

# Chapter1：程式語言簡介

## 一、 程式設計的流程

1. Defining the program：定義問題
2. Planning the solution：用演算法寫出解決方案
3. Coding the program：寫出程式碼
4. Testing the program：測試程式
5. Documenting the program：寫程式的發展文件

## 二、 程式語言的分類

1. 機器語言 (Machine Language)：硬體內部所使用的語言，是電腦唯一能直接辨識的語言，通常是由 0 或 1 所組合出的一連串數字所組成。
2. 組合語言 (Assembly Language)：接近機器語言的一種低階語言，是種符號式語言。組合語言必須經由組譯器 (assembler) 轉換成機器語言，才能在電腦上執行。
3. 高階語言 (High-level Language)：一種文法接近日常英文用語的語言，亦包含一般常用的數學運算符號。高階語言的原始程式碼必須轉換為機器語言才能正確執行。轉換程式包含直譯器和編譯器兩種。
4. 非常高階語言 (Very High-level Language)：別稱為第四代語言，如 SQL (Structural Query Language)。

## 三、 直譯器和編譯器的比較

1. 直譯器 (Interpreter)：將原始程式的指令逐翻譯並執行，不需要經過編譯。
  - (1) 優點：佔用的記憶體較少、修改及除錯容易
  - (2) 缺點：每次執行前才翻譯，執行速度慢、效率較低
2. 編譯器 (Compiler)：將原始程式編譯後，先產生目的檔 (.obj)，再將其他要連結的程式連結後，再執行該程式，C 語言即是使用編譯器。
  - (1) 優點：執行時不需要重複編譯，執行速度及效率較高
  - (2) 缺點：原始程式一經過修改就必須重新編譯、較佔用記憶體空間