Chapter 1:
Overview of Computers and Programming

Problem Solving & Program Design in C

8th Edition

By Jeri R. Hanly &
Elliot B. Koffman

Addison Wesley
is an imprint of

PEARSON

Outline

1.1 ELECTRONIC COMPUTERS THEN AND NOW

1.2 COMPUTERS HARDWARE

1.3 COMPUTERS SOFTWARE

1.4 THE SOFTWARE DEVELOPING METHOD

1.5 APPLYING THE SOFTWARE DEVELOPMENT METHOD

CASE STUDY:

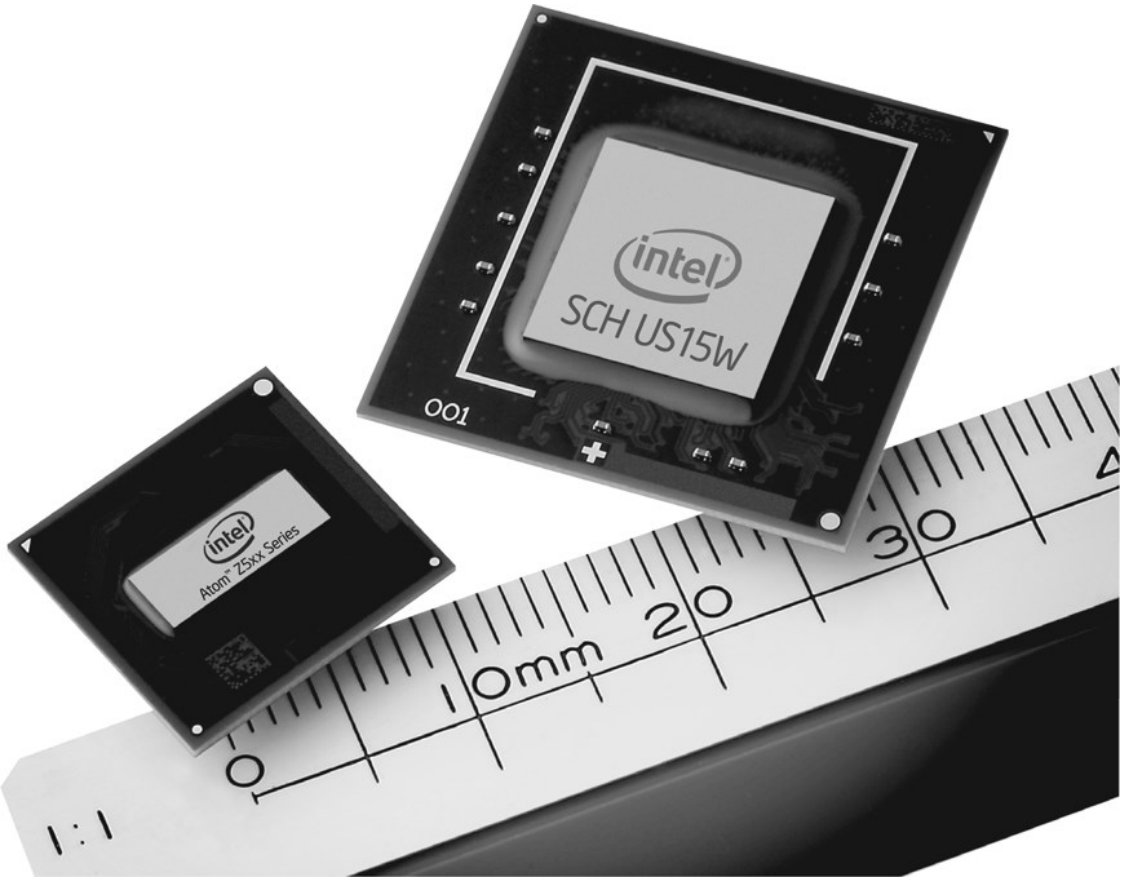
CONVERTING MILES TO KILOMETERS

1.1 ELECTRONIC COMPUTERS THEN AND NOW

- 1930 – the first electronic computer
- 1946 – ENIAC
  - ✓ weighting 30 tons and occupying a 30×50 foot space
- Alan Turing: During the second world war, he devised techniques for speeding the breaking of German ciphers.
  - ✓ See imitation game
- Today – microprocessor chip (Figure1.1)

Figure 1.1

The Intel Atom processor chip contains the full circuitry of a central processing unit in an integrated circuit whose small size and low power requirements make it suitable for use in mobile internet devices. (Intel Corporation Pressroom Photo Archives)



Modern computer categories

- Personal computer (Figure1.2)
- Mainframe
  - Used in large real-time transaction processing
- Supercomputer
  - Used in computationally intensive applications

Figure 1.2

(a) Notebook Computer (HP Pavilion dv5©, Courtesy of Hewlett-Packard).

(b) Palmtop Computer (iPhone 3G©, Courtesy of Apple, Inc.)

(c) Desktop Computer (iMac©, Courtesy of Apple, Inc.)

(a)

(b)

(c)

History of Apple (1/2)

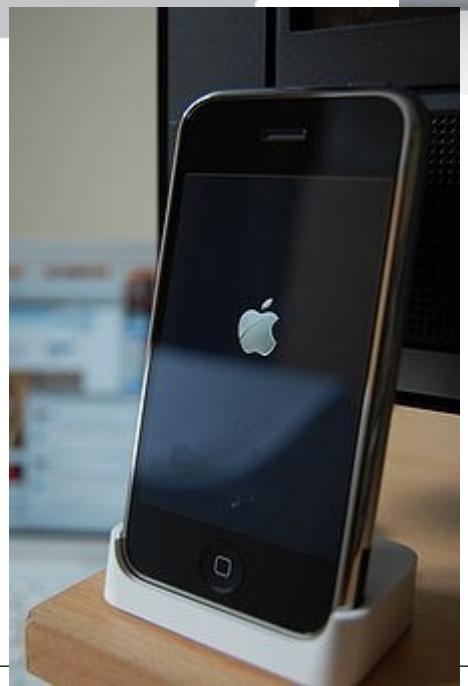
- Apple was established on April 1, 1976
  - By Steve Jobs, Steve Wozniak, and Ronald Wayne
  - Microsoft was established on April 4, 1975
- In 1984, Apple launched the Macintosh

History of Apple (2/2)

- Apple introduced the PowerBook in 1991,which established the modern form of the laptop computer
  - Sold a billion dollars worth within the first year
- 1998, Apple introduced a new all-in-one computer: the iMac

- **iXXX generation**
  - iBook, iPhone, iPod, iPad,…

PRODUCTS



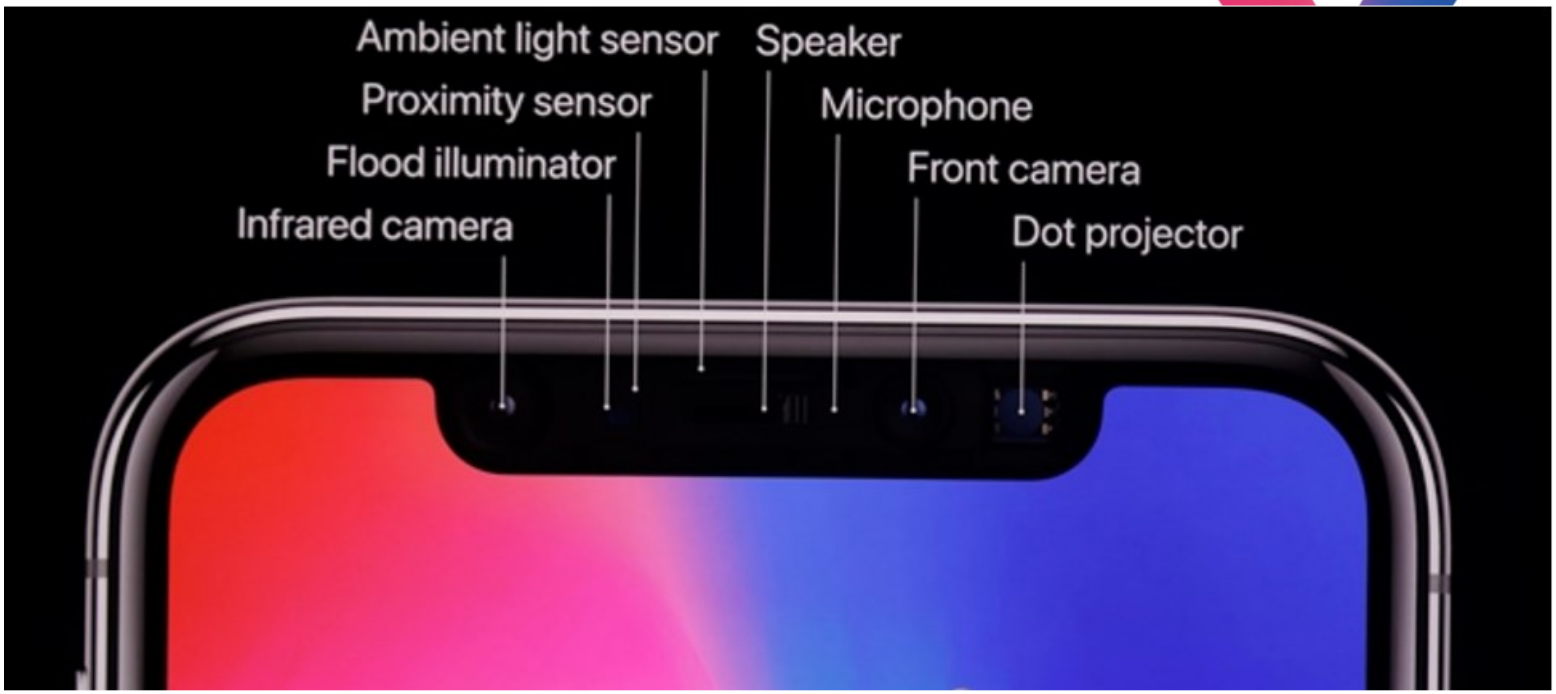
iPod

- The iPod is a portable media player launched on October 23, 2001.
- In 2003, Apple's iTunes Store was introduced, offering online music downloads for $0.99 a song.
- Price: NT$ 1,600~9,500

iPhone

- The first iPhone was introduced on January 9, 2007.
- An iPhone functions as a camera phone, including visual voicemail, a portable media player, and an Internet client, with e-mail, web browsing, and Wi-Fi connectivity.
- Price: NT$ 16,500~19,500

iPhone X



- 12MP wide-angle and telephoto cameras
- Face ID: TrueDepth camera for facial recognition
- History of iPhone

iPhone XS

- 5.8 and 6.5-inch screen sizes
- Up to 512GB storage
- Faster Face ID and edge-to-edge display
- A12 Bionic chip
- Faster Gigabit LTE
- Priced starting at $999 and $1099

iPhone 11

- Three cameras
- Wide angle capture
- 12 megapixel for each
- Deep fusion

Foldable Smartphone

- At the 2019 CES show, a foldable smartphone was first shown.
- The foldable smartphone can be seen as a combination of a smartphone and a tablet.
- Screen size: 7.8 inch
- Screen resolution: 1920 × 1440

VR Technique in Car

- VR helmet can interact with the driver by sound and image.
- The remote traveler meanwhile enjoys an immersive experience thanks to a helmet and virtual reality controllers.

Disney and Audi in-car VR

- Playing VR games or watching VR movies in a moving car.
- The game or the movie will be automatically tailored to the length and movements of your drive route.

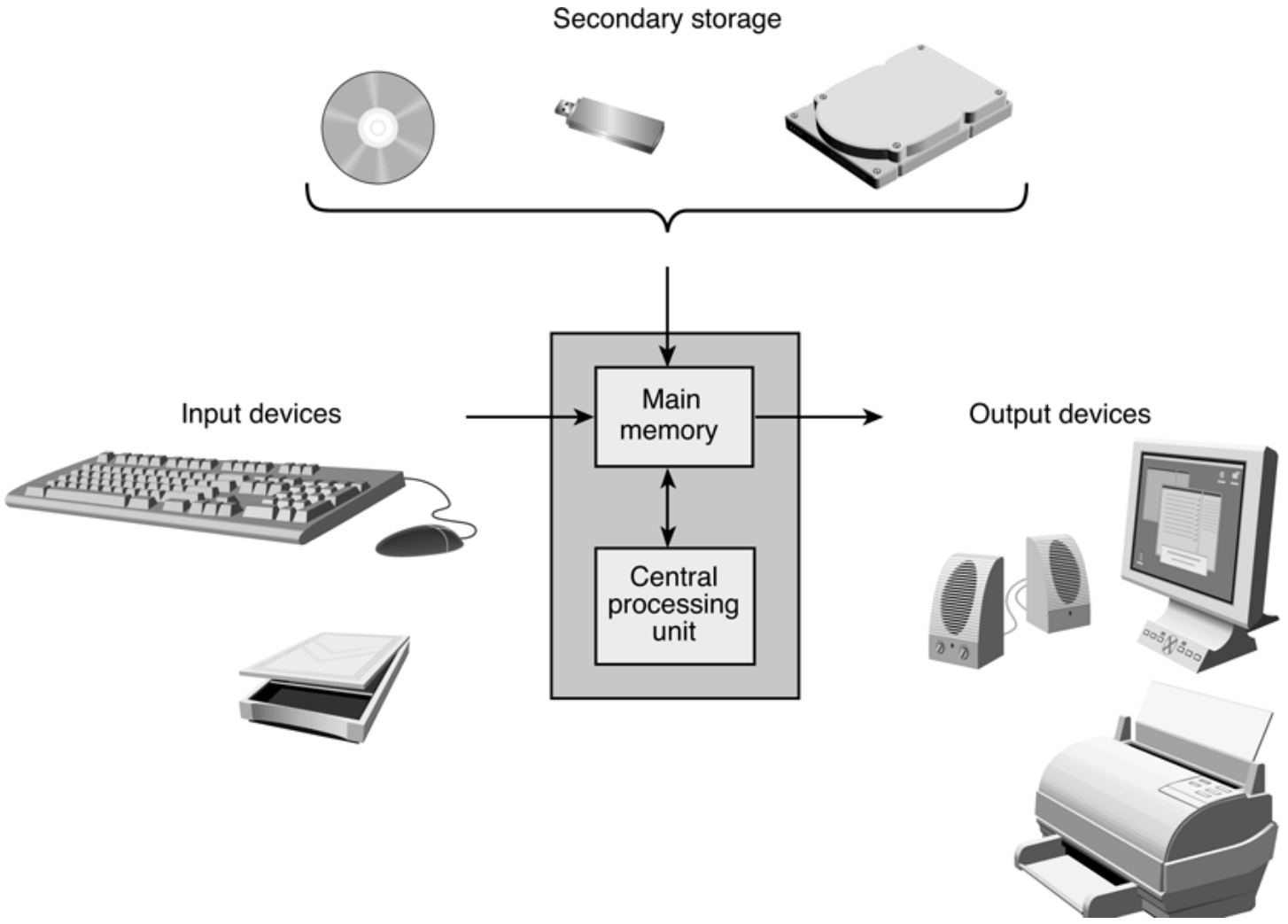
Computer System Major Category

- Hardware – actual computer equipment
- Software – the set of programs

1.2 COMPUTERS HARDWARE

- Computer essential component
  - Main memory
    - RAM (random access memory)
    - ROM (read-only memory)
  - Secondary memory
  - CPU
  - Input devices
  - Output devices

Components of a Computer

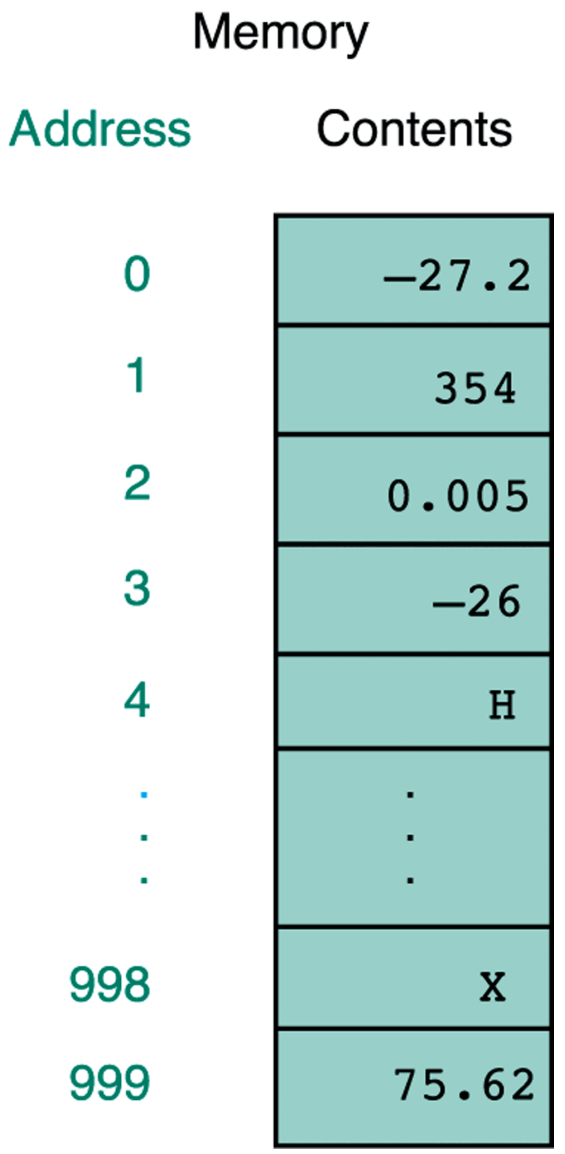


Memory

- Memory cell
  - an individual storage location
- Address
  - relative position in memory
- Contents
  - the data stored in a memory cell
- Stored program concept
  - an ability to store program instructions in main memory for execution

Figure 1.4

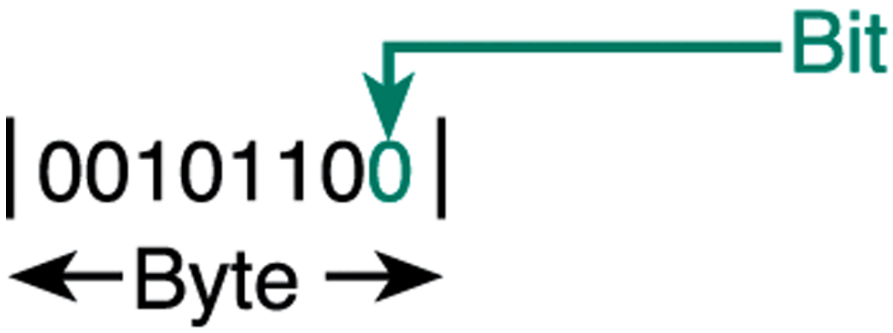
1000 Memory Cells in Main Memory



Bytes and Bits

- Byte
  - the amount of storage required to store a single character
- Bit (binary digit)
  - The smallest element a computer can deal with
  - 8 bits = 1 byte

Figure 1.5 Relationship Between a Byte and a Bit



Storage and Retrieval of Information in Memory

- Data storage
  - Setting the individual bits of a memory cell to 0 or 1, destroying its previous contents
- Data retrieval
  - Copying the contents of a particular memory cell to another storage area

Main memory

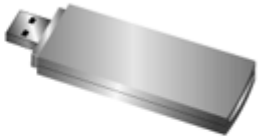
- RAM (random access memory)
  - stores programs and data temporarily
  - volatile memory
- ROM (read-only memory)
  - stores programs and data permanently

Secondary Storage Devices

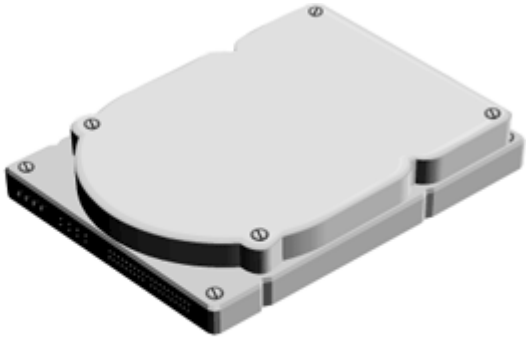
- Disk
  - Thin platter of metal or plastic on which data are represented by magnetized spots arranged in tracks
  - 680 MB of data storage for one CD.
- CD drives
  - Uses a laser to access or store data in CD.
- DVD
  - Silver plastic platter with up to 17GB of data storage
- Blu-ray Disc
  - 25GB for single layer, and 128GB for 4 layers.

Figure 1.6 Secondary Storage Media

CD



Flash drive



Hard disk

CPU

- CPU two roles
  - Control unit: coordinating all computer operations
  - Arithmetic/logical unit: performing arithmetic and logical operations on data
- Fetching an instruction
  - Retrieving an instruction in sequence from main memory

Input/Output Devices

- Cursor
  - A moving place maker that appears on the monitor
- Function keys
  - Special keyboard keys used to select a particular operation
- Icon
  - A picture representing a computer operating

Computer Networks(1/2)

- LAN (Figure 1.7~1.8)
- File server
  - The computer in a network that controls access to a secondary storage device such as a hard disk
- Wide Area Network (WAN)
  - A network such as the Internet that connects computers and LANs over a large geographic area
- World Wide Web (WWW)
  - A part of the Internet whose graphical user interfaces make associated network resources easily navigable.

Computer Networks(2/2)

- GUI
  - Pictures and menus displayed to allow user to select command and data
- MODEM
  - A device that converts binary data into audio signals that can be transmitted between computers over telephone lines.
- Digital Subscriber Line (DSL)
  - A high-speed Internet connection that uses a telephone line and does not interfere with simultaneous voice communication on the same line.
- Cable Internet access
  - Two-way high-speed transmission of Internet data through two of the hundreds of channels available over the coaxial cable that carries cable television signals.

Figure 1.7 Local Area Network

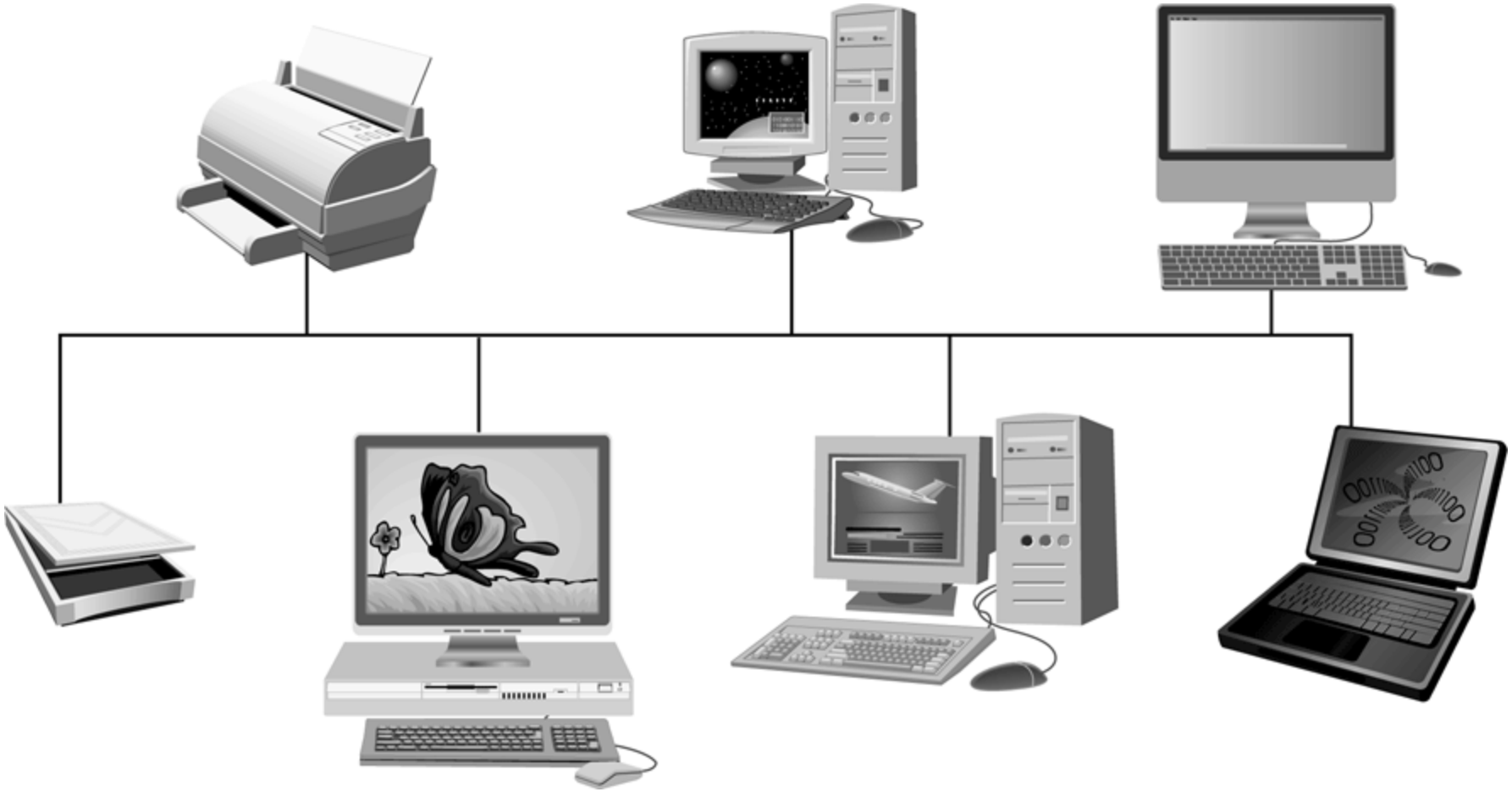
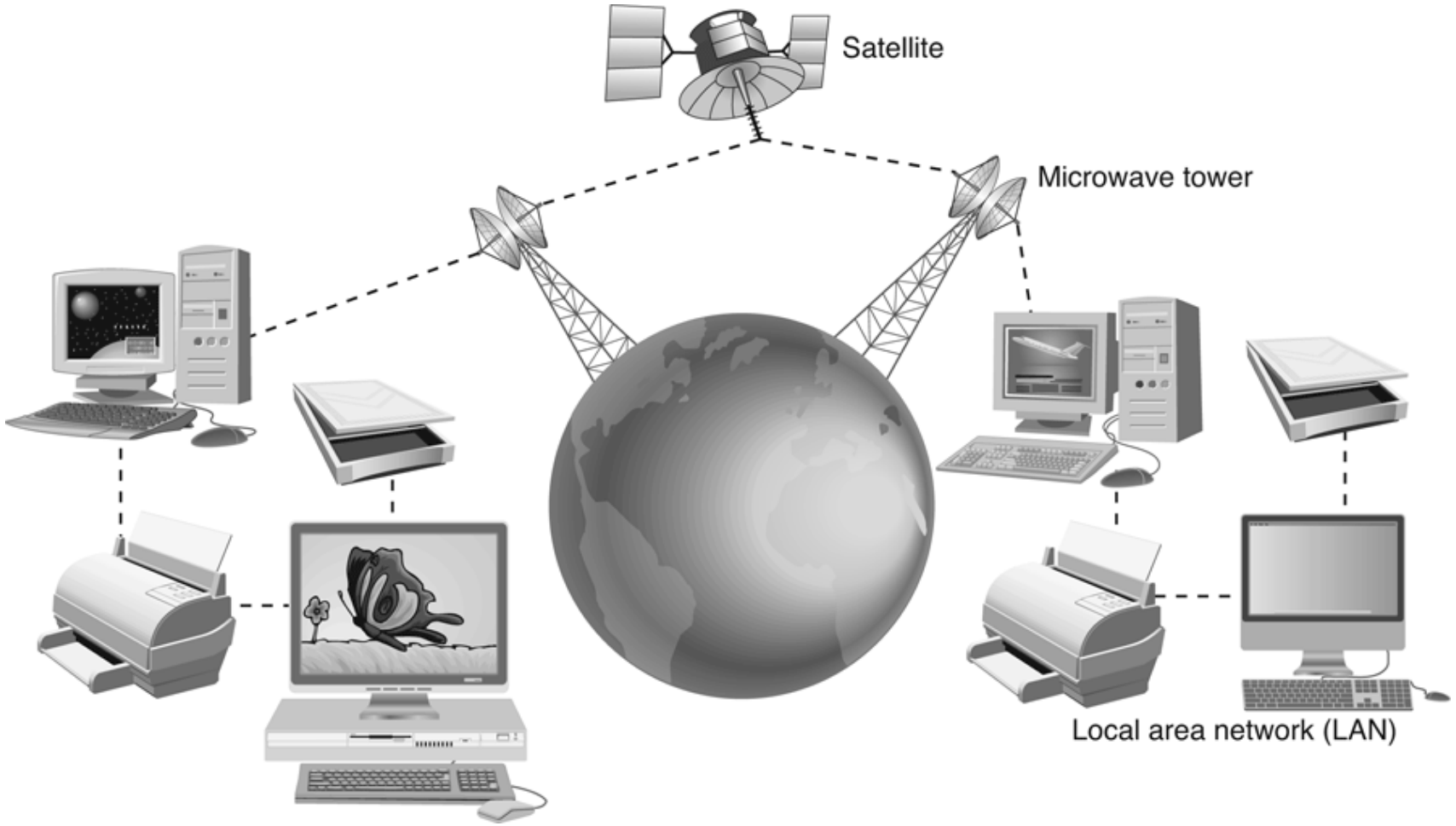


Figure 1.8

A Wide Area Network with Satellite Relays of Microwave Signals



1.3 COMPUTERS SOFTWARE

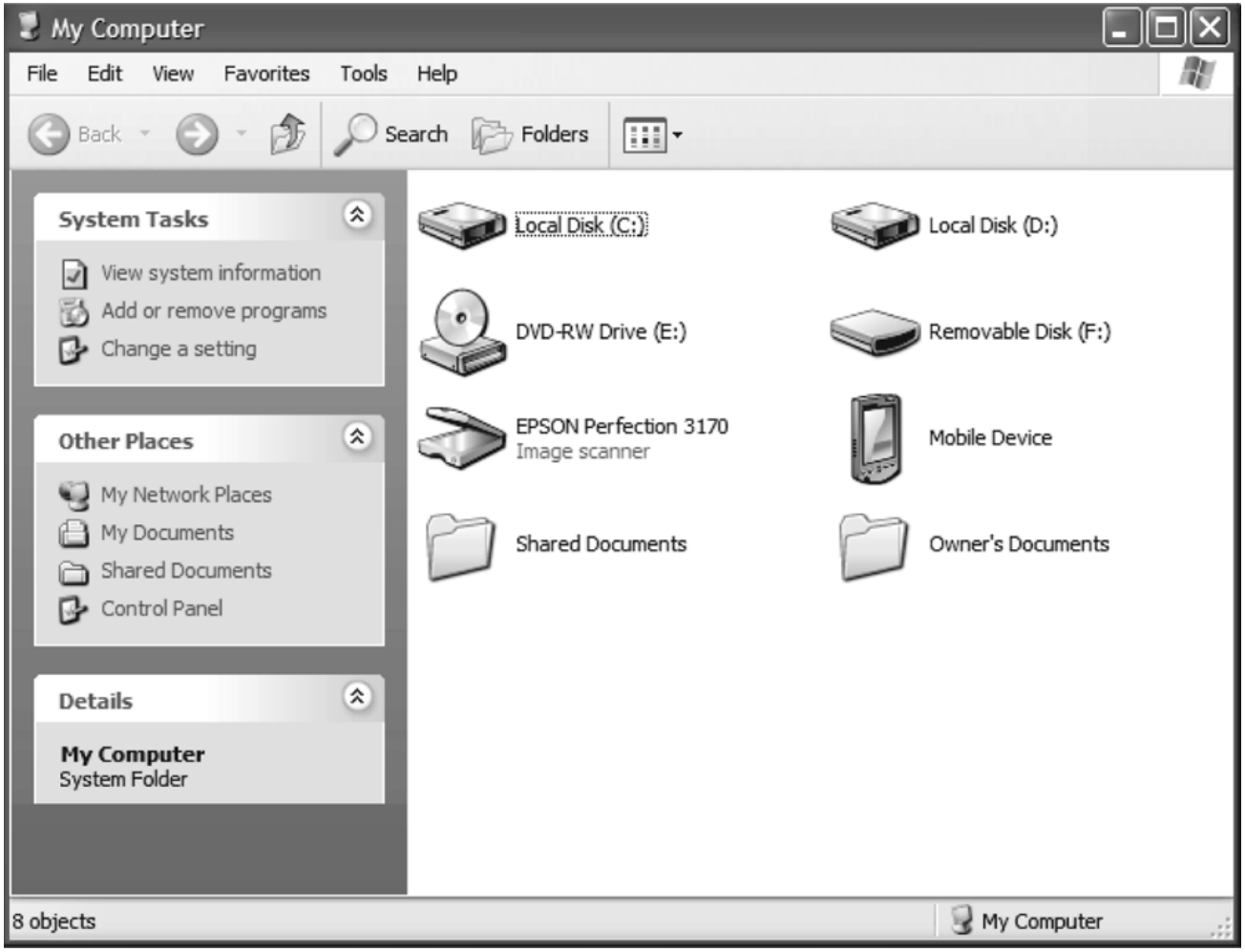
- Operating system
- Booting the computer
- Operating system's responsibility
  - Communicating with the computer user
  - Managing allocation of memory
  - Collecting input from input devices
  - Conveying program output to the output devices
  - Accessing data from secondary storage
  - Writing data to secondary storage

Figure 1.9 Entering a UNIX Command for Directory Display

```
1. mycomputer:~> ls temp/misc
2. Gridvar.c      Gridvar.exe    Gridok.dat
3.
4. mycomputer:~>
```

Figure 1.10

Accessing Secondary Storage Devices through Windows



Application Software

- Application programs are developed to assist a computer user in accomplishing specific tasks.
  - Ex: Microsoft Word, Excel,…etc

Computer Languages

- Machine language
  - Binary number codes understood by a specific CPU
  - Example: 0100110101111000…etc.
- Assembly language
  - Mnemonic codes that corresponding to machine language instructions
  - Example: LOAD A, ADD B, STORE SUM, …etc.
- High-level language (Table 1.4 @ p.17)
  - Machine-independent programming language that combines algebraic expressions and English symbols. Such as FORTRAN, COBOL, LISP, C, Prolog, Ada, Smalltalk, C++, Java
  - Most favorite language ?

Computer Languages

- compiler
  - software that translates a high-level language program into machine language
- source file
  - file containing a program written in a high-level language; the input for a compiler
- syntax
  - grammar rules of a programming language
- object file
  - file of machine language instructions that is the output of a compiler

Computer Languages

- linker
  - software that combines object files and resolves cross-references to create an executable machine language program
- integrated development environment (IDE)
  - software package combining a word processor, compiler, linker, loader, and tools for finding errors

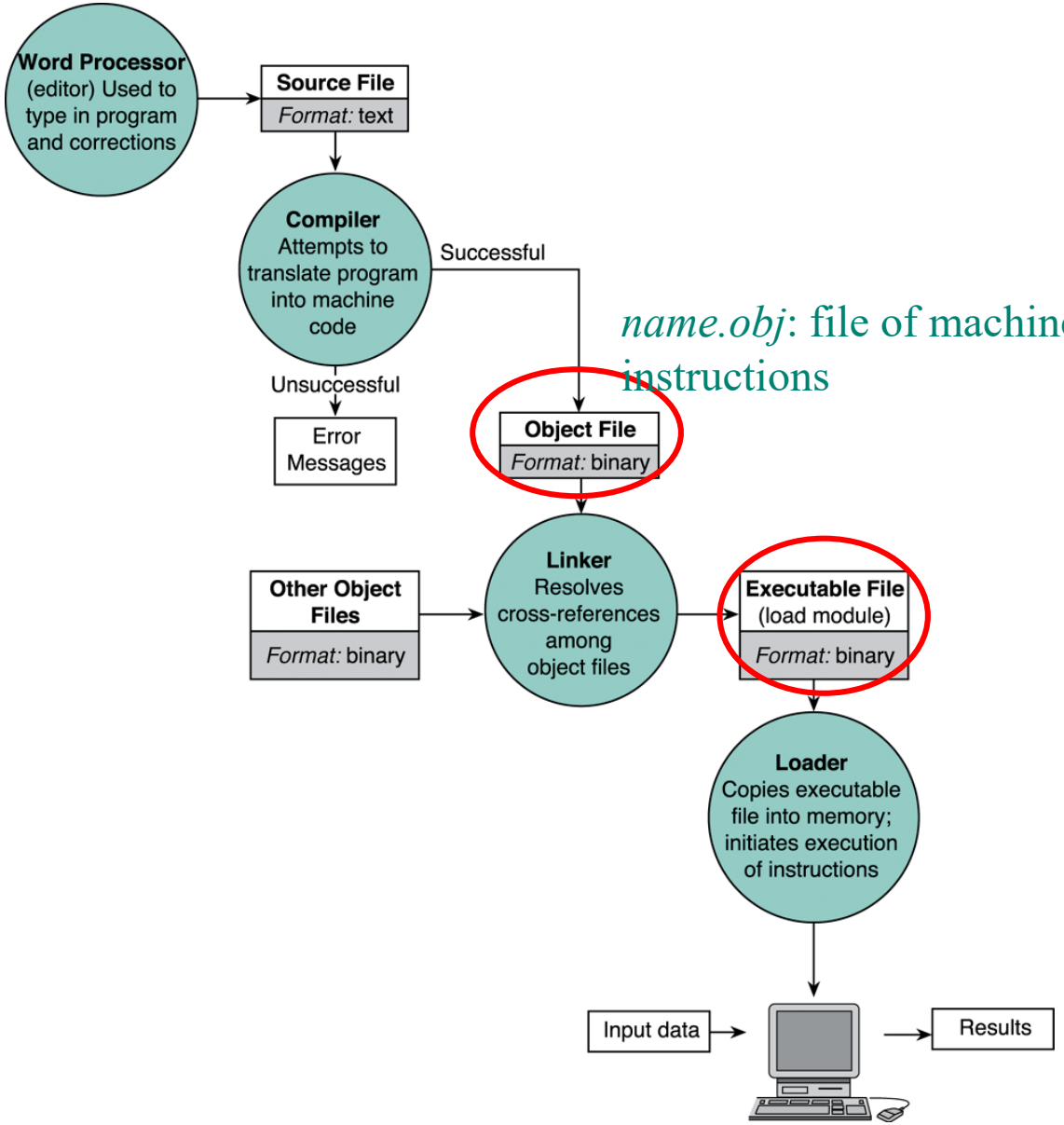
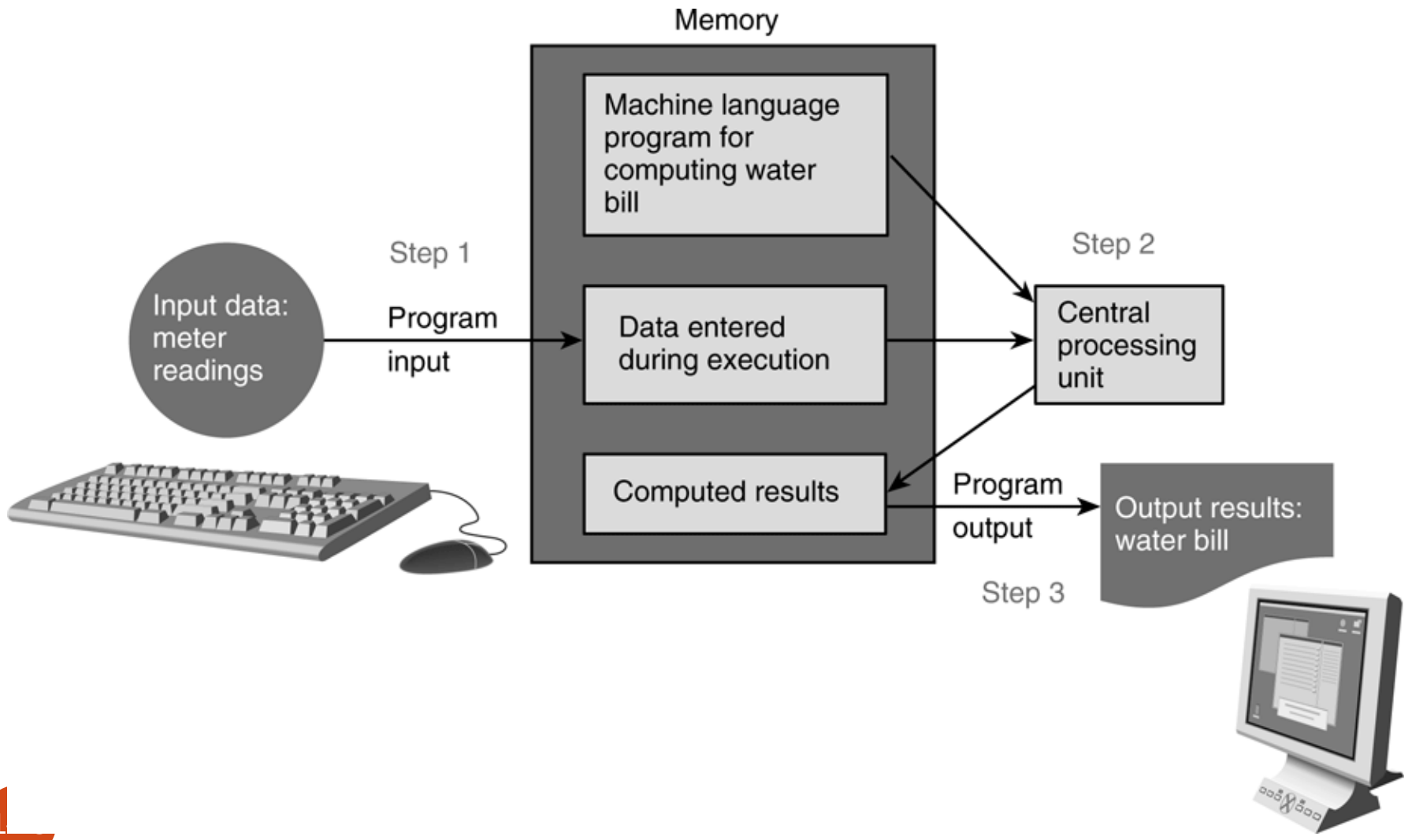


Figure 1.11 Entering, Translating, and Running a High-Level Language Program

Figure 1.12 Flow of Information During Program Execution



1.4 THE SOFTWARE DEVELOPMENT METHOD (1/5)

1. Specify the problem requirements
2. Analyze the problem
3. Design the algorithm to solve the problem
4. Implement the algorithm
5. Test and verify the completed program
6. Maintain and update the program

1.4 THE SOFTWARE DEVELOPMENT METHOD (2/5)

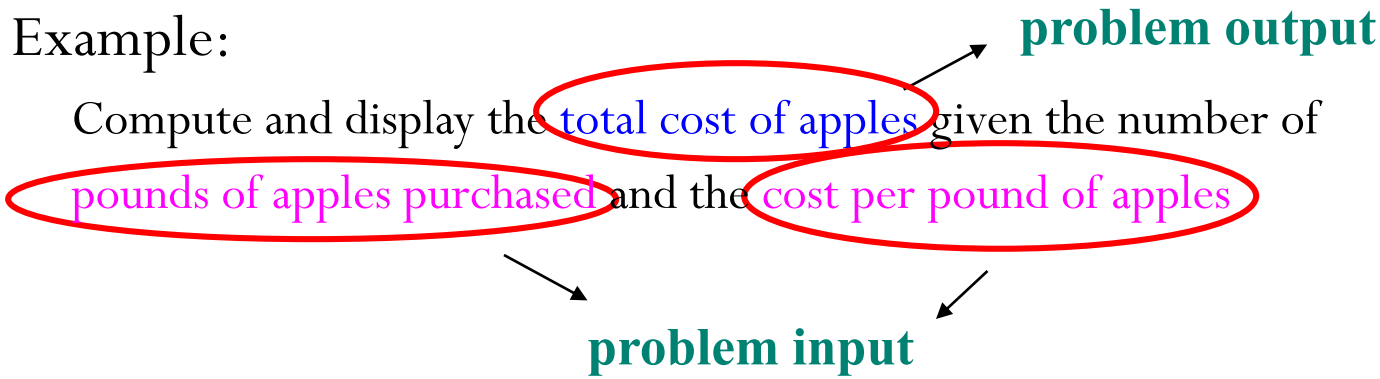
1. Specify the problem requirements
   - State the problem clearly and gain a clear understanding of what is required for its solutions
   - Eliminate unimportant aspects

1.4 THE SOFTWARE DEVELOPMENT METHOD (3/5)

2. Analyze the problem

- Identify the problem
  - Input : The data you have to work with
  - Output : The desired result
  - Additional requirements or constraints

Example:



Total cost = Unit cost × Number of units

Total cost of apples= Cost per pound × Number of apples

1.4 THE SOFTWARE DEVELOPMENT METHOD (4/5)

3. Design the algorithm to solve the problem

- Develop a list of steps (called **algorithm**) to solve the problem and to then verify that the algorithm solves the problem as intended.
- Algorithm for a programming problem
  - Get the data
  - Perform the computations
  - Display the results
- Writing the algorithm is often the most difficult part of the problem-solving process.
- **Top-down design** : Start with the list of major steps. Followed by develop a more detail steps for each major step.
  - **Algorithm refinement** – Development of a detailed list of steps to solve a particular step in the original algorithm.

1.4 THE SOFTWARE DEVELOPMENT METHOD (5/5)

4. Implement the algorithm

- Convert each algorithm step into one or more statements in a programming language.

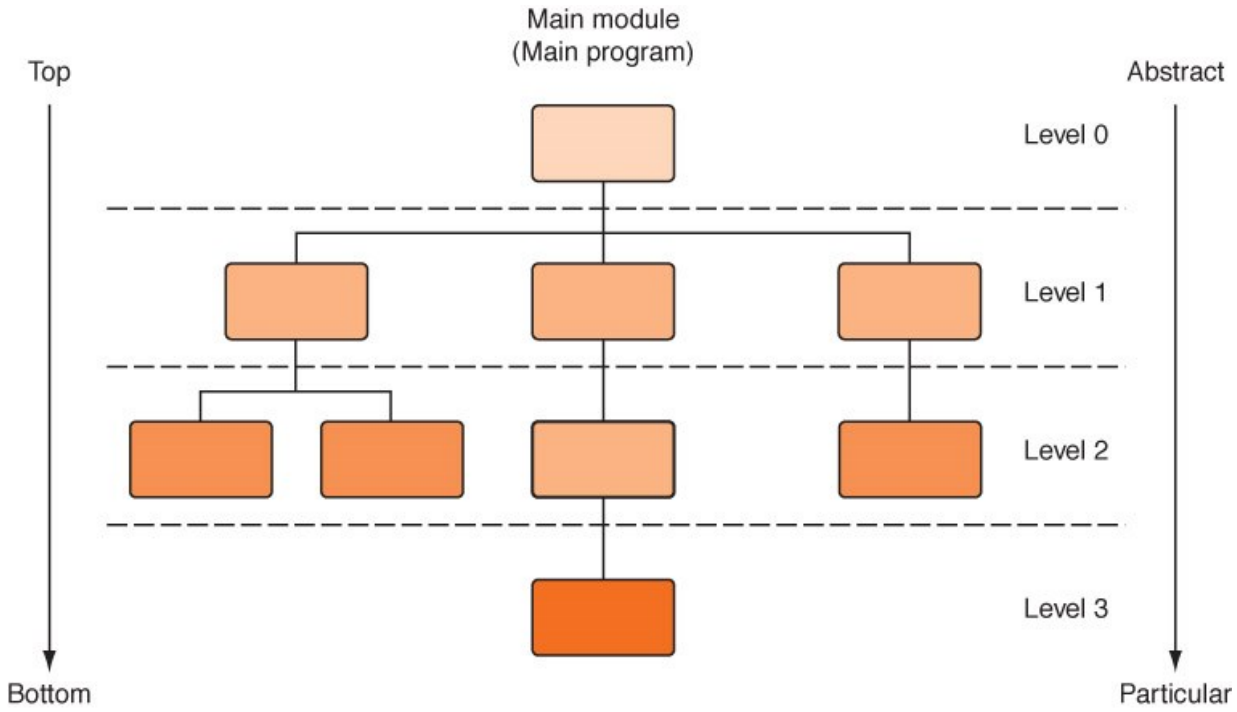
5. Test and verify the completed program

- Test the completed program to verify that it works as desired
- Run the program several times using *different set of data* to make sure that it works correctly for every situation provided for in the algorithm.

6. Maintain and update the program

- Modify a program to remove previously undetected errors and to keep it up-to-date.

Top-Down Design

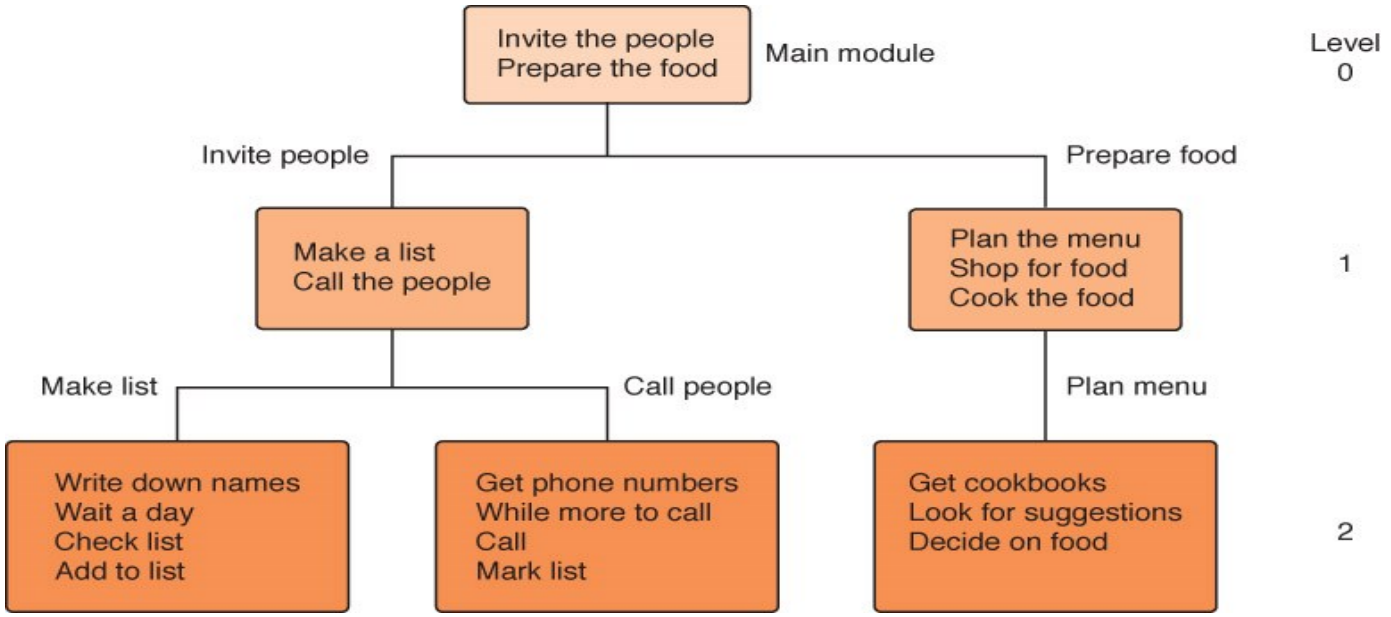


Process continues for as many levels as it takes to make every step concrete

Name of (sub)problem at one level becomes a module at next lower level

A General Example

Planning a large party



Subdividing the party planning

1.5 APPLYING THE SOFTWARE DEVELOPMENT METHOD

CASE STUDY

CONVERTING MILES TO KILOMETERS

1. Problem (Specify the problem requirements)
   - Your summer surveying job requires you to study some maps that gives distances in kilometers and some use miles. You and your coworkers prefer to deal in metric measurements. Write a program that performs the necessary conversion.
2. Analysis (Analyze the problem)
   - Purpose : Conversion from miles to kilometers.
   - To solve this problem, you need to know the relationship between miles and kilometers.
   - Data Requirements
     - Problem input ：miles /*The distances in miles */
     - Problem output：kms /*The distances in kilometers */
     - Relevant Formula : 1 mile = 1.609 kilometers

3. Design (Design the algorithm to solve the problem)
   - Algorithm
     1. Get the distance in miles
     2. Convert the distance to kilometers
     3. Display the distance in kilometers
   - Algorithm with refinements
     1. Get the distance in miles
     2. Convert the distance to kilometers
        2.1 The distance in kilometers is 1.609 times the distance in miles
     3. Display the distance in kilometers
4. Implementation (Figure 1.13)
5. Testing

Figure 1.13 Miles-to-Kilometers Conversion Program

```
1.  /*
2.   * Converts distance in miles to kilometers.
3.   */
4.  #include <stdio.h>              /* printf, scanf definitions */
5.  #define KMS_PER_MILE 1.609      /* conversion constant       */
6.
7.  int
8.  main(void)
9.  {
10.       double miles,  /* input - distance in miles.       */
11.              kms;    /* output - distance in kilometers  */
12.
13.       /* Get the distance in miles. */
14.       printf("Enter the distance in miles> ");
15.       scanf("%lf", &miles);
16.
17.       /* Convert the distance to kilometers. */
18.       kms = KMS_PER_MILE * miles;
19.
20.       /* Display the distance in kilometers. */
21.       printf("That equals %f kilometers.\n", kms);
22.
23.       return (0);
24. }
```

Step 1

Step 2

Step 3

Sample Run

```
Enter the distance in miles> 10.00
That equals 16.090000 kilometers.
```