

Paul De Palma
Department of Mathematics and Computer Science
Gonzaga University
Spokane, WA 99258-2615
depalma@gonzaga.edu
509.323.3908

Prospectus

Dim Sum for the Mind ***Reflections on the Science and Industry of Computing***

This book is a personal tour of computer science for educated non-specialists. The book has three unifying themes. The first is that microcomputer hardware and software do not exhaust the richness and depth of the field. My contention is that the principles underlying commercial hardware and software are more interesting, and at least as fun, as web surfing. The second is that no discussion of computing is complete without considering its social context. Several of the chapters investigate the sociology of computing explicitly, and almost all include at least some consideration of context. The third theme is that the sort of writing anyone wants to read does more than convey information--else we would all curl up with textbooks on cold winter nights. The chapters in the book are part exposition, part personal reflection, always opinionated. The closest examples I can think of are some of the essays in Ellen Ullman's *Technophilia and its Discontents* (City Lights, 1997), and Robert Sapolsky's *The Trouble with Testosterone And Other Essays on the Biology of the Human Predicament* (Touchstone, 1998).

Table of Contents

Introduction: Most of us don't want a whole meal of chicken feet

In the introduction, I discuss the Chinese practice of dim sum and its relationship to the book. My contention is that a taste of many foods, especially when they are unfamiliar, is sometimes more pleasing than a meal of only one. The chapters in the book, like dim sum, can be tasted in any order.

1. Microcomputers, computer literacy and other sorrows

Criticizes the current contention that those of us not outfitted with the latest hardware and software will be consigned, along with our children, to lives of penury and ignorance. An earlier version of this chapter appeared in the Winter, 1999 issue of *The American Scholar*. It has been included in Houghton-Mifflin's: *Best American Science and Nature Writing*, edited by David Quammen and Bradford Bilger, 2000 and many other anthologies.

2. Why you don't understand your computer (but love it just the same)

Uses an early analysis of the Challenger disaster, the nature of software, and the culture of programming to explain why you can count on bridges, office towers, and jet planes but not computers. This chapter appeared in the Winter, 2004 issue of *The American Scholar*.

3. Women, Computing, and Peanut Butter Sandwiches

Year after year, I see mostly male faces in my computer science courses. This chapter uses a little-known fact, namely that women receive almost half of the Bachelor's degrees in mathematics, to offer a new solution to the problem of winning women into computer science. A condensed version of this chapter appeared in the *Communications of the Association of Computing Machinery* in 2000, the *JAMA* of computer science. It has been anthologized, as well.

4. Digitizing The Library of Congress

I frequently come across the fabulous--and fabulously wrong--claim that the Library of Congress is on the Internet. This is used as a justification for why public schools should throw still more money at Microsoft. In fact, the card catalog of the Library of Congress is on the Internet, along with a handful of literary classics. This chapter explores the technical reasons behind why we won't be seeing the *collections* of the Library of Congress on the Internet anytime soon.

5. Propositional logic, automated reasoning and the woes of the profession

While arguing that computing is more than microcomputer gadgetry, this chapter presents a short history of cognitive science, and artificial intelligence. It explains the AI technique of automatic theorem proving at a level accessible to an educated non-specialist.

6. The hard problems are easy

In 1998, chess Grand Master, Gary Kasparov, was defeated by a computer. That this happened barely four decades after researchers began to investigate capturing human expertise in programs, is surely one of the more remarkable achievements of recent years. This chapter explains some of the techniques computer scientists have developed to build machines that play expert-level adversary games like chess, checkers, and othello. It also speculates about why it is that chess has given up its secrets to researchers whereas language has remained intractable.

7. A little cryptography, a lot of privacy

Recently two senior city administrators in my town were fired because their affair, discovered through email monitoring, was interfering with their work. If these star-crossed lovers had used freely available cryptographic software, everyone in town would not be tittering over their tryst today. Current cryptographic techniques are remarkably secure and even more remarkably simple. This chapter explains basic cryptographic algorithms along with an elegant technique known as double-key cryptography.

8. The only trees I'll be seeing today are binary

Uses the occasion of an assignment I gave my students to discuss binary trees, permutations, a famous problem in computing, and the creativity necessary for even simple programs.

9. Finding peace in the details

This chapter shows how to get a computer to add the first N integers. Along the way, it considers loops, recursive programming, and proofs by mathematical induction, all in an attempt to convince the reader that the details of mathematical and technical arcana are a path to peace.

10. An Italian (American) among the cyborgs

This chapter uses the Italian peasant notion of the evil eye, the *malocchio*, something that I heard a lot about as a child, to reflect on the pace of change in computing and in our cities. This chapter appeared in *VIA: A Literary and Cultural Quarterly*, 2004.