its wings in New Jersey might make the difference of whether or not rain is predicted in Peoria.

It's very possible that even with perfect computer modeling, we might never be able to measure existing weather conditions accurately enough to predict weather more than a few days in advance. The measurements simply can't be precise enough to make the predictions accurate over a longer time frame.

As you can see, this small program has a valuable lesson to teach users of computers. As amazing as computers are, the results that they give us are only as useful as the mathematical models on which the programs are based. Computers can give incorrect results because of errors in programs, but even correct programs may produce erroneous results if the models are wrong or the initial inputs are not accurate enough.

1.9 Chapter Summary

This chapter has introduced computers, computer science, and programming. Here is a summary of some of the key concepts:

- A computer is a universal information-processing machine. It can carry out any process that can be described in sufficient detail. A description of the sequence of steps for solving a particular problem is called an algorithm. Algorithms can be turned into software (programs) that determines what the hardware (physical machine) can and does accomplish. The process of creating software is called programming.

- Computer science is the study of what can be computed. Computer scientists use the techniques of design, analysis, and experimentation. Computer science is the foundation of the broader field of computing which includes areas such as networking, databases, and information management systems, to name a few.

- A basic functional view of a computer system comprises a central processing unit (CPU), main memory, secondary memory, and input and output devices. The CPU is the brain of the computer that performs simple arithmetic and logical operations. Information that the CPU acts on (data and programs) is stored in main memory (RAM). More permanent information is stored on secondary memory devices such as magnetic disks, flash memory, and optical devices. Information is entered into the computer via input devices, and output devices display the results.

- Programs are written using a formal notation known as a programming language. There are many different languages, but all share the property of having a precise syntax (form) and semantics (meaning). Computer hardware understands only a very low-level language known as machine language. Programs are usually written using human-oriented, high-level languages such as Python. A high-level language must either be compiled or interpreted in order for the computer to understand it. High-level languages are more portable than machine language.

- Python is an interpreted language. One good way to learn about Python is to use an interactive shell for experimentation. The standard Python distribution includes a program called IDLE that provides a shell as well as facilities for editing Python programs.

- A Python program is a sequence of commands (called statements) for the Python interpreter to execute. Python includes statements to do things such as print output to the screen, get input from the user, calculate the value of a mathematical expression, and perform a sequence of statements multiple times (loop).

- A mathematical model is called chaotic if very small changes in the input lead to large changes in the results, making them seem random or unpredictable. The models of many real-world phenomena exhibit chaotic behavior, which places some limits on the power of computing.

1.10 Exercises

Review Questions

True/False

1. Computer science is the study of computers.

2. The CPU is the “brain” of the computer.

3. Secondary memory is also called RAM.

4. All information that a computer is currently working on is stored in main memory.

5. The syntax of a language is its meaning, and semantics is its form.

6. A function definition is a sequence of statements that defines a new command.

7. A programming environment refers to a place where programmers work.

8. A variable is used to give a name to a value so it can be referred to in other places.

9. A loop is used to skip over a section of a program.

10. A chaotic function can't be computed by a computer.

Multiple Choice

1. What is the fundamental question of computer science?
   a) How fast can a computer compute?
   b) What can be computed?
   c) What is the most effective programming language?
   d) How much money can a programmer make?

2. An algorithm is like a
   a) newspaper b) venus flytrap c) drum d) recipe

3. A problem is intractable when
   a) you cannot reverse its solution
   b) it involves tractors
   c) it has many solutions
   d) it is not practical to solve

4. Which of the following is *not* an example of secondary memory?
   a) RAM b) hard drive c) USB flash drive d) DVD

5. Computer languages designed to be used and understood by humans are
   a) natural languages
   b) high-level computer languages
   c) machine languages
   d) fetch-execute languages

6. A statement is
   a) a translation of machine language
   b) a complete computer command
   c) a precise description of a problem
   d) a section of an algorithm

7. One difference between a compiler and an interpreter is
   a) a compiler is a program
   b) a compiler is used to translate high-level language into machine language
   c) a compiler is no longer needed after a program is translated
   d) a compiler processes source code

8. By convention, the statements of a program are often placed in a function called
   a) import b) main c) program d) IDLE

9. Which of the following is *not* true of comments?
   a) They make a program more efficient.
   b) They are intended for human readers.
   c) They are ignored by Python.
   d) In Python, they begin with a pound sign (#).

10. The items listed in the parentheses of a function definition are called
   a) parentheticals
   b) parameters
   c) arguments
   d) both b) and c) are correct

Discussion

1. Compare and contrast the following pairs of concepts from the chapter:

   a) Hardware vs. Software

   b) Algorithm vs. Program

   c) Programming Language vs. Natural Language

   d) High-Level Language vs. Machine Language

   e) Interpreter vs. Compiler

   f) Syntax vs. Semantics

2. List and explain in your own words the role of each of the five basic functional units of a computer depicted in Figure 1.1.

3. Write a detailed algorithm for making a peanut butter and jelly sandwich (or some other everyday activity). You should assume that you are talking to someone who is conceptually able to do the task, but has never actually done it before. For example, you might be telling a young child.

4. As you will learn in a later chapter, many of the numbers stored in a computer are not exact values, but rather close approximations. For example, the value 0.1 might be stored as 0.10000000000000000555. Usually, such small differences are not a problem; however, given what you have learned about chaotic behavior in Chapter 1, you should realize the need for caution in certain situations. Can you think of examples where this might be a problem? Explain.

5. Trace through the `chaos` program from Section 1.6 by hand using 0.15 as the input value. Show the sequence of output that results.

Programming Exercises

1. Start up an interactive Python session and try typing in each of the following commands. Write down the results you see.

   a) `print("Hello, world!")`

   b) `print("Hello", "world!")`

   c) `print(3)`

   d) `print(3.0)`

   e) `print(2 + 3)`

   f) `print(2.0 + 3.0)`

   g) `print("2" + "3")`

   h) `print("2 + 3 =", 2 + 3)`

   i) `print(2 * 3)`

   j) `print(2 ** 3)`

   k) `print(7 / 3)`

   l) `print(7 // 3)`

2. Enter and run the `chaos` program from Section 1.6. Try it out with various values of input to see that it functions as described in the chapter.

3. Modify the `chaos` program using 2.0 in place of 3.9 as the multiplier in the logistic function. Your modified line of code should look like this:

```
x = 2.0 * x * (1 - x)
```

Run the program for various input values and compare the results to those obtained from the original program. Write a short paragraph describing any differences that you notice in the behavior of the two versions.

4. Modify the `chaos` program so that it prints out 20 values instead of 10.

5. Modify the `chaos` program so that the number of values to print is determined by the user. You will have to add a line near the top of the program to get another value from the user:

```
n = eval(input("How many numbers should I print? "))
```

   Then you will need to change the loop to use `n` instead of a specific number.

6. The calculation performed in the `chaos` program can be written in a number of ways that are algebraically equivalent. Write a version of the program for each of the following ways of doing the computation. Have your modified programs print out 100 iterations of the calculation and compare the results when run on the same input.

   a) `3.9 * x * (1 - x)`

   b) `3.9 * (x - x * x)`

   c) `3.9 * x - 3.9 * x * x`

   Explain the results of this experiment. *Hint*: See discussion question number 4, above.

7. (Advanced) Modify the `chaos` program so that it accepts two inputs and then prints a table with two columns similar to the one shown in Section 1.8. (*Note*: You will probably not be able to get the columns to line up as nicely as those in the example. Chapter 5 discusses how to print numbers with a fixed number of decimal places.)