Recovering from an Computer Science Education

I had originally planned to name this "Undoing the Damage of the Computer Science Education," but that was too link-baity and too extreme. https://huliypin.com/ There is real value in having a degree in computer science. First, you'll be able to find an occupation that pays well. You've also been able make amazing and useful items. There is a downside to this, too. You can get so involved in the theoretical and technical aspects that it's easy to forget the joy of it is making beautiful and useful things. It happened to me, and it took some time to recover.

These are just a few of the things that helped me.

If it is directly related to what you are working on, stay clear of technical forums. It's all too easy to get wrapped up in discussions of the legitimacy of functional programming, whether or whether Scheme can be used to develop commercial applications or how terrible PHP is. It's easy to lose your way the more you go into this.

Continue to work on real projects that align with your interests. If you like designing games, write games. If you like photography write a photo organizer or camera application. Don't approach things wrong-way-around, thinking that "a photo organizer written in Haskell" is more important than "a photo organizer which solves a specific problem using photo organizers."

If you find yourself repeatedly making excuses for technology, then it is worthwhile to understand it and apply it. All jokes and snide remarks aside Perl is tremendously useful. It's the same for PHP, Java and C++. Who wins, the person who has been screaming Java for 10 years or the creator of Minecraft who just used the language and made hundreds of millions of dollars?

Don't become an advocate. This is the reverse of the first point. If Linux or Android or Scala can be helpful in the project you're creating, then that's great! The fact that you're using it is proof of its usefulness. It doesn't matter if everyone else uses it.

Concentrate on the end product and not the "how" of your passion. Woodworkers can transform into tool collectors. Photographers can be obsessed with comparison of spec. Don't get caught up in that and concentrate on the images you're creating.

Do something creative. Learn to pixel art, write songs, or short stories. These projects also have a faster turnaround time than other type of software project.

Be widely read. There are endless books about architecture, books by naturalists, both classic and popular modern novels, and most of them do not have anything to do with computers, programming or science fiction.

permalink January 15 2012

In the past

Follow-up to "A Programming Idiom You've Never Heard Of"A Programming Idiom You've Never Heard Of2011 Retrospective User Experience Intrusions in iOS 5Photography as a Non-Technical Hobby archive

Twitter /Mail

James Hague is a recovering programmer who has been making video games since the 1980s. Programming Without Being Obsessed with Organizational and Programming Skills beat Algorithmic Wizardry are good starting points. The 2012 Retrospective is a good option for older material.