Whenever possible, programs should be written so that when they don't work properly, it is self-evident. We will discuss how to do this throughout the book.

Finger exercise: Computers can be annoyingly literal. If you don't tell them exactly what you want them to do, they are likely to do the wrong thing. Try writing an algorithm for driving between two destinations. Write it the way you would for a person, and then imagine what would happen if that person were as stupid as a computer, and executed the algorithm exactly as written. (For an amusing illustration of this, take a look at the video https://www.youtube.com/watch?v=FN2RM-CHkuI&t=24s.)

1.1 Terms Introduced in Chapter

declarative knowledge

imperative knowledge

algorithm

computation

fixed-program computer

stored-program computer

interpreter

program counter

flow of control

flowchart

programming language

Universal Turing machine

Church-Turing thesis

halting problem

Turing completeness

literals

infix operators

syntax

static semantics

semantics

2 Not everything is more ex
1960, a bit of computer mer
costs about $0.000000004.