as named on the problem sheet. Send your copy of your drawing and code for verification to the authors of these problems to have your name and location posted on the website.**

**Figure 8.5 – Sample World Class CAD Challenge**

## Other Chapters in this Textbook

---

Successes of the World Class Visual Auto LISP Training Method are measured by completing one chapter at a time. As in the World Class CAD textbooks, 2D Fundamentals of Drawing and Fundamentals of 3 Drawing, each unit builds on another and the World Class CAD Challenge will evaluate our ability to do the subject efficiently. We want to get into a regular

training cycle when learning to code. Do the orientation eight problems and make up a few of your own to check your aptitude. Jump into each chapter with the motivation to learn each function. Do the examples on the Command Line in AutoCAD to practice the programming syntax. Build a programming notebook that contains the sketch for each problem, the printed routine and a printout of the drawing made. Place the finished program on a floppy disk or Flash drive. Believe that these skills are important in your upcoming career and to the success of your future employer. The portfolio that you build today will be the beginning of the standards manual at your potential organization.

Week	Subject
3	More Math
4	Input and String Handling
5	While Loops
6	Geometric Functions
7	Conversions
8	List Manipulation
9	Selection Sets
10	Object Handling

Figure 8.6 – The Chapters by Week of Training

## Exploring the 100 Programming Functions

Function	Name	Description
<b>setq</b>	<b>Set Quotient</b>	Allows the user to assign a real number, integer, string or list to a variable
<b>Examples</b>		
Set the variable <b>a</b> the text string <b>World Class CAD</b>	<b>(setq a "World Class CAD")</b>	Answer: <b>"World Class CAD"</b>
Set the variable <b>counter</b> the integer <b>0</b>	<b>(setq counter 0)</b>	Answer: <b>0</b>
Set the text height variable <b>txht</b> the real number <b>0.125</b>	<b>(setq txht 0.125)</b>	Answer: <b>0.1250</b>
Set the point variable <b>sp</b> the list of 0,0,0	<b>(setq sp (list 0.0 0.0 0.0))</b>	Answer: <b>(0,0,0)</b>

Figure 8.5 – The Quick Reference Guide

Most programming languages have about a hundred functions in which we will need to be familiar. These functions need to be typed correctly in the Visual LISP Editor, but we can make mistakes by leaving out parenthesis or quotes. So throughout the Visual LISP Training Method chapters, we will see gray blocks that review a LISP function. The review box is a

really helpful tool in learning how to type the programming syntax. After the function name, there is a short description, and then a few examples concerning the coding. Each example expression can be typed on the Command Line right in the AutoCAD program as shown below. We should attempt to type each example without error.

```
Command: (setq a "World Class CAD")  
Command: "World Class CAD"
```

After typing the expression where we assign a value to a variable like we did with (setq a "World Class CAD"), then we can type !a at the command line to check the assignment. After the !a, the value will appear on the next line.

```
Command: !a  
Command: "World Class CAD"
```

After developing the basic understanding of writing a successful program, we will study how to change many drawings in a short time and how to link the data to another program.

## Object Selection and Rip Code

---

If the first two sections of the textbook give us the knowledge of creating an entire drawing from a list of user input data, then the last section in the Visual LISP Training Method will help us to check and modify hundreds or even thousands of drawings in a folder using automated tools.

By the time we are ending the textbook, we will learn to write code that will Rip the information from existing drawing that we receive or have drawn ourselves. In the mid 1990's, as we called the first template the Construction Code, the second easily was termed the Rip Code. That is precisely what we are doing when we use the selection set function **ssget** and by learning to apply filters, we grab data, analyze the information, and change drawing entities based upon the criteria in the routine.

We start by programming custom functions to change or match text, place dimensions automatically, count items or change layout positions of hundreds of parts based on design criteria. Although text manipulation is considered very basic, many companies spend hundred of drafting hours doing just that. By learning how to control hundreds of lines of text, our graduates have successfully altered hundreds of drawings without human intervention in minutes rather than manually opening each drawing, searching for certain text and making the adjustment. Other routines we will put together will give us the quantities in the bill of materials without physically counting items such as light fixtures or fire extinguishers. We expand our efforts to changing the position or direction of single or complex entities. Eventually we reach the moment of writing bill of materials internally or externally to another file using the facts gained in this unit. With these abilities, both our routines and our programmers become even better since handling such volumes of material is unique to the industry and seeing drawing modified in seconds is a great motivator.

## Programs That Do It All

We are not mistaken that by the time we finish this one textbook, we will have every skill to draw an entire drawing after answering a series of questions. Our instructors have the experience from working on previous professional projects to create entire drawings and then after more work, entire set of prints to finish a complete release of architectural or engineering documents. We have been in offices where our code made 400 drawings completely after answering 15 to 20 minutes of questions. The entire morning the computers are running and sending files to the printers, so the design team and customer can review the work. When changes are made, either the input data can be reinserted and recompiled or simple changes can be made in the drawing set physically or by a smaller routine. Companies that utilize these capabilities can increase their percent of custom work since the design team has the luxury to improve on their products.

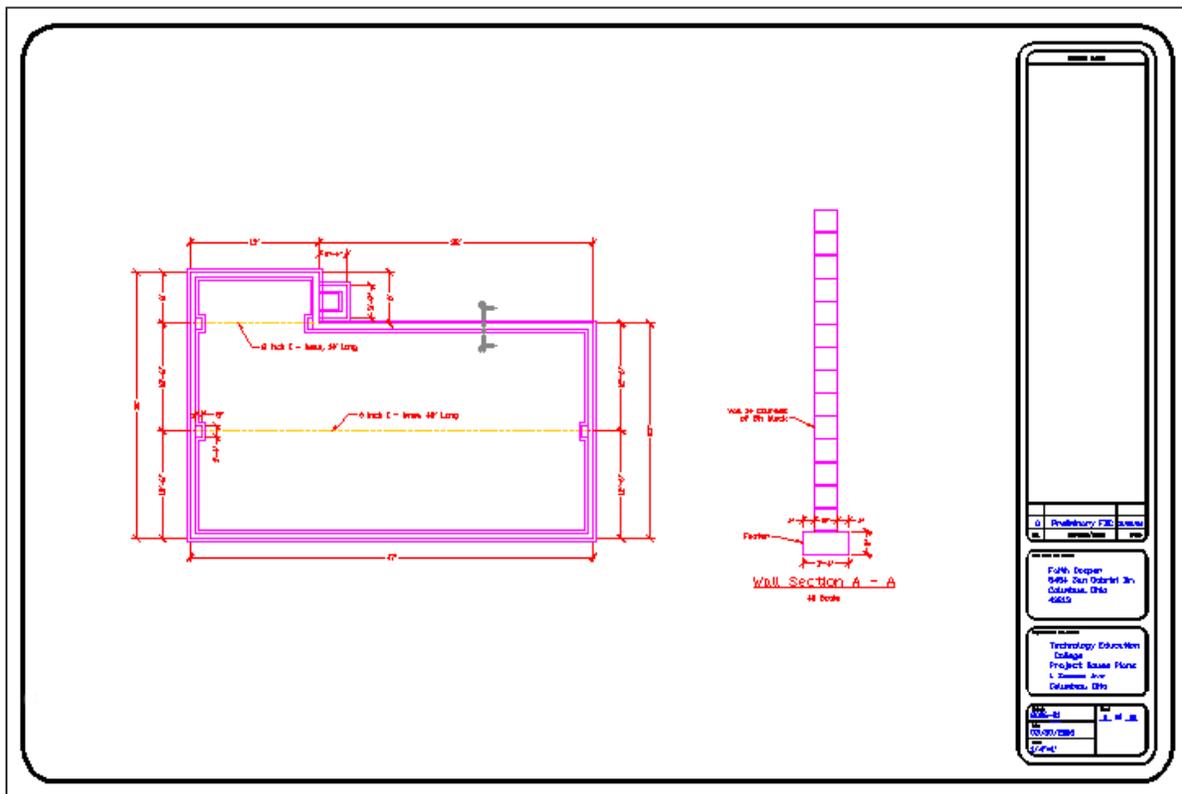


Figure 8.7 – Drawing Create Entirely from a Visual AutoLISP Program

## Companies using Custom Engineering Programs

Are companies using Visual AutoLISP programs to create new designs? Yes, we have graduates that have automated the creation of drawings devising routines for title blocks, bill of materials, designing manufacturing robots, making production drawings for custom windows, detailing NEMA enclosures for electrical control panel and making electrical wiring

diagrams for Programmable Logic Controllers. All of these organizations are using basic Construction Code as the root of their company's success.

When a person introduces automation in a department by writing an LISP routine to create an entire drawing, the other individuals in the department will probably be somewhat skeptical thinking that only someone with special insight can accomplish such a task. The fact is that anyone who can organize their thoughts and use the Construction code process is capable of accomplishing the entire project unaccompanied. We have seen third party organizations attempt to write code for a business where they will slowly learn the procedures that the company goes through to make a drawing. Honestly, a properly trained programmer, who finishes the material in this textbook, will do a better and faster job than an outside organization. Remember, just like any other process that an architectural or engineering group attempts, the technology must be understood by the members of our department and revised periodically as the organization changes. Purchasing third party or in the box applications from outside suppliers usually does not meet most company's needs.

Company managers typically want a group of Computer Aided Design (CAD) programmers creating custom routines versus having one individual with the knowledge and capability to accelerate their drawing processes and give their customers choices. So when we learn to program engineering code, we want to develop a programming partner simultaneously in our organization. That individual will grow in the discipline with us, and the new skill set is now more likely to spread among others in the department. Next, document the code and place plenty of comments in the routines that are written in simple English so that anyone one in the work area can understand the phrase. Check the finished drawing as per our normal company standard operating procedure, and when the print returns from the plant floor or work site with any corrections, determine whether these were the result of a programming error. If the mistake was in the code, correct the infraction and retest the program. As in any successful area of learning, the more hands and eyes on the finished product, the better the machine or code will become. For years to come, training Engineering programmers will be imperative to the success of our organizations.

## **The Future of Engineering Programming**

---

When we look back, our first large undertaking was taking a engineering group through the process of writing routines that would make hundreds of outline drawings that were complete with a front and side orthographic view of an complex assembly, showing dimensions, notes, border and title block. On each assembly drawing, the salesperson could input custom text and the routine would refer to data from a federal standard to space the text across the face of the assembly. There were less training material for engineering programmers in those years, and we would telephone the experts in the field to get their opinions on different approaches. With a lot of hard work, we released the first code and the sales department made thousands of drawings in the first year of the program release, which was a higher production number of prints than coming out of our main engineering departments.

For five years after that initial set of routines, our group of coders released hundreds of programs that would make many of the drawings in the department. There are a few reasons

which might push an architectural or engineering group into programming. In our case, productivity in the business was increasing faster than we could hire and train personnel, so we bet on this newer technology to counter for office space and availability of trained designers. Another rationale was the routines would do repetitive types of drawings to support custom product lines that engineers and designers felt were unexciting to do. Also, the sales and marketing team in our region wanted quotes and drawings in 24 hours.

Our branch of a major corporation was growing exponentially and we were always just in time or slightly behind in our print release schedule. As the company was growing, especially in the area of custom work, the production of print sets was paramount to keep the business running. We could spend a few months to get new personnel acquainted with a certain product line and multiple years training them in all the aspects of computer aided design. We need to remember that most universities spend the smallest amount of time training students in computer aided design, so this is a very common problem in the workplace. One could have to say that writing custom code to create our unique drawings was forced upon us. So a group of three designers and one electrical engineer became the core group in writing the routines.

One of the smartest steps taken early in the process was to separate ourselves from software development in the other departments. After many years of study in this technology, our group is convinced more than ever that engineering programmers need to be right in the mix of the other design disciplines in order to create high-quality code. We have seen other programming groups struggle when they are separated from the main engineering departments they support. Also, writing the routines today and having the code working in the department tomorrow is a real world solution to the demand of print output. When a professional receives the finished code that has nearly unlimited potential to make thousands of drawings, they can be successfully integrating the new routine in minutes. The difference is astounding and shows in the weekly engineering production report.

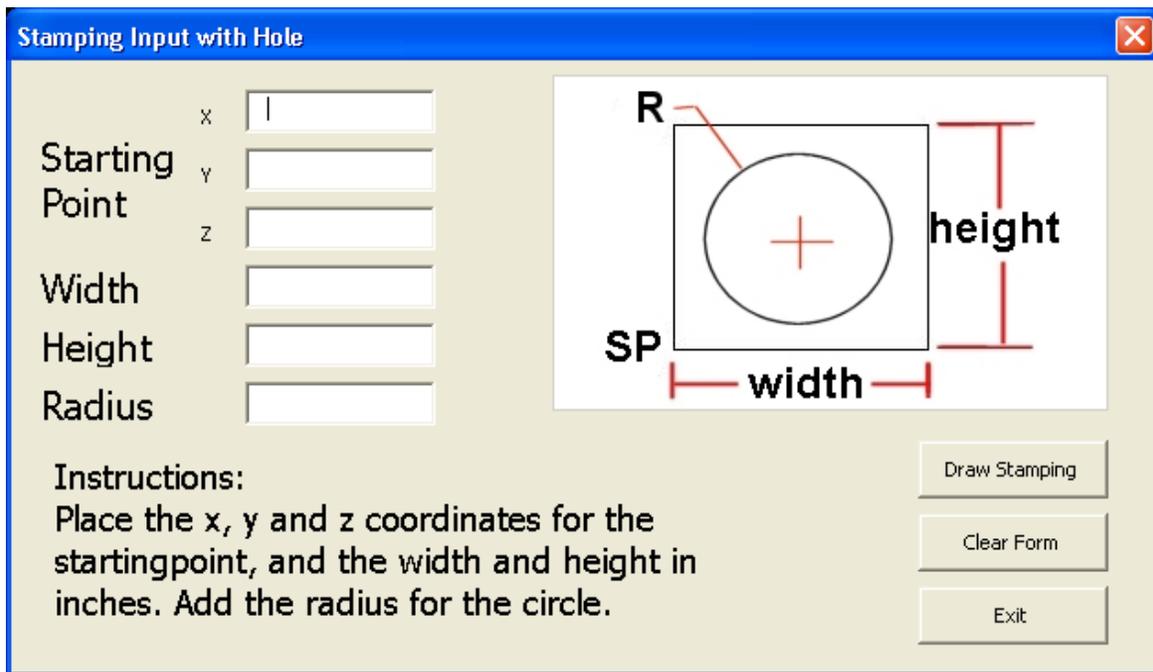


Figure 8.8 –Visual Basic Application Program Form for Drawing a Stamping with a Hole

Sales departments need color prints with actual data for their customers and not just an advertising sheet with a table displaying possible variations. Many of the clients have a difficult time determining what they will be receiving and expect more service for their money. Our sales force wanted quotes and drawings in 24 hours. The Visual AutoLISP routines written years ago had questions appearing on the Command Line in AutoCAD and the salesperson would need a little Computer Aided Design training to make their drawings. Some programmers would place C language forms in the LISP routine to assist the user. Today, in Figure 8.8, we see a Visual Basic form that allows the computer users to place data into AutoCAD without having any specific CAD skills. These days, programmers are using the Visual Basic Application routines to make that step simpler. Creating forms as we see above is a simple task using the Control Toolbox, resulting in an input method that would enable any salesperson to create a detail drawing for the customer that meets all of the organization's and industry's standards.

Finally, we would be remiss in our CAD training method not to train every student in a feature that is in every version of AutoCAD, especially since the year 2000, where Autodesk has made very capable software, but we also recognize that many companies want to expand into Visual Basic Applications (VBA). Earl Williams was our first AutoLISP programmer trained in 1990 and Gordon Dumas was our first graduate to complete a major VBA routine for a local robotics company in 1998. Since then, World Class CAD has continued to explore the wide regions of using simple techniques to program in multiple languages. After completing this text, the next, World Class CAD – Applied VBA, one will be able to build on their abilities that they obtained in this instructional material.

Welcome to the world of programming.

Charles Robbins

**\* World Class CAD Challenge 04-00 \* - Complete this textbook in 40 hours of classroom training. Pass your Programming Levels 1, 2 and 3 certifications to be ranked among the best in the world.**