

Chapter 30

Advice for Future Steps



Congratulations! You have reached the end of your first step into the world of problem solving and program design. There is still much more you can learn about problem solving and programming, but there is no doubt that now you have a solid foundation to continue on this journey. This journey is inevitable even for those who do not aspire to become Computer Scientists. Remember that problem solving is at the heart of many human activities.

Although the book builds on much of your background knowledge (e.g., high school algebra), you likely feel you have a different understanding now of this knowledge. The book has emphasized the use of types to organize your thoughts during the problem solving process. At the heart of this process is how elements in the real or an imaginary world are represented in a program. The chosen representation may (and should) be exploited to find solutions to problems. Remember that if you know the type of data to be processed, then you know something about what the solution to a problem may look like.

All good things must continue . . .

168 Advice for Computer Science Students

Be patient and apply the skills you have learned in the future. As your studies progress, you will discover that few textbooks on programming emphasize design. This makes it difficult sometimes to understand the programs presented. When you see such code, tease out the details. Try to formulate the steps of the design recipe to enlighten you about what the program does. This a good skill to develop given that large pieces of software evolve over years of development in which programmers come and go. Documenting the design of a program is a service that you and others will appreciate when you have to maintain code.

Where can you go from here? The most fundamental piece of advice is to read and learn more about problem solving using a computer. You will be well-served if you practice your design skills by learning about a new programming language

every semester and summer. The skills you have developed are directly applicable to solving problems using programming languages beyond BSL, BSL+, and ISL+. As you advance, you will discover abstractions that are not covered in this textbook. You are, however, well-prepared to learn about them. You will also be well-served to explore topics covered in this book in more depth, such as big-O notation (i.e., complexity), generic programming, and distributed programming. I cannot recommend strongly enough to take courses in the implementation of programming languages: truly understand the technology that is central to Computer Science.

169 Advice for Non-Computer Science Students

You may feel excited, overwhelmed, or both after completing this textbook. There is no doubt that you are a better problem solver now. The truth is that you are likely to program throughout your life. Perhaps not using a programming language as done in this textbook, but if you are problem solving then you are programming. You may write essays, diagnose a patient, or create a piece of music. What do these activities have in common with programming? They process data and are refined until you are satisfied with the result. Is this truly different than finally designing Aliens Attack 8? Do you not refine several drafts of an essay? Use the lessons you have absorbed and apply them to domains other than programming. The famous popular saying *don't reinvent the wheel* is a call for abstraction. If you think about it carefully, you will see that the steps of the design recipe are applicable to problem solving in any context.

You are now in a much better position to understand and, therefore, use software. If you use a spreadsheet, you are programming. If you adjust the settings of a thermostat, you are programming. If you are driving, you are programming. No? When you put your blinker on before turning, are you not sending a message to other drivers? If you begin to realize that problem solving and programming are fully intertwined with life, then you are now in a position to be a better problem solver.