C++程式設計：函數

**目錄**

[C++程式設計：安裝環境與Bug紀錄 1](#_Toc204167977)

[1. 建立編譯環境 1](#_Toc204167978)

# 基本觀念

函式是組成程式的基本元件。每個程式中至少包含一個主程式（main function），也就是主程序（main program）。

函數可以分為以下兩種：

|  |  |
| --- | --- |
| 系統定義函式 （System-defined Function） | * 由系統或標準函式庫（如 C++ STL）預先定義，使用者直接使用，無須自己撰寫。 * e.g., sqrt() |
| 使用者定義函式 （User-defined Function） | 使用者自行定義與撰寫的函式，依需求設計功能。像是使用者自己撰寫 main() 函式，或自定義一個加法函式。 |

函數整體可以分成五個部分：

* 標頭檔（Header File）
* 函式名稱（Function Name）
* 函式參數（Function Parameter）
* 函式回傳值（Return Value）
* 程式庫（Library）

## 範例：呼叫函數

#include <iostream> *// 標頭檔，用來輸出結果*

#include <cmath> *// 標頭檔，提供 sqrt 函式*

using namespace std;

int main() {

double number = 25.0;

double result;

result = sqrt(number);

cout << "平方根是：" << result << endl; *// 平方根是：5*

return 0;

}

# Function Declaration/Definition/Calls

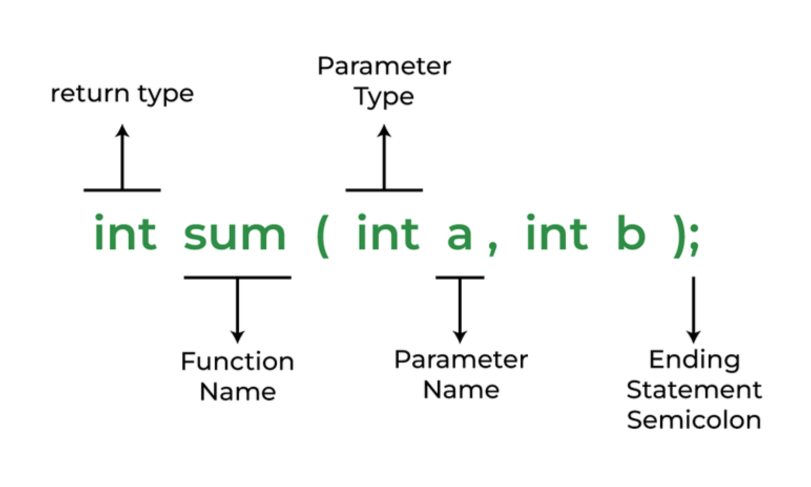
在程式碼段，可以分三個方面：

* Function Declaration 函數聲明
* Function Definition 功能定義
* Function Calls 函數調用

## Function Declaration 函數聲明

Function Declaration是告訴編譯器，程式中某處有一個具有該名稱的函式被定義。在Function Declaration中，我們必須提供函式名稱、其回傳型別，以及參數的數量與型別。

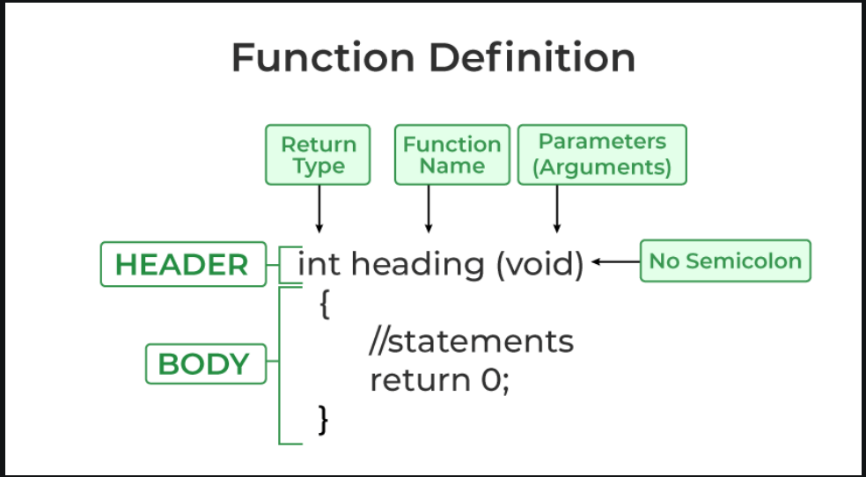
Figure ：Function Declaration



## Function Definition

Function Definition包含實際的敘述（statements），這些敘述會在函式被呼叫時執行（也就是當程式的控制流程進入該函式時）。

Figure ：Function Definition



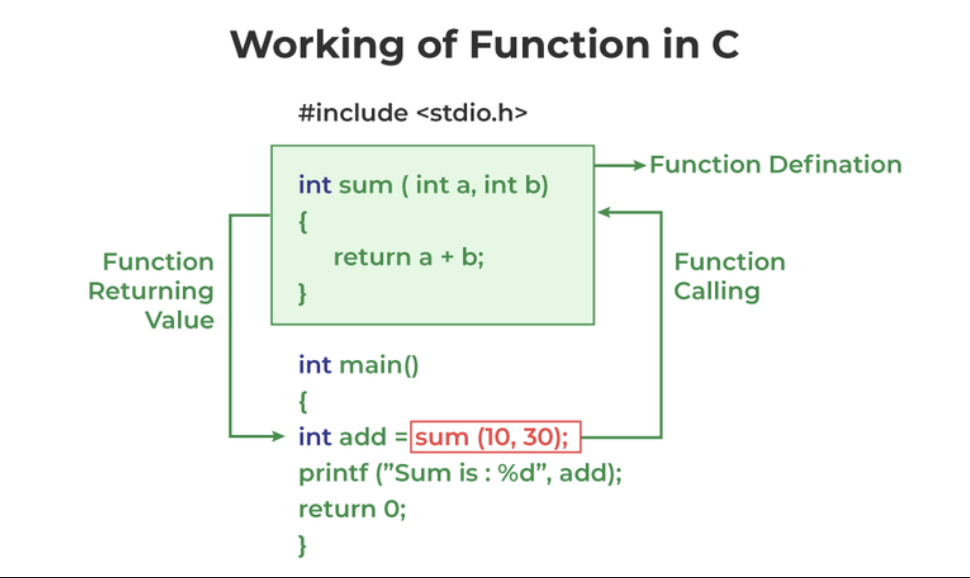
## Function Call

函式呼叫是一種敘述，用來指示編譯器去執行該函式。我們在呼叫時會使用函式名稱與對應的參數。

Q：函式宣告（declaration）與函式定義（definition）有什麼不同？

* 函式宣告包含函式名稱、回傳型別與參數等資訊。
* 函式定義則包含該函式的實際程式區塊（函式本體）

Figure ：Function Call



## 範例：使用Function Declaration、Function Definition跟Function Call。

#include <iostream>

using namespace std;

*// Function Declaration（函式宣告）*

int add(int a, int b);

*// Function Definition（函式定義）：定義函式本體：實際執行加法並回傳結果*

int add(int a, int b) {

return a + b;

}

int main() {

int x = 10;

int y = 20;

int result = add(x, y); *// Function Call（函式呼叫）*

cout << "x + y = " << result << endl; *// x + y = 30*

return 0;

}

## 向前申明(forward declaration)