

The Virtues of COMPUTING

An unusual juxtaposition, don't you think? Virtues and computing? Frustrations and computing, sure. Particularly if it's your happy lot in life to teach the fine art of programming to your little charges. But, dear reader, I'm here to tell you there are indeed virtues to be claimed if you are so inclined.

The first of these is self-confidence. "Surely you jest," I can hear you say. Self-confidence dwindles at the very thought of a keyboard and screen. Sad, but true.

Yet, it doesn't have to be. Your students' fragile little selves are really quite adept at leveraging a modicum of control into a bastion of confidence. Note the plethora of video salons, Nintendo and Sega systems, and those few, but highly addictive, Apple and IBM games of graphic intrigue and mental dexterity.

The key to fostering self-confidence is simplicity itself. Remember the old army slogan, "K.I.S.S." ("Keep It Simple Stupid.") That's also a useful motto when it comes to computing. Say what you're going to do. Say why you're going to do it. Say what you think will happen. Then, see if you're right.

Fairly early in the game, find out what happens if you do something wrong. The important point here is that very little of a devastating nature happens if you do do something wrong.

Anyone who has tried to land the airplane on a flight simulator knows that if you crack up the plane you simply press Return and are given a new one. No paperwork to fill out; no lengthy lawsuits. You just try it again. The same goes for almost all computing.

But how often (particularly if your school has only one computer) is the opposite attitude instilled. "Don't touch anything unless I tell you to!" With that one fell swoop goes any chance of build-

ing self-confidence through computing.

Usually, if you keep food and drinks away from disks and keyboard and pay close attention to backup procedures, computers are about as indestructible as chalkboard erasers. However, it helps if you yourself have a somewhat *laissez faire*, cavalier attitude toward all things computed.

The other often hidden virtue comes under the general heading of fair competition. Set aside, for the moment, your prejudice against the word *competition*. One of the reasons competition got a bad name is because most competitions aren't fair. One side is usually stronger or

smarter or faster or more agile at the start.

We try to make competitions fairer by designating leagues or letting each captain choose in turn so that talent is more evenly distributed. But in the end one side or team or person generally has a decided advantage.

That's why people are so intimidated by competition. The only competitions that really excite us are those that are close to fair.

Computing offers a competition that is truly fair. It's you against you. How much better can you get? Can you beat your highest score? The vast majority of video games are made for only one player. The only difference between learning how to program a computer and learning how to play PAC-MAN is that programming takes a little longer and pays a lot better.

Once this concept of you against you sinks in, you can share its potential with your students. When they understand its limitless possibilities, they'll tackle the challenge of making that computer do whatever they want it to do.

But there's a catch. How do you—and your students—react to frustration? To get the computer to perform, you have to be willing to overcome problems and persevere until you've succeeded. You will reach your goal only if you want it badly enough.

This can be a much more challenging competition than beating someone else in the 100-yard dash. Do you persevere despite difficulties? Or do you quit before you reach your goal? Do you reestablish your priorities by lowering your expectations? Or do you press on—read, ask, explore, compare, until you wrench out of that computer what you want it to do.

We'll look at other virtues in future columns. But through the years these two have surfaced as the most meaningful to me. Next time, the proper attitude in which to compute.—Dave Ruskjer. □

Dave Ruskjer is President of Touch Talk Technologies, Mayo, Maryland.