Chapter 1

Introduction to Computers

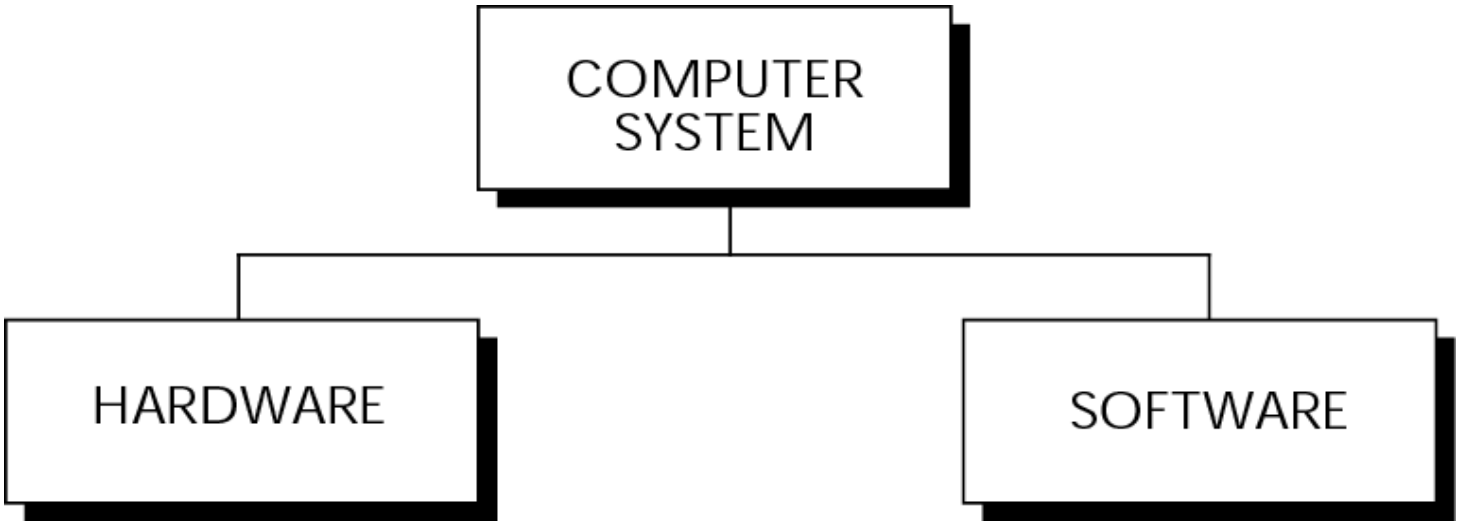
OBJECTIVES

After studying this chapter you will be able to:

- Identify the components of a computer system.
- Identify the components of system and application software.
- Trace the evolution of programming languages.
- Place program development in the system development life cycle.
- Differentiate between blackbox and whitebox testing.
- Understand the role of software engineering in system development.

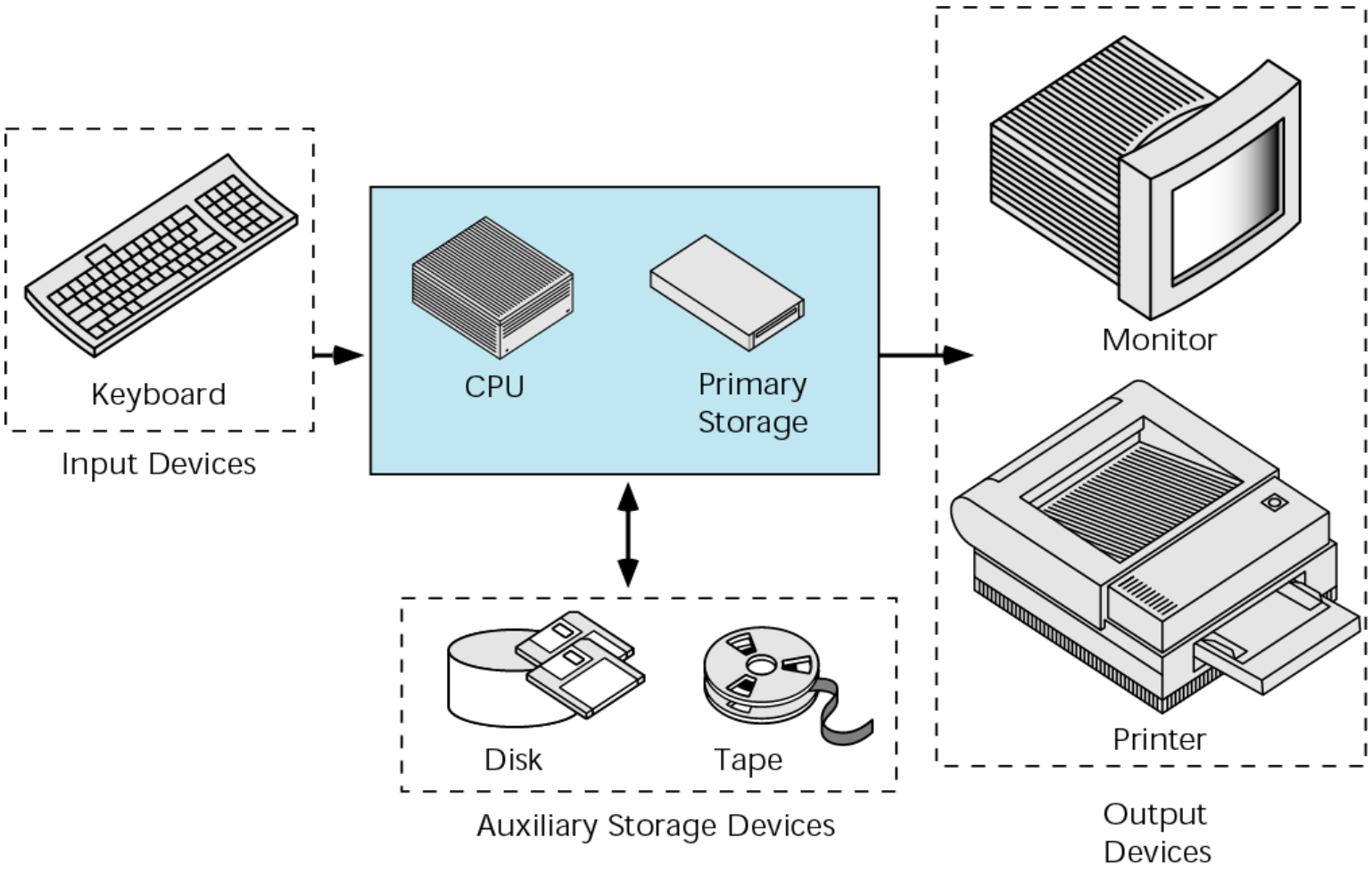
COMPUTER SYSTEMS

Figure 1-1 A computer system



COMPUTER HARDWARE

Figure 1-2 Basic hardware components



COMPUTER SOFTWARE

Figure 1-3 Types of software

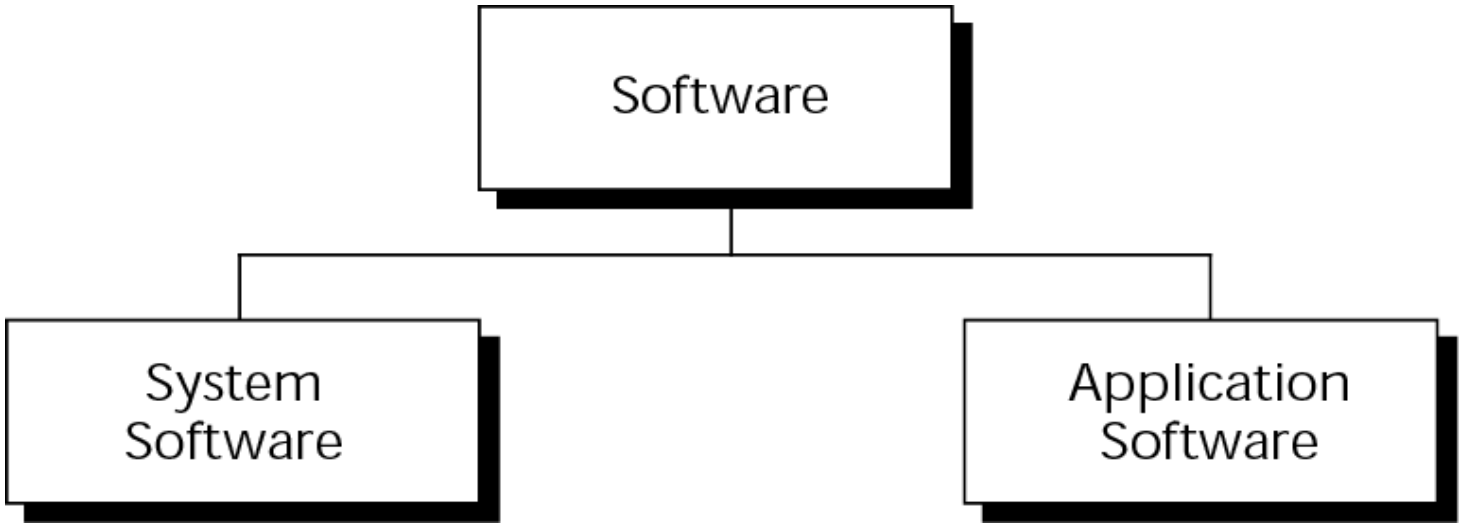
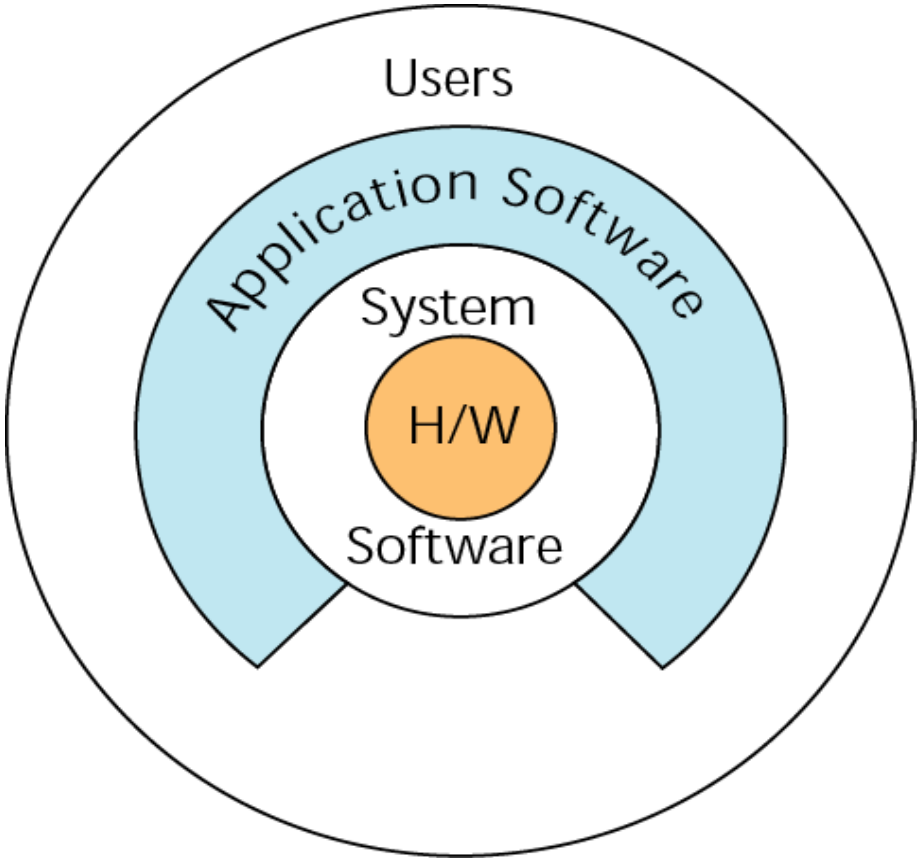


Figure 1-4 Software



COMPUTING ENVIRONMENTS

Figure 1-5 Personal computing environment

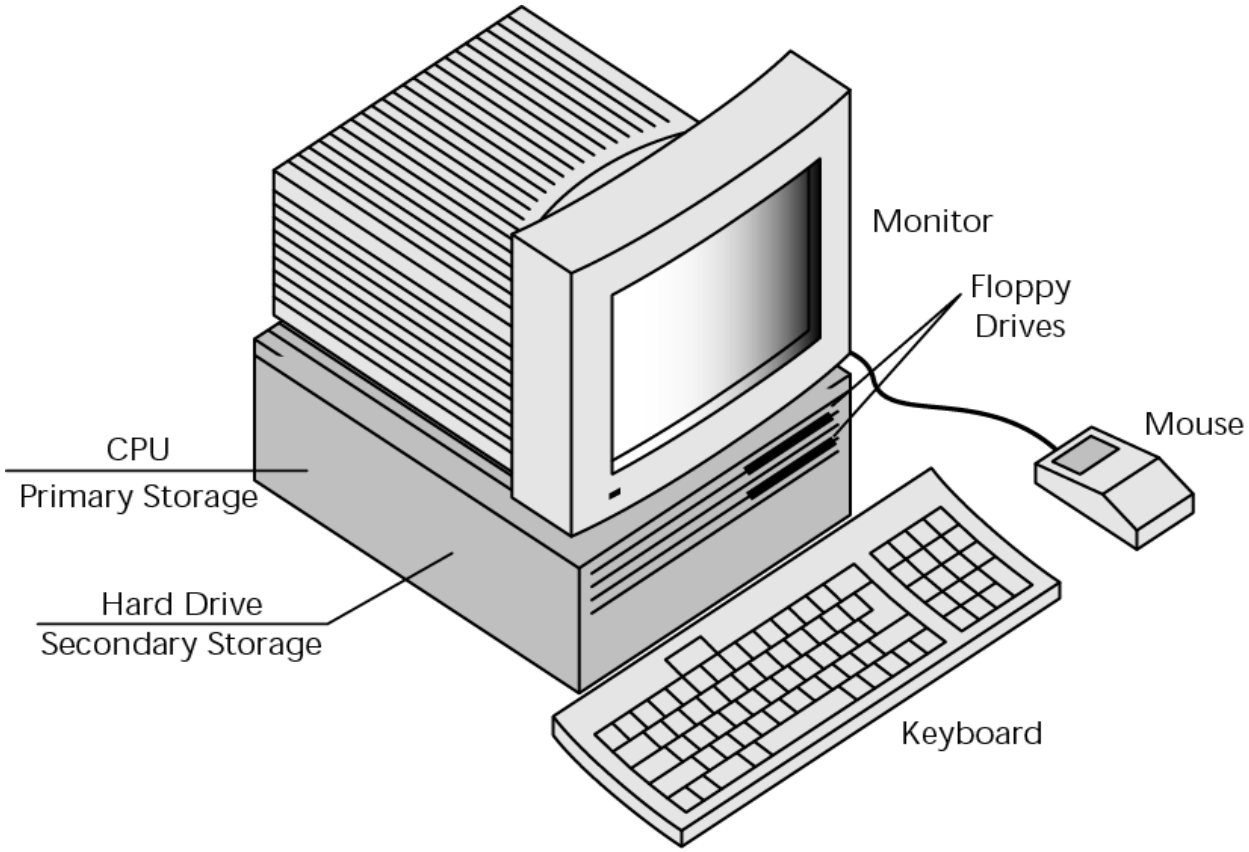


Figure 1-6 Time-sharing environment

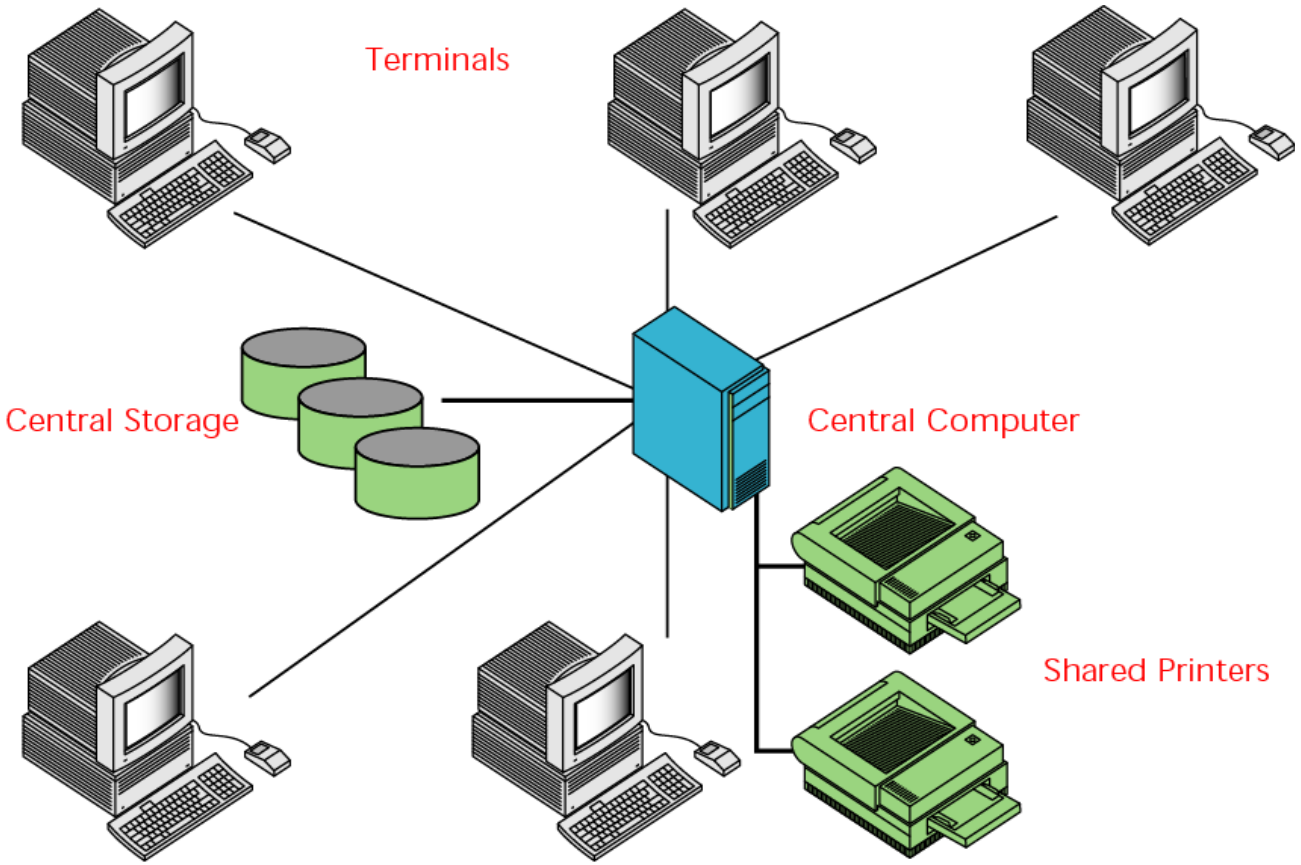


Figure 1-7 Time-sharing environment

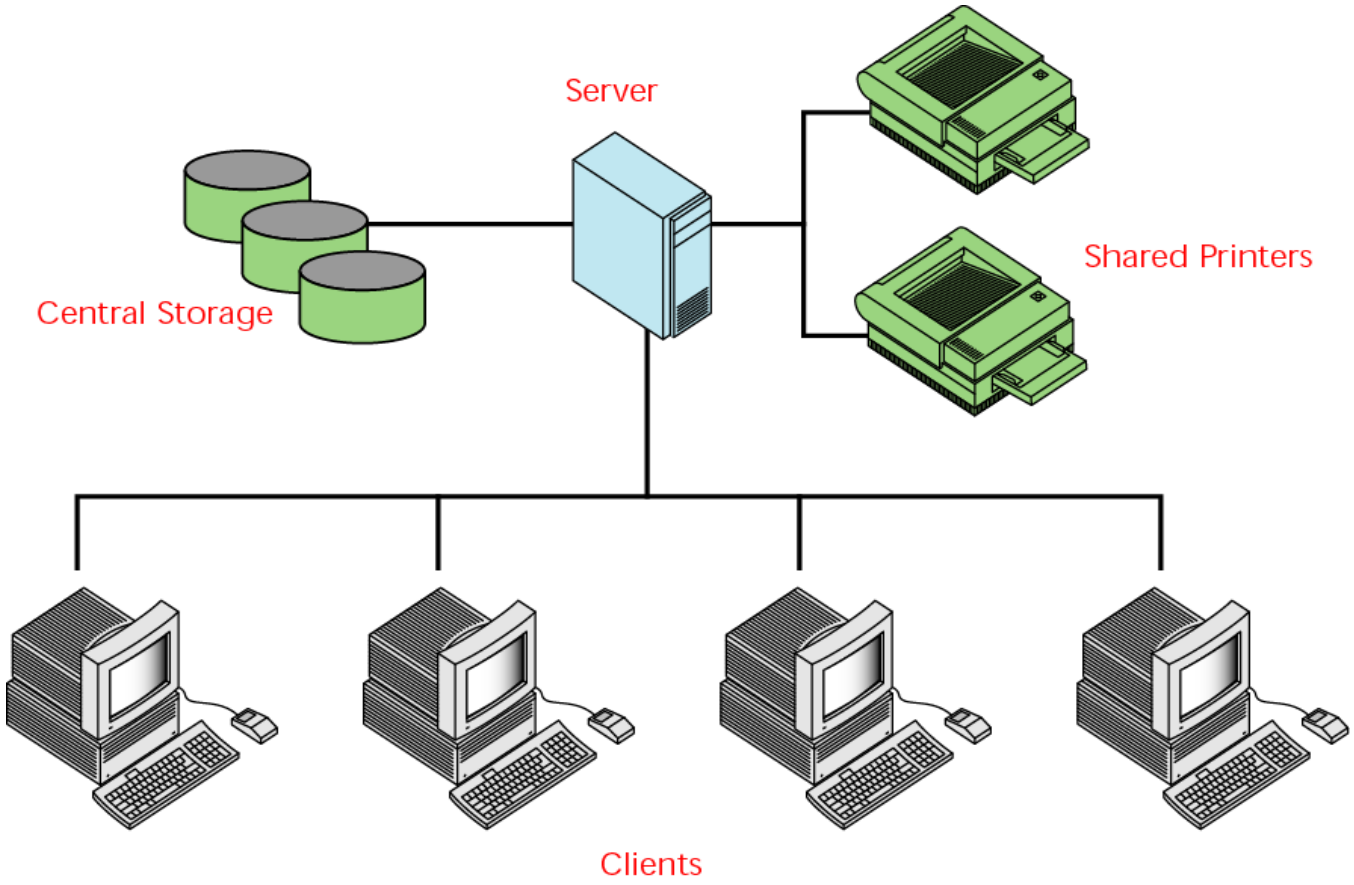
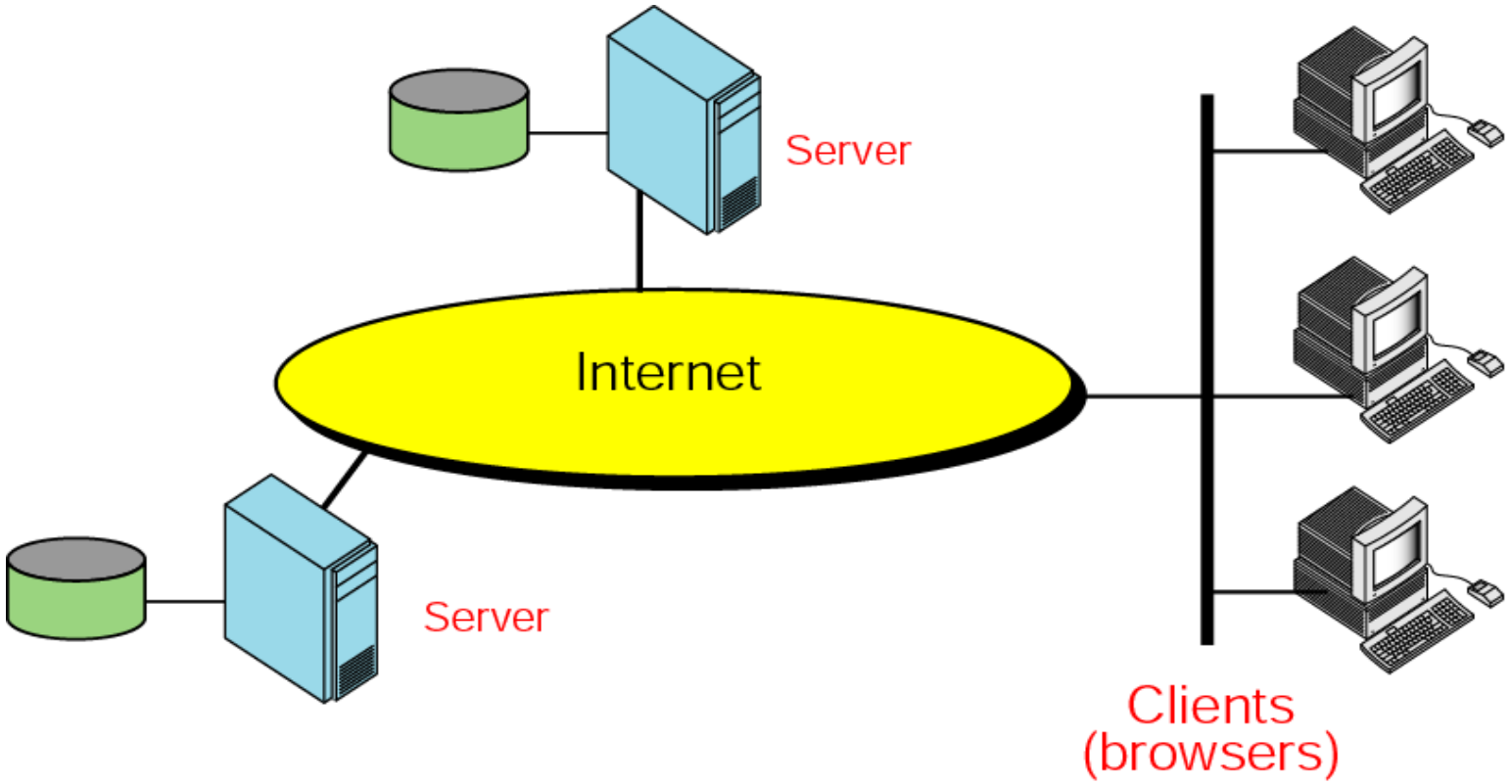
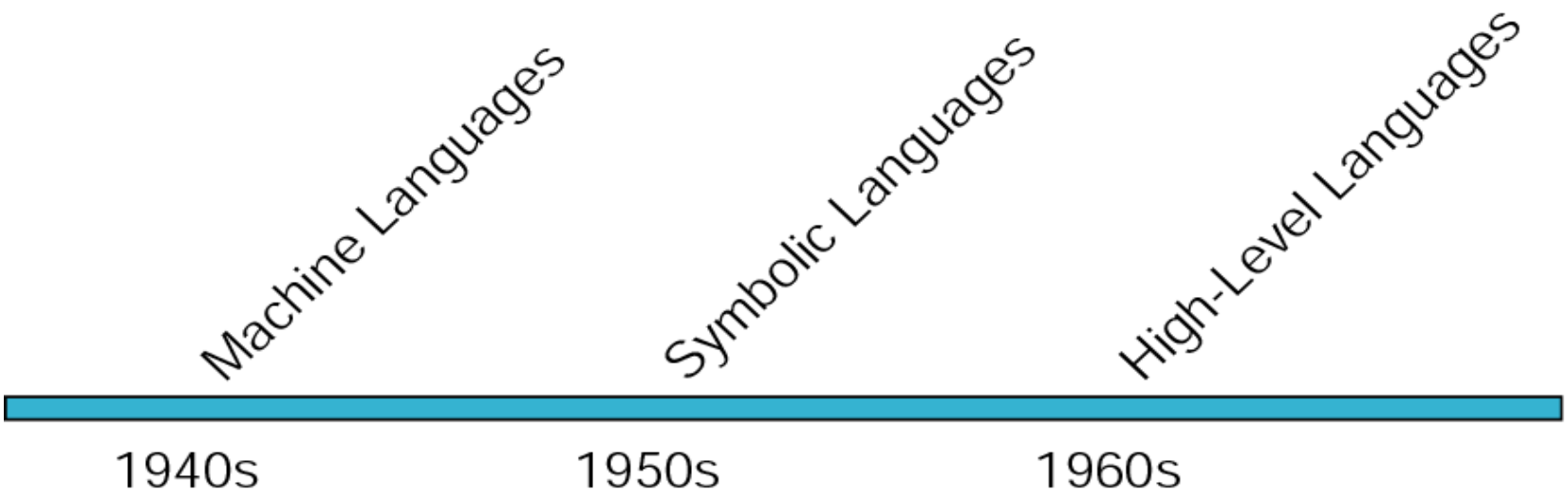


Figure 1-8 Distributed computing



COMPUTER LANGUAGES

Figure 1-9 Computer language evolution

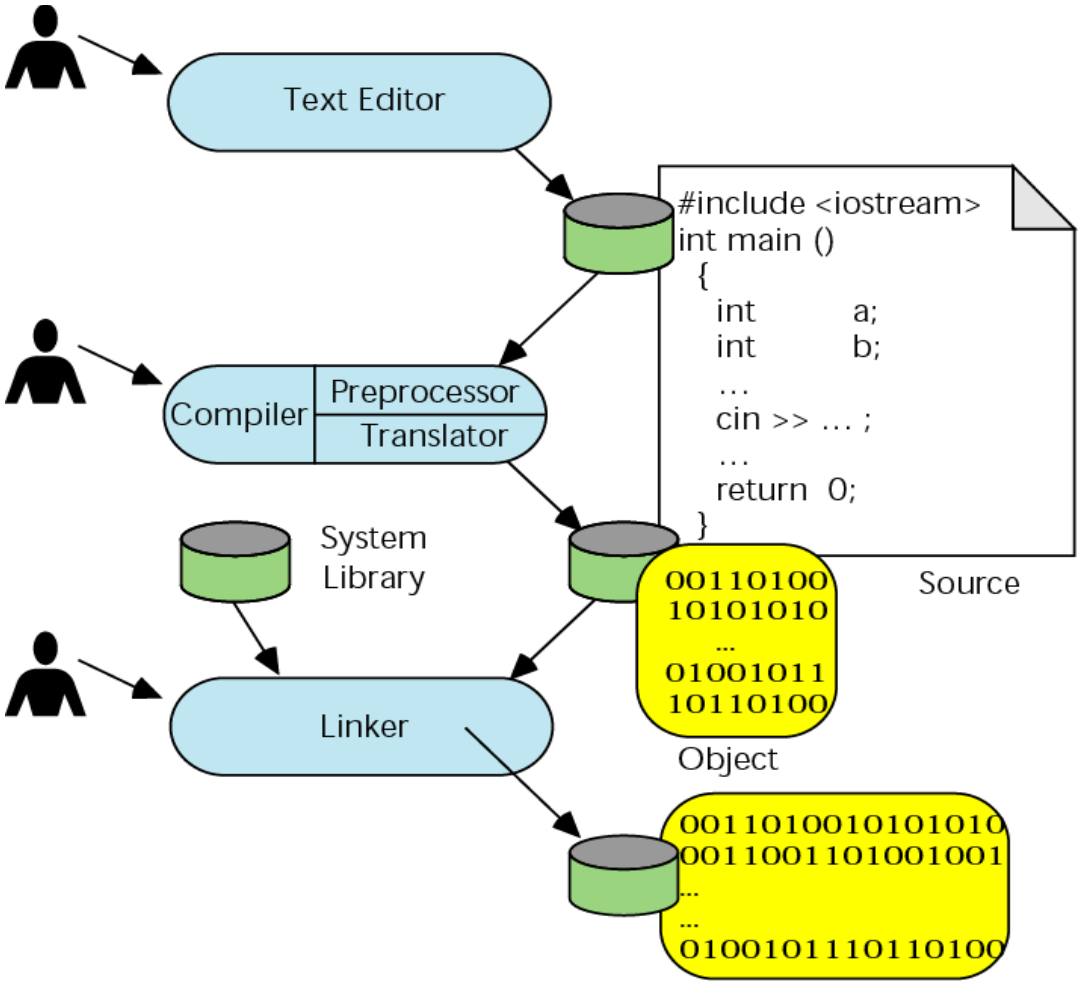


Note:

The only language understood by a computer is machine language.

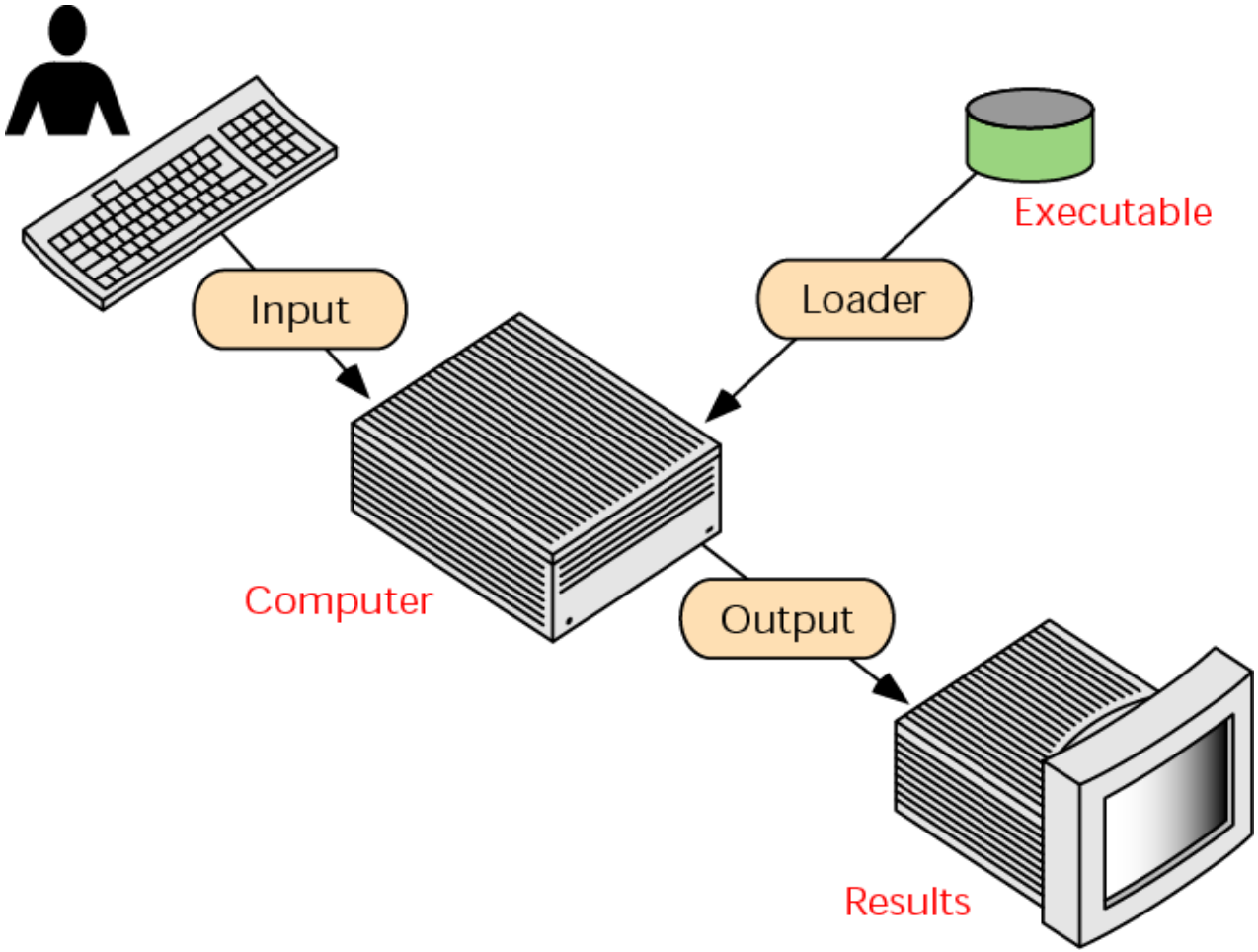
Writing, Editing, Compiling, and Linking Programs

Figure 1-10 Building a C++ program



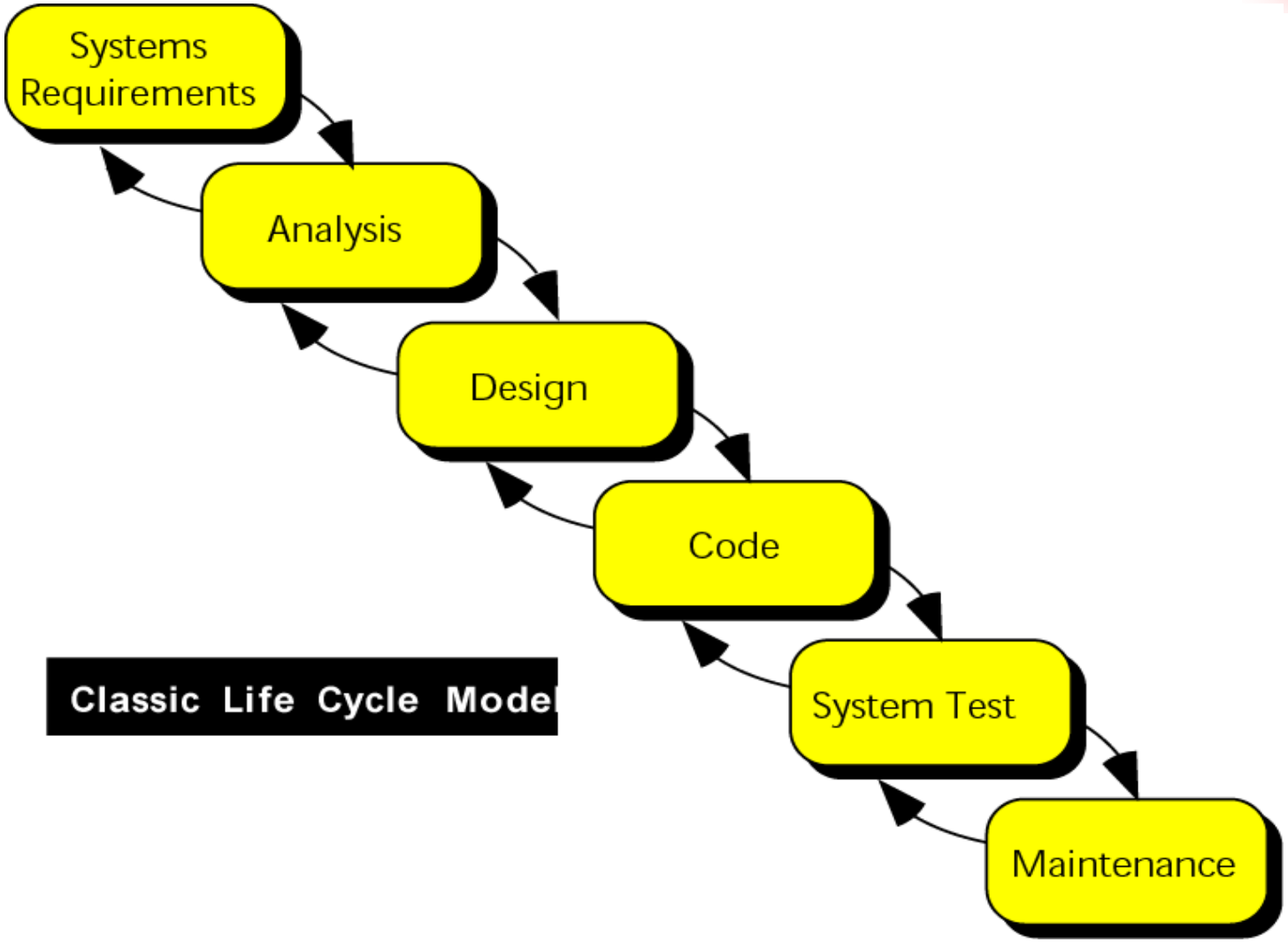
PROGRAM EXECUTION

Figure 1-11 Executing programs



SYSTEM DEVELOPMENT

Figure 1-12 System development model



Note:

An old programming proverb:

Resist the temptation to code.

Note:

Pseudocode

A precise algorithmic description of program logic.

Figure 1-13 Structure chart for calculating footage

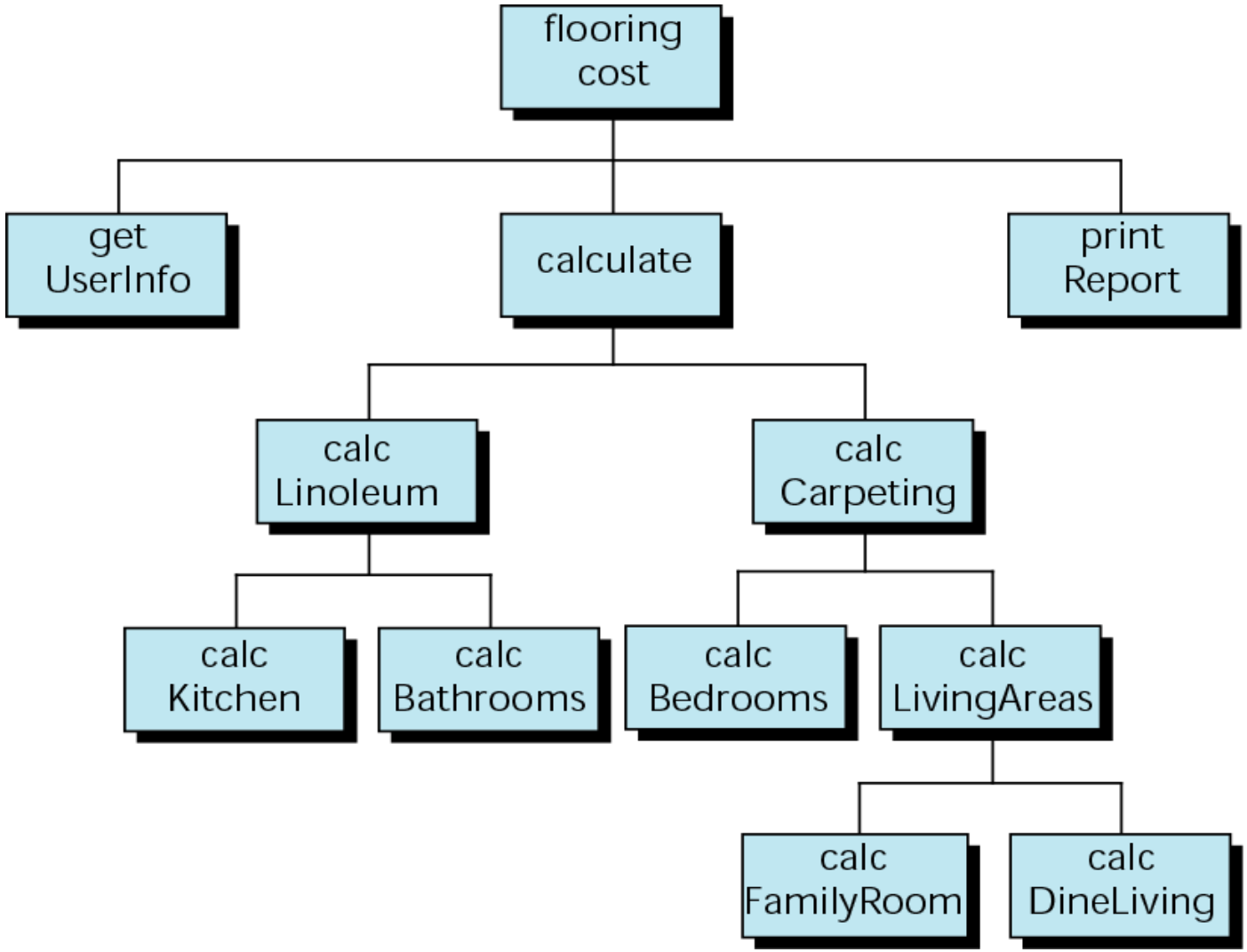
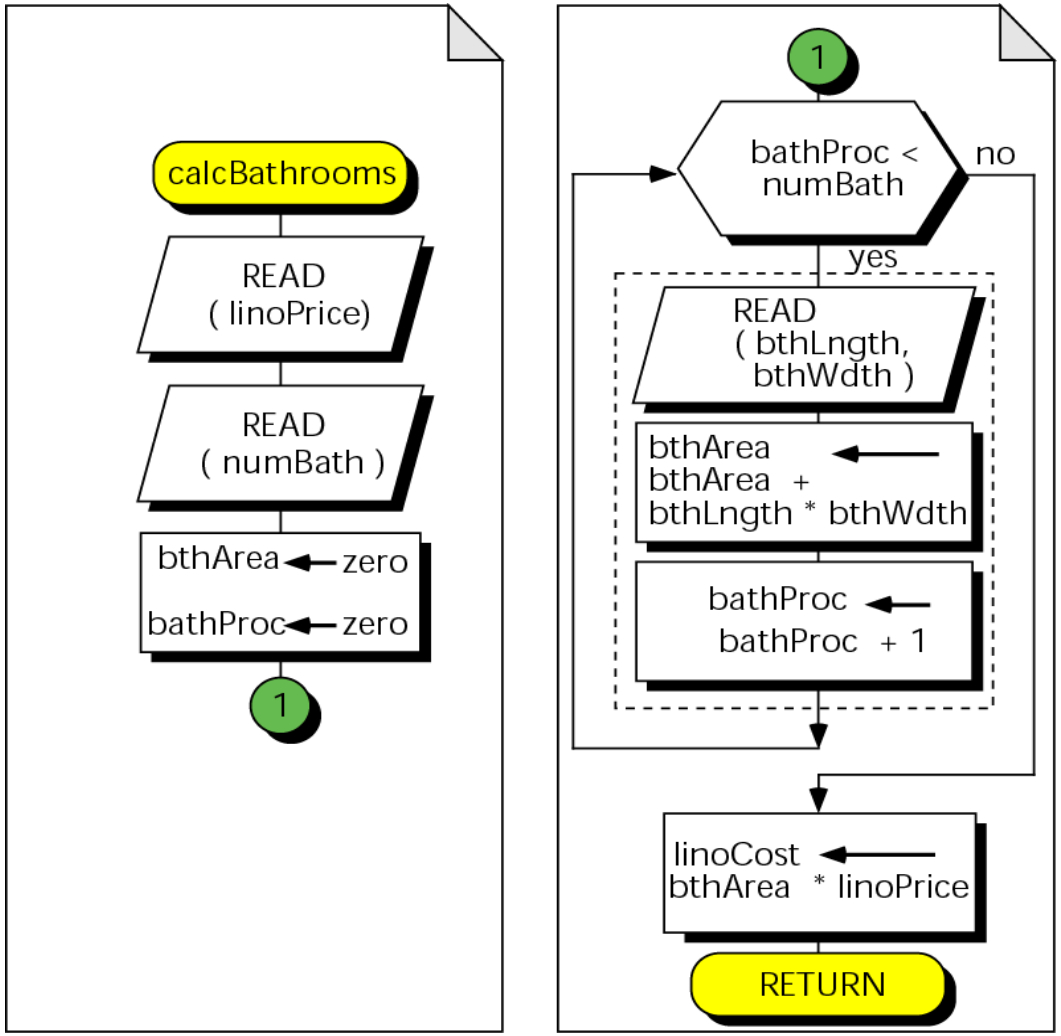


Figure 1-14 Flowchart for calcBathrooms



Note:

One set of test data will never completely validate a program.

SOFTWARE ENGINEERING AND PROGRAMMING STYLE