

《C++程式設計-教學與自習最佳範本》

勘誤日期：2026/1/21

頁號	位置	原內容	修改後
P1-2	最後一行	然後計算其總	然後計算其總和
P1-4	表 1-1 下面內文第 1 行	你就可以在從容的要給予變數適當的資料型態。	你就可以從容地為變數指定適當的資料型態
P1-5	範例程式 maximum_int.cpp 在第 1 行下面加一敘述	#include <iostream> using namespace std;	#include <iostream> #include <climits> using namespace std;
P1-6	範例程式 minimum_int.cpp 在第 1 行下面加一敘述	#include <iostream> using namespace std;	#include <iostream> #include <climits> using namespace std;
P1-7	範例程式 unsignedInteger.cpp 在第 1 行下面加一敘述	#include <iostream> using namespace std;	#include <iostream> #include <climits> using namespace std;
P1-7	此頁上面範例程式 unsignedInteger.cpp 的第 8 行去掉 << endl。並且刪除行號 9, 其餘行往上提	cout << "maximum of unsigned long int: " << ULONG_MAX << endl << endl;	cout << "maximum of unsigned long int: " << ULONG_MAX << endl;
P1-7	中間哪一段後面	程式分別利用 USHRT_MAX....., int 以及 long int 的最大值。	程式分別利用 USHRT_MAX....., int 以及 long int 的最大值。上述三個程式記得載入 climits 標頭檔。
P1-8	表 1.1 後的那一段後面	你就可以在從容的要給予變數適當的資料型態	你就可以從容地給予變數適當的資料型態
P1-14	倒數第 1 行	值得一提的是，...準確度來得高。	值得一提的是，...準確度來得低。若要解除此設定，則可使用 cout.unsetf(ios::fixed) 函式呼叫來完成。
P1-15	1-6-2 節程式 scientific.cpp 輸出結果下的第 2 行	注意，小數點後面也有 6 位。e+01 表示 10 ¹ 。	注意，小數點後面也有 6 位。e+01 表示 10 ¹ 。若要解除此設定，則可使用 cout.unsetf(ios::scientific) 函式呼叫來完成。

頁號	位置	原內容	修改後
P1-17	1-6-4 節 showpoint 格式調整器，內文的最後加上一段文字	輸出結果將強迫印出小數點及其後面三位是 0 的數字。	輸出結果將強迫印出小數點及其後面三位是 0 的數字。若要解除此設定，則可使用 noshowpoint 格式調整器。
P1-18	1-6-5 節 showpos 格式調整器，程式的輸出結果加上一段文字	123456 +123456 +7890	23456 +123456 +7890 若要解除此設定，則可使用 noshowpos 格式調整器。
P1-19	表 1-2 的 setfill('c') 的時效性	一次性	永久性
P1-22	程式下第 2 行	即美觀又好看	既美觀又好看
P1-27	中間輸出結果第 1 行	Enter your last name: Ming jyh	Enter your first name: Ming jyh
P1-27	中間輸出結果第 2 行	Enter your first name: Tsai	Enter your last name: Tsai
P1-37	習題第 3 題第 1 行位置好像歪了	//exercise1-3.cpp(書上排版有問題)	//exercise1-3.cpp
P3-7	練習題 3.6 倒數第 3 行後面	salary	area
P3-21	練習題 3.16 的(b) else 下的	countet	counter
P3-46	倒數第 2 行前面空一格	是正整數	是正整數
P4-13	範例程式 randNum-1.cpp 第 2 行刪除，改為	#include <iomanip>	#include <cstdlib>
P4-14	範例程式 randNum-2.cpp 第 2 行刪除，改為	#include <iomanip>	#include <cstdlib> #include <ctime>
P4-15	範例程式 randNum3.cpp 第 2 行刪除，改為	#include <iomanip>	#include <cstdlib>
P4-19	練習題 4.10 倒數第 7 行	if (i%2 != 0) continue; }	if (i%2 != 0) { continue; }
P4-25	4-8-2 小節內文第 1 行	每一列印每十個質數	每一列印出十個質數
P4-41	練習題解答 4.11	程式內的內迴圈 for(int i=1; i<=10; i++) { for(int i=1; i<=6; i++) { }	for(int i=1; i<=10; i++) { for(int j=1; j<=6; j++) { }

頁號	位置	原內容	修改後
P4-43	第 1 行	Enter #1 character: pqr	Enter #1 character: pq
P4-49	第 1 行	承第 3 章選擇敘述的習題第 5 題	承第 3 章選擇敘述的習題第 10 題
P5-24	第 12 行	int sum(int, double)	double sum(int, double)
P5-31	程式第 57 行	strcpy(x2.course, x1.course);	strcpy(x1.course, x2.course);
P5-31	程式第 58 行	strcpy(x1.course, course);	strcpy(x2.course, course);
P5-32	輸出結果倒數第 2 行	name = Bright Tsai, course = C++, score = 88.6	name = Bright Tsai, course = C, score = 88.6
P6-7	第 7. 項目裏的程式 第 3 行刪除，與最後 1 行	... int max = myArray[0]; cout << max << endl; int index = 0; for (...) { ... }	... int max = myArray[0]; int index = 0; for (...) { ... }
P6-7	倒數第 2 行	此程式輸出結果是 2	此程式輸出結果索引是 2
P6-7	倒數第 1 行	試問結果會是如何?答案是 5	其結果會是 5
P6-7	倒數第 1 行	答案是 5	答案是 9 和 5
P6-9	第 10. 項目裏的程 式第 3 行移	cout << "Enter a seasons number (1~12): ";	cout << "Enter a seasons number (1~4): ";
P6-10	第 3 行下面的程式	for (int x=0; x<=SIZE; x++) { cout << arr[x] << " ";	for (int i=0; i<=arraySize; i++) { cout << myArray[i] << " ";
P6-15	範例程式 lottoNum- 1.cpp 在第 2 行後面 加兩行	#include <iomanip>	#include <iomanip> #include <cstdlib> #include <ctime>
P6-19	範例程式第 22 行	}4	}
P6-24	範例程式 binarySearch.cpp 第 11 行	cout << "Data array: " << endl;	cout << "dataArr[]: " << endl;
P6-24	範例程式 binarySearch.cpp 第 15 行	cout << "\n\dataArr size is : " << size << endl;	cout << "\n\dataArr[] size is : " << size << endl;
P6-34	程式第 04 行 main() 括號不要有空白	int main()	int main()
P6-38	程式第 98 行往右移 4 格	return 0;	return 0;
P6-44	6.5 (e)	double max = arr2[0] for (int i = 1; i < 10; i++) { if (arr2[i] > max) { max = list[i]; } }	double min = arr2[0] for (int i = 1; i < 10; i++) { if (arr2[i] < min) { min = arr2[i]; } }
P9-6	第 11 行後面	若指標變數為，它以 NULL 表示 之。	若指標變數為空值，則以 NULL 表示之。

頁號	位置	原內容	修改後
P9-10	第 1 行	避免此錯誤的宣告是，使用同義型態 <code>intPointer</code> 型態，如下所示:	要避免此錯誤，請使用同義型態 <code>intPointer</code> ，如下所示
P9-14	中間內文	看那一個較方便	看哪一個較方便
P10-8	圖 10-3 的方法	//方法 <code>Circle(int x=1)</code>	//方法 <code>+Circle(int x=1)</code>
P10-11	練習題 10.1 第 5 行	<code>c4.getRadius()</code>	<code>c3.getRadius()</code>
P10-12	練習題 10.2 (b) 第 5 行	<code>c5.getRadius()</code>	<code>c6.getRadius()</code>
P10-12	倒數第 6 行	<code>cout << "radius = " << c.radius;</code>	<code>cout << "radius = " << c2.radius;</code>
P10-21	圖 10-4 中的最後一行	<code>+show(): vod</code>	<code>+show(): void</code>
P10-28	圖 10-4 的方法	//方法 <code>Rectangle(int len=1, int wid=1)</code> <code>~Rectangle()</code>	//方法 <code>+Rectangle(int len=1, int wid=1)</code> <code>+~Rectangle()</code>
P10-25	10-6 小節內文第 3 行	<code>className(class className&)</code>	<code>className(const className&)</code>
P10-28	圖 10-5 的方法	//方法 <code>Course(string, int)</code> <code>~Course()</code> <code>insertStudent(string):void</code> <code>printStudent(): void</code>	//方法 <code>+Course(string, int)</code> <code>+~Course()</code> <code>+insertStudent(string):void</code> <code>+printStudent(): void</code>
P10-28	圖 10-6 的方法，加上+和去掉;	//方法 <code>Student();</code> <code>Student(string, double);</code> <code>largest(Student&): Student&</code> <code>show(): void</code>	//方法 <code>+Student()</code> <code>+Student(string, double)</code> <code>+largest(Student&): Student&</code> <code>+show(): void</code>
P11-15	倒數第 5 行	<code>friend ostream& operator</code> <code>>>(ostream&, Rational&);</code>	<code>friend istream& operator</code> <code>>>(istream&, Rational&);</code>
P12-4	倒數第 3 行	現將上一章的討論的 <code>Circle</code> 類別	現將上一章討論的 <code>Circle</code> 類別
P12-29	第 3 行	隱秘性	隱密性
P12-39	圖 12-5 右邊	<code>C.derived</code>	<code>C_derived</code>
P12-39	圖 12-6 標題刪除類別兩個字	<code>A_base</code> 類別	<code>A_base</code> 類別
P12-40	圖 12-9 中間	<code>#dData</code>	<code>-dData</code>
P12-65	圖 12-11 標題刪除類別兩個字	<code>Person</code> 類別	<code>Person</code> 類別
P12-65	圖 12-12 標題修改	<code>Staff</code> 類別: <code>Person</code> 類別	<code>Staff: public Person</code>
P12-66	圖 12-13 標題修改	<code>Faculty</code> 類別: <code>Person</code> 類別	<code>Faculty: public Person</code>
P12-67	圖 12-15 標題刪除類別兩個字	<code>Person</code> 類別	<code>Person</code> 類別

P12-67	圖 12-16 標題修改	Student 類別	Student: public Person
P12-67	圖 12-17 標題修改	Professor 類別	Professor: public Person
P13-22	倒數第 2 行中間	建構函式的處理順序是對	建構函式的處理順序是對的
P13-47	圖 13-5 標題刪除類別兩個字	Triangle 類別	Triangle: public Point
P13-47	圖 13-5 內的加上+	<pre>//方法 Triangle(int x, int y, string color, string filled, int s1=1, int s2=1, int s3=1) //seter getS1(): int getS2(): int getS3(): int //setter setS1(int): void setS2(int): void setS3(int): void getPerimeter(): double getArea(): double</pre>	<pre>//方法 +Triangle(int x, int y, string color, string filled, int s1=1, int s2=1, int s3=1) //seter +getS1(): int +getS2(): int +getS3(): int //setter +setS1(int): void +setS2(int): void +setS3(int): void +getPerimeter(): double +getArea(): double</pre>
P14-2	內文倒數第 1 行	catch 補捉此訊息	catch 捕捉此訊息
P14-3	中間內文第 3 行	補捉	捕捉
P14-14	倒數第 3 行	有補捉	有捕捉
P14-19	第 3 行	它是預定的異常補捉器	它是預定的異常捕捉器
P14-20	倒數第 8 行	異常補捉器	異常捕捉器
P14-20	倒數第 2 行	異常補捉器	異常捕捉器
P14-22	第 1 行	異常補捉器	異常捕捉器
P14-22	中間第 1 行	異常補捉器	異常捕捉器
P14-22	範例程式第 10 行	異常在 test()函式被補捉	異常在 test()函式被捕捉
P14-22	範例程式第 22 行	異常在 test()函式被補捉	異常在 test()函式被捕捉
P14-23	範例程式的輸出結果第 1 行	異常在 test()函式被補捉	異常在 test()函式被捕捉
P14-23	範例程式的輸出結果第 3 行	異常在 test()函式被補捉	異常在 test()函式被捕捉

- P12-68 圖 12-18，請以下圖取代

TA : public Student, public Professor	說明
-course: double	助教科目
+TA() +show()	TA 類別建構函式 顯示 TA 物件的資訊

圖 12-18 TA 類別的 UML 類別圖

● P1-41 習題第 10 題程式改為

10. 試問下一程式的輸出結果為何?

```
//exercise1-10.cpp
#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
    int a = 100;
    int b = 0x65;
    cout << setbase(16) << a << endl;
    cout << setbase(8) << a << endl;
    cout << setbase(10) << b << endl << endl;
    cout << hex << a << endl;
    cout << oct << a << endl;
    cout << dec << b << endl;
    return 0;
}
```

● P1-41 習題增加一題，第 11 題題目為

11. 試問下一程式的輸出結果為何?

```
//exercise1-11.cpp
#include <iostream>
using namespace std;

int main()
{
    int i = 123;
    double j = 123, k = 456;
    cout << showpos;
    cout << "#1: " << i << endl;
    cout << "#2: " << j << endl;
    cout << "#3: " << noshowpos << i << endl;
    cout << showpoint;
    cout << "#4: " << j << endl;
    cout << "#5: " << k << endl;
    cout << "#6: " << noshowpoint << j << endl;
    cout << "#7: " << fixed << j << endl;
    cout << "#8: " << k << endl;
    cout.unsetf(ios::fixed);
    cout << "#9: " << j << endl;
    return 0;
}
```

}

● P2-16 習題第 4 題的題目和輸出結果修正如下:

假設您每個月在戶頭裡存\$10000，銀行年利率為 8%。也就是說，月利率為 $0.08 / 12 = 0.00666667$ 。一個月過後，戶頭裡的存款會是：

$$10000 * (1 + 0.00666667) = 10066.666667$$

兩個月過後，戶頭裡的存款會變成：

$$(10000 + 10066.666667) * (1 + 0.00666667) = 20200.444444$$

三個月過後，戶頭裡的存款則變成：

$$(10000 + 20200.444444) * (1 + 0.00666667) = 30401.780741$$

依此類推。

請撰寫一程式，提示使用者輸入每個月存進戶頭的款項，並顯示前六個月帳戶裡的金額。以下是其執行結果(因為浮點數預設輸出的有效位數共六位，所以只到小數點後 1 位)：

月利率: 0.00666667
輸入每月存多少金額: 10000

一個月後帳戶金額: 10066.7
二個月後帳戶金額: 20200.4
三個月後帳戶金額: 30401.8
四個月後帳戶金額: 40671.1
五個月後帳戶金額: 51008.9
六個月後帳戶金額: 61415.7

● P4-49 習題第 7 題的題目修正如下:

假設您每個月在戶頭裡存\$10000，銀行年利率為 8%。也就是說，月利率 $0.08 / 12 = 0.00666667$ 。一個月過後，戶頭裡的存款大約會是：

$$10000 * (1 + 0.00666667) = 10066.666667$$

兩個月過後，戶頭裡的存款會變成：

$$(10000 + 10066.666667) * (1 + 0.00666667) = 20200.444444$$

三個月過後，戶頭裡的存款則變成：

$$(10000 + 20200.444444) * (1 + 0.00666667) = 30401.780741$$

依此類推。

此題目與第 2 章運算子的習題第 4 題相同，但請利用迴圈敘述撰寫之，先提示使用者輸入要存幾個月，年利率是多少，輸入每個月存入的款項，然後顯示一年中每個月後戶頭裡的金額(取小數點後 3 位數)。輸出結果如下：

- 內文中程式的修改之處

p10-28 將範例程式：`copyConstructor-1.cpp` 以下面程式取代

```
//copyConstructor-1.cpp
#include <iostream>
#include <cstring>
using namespace std;

class Course{
private:
    string courseName;
    int size;
    string *students;
    int numberOfStudents;

public:
    Course(string, int);
    ~Course();
    void insertStudent(string);
    void printStudent();
};

Course::Course(string courseName2, int size2)
{
    numberOfStudents = 0;
    courseName = courseName2;
    size = size2;
    students = new string[size2];
}

Course::~~Course()
{
    delete [] students;
}

void Course::insertStudent(string name2)
{
    if (numberOfStudents < size) {
        students[numberOfStudents] = name2;
        numberOfStudents++;
    }
    else {
        cout << "超過人數" << endl;
        exit(1);
    }
}
```

```

void Course::printStudent()
{
    cout << "course name: " << courseName << endl;
    cout << "capacity: " << size << endl;
    cout << "number of students: " << numberOfStudents << endl;
    for (int i=0; i<numberOfStudents; i++) {
        cout << students[i] << endl;
    }
}

```

```

int main()
{
    Course course1("C++ Programming", 20);
    course1.insertStudent("Bright");
    course1.insertStudent("Jennifer");
    course1.printStudent();
    cout << endl;

    Course course2(course1);
    course2.printStudent();

    return 0;
}

```

~~~output~~~

```

course name: C++ Programming
capacity: 20
number of students: 2
Bright
Jennifer

```

```

course name: C++ Programming
capacity: 20
number of students: 2
Bright
Jennifer

```

//////

P10-30 將範例程式：[copyConstructor-2.cpp](#)以下面程式取代

```

//copyConstructor-2.cpp
#include <iostream>
#include <cstring>
using namespace std;

class Course{
private:
    string courseName;
    int size;
    string *students;
    int numberOfStudents;

```

```

public:
    Course(string, int);
    ~Course();
    Course(const Course &);
    void setCourseName(string);
    void insertStudent(string);
    void printStudent();
};

Course::Course(string courseName2, int size2)
{
    numberOfStudents = 0;
    courseName = courseName2;
    size = size2;
    students = new string[size2];
}

Course::~Course()
{
    delete [] students;
}

void Course::setCourseName(string courseName2)
{
    courseName = courseName2;
}

//客製化複製建構函式
Course::Course(const Course &st)
{
    courseName = st.courseName;
    numberOfStudents = 0;
    size = st.size;
    students = new string[size];
}

void Course::insertStudent(string name2)
{
    if (numberOfStudents < size) {
        students[numberOfStudents] = name2;
        numberOfStudents++;
    }
    else {
        cout << "超過人數" << endl;
        exit(1);
    }
}

void Course::printStudent()
{
    cout << "course name: " << courseName << endl;
    cout << "capacity: " << size << endl;
    cout << "number of students: " << numberOfStudents << endl;
}

```

```
        for (int i=0; i<numberOfStudents; i++) {
            cout << students[i] << endl;
        }
    }

int main()
{
    Course course1("C++ Programming", 20);
    course1.insertStudent("Bright");
    course1.insertStudent("Jennifer");
    course1.printStudent();
    cout << endl;

    Course course2(course1);
    course2.setCourseName("Python programming");
    course2.insertStudent("Linda");
    course2.insertStudent("Amy");
    course2.insertStudent("Chloe");
    course2.printStudent();

    return 0;
}
```

~~~output~~~

course name: C++ Programming
capacity: 20
number of students: 2
Bright
Jennifer

course name: Python programming
capacity: 20
number of students: 3
Linda
Amy
Chloe