

Preface

Six blind sages were shown an elephant and met to discuss their experience. “It’s wonderful,” said the first, “an elephant is like a rope: slender and flexible.” “No, no, not at all,” said the second, “an elephant is like a tree: sturdily planted on the ground.” “Marvelous,” said the third, “an elephant is like a wall.” “Incredible,” said the fourth, “an elephant is a tube filled with water.” “What a strange piecemeal beast this is,” said the fifth. “Strange indeed,” said the sixth, “but there must be some underlying harmony. Let us investigate the matter further.”

– Freely adapted from a traditional Indian fable.

“A programming language is like a natural, human language in that it favors certain metaphors, images, and ways of thinking.”

– Mindstorms: Children, Computers, and Powerful Ideas [141], *Seymour Papert* (1980)

One approach to study computer programming is to study programming languages. But there are a tremendously large number of languages, so large that it is impractical to study them all. How can we tackle this immensity? We could pick a small number of languages that are representative of different programming paradigms. But this gives little insight into programming as a unified discipline. This book uses another approach.

We focus on programming *concepts* and the *techniques* to use them, not on programming languages. The concepts are organized in terms of computation models. A computation model is a formal system that defines how computations are done. There are many ways to define computation models. Since this book is intended to be practical, it is important that the computation model should be directly useful to the programmer. We will therefore define it in terms of concepts that are important to programmers: data types, operations, and a programming language. The term computation model makes precise the imprecise notion of “programming paradigm”. The rest of the book talks about computation models and not programming paradigms. Sometimes we will use the phrase programming model. This refers to what the programmer needs: the programming techniques and design principles made possible by the computation model.

Each computation model has its own set of techniques for programming and