

Standard Version of Starting Out with C++, 4th Edition

Chapter 1 Introduction to Computers and Programming



Contents

- 1.1 Why Program?
- 1.2 Computer Systems: Hardware and Software
- 1.3 Programs and Programming Languages
- 1.4 What Is a Program Made of?
- 1.5 Input, Processing, and Output
- 1.6 The Programming Process
- 1.7 Procedural and Object-Oriented Programming

1.1 Why Program?

Computer – programmable machine designed to follow instructions

Program – instructions in computer memory to make it do something

Programmer – person who writes instructions (programs) to make computer perform a task

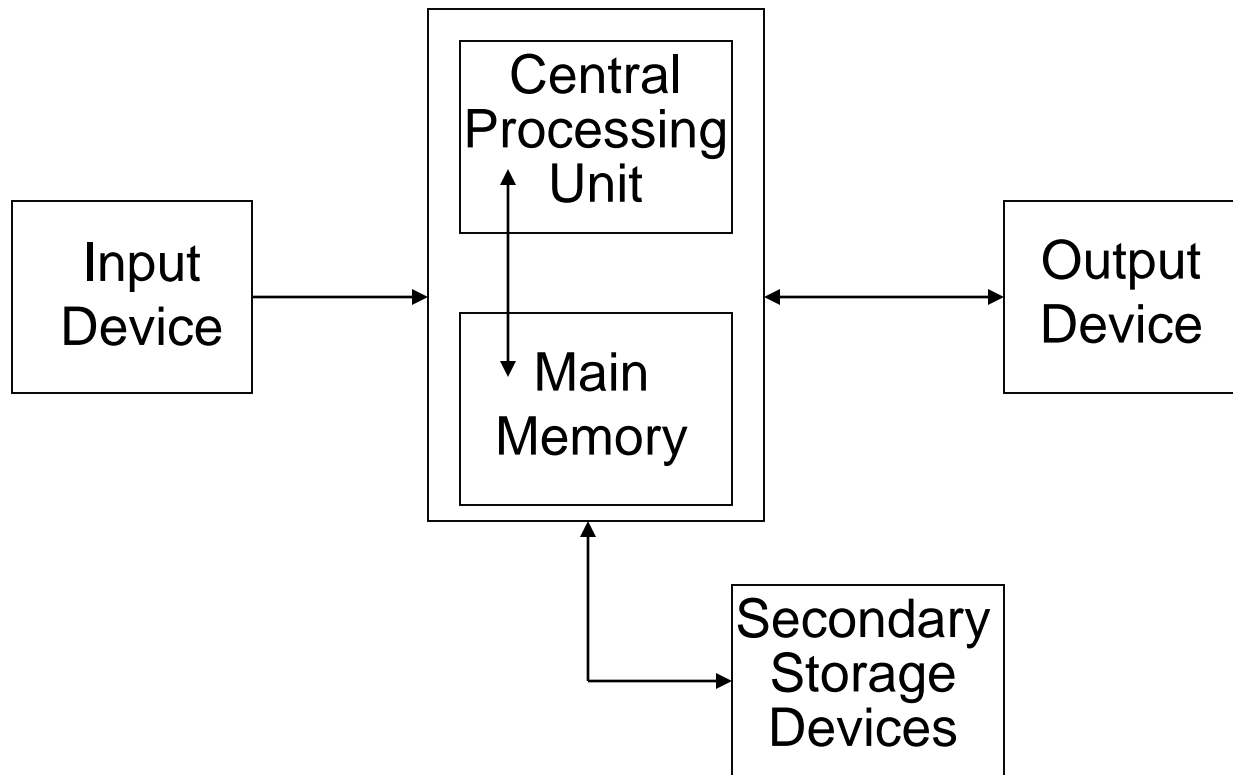
SO, without programmers, no programs; without programs, computer cannot do anything

1.2 Computer Systems: Hardware and Software

Main Hardware Component Categories:

1. Central Processing Unit (CPU)
2. Main Memory
3. Secondary Memory / Storage
4. Input Devices
5. Output Devices

Main Hardware Component Categories



Central Processing Unit (CPU)

Comprised of:

Control Unit

- Retrieves and decodes program instructions

- Coordinates activities of all other parts of computer

Arithmetic & Logic Unit

- Hardware optimized for high-speed numeric calculation

- Hardware designed for true/false, yes/no decisions

Main Memory

- Volatile – erased when program terminates or computer is turned off
- Also called Random Access Memory (RAM)
- Organized as follows:
 - bit: smallest piece of memory. Has values 0 (off, false) or 1 (on, true)
 - byte: 8 consecutive bits. Bytes have addresses.

Secondary Storage

- Non-volatile: data retained when program is not running or computer is turned off
- Comes in a variety of media:
 - magnetic: floppy disk, zip disk, hard drive
 - optical: CD-ROM

Input Devices

- Used to send information to computer from outside
- Many devices can provide input:
 - Keyboard, mouse, scanner, camera
 - Disk drives

Output Devices

- Used for information sent from a computer program
- Many devices can be used for output:
 - Computer monitor and printer
 - Floppy, zip disk drives
 - Writable CD drives

Software – Programs That Run on a Computer

- Categories of software:
 - Operating system: programs that manage the computer hardware and the programs that run on them. Ex: Windows, UNIX, Linux
 - Application software: programs that provide services to the user. Ex: word processing, games, programs to solve specific problems

1.3 Programs and Programming Languages

- Program: a set of instructions to a computer to perform a task
- Programming Language: a language used to write programs

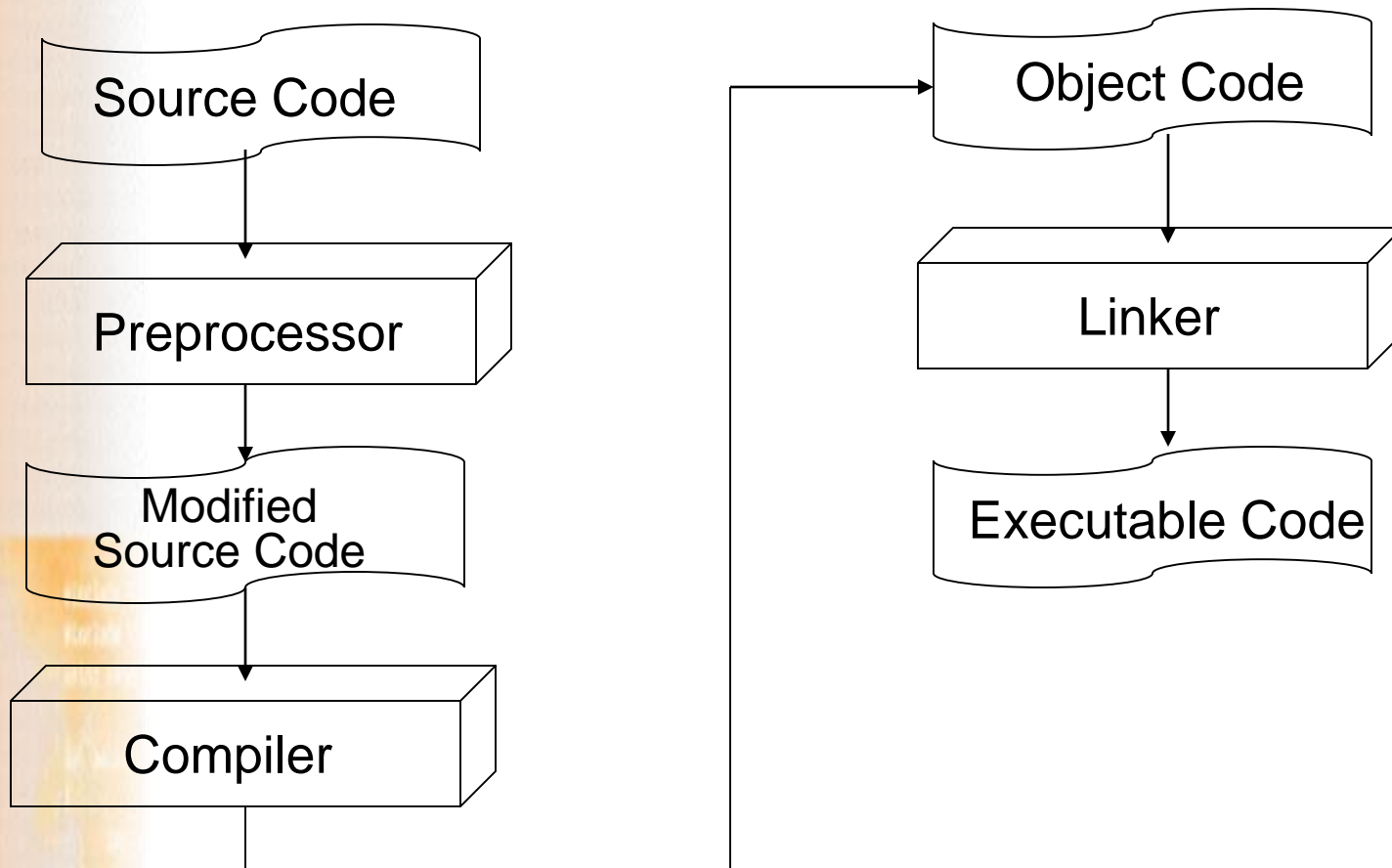
Programs and Programming Languages

- Types of languages:
 - Low-level: used for communication with computer hardware directly. Often written in binary machine code (0's/1's) directly.
 - High-level: closer to human language

From a High-level Program to an Executable File

- a) Create file containing the program with a text editor.
 - b) Run preprocessor to convert source file directives to source code program statements.
 - c) Run compiler to convert source program into machine instructions.
 - d) Run linker to connect hardware-specific code to machine instructions, producing an executable file.
- Steps b–d are often performed by a single command or button click.
 - Errors detected at any step will prevent execution of following steps.

From a High-level Program to an Executable File



1.4 What Is a Program Made Of?

- Common elements in programming languages:
 - Key Words
 - Programmer-Defined Symbols
 - Operators
 - Punctuation
 - Syntax

Example Program

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    string name;
    cout << "What is your name? ";
    cin >> name;
    cout << "Hello there, " << name;
    return 0;
}
```

Key Words

- Also known as reserved words
- Have a special meaning in C++
- Can not be used for another purpose
- Examples in program: `using,`
`namespace, int, main`

Programmer-Defined Symbols

- Names made up by the programmer
- Not part of the C++ language
- Used to represent various things: variables (memory locations), functions, etc.
- Example in program: `name`

Operators

- Used to perform operations on data
- Many types of operators:
 - Arithmetic - ex: $+$, $-$, $*$, $/$
 - Assignment – ex: $=$
- Examples in program: $<<$, $>>$

Punctuation

- Characters that mark the end of a statement, or that separate items in a list
- Examples in program: ;

Syntax

- The rules of grammar that must be followed when writing a program
- Controls the use of key words, operators, programmer-defined symbols, and punctuation

1.5 Input, Processing, and Output

Three steps many programs perform:

1) Gather input data:

- from keyboard
- from files on disk drives

2) Process the input data

3) Display the results as output:

- send it to the screen
- write to a file

1.6 The Programming Process

1. Define the purpose of the program
2. Visualize the running program; sketch the user interface
3. Model the program using design tools
4. Check the model for errors
5. Enter the program into the computer and compile it

The Programming Process

6. Correct errors and recompile until there are none
7. Execute (run) the program
8. Correct execution errors, recompile, rerun
9. Validate results: does the output meet what was defined in step 1?

1.7 Procedural and Object-Oriented Programming

- Procedural programming: focus is on the process. Procedures/functions are written to process data.
- Object-Oriented programming: focus is on objects, which contain data and the means to manipulate the data. Messages sent to objects to perform operations.

Standard Version of Starting Out with C++, 4th Edition

Chapter 1 Introduction to Computers and Programming

