Computer Literacy

Daniel J. Okuniewicz Jr.

Computers are the most prolific tools in modern society. Indeed it seems that nearly everything is powered by computers, from cars to refrigerators, and the myriad of applications for computers is expanding at an ever-increasing rate. So it seems counter-intuitive that, despite the growing number and scope of computers in every-day life, that general computer knowledge is *decreasing*. But while counter-intuitive at first, at a second glance it's quite obvious that this phenomenon would occur, as it does with everything else.

Taking cars as an example, it used to be that the people who owned a car would do their own tune-ups. But as car technology grew beyond its original purpose, from a simple mode of transportation to a full-fledged luxury carriage (and it's worth an aside mention that, strangely enough, the status of a car is inversely related to its proliferation, i.e., although it has all the traits of a luxury it is also necessary for survival), the car became too abstract for the everyman.

I suppose that explanation doesn't really do it justice, though. Although they are more abstract than they were forty years ago, cars are still machines that we can understand with a little effort. The other half of the issue is a culture shift in opposition to self-service. There has been a dramatic shift away from "do it yourself" towards "ask a professional," at least when it comes to commercial products. People who do exhibit a DIY attitude tend to be written off as Hipsters or wannabes for daring to emulate the work of a professional. And in cases where they are not, such as in cars, mentioning that one does their own work is met with "Learned Ignorance," where people assume that they cannot possibly understand the subject matter, and so the conversation ends prematurely.

Learned Ignorance is prevalent in every mode of specialization. We now live in a society where it is more beneficial to master one thing than to be proficient at several things. The consequence is that any time there is an issue that falls outside of a person's area of knowledge they are completely lost. It is ironic that, in an era where information is instantly accessible anywhere at any time, people can no longer seem to function independently or use necessary problem-solving skills to solve their own problems. Of course specialization is not fully without benefits. Because of specialization we are able to advance areas of knowledge more quickly and efficiently. But the combination of Learned Ignorance and specialization produces a people which is totally reliant on the producers of their goods to hold their hands and direct them when a problem comes up.

The high level of dependence on manufacturers and producers is most prevalent in the computer industry; and I must specify that when I say computer industry I am also including "smart" products such as phones and televisions, as well as personal computers. STEM professions are so loudly and boldly professed as the most important areas of work that the proponents of these claims are blind to the fact that more STEM people will not solve the problem at hand, which is that most people do not know, nor are they willing to know, how to use a computer. And I'm not just talking about using the interface of a computer. Most everyone knows how to point and click their way through the graphical user interface of a personal computer. What's missing is the

other half of computer literacy: maintenance.

To use the car analogy again, it is ludicrous to the point of ridicule to teach people how to operate a car without also teaching them basic car maintenance. Although there are hundreds of different cars, there are still common things to do in order to maintain them: checking the oil, tire pressure, transmission fluid, mileage, and checking for signs of wear such as rust, worn tires, and headlight brightness. It is ludicrous, yet people are taught only how to drive a car and not how to take care of their car; and the little points of self-inspection go a long way to preserving the car's life. I can't imagine how many times a mechanic has to deal with problem customers who could have prevented all of their issues with some common sense, and yet insist that the mechanic is the one screwing them.

It is the same with computers. Schools that teach computer literacy (which at this point is probably all of them) focus too much on operation and not enough on maintenance. As a result computers which should last more than ten years die in half the time or faster. There is greater emphasis on having the latest and greatest than in having something that *still works fine*. It is an especially criminal attitude because the differences between the current year's line and the next year's line are so minimal that it isn't even worth the trouble of "upgrading."[1] I wager that if people were taught how to take care of their computers that the drive to "upgrade" would be greatly diminished.

The main argument used for having programming classes is that it is akin to teaching writing one hundred years ago. The game has changed so much that we now need a citizenry that is capable of programming. Assuming that is true, it isn't the full picture. What good are people who can program who don't know their way around the tools? It would be like a fisherman who can't fix his own boat. Pathetic. It might be said that computers are changing so quickly that it's impossible to teach basic maintenance, but I say that's completely wrong. Computers are hardly changing at all–aside from their size. The construction and overal design has been the same for personal computers for decades. Things just get smaller, and parts may be slightly different, but it all fits together the same way. Ah, but that is just the hardware, what about the software? It is true that software is changing with greater rapidity than hardware is. Windows 10 is miles different than Windows 8 and the two operating systems are just a few years apart.

But once again it is much the same with cars. Two cars may have totally different locations for the dipstick, but both of them have a dipstick. There are tools available for diagnosis and maintenance no matter what the operating system is. Diagnosing problems and implementing solutions is an essential skill for programmers anyway, so why not kill two birds with one stone?

In the end it really comes down to priorities. It would be absurd to suggest that we can teach everyone how to program but we cannot teach them how to be computer literate. And we must recognize that computer literacy *includes* maintenance and protecting oneself from malware. Going straight to programming, or use of office programs, or web development is jumping the gun. Without a solid foundation of the basics there is nothing for the rest to stand on. We are heading towards a society of educated ignoramuses who are utterly dependent on a "lower" class of people who hate them.

[1] That isn't to say that manufacturers should not be releasing new things, but that people should not be so quick to buy them.