Our past

Here's how computers arose.…

Ancient history

The first programmable computers were invented in the 1940's. Before then, people were stuck with the abacus, adding machine, and slide rule.

During the 1950's, 1960's, and 1970's, most computers used punched cards — whose history is weird. The cards were first used for *weaving tapestries*. Where the cards had holes, rods could move through the cards; those moving rods in turn made other rods move, which caused the threads to weave pictures. That machine was called the **Jacquard loom**.

Charles Babbage

Charles Babbage was a wild-eyed English mathematician who, in the 1800's, believed he could build a fancy computing machine. He convinced the British government to give him lots of money, then bilked the government for more. Many years later — and many British pounds later — he still hadn't finished his machine. So he dropped the idea and — can you believe this? — tried to build an even fancier machine. He didn't finish that one either. You might say his life was a failure that was expensive for the British government.

But Charlie (as I'll call him) is admired by all us computerniks (in spite of his face, which was even sterner than Beethoven's), because **he was the first person to realize that a computing machine must consist of 4 parts:**

an input device (he used a card reader)
a memory (which he called "The Store")
a central processing unit (which he called "The Mill")
an output device (he used a printer)

Lady Lovelace

Lady Lovelace was one of Charlie's great admirers, but he never noticed her until she translated his stuff. And boy, it was impossible for him not to notice her translations. Her "footnotes" to the translation were three times as long as what she was translating!

She got very intense. She wrote to Charlie, "I am working very hard for you — like the Devil in fact (which perhaps I am)."

The two became lovebirds, though he was old enough to be her dad. (By the way, her dad was Lord Byron, the poet. She was Lord Byron's only "official" daughter. His other daughters were out-of-wedlock.) Some people think she was actually brighter than Charlie, despite Charlie's fame. She was better at explaining Charlie's machines and their implications than Charlie was. **Some people have dubbed her "the world's first programmer".**

Stunning She stunned all the men she met. She was so bright and… a woman! Here's how the editor of The Examiner described her (note the pre-Women's-Lib language!):

"She was thoroughly original. Her genius, for genius she possessed, was not poetic, but metaphysical and mathematical. With an understanding thoroughly masculine in solidity, grasp, and firmness, Lady Lovelace had all the delicacies of the most refined female character. Her manners, tastes, and accomplishments were feminine in the nicest sense of the word; and the superficial observer would never have divined the strength and knowledge that lay hidden under the womanly graces. Proportionate to her distaste for the frivolous and commonplace was her enjoyment of true intellectual society. Eagerly she sought the acquaintance of all who were distinguished in science, art, and literature."

Mad Eventually, she went mad. Mattresses lined her room to prevent her from banging her head. Nevertheless, she died gruesomely, at the ripe young age of 36, the same age that her dad croaked. (I guess premature death was popular in her Devilish family.)

Who's the heroine? I wish feminists would pick a different heroine than Lady Lovelace. She was not the most important woman in the history of computing.

Far more important were Grace Hopper and Jean Sammet. In the 1950's Grace Hopper invented the first programming languages, and she inspired many of us programmers until her recent death. Jean Sammet headed the main committee that invented Cobol; she's the world's top expert on the history of programming languages, and she's been president of the computer industry's main professional society, the ACM.

Lady Lovelace was second-string to Babbage. Grace Hopper and Jean Sammet were second-string to nobody. Since Hopper was an Admiral in the Navy, she irked some of us doves; but whenever she stepped in front of an audience, she got a standing ovation because we all realize how crucial she was to the computer industry.

But I'm straying from my story.…

Herman Hollerith

The U.S. Bureau of the Census takes its census every ten years. To tabulate the results of the 1880 census, the Bureau took *7 years*: they didn't finish until 1887. When they contemplated the upcoming 1890 census, they got scared; at the rate America was growing, they figured that tallying the 1890 census would take 12 years. In other words, the results of the 1890 census wouldn't be ready until 1902. So they held a contest to see whether anyone could invent a faster way to tabulate the data.

The winner was Herman Hollerith. **He was the first person to successfully use punched cards to process data.**

Hermie (as I'll call him) was modest. When people asked him how he got the idea of using punched cards, he had two answers. One was, "Trains": he had watched a train's conductor punch the tickets. His other, more interesting answer was, "Chicken salad". After saying "Chicken salad", he'd pause for you to ask the obvious question, "Why chicken salad?" Then he'd tell his tale:

One day, a girl saw him gulping down chicken salad. She said, "Oh, you like chicken salad? Come to my house. My mother makes excellent chicken salad." So he did. And her father was a head of the Census. (And he married the girl.)

By the way, Herman Hollerith hated one thing: spelling. In elementary school, he jumped out a second-story window, to avoid a spelling test.

In some versions of Fortran, every string must be preceded by the letter H. For example, instead of saying —

```
'DOG'
```

you must say:

```
3HDOG
```

The H is to honor Herman Hollerith.

The Census used Hollerith's punched-card system in 1890 and again in 1900.

In 1910 the Census switched to a fancier system created by a Census Bureau employee, James Powers, who later quit his job and started his own company, which merged into **Remington-Rand-Sperry-Univac**. Meanwhile, Herman Hollerith's own company merged into **IBM**. That's how the first two computer companies began doing data processing.

World War II

The first programmable computers were invented in the 1940's because of World War II. They could have been invented sooner — most of the know-how was available several decades earlier — but you can't invent a computer unless you have big bucks for research. And the only organization that had big enough bucks was the Defense Department (which in those days was more honestly called the "War Department"). And the only event that was big enough to make the War Department spend that kind of money was World War II.

Of course, the Germans did the same thing. A German fellow, Konrad Zuse, built computers which in some ways surpassed the American ones. But since the Germans lost the war, you don't hear much about old Konrad anymore. Fortunately, throughout World War II the German military ignored what he was doing.

During the 1940's, most computers were invented at universities, usually funded by the War-Defense Department.

Some of the most famous computers were the **Mark I** (at Harvard with help from IBM), the **Eniac** and the **Edvac** (both at the University of Pennsylvania), the **Whirlwind** (at the Massachusetts Institute of Technology, M.I.T.), and the **Ferranti Mark I** (at the University of Manchester, in England). Which of those computers deserves to be called "the first programmable computer"? The answer's up for grabs. Each of those machines had its own peculiar hang-ups and required years of debugging before working well.

Each of those computers was unique: no two were alike.

First generation (1951-1958)

The first computer to be mass-produced was the Univac I, in 1951. It was made by the same two guys (Eckert & Mauchly) who'd built the Eniac and Edvac at the University of Pennsylvania. (Mauchly was an instructor there, and Eckert was the graduate student who did the dirty work.) While others at the school were helping build the Edvac, Eckert & Mauchly left and formed their own company, which invented and started building the Univac. While building the Univac, the Eckert-Mauchly company merged into Remington Rand (which later merged into Sperry-Rand, which later merged into Unisys).

The Univac I was so important that historians call it the beginning of the "first generation". As for computers before Univac — historians disparagingly call them the "zeroth generation".

So the first generation began in 1951. It lasted through 1958. Altogether, from 1951 to 1958, 46 of those Univacs were sold.

46 might not sound like many. But remember: in those days, computers were very expensive, and could do very little. Another reason why just 46 were sold is that newer models came out, such as the Univac 1103, the Univac 80, and the Univac 90. But the biggest reason why only 46 of the Univac I were sold is IBM.

The rise of IBM Although IBM didn't begin mass-marketing computers until 1953 — two years after Univac — the IBM guys were much better salesmen, and soon practically everybody was buying from IBM. During the first generation, the hottest seller was the **IBM 650**. IBM sold hundreds and hundreds of them.

There were many smaller manufacturers too. People summarized the whole computer industry in one phrase: **IBM and the Seven Dwarfs**.

Who were the dwarfs? They kept changing. Companies rapidly entered the field — and rapidly left when they realized IBM had the upper hand. By the end of the first generation, IBM was getting 70% of the sales.

Primitive input and output During the first generation, there were no terminals. To program the Univac I, you had to put the program onto magnetic tape (by using a non-computerized machine), feed that tape to the computer, and wait for the computer to vomit another magnetic tape, which you had to run through another machine to find out what the tape said.

One reason why the IBM 650 became more popular was that it could read cards instead of tapes. It really liked cards. In fact, the answers came out on cards. To transfer the answers from cards to paper, you had to run the cards through a separate non-computerized machine.

Memory At the first generation's beginning, there were no RAM chips, no ROM chips, and no "core memory". Instead, the Univac's main memory was banks of liquid mercury, where the bits were stored as ultrasonic sound waves. It worked slowly and serially, so the access time ranged from 40 to 400 microseconds per bit.

Univac's manufacturer and IBM started playing around with a different kind of memory, called the Williams tube, which was faster (10 to 50 microseconds); but since it was less reliable, it didn't sell well.

In 1953, several manufacturers started selling computers that were much cheaper, because they used super-slow memory: it was a drum that rotated at 3600 rpm, giving an average access time of 17000 microseconds (17 milliseconds). (During the 1970's, some computers still used drums, but for *auxiliary* memory, not for *main* memory.) The most popular first generation computer, the IBM 650, was one of those cheap drum computers.

Eventually, computer manufacturers switched to a much better scheme, called **core memory**. It consists of tiny iron donuts strung on a grid of wires, whose electrical current magnetizes the donuts. Each donut is one bit and called a **core**. The donuts are strung onto the wire grid by hand, by women knitting.

Core memory was first conceived in 1950. The first working models were built in 1953 at MIT and RCA, which argued with each other about who owned the patent. The courts decided in favor of MIT, so both RCA *and IBM* came out with core-memory computers. Core memory proved so popular that most computers used it through the 1970's, though in the 1980's RAM chips finally overshadowed it, since RAM chips don't require hiring knitters.

Languages During the first generation, computer programming improved a lot. During the early 1950's, all programs had to be written in **machine language**. In the middle 1950's, **assembly language** became available. By 1958, the end of the first generation, 3 major high-level languages had become available: **Fortran**, **Algol**, and **Apt**.

Fancy programs Programmers tried to make computers play a decent game of chess. All the attempts failed. But at IBM, Arthur Samuel had some luck with checkers:

He got his first checkers program working in 1952 and then continually improved it, to make it more and more sophisticated. In 1955, he rewrote it so that it learned from its own mistakes. In 1956, he demonstrated it on national TV. He kept working on it. Though it hadn't reached championship level yet, it was starting to look impressive.

Computer music scored its first big success in 1956, on the University of Illinois' Illiac computer:

Hiller & Isaacson made the Illiac compose its own music in a style that sounded pre-Bach. In 1957, they made the program more flexible, so it produced many styles of more modern music. The resulting mishmash composition was dubbed "The Illiac Suite" and put on a phonograph record.

In 1954, IBM wrote a program that translated simple sentences from Russian to English. Work on tackling harder sentences continued — with too much optimism.

Second generation (1959-1963)

Throughout the first generation, each CPU was composed of vacuum tubes. Back in 1948, Bell Telephone had invented the transistor, and everybody realized that transistors would be better than vacuum tubes; but putting transistors into computers posed many practical problems that weren't solved for many years.

Finally, **in 1959, computer companies started delivering transistorized computers. That year marked the beginning of the second generation.** Sales of vacuum-tube computers immediately stopped.

All second-generation computers used core memory.

IBM The *first* company to make transistors for computers was Philco, but the most *popular* second-generation computer turned out to be the **IBM 1401**, because it was business-oriented and cheap.

IBM announced it in 1959 and began shipping it to customers in 1960.

Its core memory required 11½ microseconds per character. Each character consisted of 6 bits. The number of characters in the memory could range from 1.4K up to 16K. Most people rented the 1401 for about $8,000 per month, but you could spend anywhere from $4,000 to $12,000 per month, depending on how much memory you wanted, etc.

Altogether, IBM installed 14,000 of those machines.

IBM also installed 1,000 of a faster version, called the **1410**.

It required just 4½ microseconds per character, had 10K to 80K, and rented for $8,000 to $18,000 per month, typically $11,000.

Altogether, IBM produced six kinds of computers.…

small business computers:	the 1401, 1410, 1440, and 1460
small scientific computers:	the 1620
medium-sized business computers:	the 7010
medium-sized scientific computers:	the 7040 and 7044
large business computers:	the 7070, 7074, and 7080
large scientific computers:	the 7090 and 7094

CDC Several employees left Remington-Rand-Sperry-Univac and formed their own company, called **Control Data Corporation (CDC)**. During the second generation, CDC produced popular scientific computers: the 1604, the 3600, and the 3800.

Software During the second generation, software improved tremendously.

The 3 major programming languages that had been invented during the first generation (Fortran, Algol, and Apt) were significantly improved. 6 new programming languages were invented: **Cobol**, **RPG**, **Lisp**, **Snobol**, **Dynamo**, and **GPSS**.

Programmers wrote advanced programs that answered questions about baseball, wrote poetry, tutored medical students, imitated three-person social interaction, controlled a mechanical hand, proved theorems in geometry, and solved indefinite integrals. The three most popular sorting methods were invented: the Shuffle Sort, the Shell Sort, and Quicksort.

Third generation's dawn (1964-1967)

The third generation began with a big bang, in 1964. Here's what happened in 1964, 1965, 1966, and 1967.…

Families The first modern computer families were shipped. They were the **CDC 6600**, the **IBM 360**, and DEC's families (the **PDP-6**, **PDP-8**, and **PDP-10**).

Of those families, the CDC 6600 ran the fastest. The IBM 360 was the most flexible and was the only one that used integrated circuits (chips). The PDP-6 and PDP-10 were the best for timesharing. The PDP-8 was the cheapest.

Here are the dates:

CDC began shipping the CDC 6600 in 1964. IBM announced the IBM 360 in 1964 but didn't ship it until 1966. DEC began shipping the PDP-6 maxicomputer in 1964, the PDP-8 minicomputer in 1965, and the PDP-10 maxicomputer (a souped-up PDP-6) in 1967.

New languages IBM announced it would create **PL/I**, a new computer language combining Fortran, Cobol, Algol, and all other popular languages. It was designed especially for IBM's new computer, the 360. In 1966, IBM began delivering PL/I to customers.

Programmers invented the first successful languages for *beginners* using *terminals*. Those languages were **Basic**, **Joss**, and **APL**.

Dartmouth College invented the first version of **Basic** in 1964, and significantly improved it in 1966 and 1967.

The Rand Corporation invented **Joss** in 1964 for the Johnniac computer, and put an improved version (Joss II) on the PDP-6 in 1965. In the 1970's, three popular variants of Joss arose: a souped-up version (called Aid), a stripped-down version (Focal), and a business-oriented version (Mumps).

IBM completed the first version of **APL** in 1965 and put it on an IBM 7090. IBM wrote a better version of APL in 1966 and put it on an IBM 360. IBM began shipping APL to customers in 1967.

Stanford University invented the most popular language for statistics: **SPSS**.

Artificial intelligence Researchers calling themselves "experts in artificial intelligence" taught the computer to chat in ordinary English.

For example, Bertram Raphael made the computer learn from conversations, Daniel Bobrow made it use algebra to solve "story problems", The Systems Development Corporation made it know everything in an encyclopedia, General Electric made it answer military questions, Ross Quillian made it find underlying concepts, and Joe Weizenbaum made it act as a psychotherapist.

Also, Richard Greenblatt wrote the first decent chess program. It was good enough to play in championship tournaments against humans.

Era of boredom (1968-1974)

As you can see, the first three generations — up through 1967 — were exciting, full of action. But then, from 1968 to 1974, *nothing newsworthy happened*. That was the era of boredom.

During that era, progress was made, but it was gradual and predictable. Nothing dramatic happened.

Of course, nobody actually came out and said, "Life is boring." People phrased it more genteelly. For example, in September 1971 Robert Fenichel and Joe Weizenbaum wrote this introduction to *Scientific American*'s computer anthology:

"Partly because of the recent recession in the American economy, but more for reasons internal to the field, computer science has recently relaxed its pace. Work has not stopped, but that the current mood is one of consolidation can scarcely be doubted. Just a few years ago, computer science was moving so swiftly that even the professional journals were more archival than informative. This book could not then have been produced without great risk of misfocus. Today it's much easier to put the articles that constitute this book — even the most recent ones — into context."

Since the first generation had lasted eight years (1951-1958), and the second generation had lasted four years (1959-1963), people were expecting the third generation to last at most four years (1964-1967) and some kind of "fourth generation" to begin about 1968. But it never happened.

The only "major" announcement around then came in 1970, when IBM announced it would produce a new line of computers, called the **IBM 370**, which would make the IBM 360 obsolete. But to IBM's dismay, many computer centers decided to hang onto the old 360 instead of switching to the 370.

Since the 370's advantage over the 360 was small, not even IBM claimed the 370 marked a fourth generation. Computer historians, desperate for something positive to say about the 370, called it the beginning of the "late third generation", as opposed to the 360, which belonged to the "early third generation".

No consistency Unfortunately, in the entire history of computers, there was just one year all computer manufacturers acted together to produce something new. That year was 1959, when all manufacturers switched from vacuum tubes to transistors. Since 1959, we haven't had any consistency.

Although the third generation began with a "big bang" in 1964, each manufacturer was banging on a different drum. IBM was proclaiming how great the IBM 360 would be because it would contain integrated circuits; but other manufacturers decided to ignore integrated circuits for several years, and concentrated on improving other aspects of the computer instead. For many years after the beginning of the third generation, CDC and DEC continued to use discrete transistors (a sign of the second generation) instead of integrated circuits.

Why? The era of boredom happened for 3 reasons:

1. The preceding years, 1964-1967, had been so successful that they were hard to improve on.
2. When the Vietnam War ended, the American economy had a recession, especially the computer industry, because it had depended on contracts from the Defense Department. In 1969, the recession hit bottom, and computer companies had to lay off many workers. In that year, General Electric gave up and sold its computer division to Honeywell. In 1971, RCA gave up too and sold its computer division to Remington-Rand-Sperry-Univac.
3. The world wasn't ready yet for "the era of personal computing", which began in 1975.

Quiet changes During the era of boredom, these changes occurred — quietly.…

In 1970, DEC began shipping the **PDP-11**.

The PDP-8 and PDP-11 became the most popular minicomputers — far more popular than IBM's minicomputers. So in the field of minicomputers, IBM no longer had the upper hand.

Basic became the most popular language for the PDP-8 and PDP-11 and most other minicomputers (except IBM's, which emphasized RPG). In high schools and business schools, most of the introductory courses used Basic, instead of Fortran or Cobol.

Many businesses and high schools bought their own minicomputers, instead of renting time on neighbors' maxicomputers. The typical high-school computer class used a PDP-8. The richest high schools bought PDP-11's.

In universities, **the social sciences started using computers** — and heavily — to analyze statistics.

All new computer families used 8-bit bytes, so the each word's length was a multiple of 8 (such as 8, 16, 32, or 64).

Most older computer families, invented before the era of boredom, had used 6-bit bytes, so the length of each word had been a multiple of 6: for example, the PDP-8 had a word of 12 bits; the PDP-10 , Univac 1100, and General Electric- Honeywell computers had a word of 36 bits; and the CDC 6600 had a word of 60 bits. The IBM 360 was the first computer to use 8-bit bytes instead of 6-bit; during the era of boredom, all manufacturers copied that feature from IBM.

CRT terminals (TV-like screens attached to keyboards) got cheaper, until they finally became as cheap as hard-copy terminals (which use paper).

Most computer centers switched from hard-copy terminals to CRT terminals, because CRT terminals were quicker, quieter, and could do fancy editing. Also, many computer centers switched from "punched cards and keypunch machines" to CRT terminals.

Interest in new computer languages died. Most computer managers decided to stick with the old classics (Fortran and Cobol), because switching to a progressive language (such as PL/I) would require too much time to retrain the programmers and rewrite all the old programs.

Programmers made two last-ditch attempts to improve Algol. The first attempt, called **Algol** 68, was too complicated to win popular appeal. The second attempt, called **Pascal**, eventually gained more support.

Maxicomputers were given **virtual core** — disks that pretend to be core, in case you're trying to run a program that's too large to fit into core.

Memory chips got cheaper, until they were finally cheaper than core. Most manufacturers switched from core to memory chips.

In 1971, **Intel** began shipping **the first microprocessor** (complete CPU on a chip).

It was called the **4004** and had a word of just 4 bits. In 1972, Intel began shipping an improved version, the **8008**, whose word had 8 bits. In 1973, Intel began shipping an even better version, the **8080**.

Micro history

In 1975, the first popular microcomputer was shipped. It was called the **Altair** and was built by a company called **Mits**. It cost just $395.

It was just a box that contained a CPU and very little RAM: just ¼ of a K!

It included no printer, no disk, no tape, no ROM, no screen, and not even a keyboard! The only way to communicate with the computer was to throw 25 switches and watch 36 blinking lights.

It didn't understand Basic or any other high-level computer language. To learn how to throw the switches and watch the blinking lights, you had to take a course in "machine language".

You also had to take a course in electronics — because the $395 got you just a kit that you had to assemble yourself by using a soldering iron and reading electronics diagrams. Moreover, when you finished building the kit, you noticed some of the parts were missing or defective, so that you had to contact Mits for new parts.

That computer contained several empty slots to hold PC cards. Eventually, many companies invented PC cards to put into those slots. Those PC cards, which were expensive, let you insert extra RAM and attach a printer, tape recorder, disk drives, TV, and terminal (keyboard with either a screen or paper).

Bill Gates invented a way to make the Altair handle Basic. He called his method **Microsoft Basic**. He patterned it after DEC's Basic; but he included extra features that exploited the Altair's ability to be "personal", and he eliminated features that would require too much RAM.

Gary Kildall invented a disk operating system that the Altair could use. He called that operating system **CP/M**.

Many companies built computers that imitated the Altair. Those imitations became more popular than the Altair itself. Eventually, the Altair's manufacturer (Mits) went out of business.

Computers that imitated the Altair were called **S-100 bus computers**, because they each used a Standard cable containing 100 wires.

In those days, the microcomputer industry was standardized. Each popular microcomputer used Microsoft Basic, CP/M, and the S-100 bus. The microcomputer was just a box containing PC cards; it had no keyboard, no screen, and no disk drive. A cable went from the microcomputer to a terminal, which was priced separately. Another cable went from the microcomputer to a disk drive, which was also priced separately.

Built-in keyboards

In 1977, four companies began selling microcomputers that had built-in keyboards, so you didn't have to buy a terminal. Their computers became popular immediately. The four companies were **Processor Technology**, **Apple**, **Commodore**, and **Radio Shack**.

Processor Technology's computer was called the **Sol 20**, to honor Solomon Libes, an editor of Popular Electronics.

Apple's computer was called the **Apple 2**, because it improved on the Apple 1, which had lacked a built-in keyboard.

Commodore's computer was called the **Pet** (inspired by Pet Rocks).

Radio Shack's computer was called the **TRS-80**, because it was manufactured by Tandy's Radio Shack and contained a Z-80 CPU.

For a fully assembled computer, Processor Technology charged $1850, Apple charged $970, Commodore charged $595 (but quickly raised the price to $795), and Radio Shack charged $599 (but soon lowered the price to $499).

Notice that Commodore and Radio Shack had the lowest prices. Also, the low prices from Commodore and Radio Shack *included* a monitor, whereas the prices from Processor Technology and Apple didn't. So Commodore and Radio Shack were the real "bargains".

In those days, the cheapest computers were the most popular.

The cheapest and most popular computer was Radio Shack's.
The second cheapest and second most popular was Commodore's Pet.
The third cheapest and third most popular was the Apple 2.
Processor Technology, after a brief fling of popularity, went bankrupt.

The most expensive kind of microcomputer was the CP/M S-100 bus system. It was the oldest kind, so it had accumulated the most business software.

Improvements

In 1978 and 1979, the 3 main companies (Apple, Commodore, and Radio Shack) improved their computers.

The improved Apple 2 was called the **Apple 2-plus**. The improved Commodore Pet was called the **Commodore Business Machine (CBM)**. The improved Radio Shack TRS-80 was called the **TRS-80 model 2**.

After announcing the Apple 2-plus, Apple Computer Company stopped selling the plain Apple 2.

Commodore continued selling its old computer (the Pet) to customers who couldn't afford the new version (the CBM), which cost more. Likewise, Radio Shack continued selling its model 1 to customers who couldn't afford the model 2.

Texas Instruments & Atari

In 1979, Texas Instruments (TI) and Atari began selling microcomputers and priced them low.

TI's microcomputer was called the **TI 99/4**. Atari offered *two* microcomputers: the **Atari 400** and the **Atari 800**.

TI charged $1150. Atari charged $1000 for the regular model (the Atari 800) and $550 for the stripped-down model (the Atari 400).

TI's price included a color monitor. Atari's prices did *not* include a screen; you were to attach Atari's computers to your home's TV.

TI's computer was terrible, especially its keyboard. The Atari 800 computer was wonderful; reviewers were amazed at its easy-to-use keyboard, easy-to-use built-in editor, gorgeous color output on your TV, child-proofing (safe for little kids), and dazzling games, all at a wonderfully low price! It was cheaper than an Apple (whose price had by then risen to $1195) and yet was much *better* than an Apple.

From that description, you'd expect Atari 800 to become the world's best-selling computer, and the TI 99/4 to become an immediate flop. Indeed, that's what most computer experts hoped. And so did the TI 99/4's product manager: when he saw what a mess the TI 99/4 had become, he quit TI and went to work for Atari, where he became the product manager for the Atari 400 & 800!

But even though computer experts realized that TI's computer was junk, TI decided to market it aggressively:

TI coaxed Milton Bradley and Scott Foresman to write lots of programs for the 99/4. TI paid researchers at MIT to make the 99/4 understand Logo (a computer language used by young children and very popular in elementary schools). TI improved the keyboard just enough so that people would stop laughing at it; the version with the new keyboard was named the **99/4A**. TI paid Bill Cosby to praise the 99/4A and ran hundreds of TV ads showing Bill Cosby saying "wow". TI dramatically slashed the $1150 price to $650, then $150, and then finally to just $99.50! (To bring the price that low, TI had to exclude the color monitor from the price; instead, TI included a hookup to your home's color TV.)

By contrast, Atari did hardly anything to market or further improve the Atari 400 & 800.

Atari concentrated on its other products: the big Atari game machines (which you find in video arcades) and the Atari VCS machine (which plays video games on your home TV).

The TI 99/4A therefore became more popular than the Atari 400 & 800 — even though the TI 99/4A was inherently worse.

Sinclair, Osborne, backlash

In 1980 and 1981, two important companies entered the microcomputer marketplace: Timex Sinclair (1980) and Osborne (1981).

The first complete computer selling for less than $200 was invented by a British chap named Clive Sinclair and manufactured by Timex.

The original version was called the **ZX-80** (because it was invented in 1980, contained a Z-80 CPU, and was claimed to be "Xellent"); it sold for $199.95. In 1981, Clive Sinclair invented an improved version, called the **ZX-81**. Later, he and Timex invented further improvements, called the **ZX Spectrum** and the **Timex Sinclair 1000**. When TI dropped the price of the TI 99/4A to $99.50, Timex retaliated by dropping the list price of the Timex Sinclair 1000 to $49.95, so the Timex Sinclair 1000 remained the cheapest complete computer.

In April 1981, Adam Osborne began Osborne Computer Corp. and began selling the **Osborne 1** computer, designed by Lee Felsenstein (who'd invented Processor Technology's Sol 20 computer).

The Osborne 1 computer included practically everything a business executive needed: its $1795 price included a keyboard, a monitor, a Z-80A CPU, a 64K RAM, two disk drives, CP/M, Microsoft Basic, a second version of Basic, the WordStar word processor, and the SuperCalc spreadsheet program. Moreover, it was the world's first portable business computer: the entire computer system (including even the monitor and disk drives) was collapsible and turned itself into an easy-to-carry attaché case. (Many years later, Compaq copied Osborne's idea.)

While Timex Sinclair and Osborne were entering the marketplace, Radio Shack, Apple, and Commodore were introducing new computers of their own:

In 1980, Radio Shack began selling three new computers. The **TRS-80 model 3** replaced Radio Shack's cheapest computer (the model 1) and was almost as good as Radio Shack's fanciest computer (the model 2). The **TRS-80 Color Computer** drew pictures in color and cost less than the model 3. The **TRS-80 Pocket Computer** fit into your pocket, looked like a pocket calculator, and was built for Radio Shack by Sharp Electronics in Japan.

In 1980, Apple began selling the **Apple 3**. It was overpriced; and to make matters worse, the first Apple 3's that rolled off the assembly line were defective. Apple eventually lowered the price and fixed the defects; but since the Apple 3 had gotten off to such a bad start, computer consultants didn't trust it and told everybody to avoid it.

In 1981, Commodore began selling the **Vic-20**, which drew pictures in color and cost less than Radio Shack's Color Computer. In fact, the Vic-20 was the first computer that drew pictures in color for less than $300.

The Vic-20 originally sold for $299.95. When TI lowered the price of the TI 99/4A to $99.95, Commodore lowered the price of the Vic-20. At discount department stores (such as K Mart, Toys R Us, and Child World), you could buy the Vic-20 for just $85: it was still the cheapest computer that could handle color. (The Timex Sinclair 1000 was cheaper but handled just black-and-white.)

Moreover, the Vic-20 had standard Microsoft Basic, whereas the Timex Sinclair 1000 and TI 99/4A did not; so the Vic-20 was the cheapest computer that had standard Microsoft Basic. It was the cheapest computer that was pleasant to program.

Also, the Vic-20 had a nice keyboard, whereas the keyboards on the Timex Sinclair 1000 and TI 99/4A were pathetic.

The Vic-20 became immediately popular.

IBM PC

On August 12, 1981, IBM announced a new microcomputer, called the **IBM Personal Computer (IBM PC)**.

Although IBM had previously invented other microcomputers (the IBM 5100 and the IBM System 23 Datamaster), they'd been overpriced and nobody took them seriously — not even IBM. The IBM Personal Computer was IBM's first *serious* attempt to sell a microcomputer.

The IBM Personal Computer was a smashing success, because of its amazingly high quality and amazingly low price. It became the standard against which the rest of the microcomputer industry was judged.

Rise & fall

Let's take a closer look at how 3 computer companies — Commodore, Tandy, and Atari — rose & fell.

Commodore

A computer company called **Commodore** was called "the house that Jack built" because it was started by Jack Tramiel.

How Commodore began Jack began his career by being in the wrong place at the wrong time: he was a Jew in Poland during World War 2. He was thrown into the Auschwitz concentration camp, where he learned to view life as a war to survive. When he escaped from the camp, he moved to Canada and started an aggressive, ruthless company called **Commodore**, whose motto to survive was, "Business is war!"

At first, Commodore just repaired typewriters; but it grew fast and started to manufacture pocket calculators. In those calculators, the CPU was a microprocessor chip manufactured by **MOS Technology**, a company with a troubled past:

Back in 1974, the most popular microprocessors were the Intel 8080 and the Motorola 6800. But one of the 6800's inventors, a guy named Chuck Peddle, quit Motorola in 1975 and started a new company with his friends. That start-up company, **MOS Technology**, began manufacturing the 6501 microprocessor, which resembled Motorola's 6800.

When Motorola threatened to sue, MOS Technology stopped making the 6501 and switched to the 6502, which Chuck Peddle designed differently enough to avoid a suit. That 6502 chip became very popular and was used in many devices, including Commodore's calculators. **Commodore was one of MOS Technology's biggest customers.**

Though the 6502 was legal, **Motorola sued MOS Technology** for its illegal predecessor, the 6501. The suit dragged through the courts for two years and cost MOS Technology many thousands of dollars in lawyers' fees. Finally, in 1977, Motorola won $200,000. The lawyer fees and $200,000 put MOS Technology in financial trouble.

MOS Technology wanted to be bought by some company having lots of cash. Commodore, rich by then, bought it.

Just before that sale, Canada's tax laws changed, so Commodore moved its headquarters (in theory) from Canada to the Bahamas. That's how MOS Technology became part of "Commodore Limited", a Bahamas company, and how Commodore found itself running a company that made chips. Commodore had entered the computer business.

Dealing with competitors At MOS Technology, Chuck Peddle had sold a 6502 chip for $25 to Steve Wozniak, who used that chip to create the Apple computer. When Commodore saw Apple computers become popular, Commodore offered to buy the Apple Computer Company — and almost succeeded.

Apple wanted $15,000 more than Commodore offered, so the deal never came off. If Commodore were to have offered just $15,000 more, Apple would be part of Commodore now!

Commodore hired Chuck Peddle to design a "Commodore computer", which Commodore hoped to sell through Radio Shack's stores, but Radio Shack had already started designing its own computer.

Pet Rebuffed by Apple and Radio Shack, Jack Tramiel decided to retaliate by building a computer better and cheaper than anything Apple and Radio Shack had. Commodore called its new computer the **Pet** — because Commodore's marketing director was the guy who invented the Pet Rock, and reckoned that if folks were stupid enough to buy a Pet Rock they'd love a Pet computer! He was right: sales of Commodore's Pet Computer skyrocketed.

Commodore told the press that "Pet" was an abbreviation for "Personal Electronic Transactor"; but Commodore had invented the name "Pet" first and later made up what it stood for.

Commodore announced the Pet in 1977 and said its $495 price would include *everything* (the CPU, RAM, ROM, keyboard, monitor, and tape recorder), its ROM would include a good version of Basic, and its screen would display capital letters, lower-case letters, punctuation, math symbols, and graphics symbols.

Commodore's competitors got scared — because Commodore's price was much lower than other computers, Commodore's computer offered more features, and Commodore was rich enough to spend more on ads & marketing than all other manufacturers combined. Computer magazines called the Pet "the birth of a new generation" in personal computers and treated the Pet's designer (Chuck Peddle) to many interviews.

But Commodore disappointed its customers:

Commodore raised the Pet's price from $495 to $595 before taking orders. To order the Pet, the customer had to send $595, plus shipping charges, then wait for Commodore to deliver. Many folks mailed Commodore money and waited long, but Commodore didn't ship. Folks got impatient. Computer stores that had advertised the Pet got worried: customers who'd prepaid complained to the stores, but the stores couldn't get Commodore to ship.

Meanwhile, Radio Shack entered the market with its TRS-80 model 1 priced at $599 — about the same price as Commodore's Pet. **Radio Shack was kinder than Commodore:**

Radio Shack asked customers for just a 10% deposit. Commodore required payment in full.

Radio Shack didn't charge for shipping. Commodore did.

Radio Shack set up repair centers throughout the U.S. Commodore's only repair center was in California.

Radio Shack delivered computers fast. Commodore still wasn't delivering! Finally, Commodore admitted that the $595 Pet would *not* be delivered soon; instead, Commodore would deliver a $795 version that included 4K of extra RAM. So if you already sent $595 to Commodore and wanted a computer soon, you'd have to send an extra $200. That was a rip-off, since 4K of extra RAM was *not* worth an extra $200; but desperate customers sent the $200 anyway.

Radio Shack shipped its computers on a first-come first-served basis; if you ordered a Radio Shack computer, Radio Shack gave you an accurate estimate of when you'd receive it. Commodore gave preferential treatment to its "friends"; if you ordered a computer from Commodore, you hadn't the faintest idea of when it would arrive, since you didn't know how many "friends" were on Commodore's list.

Radio Shack's computer came with a 232-page manual that was cheery and easy. Commodore's computer came with just 10 loose pages that were incomplete and hard to understand.

Commodore announced a low-cost printer but then reneged and decided to sell just an expensive printer. Commodore announced a low-cost disk drive but then reneged and decided to sell just an expensive unit containing 2 disk drives. Commodore became known as a liar.

At first, the Pet was the world's best-selling computer; but all those disappointments made its popularity drop to #3, below Radio Shack (#1) and Apple (#2).

Commodore developed a souped-up Pet, called the **Commodore Business Machine (CBM)**, but it wasn't enough to raise Commodore above the number 3 spot. As Commodore's fortunes dipped, Chuck Peddle and his friends quit. Apple hired them but treated them as second-class citizens, so they returned to Commodore.

Commodore sold several Pet versions, each containing a different quantity of RAM.

If you bought a cheap version and wanted to increase its RAM, Commodore refused to install extra RAM. Instead, Commodore insisted you buy a whole new Pet.

Customers tried buying extra RAM from chip dealers and installing the chips themselves; but to stop those tinkerers, Commodore began cutting a hole in the PC board where the extra RAM chips would go. Commodore was an asshole.

Commodore changed the Pet's tape-handling system.

Tapes created for old Pets wouldn't work on new Pets. Commodore didn't tell customers of the change. Customers who wrote programs for old Pets and then bought more Pets discovered that their programs didn't work on the new Pets. They thought their new Pets were broken. Companies who'd been selling tapes of Pet computer programs began getting angry letters from customers who bought the tapes and couldn't make them work on their new Pets: the customers thought the companies were crooks; the companies thought the customers were lying; eventually folks realized the real culprit was Commodore, who'd changed the Pet secretly.

When the companies discovered that Commodore had changed the Pet without providing a label to distinguish new Pets from old, the companies realized they'd have to give each customer two copies of each program, so the customer could try both versions. That's when many companies gave up trying to sell Pet tapes. They sold tapes for Apple and Radio Shack computers instead. Commodore programs became rare.

Vic Jack's experience at Auschwitz made him scared of Nazis and the Japanese. He feared the US would be invaded by cheap Japanese computers putting Commodore and other American companies out of business.

Paranoid, in April 1980 he called his engineers together and screamed at them, "The Japanese are coming! The Japanese are coming! So we'll become the Japanese!" He laid out his bold plan: Commodore would build the world's first under-$300 computer to display colors on an ordinary TV and produce three-part harmony through the TV's speaker.

At that time, the only under-$300 computer was Sinclair's ZX-80, which was black-and-white and crummy. Commodore's engineers said it was impossible to build a color computer cheaply, but Jack insisted. Commodore's engineers finally managed to do it. Here's how:

MOS Technology, owned by Commodore, had already invented the amazing **Video Interface Chip (Vic)**, which could handle the entire process of sending computer output to the TV screen. Since that chip was cheap, Commodore used it in the under-$300 computer. Unfortunately, it put just 22 characters per line on the screen, so the under-$300 computer would display just 22 characters per line.

Since the new computer was feminine and foxy, Commodore wanted to call it the "Vixen"; but Commodore discovered that a "Vixen" computer couldn't sell in Germany, since "Vixen" sounds like the German word "Wichsen", which means "jerk off". Commodore hastily changed the name to "Vic" and ran TV ads for the "Vic" computer; but that got Commodore into even worse trouble, since "Vic" sounds like the German word "Ficke", which means "fuck". Commodore kept calling it the "Vic" in the USA but called it the "VC" computer in Germany and pretended "VC" stood for "Volks Computer".

Commodore began shipping the Vic in 1981 at $299.95. Later, the price gradually dropped to $55.

To sell the Vic, Commodore tried 3 kinds of ads:

The first ad featured TV star William Shatner (who played Captain Kirk in Star Trek) and said the Vic was wonderful, amazing, out of this world, fun! But then people started thinking of the Vic as just a sci-fi toy. To combat the "toy" image, Commodore changed to a second kind of ad, which said the Vic was as cheap as a video-game machine but more educational for kids. When Texas Instruments began making similar claims, Commodore changed to a third kind of ad, which said Commodore's disk drives, printers, and phone hookups cost much less than Texas Instruments'.

The Vic's low price, fun colors, and effective ads made it popular in the USA, England, Germany, and Japan. Commodore quickly sold over a million Vics! **The Vic became the world's best-selling computer!**

Commodore 64 In 1982, Commodore began selling an improved Vic, called the **Commodore 64** because it included 64K of RAM. (The original Vic had just 5K.) The Commodore 64 also improved on the Vic by displaying 40 characters per line (instead of just 22) and including 20K of ROM (instead of just 16K).

The Commodore 64's price went through 4 phases:

In phase 1, the recommended list price was $599.95, which Commodore tried to force all dealers to charge. If a dealer advertised a discount, Commodore refused to send that dealer any more computers. (Commodore's policy was an example of **price fixing**, which is illegal.)

In phase 2, Commodore allowed discounts. Dealers charged just $350, and Commodore mailed a $100 rebate to anybody trading in another computer or a video-game machine. Bargain-hunters bought the cheap Timex Sinclair 1000 computer just to trade in for a Commodore 64. A New York dealer, "Crazy Eddy", sold junky video-game machines for $10 just so his customers could mail them to Commodore for the $100 rebate. Commodore donated most of the trade-ins to charities for a tax write-off but kept some Timex Sinclair 1000's for use as doorstops.

In phase 3, Commodore stopped the rebate but offered a lower price: discount dealers charged just $148.

In phase 4, the Commodore made an improved version, the **Commodore 64C**, sold by discounters for just $119. It came with a copy of the **Geos** operating system (which made it resemble a Mac), and its keyboard contained extra keys.

The Commodore 64 cost much less than an Apple 2c or IBM PC. Here's why:

Commodore's disk drive (Model 1541) was slow and unreliable and put few bytes on the disk (just single-sided single-density).

Commodore's color monitor (Model 1702) produced a blurry image, which restricted it to 40 characters per line instead of 80, and made the M look too much like an N, the B look too much like an 8.

Commodore's Basic was weak: it didn't even include a command to let you draw a diagonal line across the screen.

Commodore's printer port was non-standard: it worked just with printers built by Commodore, unless you bought a special adapter.

Eventually, Commodore developed an improved monitor (Model 1802) and improved disk drives (Models 1541C and 1541-2).

Because the Commodore 64 was cheap, Commodore sold over a million of them.

Many programmers who wrote programs for Apple computers rewrote their programs to also work on the Commodore 64. Soon the Commodore 64 ran nearly as many popular programs as the Apple 2c.

The Commodore 64's price, even after adding the price of a disk drive and a monitor, still totaled less than the price of an Apple 2e, Apple 2c, IBM PC, or IBM PC Junior. The Commodore 64 was a fantastically good value! It also contained a fancy music synthesizer chip that produced a wide variety of musical tone qualities: when it played music, it sounded much better than an Apple 2e or 2c or IBM.

Jack jumps ship After the Commodore 64 became successful, Jack Tramiel wanted to hire his sons to help run Commodore; but Commodore's other major shareholders refused to deal with Jack's sons, so Jack quit. He sold his 2 million shares of Commodore stock, at $40 per share, netting himself 80 million dollars in cash.

New computers After Jack quit, Commodore tried selling 2 new computers (the **Commodore 16** and **Commodore Plus 4**), but they had serious flaws. Then Commodore invented 2 great computers: the **Commodore 128** and **Amiga**.

The **Commodore 128** ran all the Commodore 64 software and also included a better version of Basic, better keyboard, and better video. To go with it, Commodore invented a better RGB monitor (Model 1902) and better disk drive (Model 1571). Later, Commodore invented the **Commodore 128D** computer, which included a built-in disk drive.

The **Amiga** was even newer and fancier. It contained 3 special chips that produce fast animated graphics in beautiful shades of color. Like the Mac, it used a mouse and pull-down menus. It was bought mainly by video professionals and by others interested in animated graphics. On TV, weathermen used the Amiga to show the weather moving across the weather map.

The Amiga was not compatible with the Commodore 64 or Mac. Aside from graphics, not enough good software was available for the Amiga.

Bankruptcy In 1994, Commodore filed for bankruptcy. Commodore was bought by **Escom**, which sold Amiga Technologies to **Visual Information Services Corp. (Viscorp)**, which sold it to **Gateway**, which eventually abandoned the technology.

Tandy

Tandy, which owns Radio Shack, has survived many years.

Thanks to Tandy Radio Shack helped the computer industry in many ways:

Radio Shack was **the first big chain of stores to sell computers nationally**. It was the first chain to reach rural areas.

Radio Shack invented **the first low-cost assembled computer** (the TRS-80 model 1, which cost just $599, including the monitor).

Radio Shack was **the first company to keep computer prices low without skimping on quality**.

Radio Shack sold **the first notebook computer** (the Tandy 100, invented by Tandy with help from Microsoft and a Japanese manufacturer, Kyocera).

Radio Shack sold **the first pocket computers**. They were manufactured for Tandy by Sharp and Casio.

Radio Shack invented **the first cheap computer having fancy graphics commands**. That was the Color Computer, whose Basic was designed by Microsoft as a "rough draft" for the fancier Basic in the IBM PC.

But when the IBM PC came out and became the standard, Americans suddenly decided to buy just the IBM PC and clones. Tandy tried building IBM clones innovatively, but in 1993 gave up: it stopped making computers and sold all its factories to another computer company, **AST**. Afterwards, Tandy sold computers built by AST, then switched to selling computers built by IBM. Now Tandy sells computers built by Compaq instead.

Nicknames Tandy's computers are often called "TRS" computers. The "TRS" stands for "Tandy's Radio Shack". Cynics add the letters A and H, and call them "TRASH" computers, so Tandy's customers are called "trash collectors".

How Tandy began The Tandy Leather Company was begun by Charles Tandy. Later, he acquired Radio Shack, which had been a Boston-based chain of discount electronics stores.

Under leadership from his Fort Worth headquarters, Tandy/Radio Shack succeeded and grew 30% per year, fueled by the CB radio craze. When the market for CB radios declined, he began looking for a new product to continue his 30% growth.

Don French, a Radio Shack manager whose hobby was building computers, told Radio Shack's leaders that Radio Shack should start selling computers.

The original TRS-80 computer Radio Shack hired Steve Leininger to design a Radio Shack computer and **keep the cost as low as possible:**

Steve wanted his computer to handle lower-case letters instead of just capitals; but since the lower-case chip would have added 10¢ to the cost, management rejected lower case: **Radio Shack's computer handled just capitals.**

The monitor was a modified black-and-white TV built for Radio Shack by RCA. When RCA told Radio Shack that the TV case's standard color was "Mercedes silver" and any other color would cost extra, Radio Shack accepted Mercedes silver and painted the rest of the computer to match the TV. When you use a Radio Shack computer, you're supposed to feel as if you're driving a Mercedes; but since Mercedes silver looked like gray, Radio Shack became nicknamed "the great gray monster". Californians preferred Apples, whose beige matched their living-room decors. (Later, in 1982, Radio Shack wised up and switched from "Mercedes silver" to white.)

Radio Shack's original computer listed for just $599 and consisted of 4 devices: a keyboard (in which hid the CPU, ROM, & RAM), a monitor (built for Radio Shack by RCA), a cheap Radio Shack tape recorder, and an AC/DC transformer. Wires ran between those devices, so that the whole system looked like an octopus. Radio Shack wanted to put the AC/DC transformer *inside* the keyboard, to make the computer system consist of three boxes instead of four; but that *internal* transformer would have delayed approval from Underwriters Laboratories for 6 months, and Radio Shack couldn't wait that long.

Radio Shack's first production run was for just 3000 computers, because Radio Shack's leaders doubted anybody would actually buy them. If none were sold, Radio Shack figured it could use the computers to do internal paperwork instead in its 3500 stores. To Radio Shack's surprise, 250,000 people put themselves on a waiting list to buy the computer during the first year.

Radio Shack named its computer the **TRS-80** because it was by Tandy's Radio Shack and contained a Z-80 CPU chip. Radio Shack's vice-president, John Roach, doubted anybody would buy the computers, so he built just 3500 of them, since Radio Shack had 3500 stores. He figured that if the computers didn't sell, the stores could use them for internal accounting instead.

To announce the computer, Radio Shack held a press conference in August 1977 in New York. But during the conference, a guy ran up and yelled that a bomb exploded two blocks away. Reporters ran to the bomb site, and Radio Shack couldn't get as much publicity as it wanted.

Radio Shack needed a new place to announce the computer. Radio Shack heard that the Boston Computer Society was run a computer show that week, so Radio Shack's management drove to that Boston show, got a booth, re-announced its computer there, and was shocked to discover that the whole show and Boston Computer Society were run by Jonathan Rotenberg, a 14-year-old kid!

That intro was successful: people liked and bought Radio Shack's new computer. The base price was $599.95. For a complete business system (including a souped-up base plus two disk drives and a printer), Radio Shack charged $2600, while Radio Shack's competitors charged over $4500.

Though the first production run was for just 3500 of the computers, 250,000 people put themselves on a waiting list to buy them the first year.

Problems with DOS Radio Shack hired Randy Cook to write the DOS.

My friend Dick Miller tried DOS version 1.0 and noticed it didn't work; it didn't even boot! He told Radio Shack, which told Randy Cook, who fixed the problem and wrote version 1.1. Dick noticed it worked better but still had a big flaw: it didn't tell you how much disk space was left, and when the disk got full it would self-destruct! Then came version 1.2, which worked better but not perfectly.

Since Radio Shack's DOS was still buggy, the inventors of Visicalc (the world's first spreadsheet program) put Visicalc onto the Apple instead of the TRS-80. Apple became known as the "spreadsheet machine", and many accountants began buying Apples instead of TRS-80's.

Dealing with the public In 1977, when Radio Shack began selling the TRS-80, customers didn't understand what computers were.

At a Radio Shack show, I saw a police chief buy a TRS-80. While carrying it out of the room, he called back over his shoulder, "By the way, how do you program it?" He expected a one-sentence answer.

Radio Shack gave customers an 800 number to call for free tech support. Many customers called because they were confused. For example, many customers had this gripe: "I put my mouth next to the tape recorder and yelled TWO PLUS TWO, but it didn't say FOUR!"

Radio Shack's first version of Basic gave just 3 error messages: WHAT (which means "What the heck are you talking about?"), HOW (which means "I don't know how to handle a number that big") and SORRY (which means "Sorry I can't do that — you didn't buy enough RAM yet"). Those error messages confused beginners. For example, here's a conversation between a Radio Shack customer and a Radio Shack technician (Chris Daly)….

Chris: "What's your problem?"
Customer: "I plugged in the video, then the tape recorder, then…"
Chris: "Yes, sir, but what's the problem?"
Customer: "It doesn't work."
Chris: "How do you *know* it doesn't work?"
Customer: "It says READY."
Chris: "What's wrong with that? It's *supposed* to say READY."
Customer: "It isn't ready."
Chris: "How do you *know* it isn't ready?"
Customer: "I asked it 'Where's my wife Martha?', and it just said WHAT."

Other Z-80 computers After the TRS-80, Tandy invented improved versions: the TRS-80 Models 2, 3, 4, 4D, 4P, 12, 16, & 16B, and the Tandy 6000. Like the Model 1, they included a Z-80 CPU and a monochrome monitor.

Coco To compete against the Commodore 64, Tandy invented the **Color Computer**, nicknamed the **Coco**. Like the Commodore 64, the Coco could attach to either a monitor or an ordinary TV, and it could store programs on either a disk or an ordinary cassette tape (the same kind of tape that plays music).

Tandy began selling the Coco in 1980 — the year before IBM began selling the PC.

Microsoft invented the Coco's Basic ROM and also invented the IBM PC's. The Coco's Basic ROM was Microsoft's rough draft of the ROM that went into the IBM PC, so the Coco acted as "an IBM PC that wasn't quite right yet". In the Coco's Basic, the commands for handling graphics & music were similar to the IBM PC's but more awkward. Folks who couldn't afford an IBM PC but wanted to learn how to program it bought the Coco.

Pocket computers Tandy sold 8 different pocket computers, numbered **PC-1** through **PC-8**. They fit in your pocket, ran on batteries, and included LCD screens.

Notebook computers In 1983, Tandy, Epson, and NEC all tried to sell cheap notebook computers. Just Tandy's became popular, because it was the cheapest ($499) and the easiest to learn how to use. It was called the **Model 100**.

Later Tandy sold an improved version, the **Model 102**.

It included more RAM (32K), weighed less (just 3 pounds), and listed for $599. It including a nice keyboard, a screen displaying eight 40-character lines, a 32K ROM (containing Basic, a word-processing program, some filing programs, and a telecommunications program), and a 300-baud modem (for attaching to a phone, after you bought a $19.95 cable). It was 8½ inches by 12 inches and just 1½ inches thick. Reporters used it to take notes and phone them to the newspaper.

Popularity Tandy's 7000 Radio Shack stores penetrated every major city and also remote rural areas, where few other computer stores competed.

Tandy offered "solid value". Tandy kept its quality high and its prices below IBM's and Apple's (though not as low as generic clones). Tandy's computers and prices were aimed at middle-class American consumers, not business executives (who bought from IBM) or bargain-hunting hobbyists (who bought from mail-order discounters).

Tandy's computers were built reliably. Tandy's assembly line checked them thoroughly before shipping to Tandy's stores. If a Tandy computer needed repair during the warranty period, the customer could bring it to any Radio Shack store for a free fix, even if purchased from a different store. After the warranty expired, Radio Shack was kind and charged very little for labor.

Worse attitude During the 1970's, Tandy's headquarters gave toll-free tech help. During the 1980's, Tandy switched to numbers that weren't toll-free. Later, Tandy refused to answer any questions unless the customer bought a support contract. Tandy's claim to offer better support than mail-order companies became Texas bull.

During the 1980's, Tandy established a dress code for its computer centers: employees who met the public had to wear blue or gray suits, blue or white shirts, no beards, and no moustaches. Tandy fired a center manager for refusing to shave his beard. Wasn't the personal-computing revolution supposed to give us tools to express our *individuality*?

Eventually, Tandy shut down all its computer centers.

Atari

Of all the major computer manufacturers, Atari was the most creative — and strangest! Atari was in America's strangest state (California) and had the strangest name: "Atari" is a Japanese war cry that means "beware!"

Video games In 1972, Atari invented the world's first popular video game, **Pong**. Next, Atari invented the game **Asteroids** then dozens of other games.

Atari's games were placed in arcades & bars and required you to insert quarters. In 1975, Atari invented a machine that could play Pong on your home TV. In 1976, Atari gave up its independence and was bought by Warner Communications (the conglomerate that owned Warner Brothers movies & cartoons, Warner Cable TV, and DC Comics).

In 1977, Atari invented a machine called the **Video Computer System (VCS)**, which could play *many* games on your home TV: each game came as a ROM cartridge. Later, Nintendo, Sega, and Sony invented machines that were similar but fancier.

Early personal computers In 1979, Atari began selling complete personal computers. Atari's first two computers were the **Atari 400** (cheap!) and the **Atari 800** (which had a nicer keyboard). They were far ahead of their time. Of all the microcomputers being sold, Atari's had the best graphics, best music, and best way of editing programs. Compared to Atari, the Apples looked pitiful! Yet Atari charged *less* than Apple!

But **Atari made two mistakes:**

Atari didn't hire Bill Gates to write its version of Basic. Instead, it hired the same jerk who invented Apple's DOS. Like Apple's DOS, Atari's Basic looked simple but couldn't handle serious business problems.

Atari believed personal computers would be used mainly for games. Atari didn't realize that personal computers would be used mainly for work. Atari developed spectacular games but not enough software to handle word processing, accounting, and filing.

Atari developed some slightly improved computers (the **600 XL**, **800 XL**, and **1200 XL**) but still lost lots of money.

Jack attack Atari got bought by Jack Tramiel, who'd headed Commodore. Here's why:

When Jack quit being the head of Commodore, he sold his Commodore stock for 80 million dollars. He spent some of that cash to take his wife on a trip around the world.

When they reached Japan, the heads of Japanese computer companies said, "Jack, we're glad you quit Commodore, because now we can enter the American computer market without having to fight you."

That comment scared Jack. To stop the Japanese from invading the U.S. computer market, he started a new computer company, Tramiel Associates, which bought Atari from Warner. Since Jack was rich and Atari was nearly worthless (having accumulated lots of debt), Jack managed to buy all of Atari at 4PM one afternoon by using his Visa card.

Jack and his sons ran Atari. Jack replaced Atari's old computers by two new computers (the **65 XE** and the **130 XE**), which ran the same software as Atari's old computers but cost less.

In 1985, Jack began selling the **Atari 520ST**, which imitated Apple's Mac computer cheaply and nicknamed the "Jackintosh".

It used the **Gem operating system** (invented by **Digital Research** for the Atari and the IBM PC), which made the 520ST computer look like a Mac but did *not* run Mac software: you had to buy software specially modified to work on the 520 ST.

When the 520 ST first came out, its price was about half as much as the Mac and Amiga so that, by comparison, the Mac and Amiga looked overpriced. To fight back, Apple lowered the Mac's price, and Commodore lowered the Amiga's; but Atari's 520 ST remained the cheapest of the bunch.

When Apple announced the Mac Plus, which contained a whole megabyte of RAM, Atari retaliated with the **1040 ST** (which contained a megabyte also), then a 2-megabyte version (the **Mega-2**) and 4-megabyte version (the **Mega-4**).

Atari's had difficulty competing in the U.S., but Atari computers were popular in Europe. Eventually, Atari's fortunes declined. In 1996, Atari died: it got merged into another company, **JTS**, which made disk drives.

Cycles

Every 8 years, the country's mood about computers has changed. After 8 years of dramatic revolution, we switched to 8 years of subtle evolution, then back again.

Pivotal years

The pivotal years were 1943 (beginning the first revolution), 1951 (beginning the first period of *ev*olution), 1959 (revolution), 1967 (evolution), 1975 (revolution), 1983 (evolution), 1991 (revolution), 1999 (evolution), 2007 (revolution), and 2015 (evolution). Here are the details.…

Revolution From 1943 to 1950, researchers at universities were building the first true computers, which were big monsters. Each was custom-built; no two were alike.

Evolution In 1951, Sperry began selling the first mass-produced computer: the **Univac I**. Sperry built 46 of them. During the 8-year era from 1951 to 1958, computers gradually became smaller and cheaper and acquired more software. That evolutionary era was called the **first generation**.

Revolution The next computer revolution began in 1959, when IBM began selling the **IBM 1401**, the first IBM computer to use transistors instead of vacuum tubes.

During that 8-year revolution from 1959 to 1966, computerists polished Fortran and Algol (which had been begun earlier), invented 9 other major computer languages (Cobol, Basic, PL/I, Lisp, Snobol, APL, Dynamo, GPSS, and RPG), and began developing Forth and SPSS. They created many amazing programs for artificial intelligence, such as Weizenbaum's Eliza program, which made the computer imitate a therapist. During that same eight-year period, IBM invented the **IBM 360**: it was the first popular computer that used integrated circuits, and all of IBM's modern mainframes are based on it.

Evolution The years from 1967 to 1974 showed a gradual evolution. Computer prices continued to drop and quality continued to improve. DEC began selling PDP-10 and PDP-11 computers, which became the favorite computers among researchers in universities.

Revolution In 1975, MITS shipped the first popular microcomputer, the **Altair**, which launched the personal computer revolution. Soon Apple, Commodore, Tandy, and IBM began selling microcomputers also. Programmers developed lots of useful, fun software for them. The revolution climaxed at the end of 1982, when many Americans bought microcomputers as Christmas presents.

Evolution In January 1983, the cover of *Time* magazine declared that the 1982 "man of the year" was the personal computer. But consumers quickly tired of the personal-computer fad, chucked their Commodore Vic and Timex Sinclair computers into the closet, and shifted attention to less intellectual pursuits. Many computer companies went bankrupt. In 1983, Lotus announced **1-2-3** (a spreadsheet program), but that was the computer industry's last major successful new product. After that, prices continued to fall and quality gradually increased, but no dramatic breakthroughs occurred. The computer industry became boring. During that time, if you were to ask "What fantastically great happened in the computer industry during the past year?" the answer was: "Not much".

Revolution In 1991, the computer industry became exciting again. Here's why.…

Part of that excitement came from revolutionary influences of the previous two years: in 1989 & 1990 the Berlin Wall fell, the Cold War ended, a new decade began, Microsoft finally invented a version of Windows that worked well (version 3.0), and Apple invented a color Mac that was affordable (the LC). In 1991, Microsoft put the finishing touches on Windows (version 3.1) and DOS (version 5).

In 1991 and 1992, price wars made the cost of computers drop 45% per year instead of the customary 30%. Those lower prices made people spend *more* money on computers, because the ridiculously low prices for fancy stuff encouraged people to buy fancier computers: 486 instead of 286, Super VGA instead of plain VGA, 8M RAM instead of 1M, 200M hard drives instead of 40M.

The sudden popularity of Windows whetted the public's hunger for those muscle machines, since Windows requires lots of muscle to run well. That growing American muscle (bigger and bigger!) then made Windows practical enough to become desirable. All big software companies hastily converted their DOS and Mac software to Windows.

The challenge of doing that conversion forced them to rethink the twin questions of software wisdom: "What makes software easy to use?" and "What kinds of software power do users want?" Many creative solutions were invented to those questions.

During the 1992 Christmas season, fast CD-ROM drives finally became cheap enough to create a mass market: many American bought them, and CD-ROMs became the new standard way to distribute encyclopedias, directories, other major reference works, and software libraries (full of fonts and shareware). The attention given to CD-ROMs made customers think about the importance of sound, and many customers bought sound cards such as the Sound Blaster.

In 1995, Windows 95 was invented, Netscape Navigator 2.0 was invented, and the Internet began to become popular. During the next few years, the Internet's popularity grew wildly.

Evolution In 1999, interest in the Internet peaked, then declined, as Internet companies began running out of clever ideas.

Microsoft stopped coming out with major new products, partly because Microsoft got distracted by lawsuits against it. In the fall of 1999, RAM prices shot up. In November 1999, Packard Bell went out of business. In December 1999, many companies selling on the Internet developed bad reputations by not shipping goods in time for Christmas. Companies prepared for computer problems that the year 2000 might cause.

The year 2000 began boringly, a disappointing way to begin a new millennium. In January 2000, IBM and Acer stopped selling desktop computers through retail stores. In March 2000, the Internet part of the stock market crashed. In June 2000, a judge ruled that Microsoft should be split into two companies.

Revolution In 2007:

Microsoft completely changed the way Microsoft Office looked, by coming out with **Windows Vista** (a major change from Windows XP) and **Office 2007** (which used a ribbon instead of a menu bar). Apple came out with the **iPhone**. Many other innovations arose afterwards.

Evolution In 2015, Microsoft stabilized Windows, by coming out with Windows 10, which was a political compromise between Windows 7 (traditional) and Windows 8.1 (wild).

Presidential politics

The 8-year computer cycle coincides with the American cycle of switching political parties:

After years of Roosevelt & Truman, the presidential election of 1952 ushered in 8 years of a Republican (Eisenhower); 1960 brought 8 years of Democrats (Kennedy & Johnson); 1968, 8 years of Republicans (Nixon & Ford).

1976 began another 16-year experience of "Democrat followed by Republicans"; but the Democrat (Carter) got just 4 of those years (because he lost face in the middle of the Iran hostage crisis, oil crisis, and recession); the Republicans (Reagan and Bush the elder) got the remaining 12.

1992 began another 8-year experience of "Democrat followed by Republican". The Democrat was Clinton (8 years). The Republican was George W. Bush (8 years).

2008 began another experience of "Democrat followed by Republican". The Democrat was Obama (8 years). The Republican is apparently Trump.

When Americans love liberals and revolution, they vote for Democrats; As historian Michael Krigsman remarked, "An excitable mood in the country causes a computer revolution, and the next year the Democrats grab power." When Americans prefer conservative evolution, to go back to the "good old days", they vote for Republicans.

Events

9 events dramatically changed the public's perception of what a computer is.

Powerful computers

In the **1940's**, universities built the first powerful computers, to help World War II Allies calculate ballistics (trajectories of bullets and bombs). Before then, "powerful computers" were just science fiction; suddenly they'd become reality!

Mass-produced computers

The first computer to be mass-produced was the Univac I, in **1951**. Before then, computers were just military research projects; suddenly they'd become practical commercial tools!

46 of the Univac I computers were built, and competitors such as IBM began building computers in much bigger quantities.

Transistors & high-level languages

In **1959**, computer manufacturers began using transistors (instead of vacuum tubes), so that computers became much smaller, cheaper, more reliable, and more powerful. About the same time, the first reasonable computer languages were invented: Fortran, Cobol, and Algol.

For the first time, computers became cheap enough and easy enough to program so that colleges could encourage students to take computer courses.

Chips & Basic

The first computer to contain integrated circuits (chips) was the IBM 360, which IBM began selling in **1966**.

Chips had been invented by other companies earlier, but chips weren't used in complete computer systems until 1966. Afterwards, other computer brands began using chips also. The chips made computers even smaller, cheaper, more reliable, and more powerful.

About the same time, the first easy full-featured computer language was invented: Basic.

For the first time, computers became cheap enough and easy enough so that high schools could encourage students to take computer courses.

Personal computers

In **1975**, Mits began selling the first popular personal computer, the Altair, for $395. Before then, computers were too expensive for individuals to afford.

Unfortunately, the Altair came as a kit that was hard to assemble, and it contained inadequate hardware and software. But soon afterwards, in 1977, came personal computers that were easy to set up and contained reasonable hardware, built by Apple, Commodore, and Radio Shack. For the first time, computers became easy & cheap enough to put in the typical American home.

IBM PC

In **1981**, IBM began selling the IBM PC. It was slightly better than earlier personal computers and set the standard for all future personal computers.

Mouse & graphical interfaces

In **1984**, Apple began selling the Macintosh computer, nicknamed the "Mac." Priced at $2495, it was the first affordable computer to use a mouse. It was a stripped-down version of Apple's Lisa computer and Xerox's Alto computer, which had been invented earlier but were too expensive.

The Mac became immediately popular and led Microsoft to create Windows, which made the IBM PC try to act like a Mac. Versions 1 and 2 of Windows worked terribly, but Windows 3 (which came out in 1990) worked well. Then came further improvements: Windows 3.1, 95, 98, Me, XP, Vista, 7, 8, and 8.1.

Now every desktop personal computer comes with a mouse, and every notebook computer comes with a mouse or an imitation (such as a Touchpad).

CD-ROMs & multimedia

During the Christmas season of **1992**, many folks bought CD-ROM drives. The drives were available before then, but the public had to wait until 1992 for the drives to become cheap and the disks to become plentiful. Now most software comes on CD-ROM disks instead of floppy disks.

CD-ROM disks can hold enough bytes to store music, so now most computers come with nice sound cards and speakers, and entertainment software produces nice music. CD-ROM disks can also hold short video clips; longer video clips are available on souped-up CD-ROM disks called DVD.

Internet

In **1995**, the Internet suddenly became popular, as Netscape 2 came out. (Earlier browsers and e-mail systems were awkward and less powerful.) Also in 1995, Windows 95 came out, which was the first version of Windows that could attach to the Internet well. That year, Americans took crash courses in how to use the Internet. Now most computers connect to the Internet.

Your future

Let's look ahead.…

Become an expert

To become a computer expert, you need a computer, literature, and friends.

A computer to practice on

If possible, buy a computer to practice on. You can buy a decent one for about $300. If you can't afford even $300, get a used computer. Ask your computer friends whether they want to get rid of any "used junky obsolete computers" for under $100, or ask them whether they can lend you a computer for a weekend.

Another way to save money is to join your friends for a group purchase. For example, if 9 of you each chip in $25, you can buy a $225 computer. Divide the 9 of you into 3 trios, and rotate the computer from trio to trio every day, so that you get to use the computer every third day.

Literature to read

Begin by reading *The Secret Guide to Computers*. Then read the manuals that came with your computer.

Find out what's new by subscribing to computer magazines or reading them in your town's library.

You can get computer books and magazines from the bookstore at your local college. You can also try your local branch of the country's biggest bookstore chain: **Barnes & Noble**. If you live near Denver, visit **Tattered Cover** (America's largest independent bookstore, at 303-322-7727). You can find a huge collection of computer books at **Micro Center** (a chain of computer stores).

You can get discounts from mail-order booksellers such as **Amazon.com** and **Walmart.com**.

Since *The Secret Guide to Computers* is an underground book, you won't find it in most stores. To find out whether any stores or consultants near you carry the *Secret Guide*, phone me at 603-666-6644, and I'll look up your ZIP code in my computer.

Friends to chat with

When you have a computer question, phone me at 603-666-6644. Another way to get help is to join a computer club.

The biggest and best computer club was the **Boston Computer Society (BCS)**, which had about 30,000 members, held over 1,000 meetings per year, published many magazines and newsletters, and had hundreds of volunteers who gave free phone help on technical topics. It began in 1977 but shut down in 1996. Its founder and first president was a 13-year-old kid. I hope another kid starts something equally wonderful someday!

If you live near Philadelphia, join a computer club called the **Philadelphia Area Computer Society (PACS)**. Membership costs $10 per year. Details are at PacsNet.org.

The biggest and best computer clubs are in retirement communities in **Arizona** (near Mesa) and **Florida**.

To find computer clubs near *you*, ask employees at your local computer stores, high schools, and colleges. You can also check the list put out by the **Association of PC User Groups (APCUG)** at:

http://ugls.apcug.net/FindUserGroup.aspx

If you take a computer course, get personal help by chatting with your teacher and classmates. To save money, sign up for the cheap courses given by your high school's "adult education" evening program and your local community college.

I've occasionally traveled around the world and given courses inexpensively or for free. Heads of the computer industry got their training from my courses. To get on our mailing list, use the coupon on the back page.

Computer careers

To become a lawyer, you must graduate from law school and pass the Bar Exam. But to become a computer expert, there's no particular program you must graduate from, no particular exam to pass, and no particular piece of paper that "proves" you're an expert or even competent.

You can get a job in the computer industry even if you've never had any training. Your job will be sweeping the floor.

To become a top computer expert, study hard, day & night.

Read lots of computer manuals, textbooks, and magazines. Practice using various computers, operating systems, languages, word-processing programs, spreadsheets, database systems, graphics packages, and Web browsers. Study the human problems of dealing with computers. No matter how much you know, learn *more!*

When I surveyed computer experts, I found that the typical expert spends 2 hours per day **reading** about computers, to fill holes in the expert's background and learn what happened in the computer industry that day! The expert also spends many hours **practicing** what was read and swapping ideas in chats with other computerists.

As a computer expert, you can choose your own hours, but they must be many: if your interest in computers lasts just from 9 AM to 5 PM, you'll never become a computer expert.

Break into the field

To break into the computer field, you can use 6 tools: college, home consulting, home programming, salesmanship, job expansion, and on-the-job training.

College The traditional way to get a computer job is to attend college and get an M.A. or Ph.D. in computer science. Unfortunately, that takes a lot of time.

Home consulting The fastest way to break into the field is to keep your current job but spend weekends and evenings helping neighbors, friends, and colleagues learn about computers.

Help them buy hardware & software. Customize their systems to meet their own personal needs. Teach them in how to use it all. Many folks want training in how to get the most out of Windows, Microsoft Office, other popular software, and the Internet.

At first, do it free. When you've become an experienced expert and developed a list of happy clients who'll vouch for your brilliance, start requesting money from new clients. Start cheaply, at $10 per hour, then gradually raise your rates. Most computer consultants charge about $60 per hour, and some charge much more; but I suggest you be gentler on your clients' pocketbooks! By charging little, you'll get more clients, they'll rack up more hours with you, and you won't need to spend lots of time & money on "advertising". At $20 per hour you'll be very popular!

Home programming At home, you can write computer programs to sell to friends and software publishers, but make sure your programs serve a real need and don't duplicate what's already on the market. Be creative!

Salesmanship For a faster career path, learn enough about computers to get a job selling them in a store.

As a salesperson, you'll help people decide which hardware and software to buy; you'll be acting as a consultant.

The store will probably let you take hardware, software, and literature home with you, so you can study and practice new computer techniques every evening and become brilliant. If you wish, moonlight by helping your customers use the software they bought; design your own customized programs for them.

After working in the store several months, you'll have the knowledge, experience, contacts, and reputation to establish yourself as an independent consultant. You can call your former customers and become their advisor, trainer, and programmer — or even set up your *own* store.

Job expansion Another way to break into the field is to take a non-computer job and gradually enlarge its responsibilities, so it involves computers.

If you're a clerk, ask permission to use spreadsheet and data-management programs to manage your work more efficiently. If you're a math teacher, ask the principal to let you teach a computer course or help manage the school's computer club.

Keep your current job, but expand it to include new skills so you gradually become a computer expert.

On-the-job training The final way to break into the field is to get a job in a computer company, as a janitor or clerk, and gradually move up by using the company's policy of free training for employees.

Phone me Companies phone me when they're want computer experts. If you think you're an expert and can demonstrate your expertise, I'll be glad to pass your name along to employers.

Occasionally, I've even had job openings here at The Secret Guide to Computers. Ask!

Set your rates

If somebody's interested in hiring you to be a programmer or consultant, you must decide what rate to charge.

On your *first* job, be humble and charge very little!

Your first job's main goal should *not* be money. Instead, your goal should be to gain experience, enhance your reputation, and find somebody who'll act as your reference and give you a good recommendation. Convince your first employer you're the best bargain he ever got, so he'll be wildly enthusiastic about you and give you a totally glowing recommendation when you seek your second job.

If you can't find anyone willing to pay you, work for free, so your résumé can say you "helped computerize a company". Then you can get jobs that make you richer.

Though your first computer job might pay little or nothing, it gets your foot in the computer industry's door. After your first job, your salary will rise fast because the most valuable attribute you can have in this field is *experience*.

Since experienced experts are hard to find, they get high salaries; but there's a *surplus* of "kids fresh out of college" who know nothing. Consider your first job a valuable way to gain experience, even if the starting salary is low. When applying for your first job, remember you're still unproven, and be thankful your first employer is willing to take a risk on you.

Asking for a raise After several months on the job, when you've thoroughly proved you're worth more than your pay and your employer is thoroughly thrilled with your performance, gently ask for a slight raise. If declined, keep working at that job but keep your eyes open for a better alternative.

Negotiating a contract Never make a big commitment.

For example, if somebody offers to pay you $10,000 to write a fancy program, don't accept the offer; the commitment's too big. Instead, request $1,000 for writing a stripped-down version of the program. After writing the stripped-down version, wait and see whether you get the $1,000; if you get it without hassles, agree to make the version slightly fancier, for a few thousand dollars more. That way, if you have an argument with your employer, you've lost just $1,000 of effort instead of $10,000.

Contract headaches **Arguments between programmers and employers are common, for 6 reasons:**

1. As a programmer, you'll unfortunately **underestimate the time** to debug the program, because you're too optimistic about your abilities.

2. Your employer **won't be precise** when telling you what kind of program to write. You'll write a program you *think* satisfies the employer's request then discover he wanted something slightly different.

3. Your employer will forget to tell you about **strange cases** the company must handle. They require extra "IF" statements in your program.

4. When the employer sees your program work, he'll think of **extra** things he'd like it to do, which require extra programming effort from you.

5. When the program finally does all the employer expects, he'll want you to **teach** his staff how to use it. If his staff hasn't dealt with computers before, the training could take long. He'll also want you to write a manual about how to use the program.

6. **After the company begins using the program, the employer will want you to make more changes**, for free.

To minimize those 6 conflicts, be honest and kind to your employer. Explain to him you're worried about those 6 conflicts and you'd like to discuss them *now*, before you or he makes commitments. Then make a small commitment for a small payment for a short time; and make sure you and the employer are both happy with the way that small commitment works out before attempting bigger ones.

Life as a programmer

A **programmer** teaches the computer new tricks by feeding the computer a **program** (list of instructions explaining how to do the tricks).

Languages The program's written in the computer's limited vocabulary. For example, this book explained a vocabulary called **Basic**, which consists of words such as PRINT, INPUT, and IF. That vocabulary — Basic — is called a **computer language**. It's a small part of English.

No computer understands the whole English language. The programmer's job is to translate an English sentence (such as "do the payroll") into language the computer understands (such as Basic). So *the programmer's a translator*.

Some computers understand Basic, but other computers were fed a different vocabulary, such as Python, Java, or C#. If you're applying for a programming job, find out which language you're expected to program in.

Of the popular languages, Basic is the easiest and the most fun. To become a programmer, study Basic then learn other languages that are yuckier.

Since Basic's so easy, saying you know Basic is less prestigious than saying you know harder languages such as Java. To get lots of prestige, learn *many* languages. To convince the interviewer you're brilliant, say you know *many* languages well, even if the job you're applying for needs just one language.

The most prestigious languages to know are assembly & machine languages, because they're the hardest. If you can convince the interviewer that you know assembly & machine languages, the interviewer will assume you're smart and offer you a high salary, even if the job doesn't require a knowledge of those languages.

Specific computers Before going to the interview, learn about the specific computer the company uses — and its operating system.

Analysis versus coding Programming consists of 2 stages. In the first stage, analyze the problem to make it more specific.

For example, suppose the problem is, "Program the computer to do the payroll". The first stage is to decide exactly how the company wants the payroll done: weekly, bi-weekly, semi-monthly, or monthly? While computing payroll checks, what other reports do you want the computer to generate? For example, do you want the computer to print a report about the employees' attendance and how much money each department spends on salaries? What kind of paychecks do you want the computer to *refuse* to print? If somebody tries to make the computer print a paycheck for a

ridiculous amount (such as $1,000,000 or ½¢), you want the computer to refuse (and maybe signal an alarm).

That stage — analyzing a vague problem (such as "do the payroll") to make it more specific — is called **analysis**. A person who analyzes is called an **analyst** or, more prestigiously, a **systems analyst**.

After analyzing the vague problem and transforming it into a series of smaller tasks that are more specific, the analyst turns the problem over to a team of **coders**. Each coder takes one of the tasks and translates it into Basic or some other language.

If you're hired to be a "programmer", your first assignment will probably be as a coder. After you gain experience, you'll be promoted to a systems analyst.

The ideal systems analyst knows how to analyze a problem but has prior experience as a coder. A systems analyst who knows how to both code and analyze is called a **programmer/analyst**. An analyst who doesn't know how to code — who merely knows how to break a big problem into a series of little ones — is paid less.

3 kinds of programming Programming falls into 3 categories: **development**, **testing**, and **maintenance**.

Development means inventing a new program.

Testing means making sure the program works.

Maintenance means making little improvements to programs written long ago. The "improvements" consist of eliminating errors discovered recently, or making the program conform to changed government regulations, or adding more features so the program produces more reports or handles special cases.

Development is more exciting than testing, which is more exciting than maintenance. If you're a new programmer, the other programmers will probably "stick you" in the maintenance department, where you'll be part of the maintenance crew. Since your job will consist of "cleaning up" old programs, cruel programmers will call you a "computer janitor".

"Application program" versus "system program" Programs fall into two categories.

The usual kind of program is an **application program (app)**: it handles a specific application (such as "payroll" or "chess" or "send rocket to moon").

The other kind of program is a **system program**, whose only purpose is to help programmers write applications programs.

For example, hidden inside the computer can be a program that makes the computer understand Basic. That program explains to the computer what the words PRINT, INPUT, and IF mean. That program (called the **Basic language processor**) is an example of a system program. Another system program is the **operating system**: it tells the computer how to handle the screen, keyboard, mouse, printer, and disks.

A person who invents system programs is called a **systems programmer**. To become a systems programmer, learn C++, assembly language, and machine language. Creating a system program is hard, so a systems programmer usually gets paid more than an applications programmer.

The word "systems" is prestigious: it's used in the phrase "systems analyst" and in "systems programmer". In some companies, if your boss wants to praise you, the boss will put the word "systems" in front of your title even if your job has nothing to do with "systems".

How to learn programming To be a good programmer, you need experience. You can't become a good programmer by just reading books and hearing lectures; you must **get your hands on a computer and practice**.

If you take a computer course, spend lots of time practicing, at home or on the school's computers. Think of the course as just an excuse to get permission to use the school's computers. The ideal computer center:

has computers that understand many languages
gives you *unlimited* use of the computers (no "extra charges")
is open 24 hours a day
has enough computers so you don't have to wait for somebody else to finish
has a staff of "teaching assistants" who answer your questions
has a rack full of easy-to-read manuals explaining how to use the computers
lets you borrow books and manuals, to take home with you
has *several* kinds of computers, so you get a broad range of experience

Before enrolling in a computer course, find out whether the school's computer center has those features.

Computer courses can be expensive. To pay less, take fewer courses: buy more books and magazines instead, and buy a computer yourself! If you can't afford a fancy computer, get a cheaper one or share the cost with friends; after using it, you can get some of its cost back by selling it.

Another cheap way to get an education is to phone your town's board of education and ask whether the town offers any adult-education courses in computers. Some towns offer adult-education computer courses for under $100.

Community colleges offer cheap courses that are okay. Explore the community colleges before paying institutions that overcharge.

Starting salary For your first programming job, your salary will be "about $30,000", but the exact amount depends on which languages you know, how many programs you wrote before, whether you have a college degree, whether you've had experience on that kind of computer, and whether you know the application area. (For example, if you're a programmer for an insurance company, it's helpful to know something about insurance.)

Degrees A college degree isn't needed, but it can make you look smart! Try to get a degree in **computer science** or **management information systems** or **information technology.**

Computer science emphasizes the underlying theory, systems programming, assembly language, C++, and applications to science.

Management information systems (MIS) emphasizes Basic, databases, and applications to business.

Information technology (IT) is a modern compromise that also emphasizes networking, the Internet, and Java.

A major in "math" that emphasizes computers is also acceptable.

Discrimination If you're a woman or non-white or handicapped, great: the computer industry discriminates less than other occupations. Being a woman or non-white or handicapped works to your *advantage*, since many companies have affirmative-action programs.

But discrimination exists against older people. If you're over 40 and try to get a job as an entry-level programmer, you'll have a tough time since the stereotypical programmer is "young, bright, and a fast thinker". If you're old, they'll assume you're "slow and sluggish".

Because of that discrimination, an oldster should try entering the computer industry through a different door: as a consultant or computer salesperson or computer-center manager or computer teacher. For those positions, your age works to your *advantage*, since those jobs require *wisdom*, and people will assume that since you're old, you're wise.

Shifting careers If you're old, the best way to enter the computer field is to combine computer knowledge with other topics you knew previously.

If you already know a lot about selling merchandise, get a job selling computers. If you already know a lot about teaching, get a job teaching about computers — or helping teachers deal with computers. If you already know a lot about real estate, computerize your real estate office.

Instead of trying to "hop" to a computer career, gradually *shift* your responsibilities so they deal more with computers.

To enter the computer field safely, keep your current job but computerize it.

For example, if you're already a math teacher, keep teaching math but convince your school to let you also teach a computer course or incorporate computers into math classes or help run the computer center. If you already work for a big company and your job bores you, try transferring to a department that puts you in closer contact with computers. After a year in that transitional state, you can break into the computer field more easily since you can put the word "computer" somewhere on your résumé as "job experience".

If you're a college kid, write programs that help professors and others during your vacations.

Agree to write programs for little or no pay. Your goal is *not* money: your goal is to put "experienced programmer" on your résumé.

Interviews When applying for your first computer job, try to avoid the "personnel" office. The bureaucrats in that office will see your résumé includes too little experience and trash it.

Instead, play the who-you-know game. Contact somebody who actually works with computers. Convince that person you're brighter than your résumé indicates. Prove you've learned so much (from reading, courses, and practice) that you can *quickly* conquer any task. If you impress that person enough, you can get the job even though your paper qualifications look too short.

When you get an interview, be assertive.

Ask the interviewer more questions than the interviewer asks you. Ask the interviewer about the company's computer and why the company doesn't have a different one. Ask the interviewer how other employees feel about the computer center. Ask the same kinds of questions a computer manager would ask. That way, the interviewer will assume you have the potential to become a computer manager, so you get hired immediately. You'll also be showing you *care* enough about the company to ask questions. You'll be showing you have a vibrant personality and you're not just "another vegetable who came through the door".

When you apply for a programming job, the interviewer will *not* ask to see samples of your work. He doesn't have time to read your programs. Even if he *did* have time to read your programs, he couldn't be sure you wrote them yourself. Instead, he'll just *chat* with you about your accomplishments. You must "talk smart" by knowing computer-industry buzzwords, even if they don't help you write programs.

Later joys Your first job will pay low, but you'll learn a lot from that experience: it's a free education. After your training period is over, your salary will rise fast — especially if you do extra studying during evenings and weekends. Your *real* job is: to become brilliant!

When you've become brilliant & experienced, other companies will try to hire you. Then leave your current company and work elsewhere to gain new experiences. **Whenever you feel you're "coasting" and not learning anything new, it's time to move to a different job.** The "different job" can be in a new company — or a different department of the same company.

By moving around, you'll gain a wide variety of experiences, so you'll become a qualified, wise consultant.

Social contacts Programming can be frustrating. You'll spend long hours staring at your screen and wondering why your program doesn't work. The job is intellectual, not social. But after you've become an expert coder, you'll get to interact with people more, by doing systems analysis, consulting, teaching, and managing.

Life as a manager

Kids enjoy programming. But as you get older, you'll tire of machines and rather deal with people. As you approach retirement, you'll want to help the younger generation handle the computers you've mastered.

To be a successful manager, you need 3 skills: you must be **technically competent**, **wise**; and **know how to handle people**.

You must know how to program. Know each computer company's strengths & weaknesses and be able to compare their products. Develop a philosophy about what makes a "good" computer center. Understand people's motives and turn them into constructive energy.

Keep up to date. Read the latest books and periodicals about computers. Chat with other computer experts (by phone & e-mail and at conventions & computer clubs).

Here are hints about how to manage a computer center:

Many computer centers put 4-foot-high partitions between their programmers, to give the programmers "privacy". But those partitions are counter-productive: too low to block noise, and too high to permit helpful conversation with your neighbors. Knock the partitions down!

When putting a computer center into a school, develop a *cadre* of hotshot students who are bright, friendly, and outgoing and who'll help other students use the computers. If the hotshots are *not* outgoing — if they become an elitist, snobbish club — the rest of the school will avoid the computers.

If you've hired "support assistants" to help programmers & users, don't let the assistants hide in an office or behind a desk. The assistants should walk up to programmers & users and offer help.

Too often, managers judge their own worth by the size of the computer center's budget: the bigger the budget, the more prestigious the manager. But the best manager does *not* having a big budget; the best manager is the one clever enough to meet the company's needs on a *small* budget.

Too often, the computer center's manager decides who can use the computers. That manager becomes powerful & evil. To avoid concentrating so much power in the hands of one bureaucrat, let each department & person buy computers directly. Let the manager give *advice* about which computers would be most pleasant (compatible and hassle-free).

If you're a computer consultant, be honest: tell your client to buy cheap off-the-shelf programs instead of making the client pay you to write "customized" programs.

Life as a salesman

You can find 3 kinds of salesmen:

The "slick" kind knows how to sell but doesn't know technical details about the computers being sold. He doesn't know how to program and doesn't know much about the computers sold by his competitors. He knows just the "line" that his boss told him to give the customers. That kind of salesman usually resorts to trashy tactics, such as claiming all computers sold by competitors are "just toys".

The opposite kind of salesman is technical: he knows details about many brands but can't give you any *practical* advice about which computer best meets *your* needs.

The best kind of salesman is a consultant. He asks a lot of questions about your particular needs, tells you which of his computers meets your needs best, and even tells you the *limitations* of his computer and why another, more expensive computer sold by a competitor might be better. He's an "honest Joe". He clinches the sale because you trust him and know you won't have unpleasant surprises after the sale. While selling you a computer, he teaches you a lot. He's a true friend.

A woman can sell computers more easily than a man. That's because most computer customers are men, and men are more attracted to women. It's also because, in our society, women are more "trusted" than men. But if you're a woman, say some technical buzzwords to convince the customer you're technically competent and not just a "dumb clerk".

Life as an entrepreneur

Here are ideas that have been tried before, successfully, and *you* can try them too:

start a **rental service**, where people can rent computers
run a **camp** where kids can spend the summer playing with computers
run a **setup service**, where you help businesses create their own Web sites
write easy **manuals** explaining the most popular software

But here are the hardest things about starting your own company:

letting people know you exist
convincing people you're good and worth your price

Change your personality

As you spend time with computers, your personality will change. You'll gradually become a **hacker** (a person skilled at fiddling with the internal workings of computer hardware and software). I hope you become a **helpful hacker** instead of a **cracker** (a hacker who creates mischief by screwing up the internal workings of computer hardware & software, such as by writing a virus or by using password-evasion tricks to secretly spy at private files).

Back in 1993, 100 hackers in an Internet newsgroup got together and wrote a description of a hacker's personality. Here's the description, as edited by Eric Raymond (in his *New Hacker's Dictionary*) and then further edited by me. Not all hackers fit this description — but most do! If you hang around computers a lot, this description will probably start applying to you too! Watch yourself! As America and the world become more computerized, the hacker personality will gradually dominate our planet. If you don't like the "hacker personality", see what you can do to alter it.

Hacker intelligence

The hacker mind is intelligent but strange.

College intelligence Most hackers past their teens have a college degree or are self-taught to a similar level. Before becoming a full-fledged hacker, the typical hacker majored in computer science or electrical engineering or math or physics or linguistics (since studying human languages is a good stepping stone to studying computer languages) or philosophy (since philosophy analyzes the meaning of language and "life forms").

Read a lot Hackers read a lot, and read a wide variety, though with extra emphasis on science facts and science fiction. A hacker's home includes a big library, with many shelves full of books that the hacker has read. A hacker spends more spare time reading books & magazines than watching TV. A hacker spends as much spare time reading as the average non-hacker spends watching TV.

Bad handwriting Hackers have bad handwriting — their script is hard to read — so they usually write in simple capital block letters (LIKE THIS), as if they were junior draftsmen writing on a blueprint. The capital block letters make sense, especially when writing math equations or programming instructions that contain lots of symbols; script would be no faster.

Inhuman communication Since programming requires good organization and precise use of language, hackers are good at composing sentences, paragraphs, and compositions. But though hackers are good writers, they're bad talkers, since they don't get much practice chatting with humans. They're not skilled at arguing with humans, confronting them, and negotiating with them; they're better at communicating with computers, which don't argue.

Good at memorizing Hackers are good at memorizing details, such as computer codes.

Neat just in output Hackers produce programs, writings, and thinking that are very neat and well-organized; but a hacker is too busy to make the hacker's environment equally neat, so a hacker's desk and office floor are typically piled high with a disorganized mess of resources.

Hacker bodies

Here's what a hacker looks like, and where to find one.

Near universities Half of the USA's best hackers live within 100 miles of Boston or San Francisco. That's because, during the 1950's and 1960's, the top researchers in artificial intelligence were at two universities: the **Massachusetts Institute of Technology** (**MIT**, in Cambridge, Massachusetts, near Boston) and **Stanford University** (in Silicon Valley's Palo Alto, near San Francisco). Those researchers spawned protégés, who want to keep living near the master researchers even after graduation, to stay connected to the intellectual community.

Mostly male Most hackers are male, but females are more common in hackerdom than in other technical professions.

Mostly Caucasian In the USA, most hackers are Caucasian. On the West Coast, many hackers are Asian; on the East Coast, many hackers are Jewish.

Relatively unbigoted Hackers are less bigoted than other Americans, since hackers care more about what a person wrote than the person's appearance. Hackers believe computers can act like humans and therefore believe in the humane treatment of all computers and all people.

Casual dresser Hackers dislike "business attire". The typical hacker would quit a job if it required wearing a suit.

Hackers like to wear clothes that are casual, easy to take care of, post-hippie: T-shirts (with slogans on them), jeans, running shoes (or barefoot), and backpacks.

Scruffy appearance Hackers look scruffy. Many hackers have long hair. Men hackers often have beards and moustaches. Women hackers try to look "natural" by wearing little or no makeup.

Since hackers love computers, which are mostly indoors, hackers don't get tans.

Night owls Hackers often stay up all night, to finish work on excitingly frustrating programming challenges. Then they sleep late in the morning.

Extreme food For dinner, hackers prefer spicy ethnic food instead of "American" food. The most popular is spicy Chinese (Szechuan or Hunan style, rather than Cantonese, which is too bland). Alternatives, popular occasionally, are Thai & Mexican food. For a change, hackers like high-quality Jewish-deli food, when available.

For midnight snacks while in the middle of marathon programming sessions, hackers prefer pizza and microwave burritos. Back in the 1970's, hackers used to eat a lot of junk food, but modern hackers are more into "health food".

Hackers tend to be extreme: either too skinny or too fat. More hackers are too skinny than too fat.

Nearly drug-free Hackers need to protect their heads from drugs, so they don't do drugs. They don't smoke. Most hackers don't drink alcohol, though a few hackers experiment with fancy wines and exotic beers.

Since hackers favor experimentation, they tolerate folks who use non-addictive drugs such as pot and LSD. But hackers criticize people who take "downers" and opiates, since those drugs make you act stupid.

To help stay up late at night programming, hackers often take mild "uppers" such as caffeine (in coffee and Jolt cola) and sugar (in soft drinks and junk food).

Experimental sex Hackers are more likely than "normal" folks to experiment sexually. Many hackers openly have multiple boyfriends or girlfriends, or live in communes or group houses, or practice open marriage (where both partners agree that extra-marital relationships are okay), or are gay or lesbian.

Hacker beliefs

Here's how to make a hacker happy.

Toys better than money Hackers don't care about earning lots of money or social approval. Instead, hackers just want the intellectual pleasure of inventing beautiful programs and products — and exploring the beautiful products invented by others.

So to bribe a hacker, don't offer money or a fancy title; instead, offer a lab full of computer hardware and software for the hacker to play with, and permission for the hacker to spend time playing with and inventing fantastic technology.

Non-religious Since hackers don't like to be told what to do, they don't like organized religion. Since hackers are into facts, not beliefs, they tend not to believe in God.

When asked "What religion are you?", many hackers reply by calling themselves "atheist" or "agnostic" or "non-observant Jewish". Some hackers join "parody" religions, such as Discordianism and the Church of the SubGenius. Some hackers have fun participating in "mystical" religions such as Zen Buddhism and neo-paganism.

Libertarian politics Hackers like freedom to explore computers. They don't like restrictions. They don't like being told what to do.

They dislike authoritarians, managers, MBA's, and big government. They tend to be Libertarian. They dislike the dogmatic insistence of the far left and far right. If asked to choose between Democrats and Republicans, they tend to choose Democrats because Democrats permit more social freedoms, so hackers are classified as "left of center".

Cat lovers Hackers are more likely to have cats than dogs, because cats are like hackers: clever rather than belligerent.

No team sports Hackers don't like to watch sports. Hackers don't watch sports on TV and don't go to sports stadiums.

Hackers would rather participate than watch. Though half of all hackers don't make time to participate, the other half *do* participate, but mainly in individual sports rather than team sports. The only team sport they like is volleyball, because it's non-contact and friendly.

They prefer individual sports that involve dexterity, concentration, and stamina, rather than brute force. Their favorite sports are bicycling, hiking, rock climbing, caving, kite-flying, juggling, martial arts, roller skating, ice skating, skiing, target shooting, auto racing, and aviation.

Strange cars Hackers don't wash their cars. Hackers drive extreme cars: either beat-up heaps (unwashed because they're junk) or (if the hackers are rich) luxury sports cars (unwashed anyway).

Brainy hobbies Hackers like to play music, play board games (such as chess and Go), dabble in ham radio, learn about linguistics & foreign languages, and do "theater tech" (give technical support to theater productions).

Hate stupidity Hackers like active intelligent freedom, so they dislike dishonesty, boredom, business suits, stupid incompetent people (especially stupid incompetent managers who wear business suits), stupid music (such as "easy listening music"), and stupid culture (such as TV, except for TV's cleverly cynical cartoons & movies & the old Star Trek).

Teach your kids

Here's how to introduce kids to computers.

Teaching programming

Kids should start writing simple programs in Basic when they're in the third grade. (The brightest kids can start even younger!) Before the third grade, the typical kid should learn how to run other people's programs and maybe learn Logo (a language that's easier than Basic for beginners). More programs have been written in Basic than any other computer language.

Before graduating from high school, every kid should learn Basic — and how to create Web pages by using HTML & JavaScript.

Educational applications

The computer can help teach many topics.

English While trying to write a program, the kid learns the importance of punctuation: the kid learns to distinguish colons, semicolons, commas, periods, parentheses, and brackets. The kid also learns the importance of spelling: if the kid misspells the word PRINT or INPUT, the computer gripes. The kid learns to read technical stuff when wading through computer manuals.

Some kids "hate to write English compositions". The computer can change that attitude!

> A word-processing program makes "writing an English composition" become a fun video game, when the words appear on the screen and you can move them around by using the computer's nifty editing tools, which can even correct spelling (without forcing the kid to thumb through a dictionary) and check grammar and style. It's educational fun!

To make the kid understand why parts of speech (such as "nouns", "verbs", and "adjectives") are important, give the kid a computer program that writes sentences by choosing random nouns, random verbs, and random adjectives. Then tell the kid to invent his *own* nouns, verbs, and adjectives, feed them into the program, and see what kind of sentences the program produces.

Young kids have enjoyed a program called **Story Machine**.

> It gives you a list of nouns, verbs, adjectives, and other parts of speech that you can use to build a story. You type the story using any words on the list. As you type the story, the computer will *automatically illustrate it!* For example, if you type, "The boy eats the apple," your screen will automatically show a picture of a boy eating an apple! If you type *several* sentences, to form a longer story, the computer will automatically illustrate the entire story and produce an animated cartoon of it! The program will also criticize your story's structure. For example, if you say "The boy eats the apple" but the boy isn't near the apple yet, the program will recommend that you insert a sentence such as "The boy runs to the apple" beforehand. The program came on a $25 disk from Softkey and required an Apple 2 computer.

History The computer can make history come "alive" by throwing the student into an historical situation.

For example, a graduate of my teacher-training institute wrote a program that says, "It's 1910. You're **Kaiser Wilhelm**. What are you going to do?" Then it gives you several choices.

> For example, it asks "Would you like to make a treaty with Russia?" If you answer "yes", the computer replies, "Russia breaks the treaty. *Now* what are you going to do?" No matter how you answer the questions, there are only two ways the program can end: either "You've plunged Europe into a World War" or "You've turned Germany into a second-rate country". After running that program several times, you get a feeling for the terrible jam that the Kaiser was in and begin to pity him. Running the program is more dramatic than reading a book on the Kaiser's problems, because the program forces you to step into the Kaiser's shoes and react to his surroundings: you are there. When you finish running the program, you feel you've lived another life — the life of a 1910 Kaiser.

Such a program is called an historical **simulation**, since it makes the computer **simulate** (imitate) an historical event.

Current events The best way to teach current events is through simulation.

To teach the student to analyze the conflict between **Israel & Arabs**, let the student run a program that says "You're Israel's Prime Minister" then run a program that says "You're the Palestinian leader". By running both programs, the student learns to take both sides of the argument and understands the emotions

of both leaders. Such programs could help warring nations understand each other enough to bring peace!

When the nuclear power plant at **3-Mile Island** almost exploded, teachers wrote a program saying "You're in the control room at 3-Mile Island".

Your computer's screen shows a picture of the control room. Your goal: make as much money as possible for the electric company without blowing the place up. You can buy 2 versions of the program: one's called just "3-Mile Island"; the other's called "Scram". To teach kids about 3-Mile Island, it's easier to buy the program than to get permission from parents to "take the kids on a field trip to 3-Mile Island" (which also requires that you sit on a bus while listening to 100 choruses of "100 bottles of beer on the wall" and worrying about kids who get lost at 3-Mile Island).

The best way to teach **economics & politics** is to give the student a program that says "You're running the country" and then asks the student to input an economic and political strategy. At the program's end, the computer tells how many years the student lasted in office, how well the country fared, and how many people want to assassinate him.

The best way to learn anything is "by experience". Computer simulations let the student learn by "simulated experience", which condenses into a few minutes what would otherwise require many *years* of "natural experience".

Biology The computer can do **genetics** calculations: it can compute the probabilities of having various kinds of offspring and predict how the population's characteristics will shift.

The computer can handle **taxonomy**: it can classify different kinds of animals and plants.

The computer asks you a series of questions about an organism and finally tells you the organism's name. A popular game called **Animals** lets the student teach the computer which questions to ask.

To teach **ecology**, a graduate of my teacher-training institute wrote a simulation program that begins by saying, "You're the game warden of New Jersey. What are you going to do?"

It asks how many weeks you want the deer-hunting season to last. If you make the hunting season too *long*, hunters kill all the deer, and deer-loving environmentalists hate you. But if you make the deer-hunting season too *short*, hunters hate you; moreover, the deer overpopulate, can't find enough to eat, then die of starvation, whereupon *everybody* hates you. Your goal is to stay in office as long as possible.

Sex education When Dartmouth College (which for centuries had been all-male and rowdy) suddenly became coed in 1971, its biology department realized the importance of teaching about **birth control**. The professors wrote a program asking your age and which birth control method you wish to use this year.

You have 9 choices, such as pill, diaphragm, IUD, condom, rhythm method, and "Providence". After you type your choice, the computer computes the probability of having children and can say (if you're unlucky) that you had a ***BOY*** or ***GIRL***. The computation is based, as in nature, on a combination of science and chance (random numbers). Then the computer asks your strategy for the next year. The program continues until the computer finally says ***MENOPAUSE***. The program lets you explore how different strategies produce different numbers of children. Experimenting with the program is safer & faster than experimenting on your body, though maybe not as fun.

Fun Let the programs use the same techniques that make video games fun & exciting.

Let the programs include animated graphics and require the student to answer fast. Show a running total of the student's points, so whenever the student answers right the screen shows the score increases immediately.

At the end of the educational game, the computer shouldn't say "excellent" or "fair" or "poor". Instead, it should state the total number of points accumulated and ask whether the student wants to try again, to increase the score.

If the student's score is high, the computer should give praise and store the student's name on the disk. If the student's score is low, no criticism should be given other than asking "Would you like to try again?"

How to pay less for software

If you're a teacher, **tell your hotshot students to write software for you**.

Your students will love the opportunity to work on a project that's useful. Tell them that if their software is good you'll write them glowing recommendations saying they computerized the school.

Many software publishers give **educational discounts**. Some publishers offer **"site licenses"**, where you pay a big fee but then can make as many copies of the software as you wish. The nicest publishers of business software offer **"trial size" versions** (for $10 or even free), which let you practice the software but require you to keep your documents and files brief.

Avoid dangers

How could computers change human society? The many good ways are obvious. Here are the 8 bad ones.

Errors

Although the computer can have a mechanical breakdown, the usual reason for computer errors is *mental* breakdown — on the part of the people who run it. The usual computer blooper is caused by a programmer who writes a wrong program, or a user who inputs a wrong number. If you want the computer to write a check for $10.00 but you forget to type the decimal point, the computer will nonchalantly write a check for $1000.

The biggest computer blooper ever made:

A rocket rose majestically from its launch pad at Cape Kennedy and headed toward Venus. Suddenly it began to wobble. It had to be destroyed after less than 5 minutes of flight. The loss was put at $18,500,000. What went wrong? After much head-scratching, the answer was finally found. In one of the lines of one of the programs, a programmer omitted a hyphen.

In one city's computer center, every inhabitant's vital statistics were put on cards. One lady in the town was 107, but the number 107 wouldn't fit on the card properly, because the space allotted for AGE was just two digits.

The computer just examined the last two digits, which were 07, and assumed she was 7 years old. Since she was 7 and not going to school, the computer printed a truant notice. So city officials visited the home of the 107-year-old lady and demanded to see her mom.

Here's a story from *Time Magazine*:

Rex Reed ordered a bed from a department store. 3 months passed. Then came the long anticipated announcement: the bed will be delivered Friday.

Reed waited all day. No bed. Having disposed of his other bed, he slept on the floor.

The next day, deliverers brought the bed but couldn't put it up. No screws.

On Monday, men appeared with the screws but couldn't put in the mattresses. No slats. "That's not our department." Reed hired a carpenter to build them. The department store's slats finally arrived 15 weeks later.

Undaunted, Reed went to the store to buy sheets. 2 men came up and declared: "You're under arrest." Why? "You're using a stolen credit card. Rex Reed is dead." Great confusion. Reed flashed all his identity cards. The detectives apologized — then tore up his store charge card. Why? "Our computer's been told you're dead. And we can't change that."

At the end of 1999, people were nervous about the **year 2000 problem** (which was also called the **Y2K problem** and the **millennium bug**). Here's what those people said:

"Many people still use old computer programs that store each year as a 2-digit number. For example, the year 1983 is stored as 83. When the year 2000 comes, some of those old programs will still assume the first two digits of the year will be 19, so they'll store the year 2000 as 00 and assume it means 1900. They'll think the clock's been turned back to the year 1900, think bills are being paid at the wrong time, and think machines haven't been repaired at the right time, so they'll shut down all the machines they control, including

cars, elevators (which will plunge), airplanes (which will crash), hospital life-support systems (which will shut down and kill all their patients), utility companies (which will shut off your electricity, water, and phones), and bank machines (which will give customers no more cash)."

Programmers worked to solve that problem. January 1, 2000, came and went without major disasters.

Unemployment

Since the computer's a labor-saving device, it can make laborers unemployed. Clerks and other low-echelon workers can find themselves jobless and penniless.

Computers can create *new* jobs.

Not all computer-related jobs require abstract thinking: there's a need for mechanics, typists, secretaries, salespeople, editors, librarians, etc. There's a need for people to tell programmers what to program. Running a computer center is a business, and there's a need for business executives.

When computers do human work, will there be *enough* work left for us humans to do? Don't worry: when no work is necessary, humans have an amazing talent for inventing it.

That's Madison Avenue's purpose: to create new longings. Instead of significantly shortening the work week, Americans always opt for a work week of nearly equal length but devoted to more luxurious ends. That's the gung-ho Protestant work ethic we're so famous for. Computers change but don't reduce our work.

That's what will happen in the long run. But in the meantime, many folks will be temporarily out of a job.

Quantification

Since the computer handles numbers easily, it encourages people to reduce problems to numbers. That's both good and bad:

It's good because it forces people to be precise. It's bad because some people make quantification a goal in itself, forgetting it's but a tool to other ends. Counting the words that Shakespeare wrote is of no value in itself: it must be put to some use. Cynics say, "The problem with computers is they make meaningless research possible."

Since just quantifiable problems can be computerized, there's a danger that people will think unquantifiable problems aren't worth investigating, or unquantifiable aspects of a problem should be ignored. John Kemeny gives this example:

At an open hearing about designing a new Los Angeles freeway, some voters complained bitterly that the freeway would go right through the midst of a part of the city heavily populated by blacks and destroy the community spirit they'd slowly & painfully built. The voters' arguments were defeated by the simple statement that, according to an excellent computer, the proposed route was the best possible.

Nobody knew enough to ask how the computer had been instructed to evaluate the routes. Was it asked just to consider the costs of building & acquiring property (in which case it would have found routing through a ghetto area highly advantageous), or was it also asked to the human suffering a route would cause?

Maybe voters would have agreed it's not possible to measure human suffering in terms of dollars. But if we don't consider of human suffering, we're equating its cost to zero, which is the worst of all procedures!

Asocial behavior

The computer's a seductive toy that can wreck your social life.

When you walk up to the computer, you expect to spend just a few minutes but wind up spending hours instead. When catching bugs, playing games, or using the Internet, you'll while away lots of time. You may find yourself spending more time with the computer than with people.

Getting along with the computer is easy — perhaps *too* easy. Though it can gripe at you, it can't yell. If you don't like its behavior, you can turn it off. You can't do the same to people. Excessive time spent with the computer can leave you unprepared for the ambiguities and tensions of real life.

The computer replaces warmth by precision. Excessive time spent with it might inhibit your development as a loving individual.

Irresponsibility

Computerization is part of technological bureaucracy. Like all bureaucracy, it encourages the bureaucrat to say, "Don't blame me — I can't change the bureaucracy." But now the words read, **"Don't blame me — the computer did it."**

Computers will run governments and wars. The thought of someone saying, "I can't change that — that's the way the computer does it" is frightening.

Concentrated power

As computers amass more info about people, computers will become centers of knowledge. The people who control them — the programmers, sociologists, generals, and politicians — will gain lots of power. The thought of so much power being concentrated in the hands of a few is frightening. A handful of people, pressing the wrong buttons, could nuclear-bomb the earth.

Nobody should have complete control over a computer center. The power should be diversified. Sensitive data and programs should be protected by passwords and other devices, so no single individual can access all of it.

Crime

The computer's the biggest tool in the kit of the white-collar criminal. He just has to insert a zero, and the computer will send him a paycheck for ten times the correct amount.

To catch computer criminals, computers are programmed to double-check; but if the criminal evades the double-checks, he won't get caught. Police have a hard time finding computer criminals, since fingerprints and other traditional evidence are irrelevant.

A bright programmer can devise tricks to get around the passwords. The crudest is to bug the wires that computers communicate through. A cleverer method is to slip extra lines into innocent programs (or e-mail attachments). The cleverest is to use **social engineering**: convince users (by phone or e-mail) that you're an administrator who must verify all passwords.

Since you must be smart to be a computer criminal, if you're caught you'll be admired. Instead of saying "What a terrible thing you've done!" folks say "Gee, you must be smart. Tell me how you did it." A bright button-down computer criminal who steals $100,000 electronically gets a lighter sentence than the dude who must resort to a gun to get $1000. Is that justice?

Invaded privacy

Of all the harm computers can do, "invaded privacy" worries people the most. George Orwell's book "*1984*" warned that someday "Big Brother will be watching you" via a computer. His prediction's already reality: your whereabouts are constantly checked by computers owned by the FBI, IRS, Homeland Security, military, credit-card companies, and mail-order houses.

My brother once wrote an innocent letter asking for stamps. Instead of using his own name, he used the name of our dog, Rusty. Since then, we've received letters from many organizations, all addressed to "Mr. Rusty". Our dog's name sits in computers all across the country.

What computers have stored about you may be misleading. If you never discover the error, the consequences can haunt you the rest of your life. Examples:

A teacher saw one of the little boys in her class kiss another boy. She entered on his computerized school records, "displays homosexual tendencies".

According to computer records, a certain man had "3 lawsuits against him". In fact, the first was a scare suit 30 years before, over a magazine subscription he never ordered; the second had been withdrawn after a compromise over a disputed fee; the third case had been settled in his favor.

Many laws have been passed to give you privacy rights.

You've a right to see what info is stored about you, and change it if it's wrong. For example, if a teacher or employer writes a "confidential recommendation" about you, you've a right to examine it, to prevent misleading statements from haunting you for life.

Even if the info stored about you is accurate, you've a right to prevent its dissemination to the general public. No organization should store or disseminate info unjustifiably.

What's "justifiable"? Fearing "Big Brother", people don't want politicians to access personal info. On the other hand, fearing criminals, people want the police to have a free hand in sleuthing. How to give info to the police without giving it to politicians can be puzzling.

Outdated info should be obliterated. A person shouldn't be haunted by his distant past; he should be given a chance to turn over a new leaf.

Just facts should be stored, not opinions. It's okay to store that someone lives on Fifth Avenue but not that he lives in a "nice neighborhood".

It's unfortunate that people feel a need for privacy. If the info stored about you is correct, why argue? But many people feel a need to be secretive, and I suppose people have that right. It's called the right to be "let alone".

People don't want to feel their whole lives are on stage, recorded by a computer. It inhibits them from acting free and natural.

Even if the computer doesn't store any damaging info about you, the mere *thought* that all your actions are being recorded is damaging, because it makes you act more conservatively. You may be afraid to adopt a good but unusual lifestyle, because anything "different" about you will look bad on the computerized records used by banks, credit-card companies, insurance companies, and other conservative institutions. The harmful thing is not that Big Brother is watching, but that you *feel* he's watching. You're subjugated.

Share our knowledge

Thanks for reading *The Secret Guide to Computers*. If you have questions about what you've read, phone me at 603-666-6644, day or night.

Editions

You're reading the 33rd edition. I've been revising the *Secret Guide* for over 40 years:

Edition	Published	Pages	Price	Praised	New tutorials it included
edition 0	1972 spring	17	free	HP-2000	Basic
edition 1	1972 fall	12	free	DEC-10	DEC computers
edition 2	1972 fall	20	free	DEC-10	Fortran
edition 3	1972 fall	32	$1	DEC-10	data files
edition 4	1973 Jan.	63	$2	DEC-10	Algol
edition 5	1973 Sept.	73	$2	DEC-10	graphics
edition 6	1974 July	260	$5.20	DEC-10	artificial intelligence, numerical analysis
eds. 7-9	1976-1979	410	$16.25	TRS-80	hardware, micros, Cobol, language survey
edition 10	1980-1982	696	$29.60	TRS-80	discount dealers, video graphics, Pascal
edition 11	1983-1984	750	$28	IBM PC	IBM PC, word processing
edition 12	1986-1987	909	$24	Leading Edge	DOS, WordPerfect, spreadsheets, dBase, C, Logo
edition 13	1988 Oct.	909	$24	Tussey Swan	Q&A
edition 14	1990 June	607	$15	Gateway	Mac, Excel, Quattro
edition 15	1991 Sept.	607	$15	Gateway	Windows, advanced WordPerfect
edition 16	1992 May	607	$15	Micro Express	DOS 5, Quattro Pro
edition 17	1993 April	607	$15	Expotech	Mac System 7, MS Word, repairs
edition 18	1993 Aug.	607	$15	Expotech	DOS 6
edition 19	1994 Aug.	639	$15	Expotech	Pentium, multimedia computers, DOS 6.2
edition 20	1995 March	639	$15	Quantex	MS Word 6, AMI Bios
edition 21	1995 Nov.	639	$15	Quantex	Windows 95, QBasic
edition 22	1996 June	639	$15	Quantex	Internet, advanced Windows 95
edition 23	1997 May	639	$15	Quantex	Visual Basic, viruses, advanced Internet
edition 24	1997 Dec.	639	$15	Quantex	backup-storage devices
edition 25	1998 Dec.	639	$15	ABS NuTrend	Windows 98, iMac, MS Word 97, Works
edition 26	1999 Sept.	639	$16.50	ABS NuTrend	MS Word 2000, create Web pages
edition 27	2000 Oct.	639	$16.50	ABS NuTrend	PowerPoint, Publisher, Access, Java, C++
edition 28	2002 Aug.	639	$17.50	ABS NuTrend	Windows XP, Linux, Palm, HTML
edition 29	2004 July	607	$17.50	eMachines	Mac OS X, JavaScript
edition 30	2007 Sept.	575	$20	HP Compaq	Windows Vista, MS Office 2007, video editing
edition 31	2011 July	703	$25	HP Compaq	Windows 7, MS Office 2010, tricky living, C#
edition 32	2014 Feb.	703	$25	HP & Acer	Windows 8.1, iOS 7, Android, MS Office 2013
edition 33	2017 Jan.	703	$25	HP	Windows 10, iOS 10, MS Office 2016, Python

Editions 4 & 6-13 were each bound as a set of booklets (instead of as a single fat book).
Editions 14-33 contained 2 columns per typical page; earlier editions contained just 1 column per page.
I used a typewriter (editions 0-10), TRS-80 (ed. 11-13), WordPerfect (ed. 14-22), MS Word (ed. 23-33).

To get on the mailing list for a *free* brochure about the 34th edition, mail page 703's coupon (or a postcard with your name, address, and the words "send 34th edition info").

Let's meet

I hope to meet you someday. If you ever visit New Hampshire, drop by and say hi! My workload prevents me from chatting long, but at least we can grin.

I can visit your home town and give you and your friends courses and tutoring. The cost per person can get low if you join your friends. For more info about how I can help you at little or no charge, phone me at 603-666-6644.

How to give a course

After you practice using computers and become a computer expert, why not give your *own* courses? You too can become a guru. Here are suggestions.…

When giving a course, you won't have time to teach every detail, so just tell the students to read the details in *The Secret Guide to Computers* and other manuals. During class, instead of grinding through details, have fun:

Demonstrate hardware & software.
Argue cheerily about computer hassles.
Let the class ask lots of questions.
Provide **hands-on experience aided by tutors**.

To liven up your classes and loosen up your students, say this:

"I'm supposed to turn you all into computer experts by 5:00. I'll try."

"In this course, I'm your slave. Anything you want, you get."

"If you're boring, we'll follow the curriculum. If you ask lots of questions, we'll dig into the good stuff."

"Don't bother taking notes. If God wanted you to be a Xerox machine, He'd have made you look that way. So just relax. If you forget what I say, phone me anytime, and I'll repeat it all back to you."

"There's no attendance requirement. While we discuss a topic that bores you, leave — or better yet, play with the computers in the room, so you become super-smart."

Phone me for free help with curriculum, dramatics, and tricks of the trade. For your first course, charge little, so your students are grateful and you build your reputation.

No matter how great you think you are, your students will tire of you eventually. To keep them awake, add variety by including your friends as part of your act.

Good luck. Try hard. You can cast a spell over the audience. Courses change lives.

At your service, your computer butler,
Russ Walter, cell phone 603-666-6644

Resources

Index

For each topic, this index tells the page number where the discussion *begins*. Look at that page *and the next few pages also*.
If a page number is **bold**, that page is where the topic's **main info** begins.
To find a command in a specific programming language, *look at that language's chapter*, which begins with its own index of commands.

D

E

F

G

O

P

Q

R

S

Copy this coupon for friends

Get more copies

We offer 2 kinds of writing:

The *Secret Guide to Computers* explains computers. *Tricky Living* explains the rest of modern life.

Order more copies for yourself and friends. The books make **great presents** for Christmas, birthdays, graduations, and celebrations. Get the 33rd edition plus **classic editions**, which include extra info about classic computers & lifestyles — and cost less! (Page 9's chart explains how editions differ.) To get discounts and free brochures, use this coupon. Copy it for your friends. If questions, call **603-666-6644** or see **SecretFun.com**, which explains more and lets you read some chapters free!

Money-back guarantee

If not sure whether to order, go ahead: you can return unused books anytime for a 100% refund of what you paid us.

Many ways to order

The simplest way to order is to mail this page's coupon with a check, money order, credit-card info, or cash. Other choices:

You can **visit** us in New Hampshire to pick up the books personally: phone 603-666-6644 for directions and a pickup time.

To order by **credit card**, mail this coupon or phone 603-666-6644, day or night. We take Master Card, Visa, American Express, and Discover. Give your credit-card number, expiration date, name printed on the card, billing address (if different from shipping address), and verification code (3-digit or 4-digit code printed on the card).

We can **bill** you if you bought at least 10 books from us before (or you're employed by or retired from a school, bookstore, government agency, or established computer company). Mail this coupon or phone 603-666-6644 or email Russ@SecretFun.com. If over $700, phone for approval. The bill is due in 30 days.

We accept 5 forms of payment from other countries:

international postal money order (written in U.S. dollars)

credit-card number (Master Card, Visa, American Express, or Discover; to transmit the number, mail this coupon or phone 603-666-6644)

check (written in U.S. dollars and having a U.S. or Canadian city printed somewhere on the check)

cash (we convert foreign currency & send change)

wire transfer (in U.S. dollars, from your local outlet of Western Union or MoneyGram, plus a phone call or note telling us how you transferred)

Free book for review

If you plan to introduce the book to at least 80 people, phone 603-666-6644 to request a **free book** for review.

CUT OUT THIS COUPON

Book bargains

Secret Guide to Computers & Tricky Living, 33rd edition, 703 pages, lists for **$25**. To pay less, join friends:

20% discount if you order at least 2: pay just **$20** each.
40% discount if you order at least 4: pay just **$15** each.
60% discount if you order at least 60: pay just **$10** each.

How many copies do you want?____ They're usually printed as paperbacks.
If you want one of them to be a copyable CD-R disk instead, check this box: ☐

Say how many copies you want of these **classics** (printed as paperbacks):

Tricky Living,	edition 1	127 pages, list $8.75, your price just	**$2**: ____
Tricky Living,	edition 2	143 pages, list $10.00, your price just	**$2**: ____
The Secret Guide to Computers,	edition 27	639 pages, list $16.50, your price just	**$2**: ____
The Secret Guide to Computers,	edition 28	639 pages, list $17.50, your price just	**$2**: ____
The Secret Guide to Computers,	edition 29	607 pages, list $17.50, your price just	**$2**: ____
The Secret Guide to Computers,	edition 30	575 pages, list $20.00, your price just	**$3**: ____
Secret Guide to Computers & Tricky Living,	edition 31	703 pages, list $25.00, your price just	**$5**: ____
Secret Guide to Computers & Tricky Living,	edition 32	703 pages, list $25.00, your price just	**$7**: ____

How many copies do you want of the free **Secret Brochure** (about our services)?___

How do we reach you?

Print the name & address where you want the goods sent. If you want the shipment split to *several* addresses, list them; you still get the quantity discounts.

Your phone numbers (optional & kept private) will help if we have questions:

Your email addresses (optional & kept private) will get you our news:

Shipping (typically free)

We offer 3 shipping methods:

Standard is available just to the USA. It's **free**. It usually takes 1 week to ZIP codes under 30000, 1½ weeks to other ZIP codes. We usually recommend this shipping method, because it's free.

Rush is like standard (available just to the USA) but a bit faster (because we jump you ahead of other customers and, if reasonable, use Priority Mail or UPS or other fast service). It costs just **$5 total**, even if your order is big or split. If you want to chat about the delivery date, phone 603-666-6644.

International is required for shipping outside the USA. It costs **$10 per paperback book** to Canada (usually taking 1½ weeks), **$15 per paperback book** to other countries (usually taking 2½ weeks), so multiply the number of paperback books by the appropriate amount. Don't count disks or brochures: they ship free.

Circle the shipping method you want.

Final steps

Add the book prices and shipping charge. Write the sum: $_____
How do you wish to pay? Put × in the box:

- ❑ **cash** (we accept cash from all countries, convert foreign currency, send change)
- ❑ **check or money order** (made out to *Secret Guide*)
- ❑ **credit card** (MasterCard/Visa/AmEx/Discover; below write number, expir. date, verif. code, signature)
- ❑ **bill** (available just if you bought at least 10 books from us before or belong to organizations listed at left)

If the books are a gift to a friend, include a greeting card or note for us to give your friend.
On the back of this coupon, please scribble any comments or suggestions you have.
Mail to *Secret Guide*, 196 Tiffany Lane, Manchester NH 03104-4782.

This coupon is for you

Get more copies

We offer 2 kinds of writing:

> The *Secret Guide to Computers* explains computers. *Tricky Living* explains the rest of modern life.

Order more copies for yourself and friends. The books make **great presents** for Christmas, birthdays, graduations, and celebrations. Get the 33rd edition plus **classic editions**, which include extra info about classic computers & lifestyles — and cost less! (Page 9's chart explains how editions differ.) To get discounts and free brochures, use this coupon. Copy it for friends. If questions, call **603-666-6644** or see **SecretFun.com**, which explains more and lets you read some chapters free!

Money-back guarantee

If not sure whether to order, go ahead: you can return unused books anytime for a 100% refund of what you paid us.

Many ways to order

The simplest way to order is to mail this page's coupon with a check, money order, credit-card info, or cash. Other choices:

> You can **visit** us in New Hampshire to pick up the books personally: phone 603-666-6644 for directions and a pickup time.
>
> To order by **credit card**, mail this coupon or phone 603-666-6644, day or night. We take Master Card, Visa, American Express, and Discover. Give your credit-card number, expiration date, name printed on the card, billing address (if different from shipping address), and verification code (3-digit or 4-digit code printed on the card).
>
> We can **bill** you if you bought at least 10 books from us before (or you're employed by or retired from a school, bookstore, government agency, or established computer company). Mail this coupon or phone 603-666-6644 or email Russ@SecretFun.com. If over $700, phone for approval. The bill is due in 30 days.

We accept 5 forms of payment from other countries:

> **international postal money order** (written in U.S. dollars)
>
> **credit-card number** (Master Card, Visa, American Express, or Discover; to transmit the number, mail this coupon or phone 603-666-6644)
>
> **check** (written in U.S. dollars and having a U.S. or Canadian city printed somewhere on the check)
>
> **cash** (we convert foreign currency & send change)
>
> **wire transfer** (in U.S. dollars, from your local outlet of Western Union or MoneyGram, plus a phone call or note telling us how you transferred)

Free book for review

If you plan to introduce the book to at least 80 people, phone 603-666-6644 to request a **free book** for review.

CUT OUT THIS COUPON

Book bargains

Secret Guide to Computers & Tricky Living, 33rd edition, 703 pages, lists for **$25**. To pay less, join friends:

> **20% discount** if you order at least 2: pay just **$20** each.
> **40% discount** if you order at least 4: pay just **$15** each.
> **60% discount** if you order at least 60: pay just **$10** each.

How many copies do you want?____ They're usually printed as paperbacks.
If you want one of them to be a copyable CD-R disk instead, check this box: ☐

Say how many copies you want of these **classics** (printed as paperbacks):

Tricky Living,	edition 1	127 pages, list $8.75, your price just	**$2**: ____
Tricky Living,	edition 2	143 pages, list $10.00, your price just	**$2**: ____
The Secret Guide to Computers,	edition 27	639 pages, list $16.50, your price just	**$2**: ____
The Secret Guide to Computers,	edition 28	639 pages, list $17.50, your price just	**$2**: ____
The Secret Guide to Computers,	edition 29	607 pages, list $17.50, your price just	**$2**: ____
The Secret Guide to Computers,	edition 30	575 pages, list $20.00, your price just	**$3**: ____
Secret Guide to Computers & Tricky Living,	edition 31	703 pages, list $25.00, your price just	**$5**: ____
Secret Guide to Computers & Tricky Living,	edition 32	703 pages, list $25.00, your price just	**$7**: ____

How many copies do you want of the free **Secret Brochure** (about our services)?___

How do we reach you?

Print the name & address where you want the goods sent. If you want the shipment split to *several* addresses, list them; you still get the quantity discounts.

Your phone numbers (optional & kept private) will help if we have questions:

Your email addresses (optional & kept private) will get you our news:

Shipping (typically free)

We offer 3 shipping methods:

> **Standard** is available just to the USA. It's **free**. It usually takes 1 week to ZIP codes under 30000, 1½ weeks to other ZIP codes. We usually recommend this shipping method, because it's free.
>
> **Rush** is like standard (available just to the USA) but a bit faster (because we jump you ahead of other customers and, if reasonable, use Priority Mail or UPS or other fast service). It costs just **$5 total**, even if your order is big or split. If you want to chat about the delivery date, phone 603-666-6644.
>
> **International** is required for shipping outside the USA. It costs **$10 per paperback book** to Canada (usually taking 1½ weeks), **$15 per paperback book** to other countries (usually taking 2½ weeks), so multiply the number of paperback books by the appropriate amount. Don't count disks or brochures: they ship free.

Circle the shipping method you want.

Final steps

Add the book prices and shipping charge. Write the sum: $_____
How do you wish to pay? Put × in the box:

> ❑ **cash** (we accept cash from all countries, convert foreign currency, send change)
> ❑ **check or money order** (made out to *Secret Guide*)
> ❑ **credit card** (MasterCard/Visa/AmEx/Discover; below write number, expir. date, verif. code, signature)
> ❑ **bill** (available just if you bought at least 10 books from us before or belong to organizations listed at left)

If the books are a gift to a friend, include a greeting card or note for us to give your friend.
On the back of this coupon, please scribble any comments or suggestions you have.
Mail to *Secret Guide*, 196 Tiffany Lane, Manchester NH 03104-4782.