

Chapter 9

Programming

Languages

程式語言



9.1 Source: Foundations of Computer Science © Cengage Learning

Objectives 學習目標

After studying this chapter, students should be able to:

- ❑ Describe the evolution of programming languages from **machine language** to **high-level languages**. 程式語言演進從機器語言至高階語言
- ❑ Understand how a program in **a high-level language** is translated into **machine language**. 高階語言翻譯為機器語言
- ❑ Distinguish between four computer language paradigms. 瞭解四種電腦語言範例: 程序式、物件導向、功能式、宣告式
- ❑ Understand the **procedural paradigm** and the interaction between a program unit and data items in the paradigm.
- ❑ Understand the **object-oriented paradigm** and the interaction between a program unit and objects in this paradigm.
- ❑ Define a **functional paradigm** and understand its applications.
- ❑ Define a **declaration paradigm** and understand its applications.
- ❑ Understand **main program to call subprograms, pass by value and pass by reference**. 主程式呼叫副程式 採用數值或參考位址傳送

9.2

EVOLUTION 電腦語言之演進

To write a program for a computer, we must use a computer language. **A computer language is a set of predefined words that are combined into a program according to predefined rules (*syntax*)** 電腦語言為一組事先定義好的文字，這些關鍵文字依照事先定義好的語法而可組成為程式。Over the years, computer languages have evolved from *machine language* 機器語言 to *high-level languages* 高階語言.

9.3

High-level language

```

/* This program reads two integers from keyboard and prints their sum.
   Written by:
   Date:
*/
#include <iostream.h>
using namespace std;
int main (void)
{
    // Local Declarations
    int number1;
    int number2;
    int result;
    // Statements
    cin >> number1;
    cin >> number2;
    result = number1 + number2;
    cout << result;
    return 0;
} // main

```

Low-level languages

Code in assembly language	Hexadecimal
LOAD RF Keyboard	(1FEF) ₁₆
STORE Number1 RF	(240F) ₁₆
LOAD RF Keyboard	(1FEF) ₁₆
STORE Number2 RF	(241F) ₁₆
LOAD R0 Number1	(1040) ₁₆
LOAD R1 Number2	(1141) ₁₆
ADDI R2 R0 R1	(3201) ₁₆
STORE Result R2	(2422) ₁₆
LOAD RF Result	(1F42) ₁₆
STORE Monitor RF	(2FFF) ₁₆
HALT	(0000) ₁₆

9.4

TRANSLATION 電腦語言的翻譯

Programs today are normally written in one of the high-level languages. To run the program on a computer, **the program needs to be translated into the machine language of the computer on which it will run.** 使用高階語言寫的程式須被翻譯為機器語言才能在電腦上執行. The program in a high-level language is called the **source program**源程式. The translated program in machine language is called the **object program**目標程式. Two methods are used for translation: **compilation**編譯 and **interpretation**直譯.



Compilation 編譯

A **compiler** normally **translates the whole source program into the object program.** 編譯乃為先檢視整個源程式通過語法及語意後，再將整個源程式翻譯為目標程式而執行

Interpretation 直譯

Some computer languages use an **interpreter** to translate the source program into the object program. Interpretation refers to the process of **translating each line of the source program into the corresponding line of the object program and executing the line.** 直譯為每次只對源程式一行檢視語法及語意後，翻譯為目標程式及執行該行. However, we need to be aware of two trends in interpretation: that used by some languages before Java and **the interpretation used by Java.**

9.6

PROGRAMMING PARADIGMS

程式語言範例

Today, computer languages are categorized according to the approach they use to solve a problem. A **paradigm**, therefore, is a way in which a computer language looks at the problem to be solved. **We divide computer languages into four paradigms:** *procedural* 程序語言, *object-oriented* 物件導向語言, *functional* 功能式語言 and *declarative* 宣告式語言.

9.7

Pass by value 主程式呼叫副程式 採用數值傳送

In parameter **pass by value**, the main program and the subprogram create two different objects (variables) 數值傳送為主程式送出參數變數而呼叫副程式，副程式複製一份當區域參數用。The object created in the program belongs to the program and the object created in the subprogram belongs to the subprogram. Since the territory is different, the corresponding objects can have the same or different names. 因兩個變數領域不同，即使同名或異名變數也不影響主參數變數的值。 Communication between the **main program and the subprogram is one-way, from the main program to the subprogram.** 主程式到副程式是單一方向

9.8

Example of Pass-by-value in C

```
main ()
{   int x=10, y=15, z=20;

    PassByValue(x,y,z);
    printf("PassByValue    x=%d, y=%d, z=%d \n",x, y, z);
}
```

```
PassByValue(int x, int y, int z)
{
    y = x + z;
    z = x - y;
    x = y - z;
}
```

9.9

Pass by reference 主程式呼叫副程式 採用參考位址傳送

Pass by reference was devised to **allow a subprogram to change the value of a variable in the main program**. 參考位址傳送為主程式送出參數變數而呼叫副程式，允許副程式改變主程式參數的值。 In pass by reference, the variable, which in reality is a location in memory, is shared by the main program and the subprogram. The same variable may have different names in the main program and the subprogram, but both names refer to the same variable. 主副程式的參數變數均參考到同一位址。 Metaphorically, we can think of pass by reference as a box with two doors: one opens in the main program, the other opens in the subprogram. The main program can leave a value in this box for the subprogram, the subprogram can change the original value and leave a new value for the program in it.

9.10

Example of Pass-by-reference in C

```
main ()
{   int x=10, y=15, z=20;

    PassByReference(&x,&y,&z);
    printf("PassByReference x=%d, y=%d, z=%d \n", x, y, z);
}
```

```
PassByReference(int *x, int *y, int *z)
{
    *y = *x + *z;
    *z = *x - *y;
    *x = *y - *z;
}
```

9.11

Review Questions

- Please distinguish the **low-level** and **high-level languages** and give at least two their programming languages
- Please show the **major difference between interpreter and compiler**.
- Please show the translation process of a computer language.
- Terminologies: **pass by value** and **pass by reference**.
- Determine the result of printf ().

```
main ()
{   int x=3, y=5, z=7;
    PassByMix (x,y,&z);
    printf("x= %d, y= %d, z= %d \n", x, y, z);
}
PassByMix (int x, int y, int *z)
{
    y = *z + x;
    x = *z - y;
    *z = x - y;
}
```

9.12