

Non-Myths About Programming

Mordechai (Moti) Ben-Ari
Department of Science Teaching
Weizmann Institute of Science
Rehovot 76100 Israel

<http://stwww.weizmann.ac.il/g-cs/benari/>

Copyright 2010 by Mordechai (Moti) Ben-Ari. This work is licensed under the Creative Commons Attribution Non-Commercial No Derivs 3.0 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-nd/3.0/>; or, (b) send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.

This article will be published as a “Viewpoint” in the Communications of the ACM.

This essay is based on my keynote speech at the *Sixth International Computing Education Research Workshop*, held in Aarhus, Denmark, August 9-10, 2010. The talk began with the presentation of a short play *Aunt Jennifer*, in which Tiffany, a high school student, attributes her mother's dreary and poverty-stricken life as a checkout clerk in a supermarket to *rotten luck*, while attributing the pleasant life of her Aunt Jennifer, a software engineer, to *good luck*. Despite her high grades in mathematics, Tiffany rejects her guidance counselor's offer to help her obtain a scholarship to study computer science.¹

The decline of interest in studying computer science is usually attributed to a set of perceptions that students have about the subject. Many educators react to these perceptions as if they were *myths* and try to refute them. I believe that the perceptions of students are roughly true when looked at in isolation, and that the proper way to address these *non-myths* is to look at them within the context of “real life.” When examined in a broader context, a more valid image of computer science can be sketched, and this can be used to provide more accurate guidance to students who are deliberating whether to study computer science.

I will express the non-myths in terms of programming and justify this decision in a later article.

Non-myth #1: Programming is boring.

It is one of the unfortunate facts of life that all professions become routine and even boring once you develop a certain level of skill. Of course there are innumerable “McJobs”—inherently boring occupations in factories and service industries—that many people must do. But even prestigious professions are not exempt from boredom: I have heard physicians and attorneys complain about boredom. Consider physicians: either you become a general

¹ The script of the play can be downloaded from <http://stwww.weizmann.ac.il/g-cs/benari/articles/aunt-jennifer.pdf>.

practitioner and at least 9 out of 10 patients come to you with routine, “boring,” complaints, or you become a specialist, adept at performing a small number of procedures. After you have done them hundreds or thousand times, surely boredom sets in.

We can partly blame television for the impression that certain occupations are never routine or boring. The patient is always diagnosed and cured within 45 minutes, which is precisely the amount of time it takes to catch and convict a criminal. Occasionally, there are flashes of reality even on TV. *Law and Order* shows how detectives crack a case by following one small, frustrating clue after another. But even here, the 45-minute straight-jacket rules. Lt. Van Buren instructs her detectives: “Well, the victim was drunk, so check every bar within ten blocks.” Immediately, the scene cuts to the bartender who provides the next clue, but we don't see the hours of fruitless investigation by the detectives and the junior police officers that led to this moment.

The issue is not whether a subject is boring or not, but your ability to live with particular types of routine that can lead to boredom. Tiffany should be asking herself whether she prefers the routine of working as a psychologist—listening day-in, day-out to people complaining that their parents screwed up their lives—over the routine of constructing dozens of menu entries for the interface of an application.

Non-myth #2: You spend most of your working life in front of a computer screen.

For someone to refuse to study computer science for this reason is simply ridiculous. Most people sit in front of computers all day. Computer screens are ubiquitous in all professions in finance, administration, government offices, customer service, etc. I am certain that my travel agent spends more time looking at her computer screen than I do. From watching movies like *Wall Street* and *Working Girl*, I gather that securities traders spend their lives looking at six screens simultaneously.

Our medical system has recently undergone extensive computerization: a patient's history, test results and diagnostic images are stored on a network of computers. During a visit to a doctor, the patient sits quietly while the doctor reads the history, studies test results, orders X-rays, writes prescriptions and summarizes the visit, all on her computer. Of course doctors continue to perform physical examinations, but many modern diagnostic and surgical procedures involve “scopes” of various kinds, so that the physician is frequently looking at a computer screen.

Tiffany is free to decide that options trading is more exciting than programming, but that choice is not going to save her from the constant use of computers. Certainly, sitting in front of a computer developing software for an insurance company is preferable to sitting in front of a computer entering data from insurance claims.

Non-myth #3: You have to work long hours.

People who work in hi-tech industries complain about long hours, but this is true of many occupations, including prestigious professions, in particular, in the early stages before you achieve a high level of competence and the freedom to work independently. The

competition among young attorneys to clock hours is notorious. Young scientists work long hours in an effort to expand their list of publications during the short period before they are reviewed for tenure.

In 1984, Libby Zion, an 18-year-old student, died in a New York hospital from a fatal drug interaction. She was being cared for by young, overworked, interns and residents, who were not aware of a medication that she had been taking. New York subsequently enacted a law forbidding residents from working more than 80 hours a week. In comparison, spending 50 hours a week working as a software engineer doesn't seem so bad.

A career as an airline pilot sounds more adventurous than a career as a programmer, but Tiffany should not choose to become a pilot in the expectation of fewer hours at work. Spending long hours in a cubicle in a hi-tech firm, where your hours are flexible and you are free to go out for lunch or to the gym, is not as difficult as being cooped up in the small cockpit of an airplane for many hours at a time, on a schedule over which you have no control.

Non-myth #4: Programming is asocial.

Yes, but it depends what you mean by asocial. It is true that a programmer spends long hours by herself in front of a computer screen, although there are also meetings with team members and customers. There certainly are "social" professions where you are in constant contact with other people. The problem is that in most cases the human contact is *superficial* and *asymmetrical*, because you don't "chat" with your "clients." You may not even want to develop a warm relationship with your clients, for example, if you are a police detective interrogating hardened criminals.

A physician is almost always in contact with other people, but much of that is superficial contact with patients. A consultation may take just 15 or 20 minutes, once every few weeks or months. Certainly, the contact is asymmetrical: I tell my doctor every detail of my life that is related to my health, while she tells me nothing about hers.

Nursing is considered to be one of the most caring of professions, but the reality of modern medical care is far from the romantic image of nursing. I recall being hospitalized for tests and feeling stressed out, but Chrissie Williams and Donna Jackson (nurses from the BBC medical soap opera *Holby City*) did not come over to hold my hand and reassure me. The nurses at the hospital were themselves stressed out with the responsibility for 40 patients, and they barely had time to perform the myriad technical aspects of the job such as administering medication and measuring vital signs.

It is reasonable for Tiffany to choose to become a social worker because she likes helping people directly, but she must remember that she will not become a friend to her clients.

Non-myth #5: Programming is only for those who think logically.

Well, yes.

The nature of programming needs clarification. I define programming as any activity where a computation is described according to formal rules. Painting a picture is not programming: first, it obviously does not describe a computation, and, second, you are free to break whatever rules there are. At worst, they will call you an “Impressionist” and not buy your paintings until after you are dead. Constructing a website and building a spreadsheet are both *programming*, because you have to learn the rules for describing the desired output (even if the rules concern a sequence of menu selections and drag-and-drop operations), and you have to debug incorrect results that result from not following the rules.

Tiffany's good grades in mathematics imply that she has the ability to think logically. She may prefer to study music so that she can play violin in a symphony orchestra, but she should certainly consider studying computer science and her guidance counselor should insist that this alternative be thoroughly explored.

Non-myth #6: Software is being outsourced.

Of course it is. However, the share of software being outsourced is relatively small compared with that in manufacturing. This is not a fluke but an intrinsic aspect of software. Almost by definition, “soft”-ware is used whenever flexibility and adaptation to requirements is needed. If a machine tool is going to turn out the same screw throughout its entire lifetime, it can be outsourced and programmed in “hard”-ware.

Software development can also be a path to other professional activities like systems design and marketing, since *software reifies the proprietary knowledge of a firm*. A bank might outsource the building of its website, but it is not likely to outsource the development of software to implement algorithms for pricing options or analyzing risk, because this proprietary knowledge is what contributes directly to the bank's success.

It would be reasonable for Tiffany to prefer designing jewelry over studying computer science, but not because software is being outsourced. It is more likely that her jewelry business will fail when confronted with outsourced products than it is that her programming job at Boeing or Airbus will be outsourced.

Non-myth #7. Programming is a well paid profession.

That's great. Potential earnings shouldn't be the only consideration when choosing a profession, but it is not immoral to consider what sort of future you will be offering your family. It would be a good idea to remind Tiffany that the chasm between the lifestyles of her mother and Aunt Jennifer is not the result of luck.

I recently read the controversial book *Freakonomics* by Steven D. Levitt and Stephen J. Dubner [1]. The third chapter—*Why Do Drug Dealers Still Live with Their Moms?*—based upon the work of sociologist Sudhir Venkatesh [3] is quite relevant to the issue of potential earnings. As a graduate student, Venkatesh was able to observe and document the lives of the members of a drug gang, and he eventually obtained their financial records. These were analyzed by Levitt, an economist, who came up with the following conclusion, expressed as a question:

So if crack dealing is the most dangerous job in America, and if the salary was only \$3.30 an hour, why on earth would anyone take such a job? (p. 110)

The answer:

Well, for the same reason that a pretty Wisconsin farm girl moves to Hollywood. For the same reason that a high-school quarterback wakes up at 5 a.m. to lift weights. They all want to succeed in an extremely competitive field in which, if you reach the top, you are paid a fortune (to say nothing of the attendant glory and power). (p. 110)

The result:

The problem with crack dealing is the same as in every other glamour profession: a lot of people are competing for a very few prizes. Earning big money in the crack gang wasn't much more likely than the Wisconsin farm girl becoming a movie star or the high-school quarterback playing in the NFL. (p. 111)

Ambition to succeed in a glamour profession is not something to be deplored, but a young person must receive advice and support on what to do if she is not the 1 in 10,000 who succeeds. If Tiffany wants to become a professional singer, I would not try to dissuade her, but I would really like for her to pursue a CS degree part time while she tries to advance her singing career.

The real world is not so bad

I would like to end with a striking image that I found on the NASA website (Figure 1). It shows Margaret Hamilton sitting in a mockup of the Apollo space capsule. Hamilton was the chief software engineer for the development of the Apollo flight software. She and her team developed new techniques of software engineering, which enabled their software to perform flawlessly on all Apollo missions. Later, she went on to establish her own software company.

Hamilton looks like she is having a lot of fun checking out the programs that she and her team developed. I am sure that the long hours and whatever routine work the job involved were placed into perspective by the magnitude of the challenge, and there is no question that she felt immense satisfaction when her software successfully landed Neil Armstrong and Buzz Aldrin on the moon. I do not know if Hamilton felt locked out of the male-dominated "clubhouse" [2], but my guess is that the difficulty of the task, the short schedule and the weight of the responsibility felt by the whole team would have made such issues practically non-existent.

Teachers, parents and guidance counselors have the responsibility to explain the facts of life to talented young people: computer science and programming may seem like boring activities suitable only for asocial geeks, but a career like Margaret Hamilton's is more fulfilling and more rewarding than what awaits those who do not study science and engineering based upon superficial perceptions of these professions.

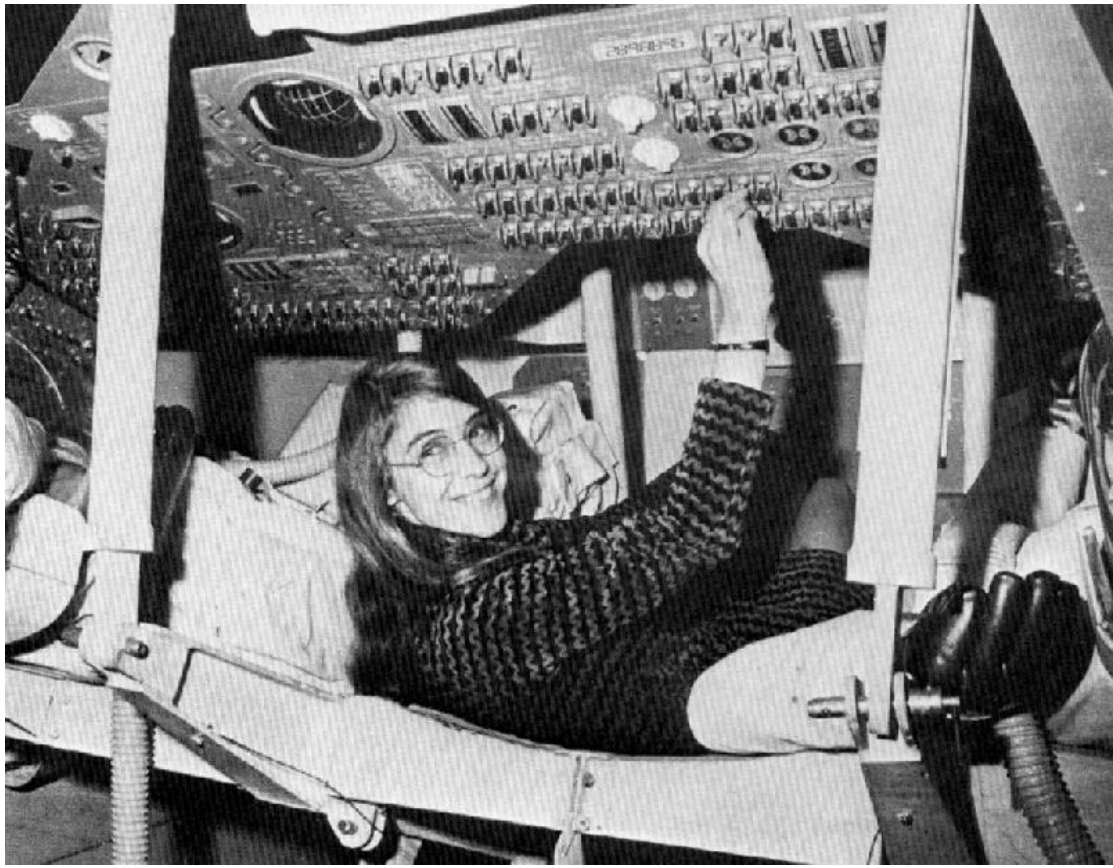


Figure 1. Margaret Hamilton

The figure appears in the 2003 report of the NASA Inventions & Contributions Board.² Hamilton received an Exceptional Space Act Award, one of only 128 awards granted from 1990 through 2003.

² http://www.nasa.gov/pdf/251093main_The_NASA_Heritage_Of_Creativity.pdf.

Acknowledgement

I would like to thank Mark Guzdial for his helpful comments on an earlier version.

References

1. Steven D. Levitt and Stephen J. Dubner. *Freakonomics: A Rogue Economist Explores the Hidden Side of Everything*. London: Allan Lane, 2005.
2. Margolis, J. and Fisher, A. *Unlocking the Clubhouse: Women in Computing*. Cambridge, MA: MIT Press, 2002.
3. Sudhir Venkatesh. *Gang Leader for a Day: A Rogue Sociologist Crosses the Line*. London: Allan Lane, 2008.