

## Chapter 1

# Learning Computer Programming for the First Time

---

### *In This Chapter*

- ▶ Learning computer programming
  - ▶ Understanding how a computer program works
  - ▶ Knowing how to program a computer
- 

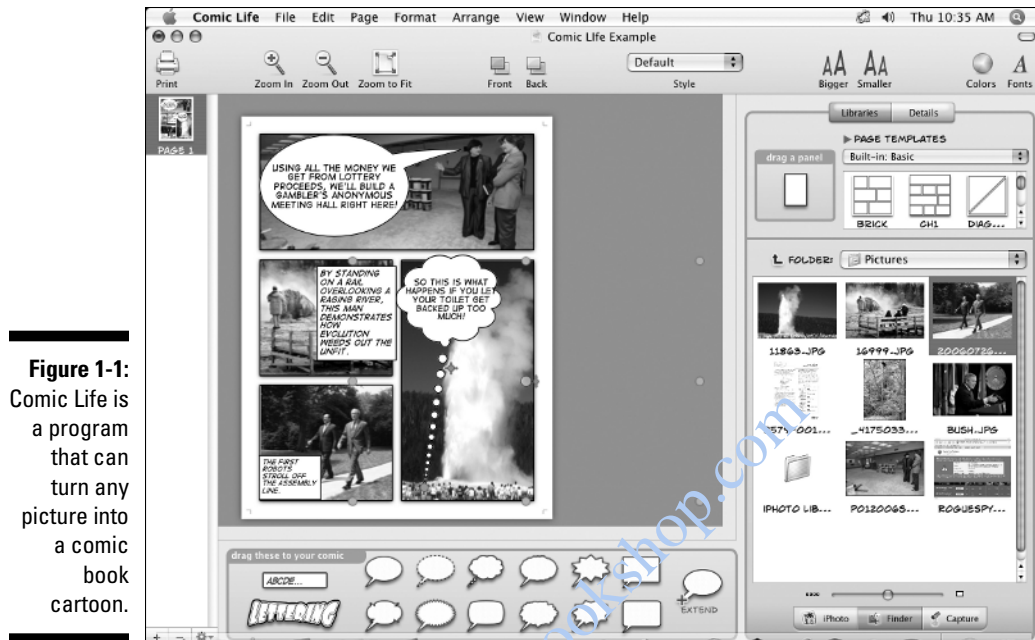
**D**espite what you may have heard, programming a computer isn't difficult. Computer programming is a skill that anyone can pick up, given enough practice, patience, and caffeinated beverages.

Although computers may seem like tremendously complex electronic beasts, relax. Few people know how internal-combustion engines work, yet people can still figure out how to drive a car. Similarly, anyone can pick up programming skills without worrying (too much) about the specific details that make a computer work.

## *Why Learn Computer Programming?*

The first question that you (or your friends, co-workers, and relatives) may ask is, "Why bother learning to program a computer?" The answer depends on your ultimate goals, but the following list offers some common answers to consider:

- ✔ **For fun:** People learn skiing, dancing, gardening, scuba diving, and painting because they enjoy the experience. They may never become professionals or experts in their chosen hobbies, but they enjoy fiddling around nevertheless. Similarly, programming a computer can prove fun because you might design a simple program that displays your boss's ugly face on the computer. More complex programs may make you a million dollars so that you never again need to work for a boss with an ugly face. Figure 1-1 shows a program known as *Comic Life*, which can turn any photograph into a comic book. This program was written in a programming language called Objective-C on a Macintosh.



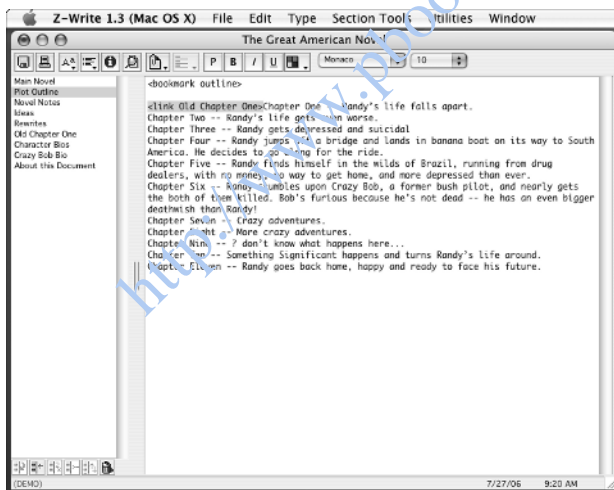
**Figure 1-1:** Comic Life is a program that can turn any picture into a comic book cartoon.

- ✔ **To fill a need:** Many people learn programming with no intention of becoming full-time, professional programmers. They just want a program that solves a particular problem, but they can't find a program that does it, so they write the program themselves. A man once needed a program to help him file his taxes, for example, but he couldn't find one, so he taught himself programming and wound up creating TurboTax, one of the most popular tax-preparation programs in the country. Similarly, a freelance writer got frustrated with trying to write a novel with an ordinary word processor, so he created a word processor, specially designed to organize a story, called Z-Write, as shown in Figure 1-2. Z-Write was written by using a program called REALbasic. Whatever your interests, you can write a program to solve a specific problem that others may find useful as well.
- ✔ **For a new or second career:** With computers taking over the world, you're never unemployed for long if you know how to program a computer. Companies are always looking to create new programs, but you also find a growing market for programmers who can maintain and modify the millions of existing programs that do everything from storing hotel reservations to transferring bank deposits electronically. If you know how to program a computer, you're in a much better position to

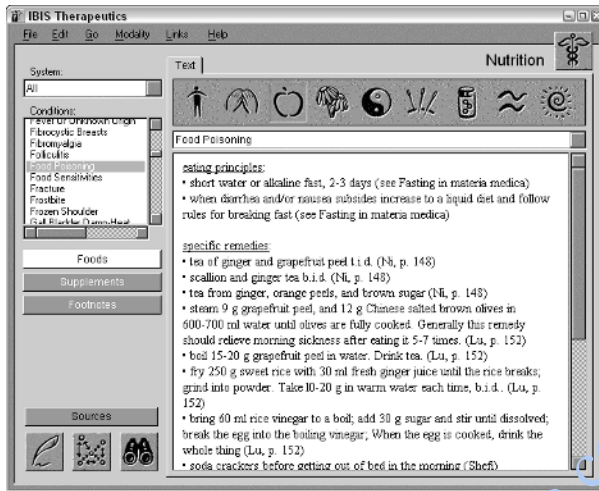
earn a lot of money and live wherever you want. You may still want to keep your current job, but programming gives you a new way to expand and share your knowledge. A group of alternative healthcare practitioners, for example, wrote *IBIS*, a program that provides information for treating a variety of ailments by using acupuncture, massage, diet, and homeopathy (see Figure 1-3). They wrote *IBIS* by using a program known as *Revolution*.

- **As an intellectual challenge:** Many people find the sheer complexity of computers as fascinating as studying a mathematical puzzle. Not surprisingly, computers tend to attract people of above-average intelligence who enjoy programming a computer to pry into the thought processes of their own minds. To help turn a computer into a thinking tool, one programmer created the *Axon Idea Processor* (see Figure 1-4) by using Prolog, a popular programming language used for researching artificial intelligence. The goal was to create a program to help people manipulate ideas, concepts, and facts so that they can devise a variety of possible solutions while better understanding their own ways of thinking in the process. If using a computer normally seems boring, try writing your own program to help you use your brain more effectively.

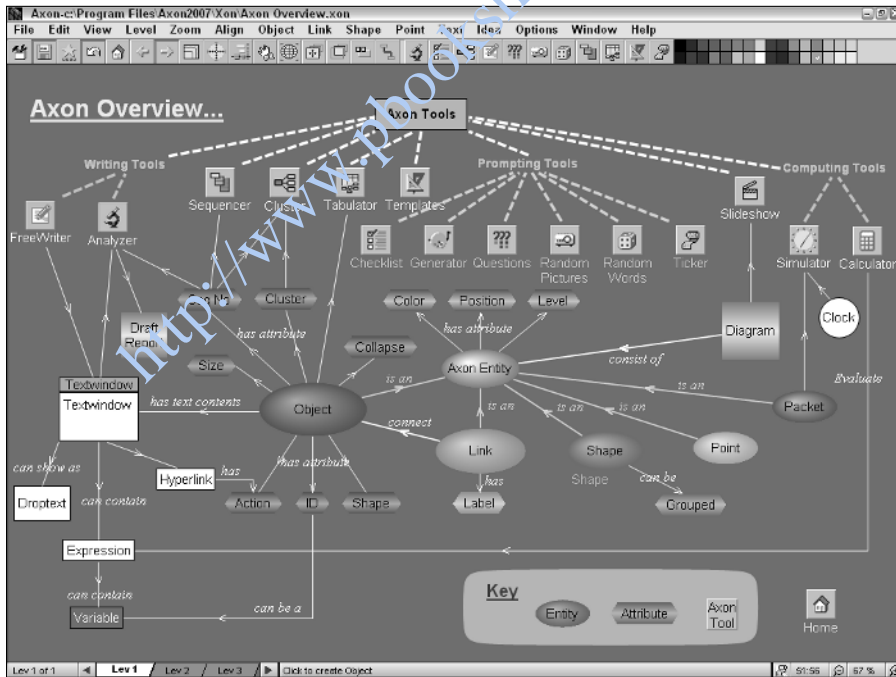
**Figure 1-2:**  
Z-Write can help you organize and write a novel or short story.



**Figure 1-3:** IBIS harnesses computer power to help healthcare workers find a variety of alternative healthcare procedures.



**Figure 1-4:** The Axon Idea Processor turns your computer screen into a canvas for organizing and manipulating ideas.



As you can see from these four examples, the programming language you use doesn't matter as much as what you want to make your computer do. The magic of computer programming doesn't come from using any particular tool, computer, or language. The real magic of programming comes from applying your own imagination and using programming as a means to achieve whatever you want to create.



Although you can make a decent living programming computers, you can also make a decent living selling paper clips, fixing leaky toilets, or raising farm animals. If you aren't doing what you truly enjoy, all the money in the world isn't going to make you happy. Choose to learn programming because you want to — not because you think that it's going to make you rich.

## How Does a Computer Program Work?

Computers don't do anything without someone telling them what to do, much like the average teenager. To make the computer do something useful, you must give it instructions in either of the following two ways:

- ✓ **Write a program** that tells a computer what to do, step by step, much as you write out a recipe.
- ✓ **Buy a program** that someone else has already written that tells the computer what to do.

Ultimately, to get a computer to do something useful, you (or somebody else) must write a program.

A program does nothing more than tell the computer how to accept some type of input, manipulate that input, and spit it back out again in some form that humans find useful. Table 1-1 lists some common types of programs, the types of input that they accept, and the output that they produce.

<i>Type of Program</i>	<i>Input</i>	<i>What the Program Does</i>	<i>Output</i>
Word processor	Characters you type from the keyboard	Formats the text; corrects spelling	Displays and prints neatly organized text
Game	Keystrokes or joystick movements	Calculates how fast and far to move a cartoon figure on-screen	Moves a cartoon figure on-screen

(continued)

<b>Type of Program</b>	<b>Input</b>	<b>What the Program Does</b>	<b>Output</b>
Stock-market predictor	Current and past prices for stocks	Tries to recognize trends in a stock's price fluctuations	Predicts the future price of a stock
Missile guidance program	Current location of the missile and the target	Calculates how to make the missile's location and the target's location coincide	Corrects the trajectory so that it stays aimed at the target
Optical character recognition (OCR)	Text from a scanner	Recognizes shapes of characters	Converts scanned text into a text file that a word processor can edit
Web browser	HyperText Markup Language (HTML) codes on other computers	Converts the HTML codes into text and graphics	Displays Web pages on-screen

## ***Programming is problem-solving***

Essentially, a program tells the computer how to solve a specific problem. Because the world is full of problems, the number and variety of programs that people can write for computers is practically endless.

But to tell a computer how to solve one big problem, you usually must tell the computer how to solve a bunch of little problems that make up the bigger problem. If you want to make your own video game, for example, you need to solve some of the following problems:

- ✓ Determine how far to move a cartoon figure (such as a car, a spaceship, or a man) on-screen as the user moves a joystick.
- ✓ Detect whether the cartoon figure bumps into a wall, falls off a cliff, or runs into another cartoon figure on-screen.
- ✓ Make sure that the cartoon figure doesn't make any illegal moves, such as walking through a wall.

- ✓ Draw the terrain surrounding the cartoon figure and make sure that if the cartoon figure walks behind an object such as a tree, the tree realistically blocks the figure from sight.
- ✓ Determine whether bullets that another cartoon figure fires are hitting the player's cartoon figure. If so, determine the amount of damage, how it affects the movement of the damaged cartoon figure, and how the damage appears on-screen.

The simpler the problem is that you need to solve, the more easily you can write a program that tells the computer how to work. A program that displays a simple Ping-Pong game with two stick paddles and a ball is much easier to write than a program that displays World War II fighter airplanes firing machine guns and dropping bombs on moving tanks while dodging anti-aircraft fire.

## *Programming isn't difficult; it's just time-consuming*

Programming really isn't that difficult or mysterious. If you can write step-by-step instructions directing someone to your house, you can write a program.

The hardest part about programming is identifying all the little problems that make up the big problem that you're trying to solve. Because computers are completely stupid, you need to tell them how to do everything.

If you're giving a friend instructions to get to your house, for example, you may write down the following information:

1. Go south on Highway I-5.
2. Get off at the Sweetwater Road exit.
3. Turn right at the light.
4. Turn left into the second driveway.

Of course, if you try giving these instructions to a computer, the computer will get confused and wants to know the following additional information:

1. Where do I start and exactly how far south do I drive down Highway I-5?
2. How do I recognize the Sweetwater Road exit, and how do I get off at this exit?
3. After I turn right at the light, how far to the right do I turn, and do you mean the traffic light or the streetlight on the corner?

4. After I turn left into the second driveway, what do I do next? Park the car? Honk the horn? Gun the engine and accelerate through your garage door?

You need to tell computers how to do everything, which can make giving them instructions as aggravating and frustrating as telling children what to do. Unless you specify everything that you want the computer to do and exactly how to do it, the computer just plain won't do what you want it to do.



## Sometimes programs never work

After spending years writing a program, people sometimes find that throwing away the whole thing and starting over is easier (and cheaper) than trying to figure out why the current program isn't working in the first place.

Back in the mid-1980s, for example, the United States government had the bright idea to develop a self-propelled, anti-aircraft weapon nicknamed *Sergeant York*. The purpose of the Sergeant York weapon was simple: Find an enemy aircraft and shoot it down.

Unfortunately, the program controlling Sergeant York never quite worked correctly. After spending millions of dollars and countless hours rewriting the program, testing it, and rewriting it again, the programmers thought that they'd finally gotten the program to work right.

To celebrate their achievement, the company that made the Sergeant York weapon staged a demonstration for the top Pentagon generals and officials. They put Sergeant York in a field, sat all the people from the Pentagon in a nearby

grandstand, and flew a remote-controlled drone overhead to demonstrate Sergeant York's capability to track and shoot down an enemy airplane.

But instead of aiming at the overhead target, rumor has it that Sergeant York leveled its twin 40-mm cannons toward the ground and swiveled its guns until they pointed directly at the grandstand where all the Pentagon officials were sitting.

Needless to say, the Pentagon officials created quite a commotion as they scrambled to get out of the line of fire. Fortunately, Sergeant York didn't fire its cannons into the grandstand, but after this disastrous demonstration, the Pentagon cancelled further development and scrapped the entire Sergeant York project.

So if you ever start writing a program and feel like giving up before it ever works, you're in good company along with the Pentagon, military contractors, Fortune 500 corporations, the FBI, and practically everyone else in the world.

## What Do I Need to Know to Program a Computer?

If you're the type who finds the idea of making a program (such as a video game) just as exciting as actually using it, you already have everything you need to program a computer. If you want to learn computer programming, you need a healthy dose of the following three qualities:

- ✔ **Desire:** If you want something badly enough, you tend to get it (although you may serve time in prison afterward if you do something illegal to get it). If you have the desire to learn how to program a computer, your desire helps you learn programming, no matter what obstacles may get in your way.
- ✔ **Curiosity:** A healthy dose of curiosity can encourage you to experiment and continue learning about programming long after you finish reading this book. With curiosity behind you, learning to program seems less a chore and more fun. And as long as you're having fun, you tend to learn faster and retain more information than does someone without any curiosity whatsoever (such as your boss).
- ✔ **Imagination:** Computer programming is a skill, but imagination can give your skill direction and guidance. A mediocre programmer with lots of imagination always creates more interesting and useful programs than a great programmer with no imagination. If you don't know what to do with your programming skill, your talent goes to waste. You need imagination prodding you onward.

Desire, curiosity, and imagination are three crucial ingredients that every programmer needs. If you possess these qualities, you can worry about trivial details such as learning a specific programming language (such as C++), studying advanced math, or attending a university where you can buy a college degree that you can just as easily make with your computer and a desktop-publishing program instead.

Learning to program a computer may (initially) seem an impossible task, but don't worry. Even the best programmers in the world were beginners once. Computer programming is actually simple to understand; everything just tends to fall apart when you write and put a program into actual use.

<http://www.pbookshop.com>