12 Keys to Success as a Programmer

By Patty Cross

I have programmed as part of my IT Management job for over 20 years. My training in programming has been self-taught and need based. Therefore I tend to use the languages I learned years ago even though there are better ones available today.

I decided to attend Rutgers and get a certification in Full Stack Development because much of what I do falls into this rough category, but self-instruction is no longer easy successful venture. I looked at Microsoft’s ASP.net in the Visual Studio product and found myself stumped. The vast amounts of languages and tools make my old methods look impossible.

I am used to stealing code snippets from the internet from JavaScript’s and without truly understanding them, I can take the code and replace some variables to make it work for me. I excelled in ASP (the old one) and VBA so with these three languages I could do nearly everything.

I also was trained to be a Unix administrator which today you might compare to Linux. Despite the variety (RED HAT, Debian, etc…) the basic operating systems are quite similar so I have tools I could use to code some Linux programming. I am old school, vi editor, where newer users would use nano or other editors. I can grep, alias, find, and a billion other useful Linux tools. I had some Pearl and Ruby scripts customized by consultants and I would be able to modify these by the same self-taught method.

No matter what language, I need to begin with a problem or issue to solve or do. Much of what I learned was based on transferring the necessary data to a program I was more comfortable with programing in.

So the 12 Keys to success as you define them are:

1. [Prepare for Pre-Work!](https://the-coding-bootcamp.gitbooks.io/pre-work-book/content/chapter1.html)

I understand this to mean be prepared for the first class by getting the bugs worked out, such as I have run into, and walking into class with every other class member at the same basic level. We need to be sure our hardware and software is ready for the class also.

When we have real code projects to write, this will take on a whole new meaning. Once assigned a task to code in real life, preparation will be using a story board or flow chart to layout the project preparation. This will be a time when we can think out of the box, if allowed and add potential features that may or may not be add-ons to our program.

1. [So You Want to Be a Web Developer](https://the-coding-bootcamp.gitbooks.io/pre-work-book/content/chapter2.html)?

Web Development is very marketable. Most companies no longer want standalone programs customized for their use.

I have found that most off the shelf programs force companies to do work the way the software package allows and most companies need customization programs from people like us to utilize these canned program data to better function for the company. For instance, my boss used to pay someone to hand draw a full year at a glance sheet on 180 plus employees. He wanted color coding to show each year basically how well the employee was doing.

I believe he would take these on vacation over Thanksgiving and lay them out on his table to determine what percentage of bonus the employee would receive. I know he would use this form in court to show a person’s work record.

Anyway, I programmed this manually function. We used ADP payroll system. I extracted a copy of each employee, the hours they worked daily, and comments like late lunch, left early, vacation, sick day etc. Then I had a web program structured that showed the work year in 4 quarters and populated each employee’s hours, vacation, and any comments for each day of the year.

I used to think I was driven to be a developer because of laziness. I might have been given a task to do daily, such as the coloring of everyone’s daily work hours and adding notes. I would think I could do it easier and with better consistent results in a program, so I would spend 1000 times more hours writing the program than it would take to do the task each given day. However, I understand that the program is a huge amount of work, which reduces a smaller amount of work.

The work of programming is often suggested by the programmer with good intensions. It usually makes the company somewhat happier because it can reduce the entry person’s time and make changes immediate and consistent. It makes data available at manager’s fingertips. But many people in the project stream WILL be less than happy.

The employee who spent hours per week would color coding and noting the employee hour’s sheets, will feel threatened. The top management will knit pick changes. Several managers wanted notes dropped or other things that finally my boss, who owned the company had to state that no one could make changes unless they came thru him. The original program (ADP) from which data was extracted changed and I had to revise the program to work with the old version and new. Then end of every year meant I needed to copy the program to a new year to adjust the start and stop dates because I had not worked out the day of the week math for global years. Employees had numbers codes which suddenly turned to alpha numeric and had to be tweaked to work correctly.

The biggest consistent issue I would find is “Programming is thankless” No one wants the change. As a programmer, I felt like a parent to this program and could not speak more highly of it, but users would complain for years about having to work with it. Suddenly, one day, and issue will take the program down, and everyone will jump on the other side complaining that they cannot do their work without the program.

I felt proud that every user who came with “bet you did not think of this issue…” found I had a straight forward answer. I felt proud that I took 5 divisions of our company who swore they could not work with one common package and I made it work with each division thinking it was customized for them. I felt proud that CAM programmers who refused to file their data as the software (document control package called ‘Living History’) demanded, would inevitably be coming to me periodically to ask if I could find a file from 3 years ago. After several times of hearing me say “If you filed it, it would be right here” or “Sorry, you should have put it away”, they realized the value of document control and started really using it.

I would always offer to make improvements if anyone could think of them. But all in all, no one wants change of any kind, so the program developer is not a popular person and it takes a great deal of programming and beta testing to replace a manual report. It is not a job that will earn you popularity.

1. [Get Your Mind Right](https://the-coding-bootcamp.gitbooks.io/pre-work-book/content/chapter3.html)

Programming is not easy. It is not popular, and it is not quick. You need to be driven to solve a problem. I like programming because I have done it for years.

I am not a social person. I like to work independently, and often I would go to work or remote onto my job in the wee hours of the morning. Whenever inspiration struck.

I get excited over the flow chart of the tasks, as well as the sales pitch on why this program will be valuable. For instance, Living History (document control system) I wrote to do many things. Basically, because we were a Job shop, we did not make “revision A” then “revision B” of our own projects. Our customers would send us 3D models, Drawings, sketches on napkins. We needed to keep track of what minute something came in because we worked on customer prototypes. They may add arms, then remove them within the same day.

As I said, a large part of Living History was CAM programs and being able to tell from the program name what revision of the customer’s part it would create. We had 5 divisions, several with CAM programmers, and programs post processed to different machines. My CAM part of Living History would allow any CAM programmer to move from one division to another and know how to find the correct work. I color coded the revisions so if Joe logged on to start work he was doing yesterday, it would show RED if the model had changes he needed to look at.

One of the biggest sales pitches on my program is it lined us up to pass ISO certification. Any Living History user could identify t5he current active item (models and drawings) instantly. There was no excuse to pick up a drawing without checking Living History to be sure it was current active.

The CAM portion of Living History took me months to program, and 7 years to get the president and vice president on board. We would meet every weekend, and they would try to see the task from the point of view of a CAM programmer, CMM Quality Controller and Project Manager.

I had programmed bells and whistles I knew would be desired. During the 7 years of showing President and VP, I stripped many of these bells and whistles out, only to reenter them as we implemented the program because each user would notice something they needed that management could not understand.

1. [Type Like a Tiger](https://the-coding-bootcamp.gitbooks.io/pre-work-book/content/chapter4.html)

I come from CAD/CAM environment where mouse work is used more than typing. I felt pretty good with the keyboard on my full sized desktop at work, but I am a bit slow on this laptop. The laptop I have was ghosting the mouse tablet so if my thumb passed over the mouse tablet screen it could move the cursor.

Since I left my job to attend this coding camp, the full keyboard is gone. I had to disable this mouse tablet and get a real mouse. I will do the best I can, but I do not feel this test will reflect my best.

1. [My Machine is ready!](https://the-coding-bootcamp.gitbooks.io/pre-work-book/content/chapter5.html)

I ordered a faster laptop and during the course I will switch over. However, I am seeing that there is a lot of different things we need installed and some web programs work better with different web browsers. So far, I am finding Edge to work best which means my “new” computer will need windows 10. Maybe I will make the new system Linux.

For now, this system is fine, and does all you require.

I am a cowboy writer though. I cannot help but to try anything and everything off the beaten path. I will keep a rebuild ready to restore in case I really mess up my system. I have to test themes, and what if I try this, kind of stuff because I often discover wondrous things. I do get in trouble this way too. Bear with me. I can live without sleep to rebuild a pc and get the homework done.

1. [Sort of Code with Pseudocode](https://the-coding-bootcamp.gitbooks.io/pre-work-book/content/chapter6.html)

I thought much of our coding would be done with these pseudocode programs. For instance, Visual Studio. I have now discovered VS is a massive processor eater, but I thought there would be many more GIT code pieces that would generate chunks of code to save time and make common errors less of a problem.

I love this scratch program. I already spent too much time with it and my grandson (9) showing him how code can be generated.

1. [Where's the $@#\*'ing Error?](https://the-coding-bootcamp.gitbooks.io/pre-work-book/content/chapter7.html)

Writing in ASP decades ago, I was able to view the HTML visually in the FrontPage editor but the SQL Queries would not show. I learned quickly how to print the SQL Query as text line in HTML, then open Access (database available) and paste the Query line in to find the error.

Since moving to Visual Studio this issue was harder. It was not uncommon to comment out lots of lines of code remembering to leave “end if, next, rm.movenext” etc. in uncommented script. I would fight to get HTML to display all the code it could, and narrow in on the trouble area. I could also use ‘document.write’ to check variables or pause the code and cursor over the variables to see if they are correct.

Being the IT Administrator, I was able to use group policy and IIS controls to assure the HTML errors would show up on my desktop. All other users would get the 404 or 505 error page or just a friendly “oops there is a problem”.

Persistence and Perseverance are the clues to errors.

1. [HTML Hotness!](https://the-coding-bootcamp.gitbooks.io/pre-work-book/content/chapter8.html)

I am quite familiar with HTML, but the new HTML5 is more demanding. Code I could ignore, like closing statements can no longer be ignored in HTML5. I guess it is good practice to write correctly so our code will work in all web browsers on all operating systems.

1. [See Me CSS](https://the-coding-bootcamp.gitbooks.io/pre-work-book/content/chapter9.html) !

When I began coding I wish I knew about CSS, or had dreamed up something similar. For me every page had to be redefined with the colors and fonts and buttons.

Before I decided to part ways with my company, using style sheets was one big reason I wanted to get training to rework the old programs. I get style sheets now, but converting ASP code to remove ‘font face=arial size=2’ does not seem to have a quick consistent program.

Like any one, I wrote a ton of back code in the beginning. I got better, but today’s codes are far better than what I have written. I need to rewrite most of my stuff, for my own sake. I know my Living History program is valuable to many companies. CSS will allow me to input even the company information.

1. [Jiggle into JavaScript](https://the-coding-bootcamp.gitbooks.io/pre-work-book/content/chapter10.html)

JavaScript writers are in high demand. I am a JavaScript thief. We all know we can steal code offered on the internet and try using it with our changes. I can honestly say anything I have written in JavaScript has been modified from a real Java coder.

Coding friends of mine, say Java is either their forte’ or not their cup of tea.

I wonder how I will feel about it. I notice this course also does not speak of OOP verse if/then programming. This could be a stupid question. Will we be using objects? Will we be using this MVC (model-view-controller) programming theories?

Do we touch on any standalone programs?

1. [Time to get Employable](https://the-coding-bootcamp.gitbooks.io/pre-work-book/content/module-11-cozy-up-to-your-new-career.html)

I look forward to this section. I need brush up on everything because I am 56 and have not looked for work in over 20 years. This can be a plus, but I understand some interviews test us now. I know a website portfolio will shine.

I want to live here in New Jersey, and I want to retire someday. I dropped everything to come do this. I do not want to start as entry level if I do not need too.

1. [What Next?](https://the-coding-bootcamp.gitbooks.io/pre-work-book/content/chapter11.html)

I have a career of programs to write in a more generic fashion and I have dreamed up markets that could easily adapt to these programs.

My parents were antique dealers. They owned a shop with consignment booths. Antique dealers have trouble inventorying their items and descriptions on point of sale systems often confuse both parties on what was sold. I envision a barcode scanning device available in the shop when a dealer brings items in.

The scan will provide an inventory code, photo and entry for the dealer to price and enter potential discount. A sticker will come out with data matrix barcode and readable price.

The point of sale software will scan this barcode and double check this item by picture as well as allow the cashier to enter password and see discount price without having to contact the dealer and hope to get ahold of them. This will tighten their loss levels. It can be secured with security beeper that goes off if an item is stolen.

This can be used at any package portion. If the dealer is not interested in having access to the inventory, he can get an immediate print out in paper. Barcode can include the picture or use cheaper scanner without photo ability.

This also solves the issue of dealers with consignment shop displays and EBay sales. Now the dealer can link his inventory to multiple ‘shops’ at the same time, so if the product comes to the checkout and has already sold on EBay, the cashier on point of sale can humbly apologize and set the item aside. This opens the dealer to sell from websites, EBay, other stores with same consignment ability… etc.

There is no end to learning new coding. I like this code school, and would hope to find some pseudo code shortcuts. I would love to make some pseudo code shortcuts of my own. I used Lynda.com for code classes for some time.