

第1課：C++程序結構與基本元素

第1課：C++程序結構與基本元素

課程目標

- 了解C++程序的基本結構
- 學習main函數的作用
- 掌握基本的輸出語句
- 理解編譯和執行的過程

本課關鍵字

- int - 整數類型
- main - 主函數
- return - 返回語句
- void - 無類型

範例1：最簡單的C++程序

```
/*
 * 範例1-1：最基本的C++程序
 * 這個程序展示了C++程序的最小結構
 * 只需要一個main函數就能構成有效的C++程序
 */
// 預處理指令：包含輸入輸出庫
#include <iostream>
/**
 * main函數是每個C++程序的入口點
 * 程序從這裡開始執行，也在這裡結束
 *
 * int main() 表示：
 * - int: 函數返回整數類型
 * - main: 函數名稱（固定不變）
 * - (): 參數列表（這裡為空）
 */
int main() {
// std::cout 用於輸出文字到螢幕
// << 是輸出運算符
```

```
// std::endl 表示換行
std::cout << "我的第一個C++程序" << std::endl;
// return 0 表示程序正常結束
// 0 是返回給操作系統的狀態碼
return 0;
}
```

編譯與執行步驟：

1. 保存為 hello.cpp
2. 編譯：g++ hello.cpp -o hello
3. 執行：
 - Windows: hello.exe
 - Linux/Mac: ./hello

範例2：多行輸出與註解

```
/*
 * 範例1-2：多行輸出與註解
 * 展示如何輸出多行文字
 * 並演示不同類型的註解
 */
#include <iostream> // 包含輸入輸出庫
// 單行註解：這是一個單行註解
// 單行註解以 // 開頭，直到行尾
/*
多行註解（區塊註解）
可以跨越多行
以 /* 開始，以 */ 結束
*/
int main() {
// 輸出第一行
std::cout << "歡迎學習C++程式設計" << std::endl;
// 輸出第二行（不加endl不會自動換行）
std::cout << "這是";
std::cout << "同一行文字" << std::endl;
// 輸出第三行
std::cout << "課程大綱：" << std::endl;
// 使用多個 << 運算符連接輸出
```

```
std::cout << "1. 基本語法" << std::endl
<< "2. 變數與數據類型" << std::endl
<< "3. 流程控制" << std::endl;
return 0; // 程序結束，返回0表示成功
}
```

範例3：main函數的不同形式

```
/*
* 範例1-3：main函數的不同形式
* C++允許main函數有不同的參數形式
* 但最常用的是無參數形式
*/
#include <iostream>
// 形式1：最常見的無參數形式
int main() {
    std::cout << "形式1：無參數的main函數" << std::endl;
    return 0;
}
/*
// 形式2：帶參數的main函數（用於接收命令行參數）
int main(int argc, char* argv[]) {
    std::cout << "參數數量：" << argc << std::endl;
    return 0;
}
// 形式3：void參數列表（C風格，不推薦在C++中使用）
int main(void) {
    std::cout << "使用void參數列表" << std::endl;
    return 0;
}
*/
```

注意：通常我們使用無參數的 `int main()` 形式。

範例4：基本數據類型展示

```
/*
* 範例1-4：基本數據類型展示
*/
```

```

* 雖然詳細的數據類型會在下一課講解
* 但這裡先展示一些基本類型
*/
#include <iostream>
int main() {
// 整數類型
int age = 25; // 整數
double height = 175.5; // 雙精度浮點數
char grade = 'A'; // 單個字符
bool isStudent = true; // 布林值 (真/假)
// 輸出變數的值
std::cout << "年齡：" << age << " 歲" << std::endl;
std::cout << "身高：" << height << " 公分" << std::endl;
std::cout << "成績：" << grade << std::endl;
std::cout << "是否學生：" << isStudent << std::endl;
// 布林值會輸出為 1(true) 或 0(false)
// 我們可以使用條件運算符來顯示"是"/"否"
std::cout << "是否學生：" 
<< (isStudent ? "是" : "否")
<< std::endl;
return 0;
}

```

範例5：簡單計算與輸出

```

/*
* 範例1-5：簡單計算與輸出
* 展示如何在輸出中進行計算
*/
#include <iostream>
int main() {
// 定義變數
int a = 10;
int b = 20;
// 輸出基本變數
std::cout << "變數 a = " << a << std::endl;
std::cout << "變數 b = " << b << std::endl;
// 在輸出中進行計算

```

```

std::cout << "a + b = " << (a + b) << std::endl;
std::cout << "b - a = " << (b - a) << std::endl;
std::cout << "a * b = " << (a * b) << std::endl;
std::cout << "b / a = " << (b / a) << std::endl;
// 注意：整數除法會捨棄小數部分
// 10 / 3 會得到 3，而不是 3.333
std::cout << "10 / 3 = " << (10 / 3) << " (整數除法)" << std::endl;
return 0;
}

```

範例6：錯誤示範與常見錯誤

```

/*
* 範例1-6：常見錯誤示範
* 展示初學者常見的錯誤
* 每種錯誤都有註解說明
*/
#include <iostream>
int main() {
// 正確：完整的語句以分號結束
std::cout << "這是正確的語句" << std::endl;
// 錯誤1：缺少分號
// std::cout << "缺少分號" << std::endl // 錯誤！
// 錯誤2：拼寫錯誤
// Std::cout << "S要大寫" << std::endl; // 錯誤！
// std:cout << "少了一個冒號" << std::endl; // 錯誤！
// 錯誤3：使用中文標點
// std::cout << "使用中文分號；" << std::endl; // 錯誤！
// 錯誤4：忘記包含頭文件
// 如果註解掉 #include <iostream>，下面這行會出錯
// 錯誤5：main函數名稱錯誤
// int Main() { ... } // 錯誤！必須是小寫main
// 錯誤6：忘記返回值
// 如果註解掉 return 0;，有些編譯器會警告
return 0;
}

```

練習題

練習1-1：修改輸出

修改範例1-1，讓它輸出你的名字和今天日期。

練習1-2：計算練習

創建一個程序，計算並顯示：

1. $15 + 27$ 的結果
2. $100 - 58$ 的結果
3. 12×13 的結果
4. $144 \div 12$ 的結果

練習1-3：個人資訊

創建一個程序，輸出你的：

1. 姓名
2. 年齡
3. 身高
4. 喜歡的顏色

練習1-4：錯誤修正

以下程序有多個錯誤，請找出並修正：

```
#include <iostrem>
int Main() {
    std::cout << "歡迎來到C++世界"
    return 0
}
```

重點摘要

1. C++程序基本結構

```
預處理指令 (#include)
主函數 (int main())
{
    語句;
    語句;
    ...
    return 0;
}
```

2. 重要規則

- 每個C++程序必須有一個 main 函數
- 語句以分號 ; 結束
- 區分大小寫 (cout ≠ Cout ≠ COUT)
- 使用英文字符 (標點符號必須是英文的)

3. 輸出語句

```
std::cout << "要輸出的內容" << std::endl;
```

- std::cout : 標準輸出物件
- << : 輸出運算符
- std::endl : 換行並刷新緩衝區

4. 註解類型

```
// 單行註解
/*
多行註解
可以跨越多行
*/
```

⌚ 常見問答

Q1: 為什麼要寫 return 0; ?

A: return 0; 告訴操作系統程序正常結束。非0值通常表示錯誤。

Q2: std:: 是什麼意思？

A: std:: 是標準命名空間 (namespace)，cout 和 endl 都定義在這個命名空間中。

Q3: 可以省略 return 0; 嗎？

A: 在 main 函數中可以省略，編譯器會自動添加。但為了清晰，建議明確寫出。

Q4: 為什麼需要 #include <iostream> ?

A: 因為 cout 和 endl 定義在 <iostream> 頭文件中，沒有包含就無法使用。

🚀 下一步

下一課我們將學習：

- 變數與數據類型 (int, double, char, bool 等)
- 變數的聲明與初始化
- 基本輸入操作 (使用 cin)

完整測試程序

```
/*
 * 第1課綜合測試
 * 把本課所有概念整合到一個程序中
 */
#include <iostream> // 必要：包含輸入輸出庫
int main() {
    // 1. 簡單輸出
    std::cout << "==== C++第一課綜合測試 ===" << std::endl;
    // 2. 變數與輸出
    int studentCount = 30;
    std::cout << "班級人數：" << studentCount << " 人" << std::endl;
    // 3. 計算並輸出
    int boys = 18;
    int girls = studentCount - boys;
    std::cout << "男生：" << boys << " 人" << std::endl;
    std::cout << "女生：" << girls << " 人" << std::endl;
    // 4. 百分比計算
    double boyPercentage = (boys * 100.0) / studentCount;
    double girlPercentage = (girls * 100.0) / studentCount;
    std::cout << "男生比例：" << boyPercentage << "%" << std::endl;
    std::cout << "女生比例：" << girlPercentage << "%" << std::endl;
    // 5. 結束訊息
    std::cout << "==== 程序執行完畢 ===" << std::endl;
    return 0; // 程序正常結束
}
```

輸出結果：

```
==== C++第一課綜合測試 ===
班級人數：30 人
男生：18 人
女生：12 人
男生比例：60%
女生比例：40%
==== 程序執行完畢 ===
```

本課檢查清單

- 能寫出完整的C++程序結構
- 理解 `main` 函數的作用
- 會使用 `std::cout` 輸出文字
- 知道語句要以分號結束
- 能區分單行註解和多行註解
- 了解 `return 0;` 的意義
- 能編譯和執行簡單的C++程序

需要幫助？

如果你在編譯或執行時遇到問題：

1. 檢查是否安裝了C++編譯器 (`g++`、`clang++`或Visual Studio)
2. 檢查代碼是否有拼寫錯誤
3. 確保使用英文標點符號
4. 檢查每條語句是否以分號結束

下一課見！ 