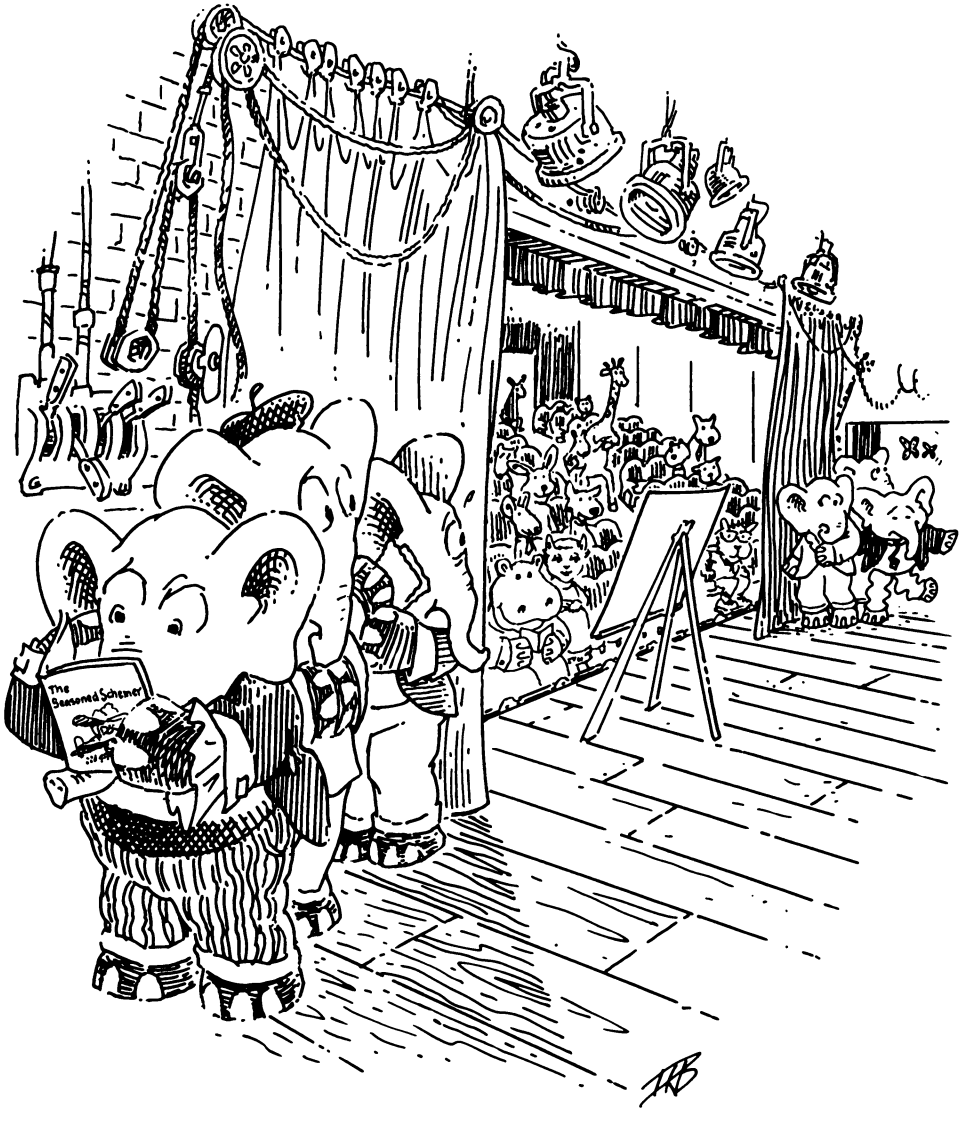
Welcome to the Show



You have reached the end of your introduction to computation. Are you now ready to tackle a major programming problem? Programming requires two kinds of knowledge: understanding the nature of computation, and discovering the lexicon, features, and idiosyncrasies of a particular programming language. The first of these is the more difficult intellectual task. If you understand the material in *The Little Schemer* and this book, you have mastered that challenge. Still, it would be well worth your time to develop a fuller understanding of all the capabilities in Scheme—this requires getting access to a running Scheme system and mastering those idiosyncrasies. If you want to understand the Scheme programming language in greater depth, take a look at the following books:

References

Abelson, Harold and Gerald J. Sussman, with Julie Sussman. *Structure and Interpretation of Computer Programs, 2nd ed.* The MIT Press, Cambridge, Massachusetts, 1996.

Dybvig, R. Kent. *The Scheme Programming Language, 2nd ed.* Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1996.

Eisenberg, Michael. *Programming in Scheme.* The Scientific Press, Redwood City, California, 1988.

Ferguson, Ian with Ed Martin and Bert Kaufman. *The Schemer's Guide, 2nd ed.* Schemers Inc., Fort Lauderdale, Florida, 1995.

Harvey, Brian and Matthew Wright. *Simply Scheme: Introducing Computer Science.* The MIT Press, Cambridge, Massachusetts, 1994.

Manis, Vincent S. and James J. Little. *The Schematics of Computation.* Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1994.

Smith, Jerry D. *An Introduction to Scheme.* Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1989.

Springer, George and Daniel P. Friedman. *Scheme and the Art of Programming.* The MIT Press, Cambridge, Massachusetts, 1989.

Steele, Guy L., Jr. *Common Lisp: The Language, 2nd ed.* Digital Press, Burlington, Massachusetts, 1990.

Afterword

In Fortran you can speak of numbers, and in C of characters and strings. In Lisp, you can speak of Lisp. Everything Lisp does can be described as a Lisp program, simply and concisely.
And where shall you go from here? Suppose you were to tinker with the programs in Chapter 20. Add a feature, change a feature ... You will have a new language, perhaps still like Lisp or perhaps wildly different. The new language may be described in Lisp, yet it will be not Lisp, but a new creation.

If you give someone Fortran, he has Fortran.
If you give someone Lisp, he has any language he pleases.

—Guy L. Steele Jr.

Index