

目錄

Contents

大道之行也（侯捷 譯序）	001
導讀（譯者的話）	003
翻譯風格（技術引導乎 文化傳承乎）	003
原文術語的採用原則	013
中英術語對照（按字母順序排列）	014
原書勘誤和修正	018
版面字形風格	018
範例程式注意事項	019
關於索引	019
重要術語的技術意義（各章重點整理與導讀）	020
目錄	i
前言	xiii
本書結構	xiv
第三版的變化	xviii
C++ 的未來	xix
致謝	xx
第二版致謝	xx
參考書目（Bibliography）	xxi
第一篇：C++ 概觀	001
第 1 章 起步走	005
1.1 問題的解決	005
1.2 C++ 程式	006
1.2.1 程式的控制流程	011
1.3 前置處理器指令（Preprocessor Directives）	013
1.4 關於註解（comments）	017
1.5 初嚐輸入和輸出（Input/Output）	019
1.5.1 檔案的輸入和輸出	021
第 2 章 縱覽 C++	023
2.1 語言內建的陣列資料型別	023
2.2 動態記憶體配置與指標	026
2.3 以物件為基礎的設計（An Object-Based Design）	030
2.4 物件導向設計（An Object-Oriented Design）	041
2.5 泛型設計（A Generic Design）	051
2.6 以「異常處理」為基礎的設計（An Exception-Based Design）	058
2.7 以任何其他名稱對 Array 命名	062

2.8 標準的 Array 其實是個 Vector.....	068
第二篇：C++ 語言的基本要素.....	073
第3章 C++ 的資料型別.....	075
3.1 字面常數 (Literal Constant)	075
3.2 變數 (Variables)	079
3.2.1 什麼是變數？.....	081
3.2.2 變數名稱.....	083
3.2.3 Object 的定義式 (Definition)	085
3.3 指標型別 (Pointer Types)	088
3.4 字串型別 (String Types)	093
3.4.1 C-Style 字元字串.....	093
3.4.2 string 型別.....	096
3.5 const 修飾詞.....	102
3.6 Reference 型別.....	105
3.7 bool 型別.....	110
3.8 Enumeration (列舉) 型別.....	111
3.9 陣列 (Arrays) 型別.....	114
3.9.1 多維度 (Multidimensional) 陣列.....	117
3.9.2 陣列型別與指標型別的關係.....	118
3.10 vector Container 型別.....	121
3.11 complex (複數) 型別.....	125
3.12 以 typedef 完成的名稱.....	126
3.13 volatile 修飾詞.....	127
3.14 pair 型別.....	128
3.15 Class 型別.....	129
第4章 算式 (Expressions)	141
4.1 什麼是算式 (Expression) ?	141
4.2 算術運算子 (Arithmetic Operators)	143
4.3 Equality (相等)、Relational (關係)、Logical (邏輯) 運算子.....	146
4.4 Assignment (指派) 運算子.....	149
4.5 Increment (遞增) 和 Decrement (遞減) 運算子.....	154
4.6 複數運算 (Complex Number Operations)	155
4.7 條件運算子 (Conditional Operator)	159
4.8 sizeof 運算子.....	160
4.9 new 算式和 delete 算式.....	162
4.10 逗號 (Comma) 運算子.....	164
4.11 位元操作 (Bitwise) 運算子.....	164
4.12 bitset 操作行為.....	168
4.13 優先權 (Precedence)	172
4.14 型別轉換.....	175
4.14.1 隱式型別轉換 (Implicit Type Conversions)	176
4.14.2 算術轉換 (Arithmetic Conversions)	177
4.14.3 顯式型別轉換 (Explicit Conversions)	179
4.15 一個 Stack Class 實例.....	185

第 5 章 述句 (Statements)	189
5.1 簡單述句和複合述句 (Simple Statement and Compound Statement)	189
5.2 宣告式 (Declaration) 述句.....	191
5.3 if 述句.....	194
5.4 switch 述句.....	202
5.5 for 迴圈述句.....	210
5.6 while 述句.....	214
5.7 do while 述句.....	216
5.8 break 述句.....	218
5.9 continue 述句.....	220
5.10 goto 述句.....	220
5.11 一個 Linked List (串列) 實例.....	222
5.11.1 供應一個一般化的 List Class.....	242
第 6 章 抽象的 Container (容器) 型別.....	249
6.1 我們的文字查詢系統.....	250
6.2 使用一個 vector 或是一個 list?	254
6.3 vector 如何自我成長.....	256
6.4 定義一個 Sequence Container (循序容器)	260
6.5 Iterators (迭代器、迭代指標、泛型指標)	265
6.6 Sequence Container (循序容器) 的操作行為	269
6.6.1 刪除 (Deletion)	270
6.6.2 指派 (Assignment) 與置換 (Swap)	271
6.6.3 泛型演算法 (Generic Algorithms)	272
6.7 儲存每一行文字	273
6.8 找出一個子字串.....	276
6.9 標點符號的處理.....	282
6.10 面對任何其他型式的字串	285
6.11 其他的字串操作行為.....	288
6.12 建立一個 Text Location Map (文字/位置 對映集)	294
6.12.1 定義一個 Map 並植入元素.....	295
6.12.2 搜尋並取出 Map 的某個元素.....	299
6.12.3 巡訪整個 Map.....	300
6.12.4 一個 Word Transformation Map (單字轉換對映集)	301
6.12.5 從 Map 之中移除 (Erasing) 元素.....	304
6.13 建立一個 Word Exclusion Set (單字排除集合)	305
6.13.1 定義一個 Set 並植入內容.....	305
6.13.2 搜尋某個元素.....	307
6.13.3 巡訪整個 Set.....	307
6.14 完整程式.....	308
6.15 Multimap 和 Multiset.....	318
6.16 Stack.....	321
6.17 FIFO Queue 和 Priority Queue.....	323
6.18 重訪我們的 iStack Class.....	324

第三篇：以程序為基礎的程式設計（Procedural-Based Programming）	329
第 7 章 函式（Functions）	331
7.1 概觀	331
7.2 函式原型（Function Prototype）	334
7.2.1 函式回返值型別（Function Return Type）	334
7.2.2 函式的參數列（Function Parameter List）	335
7.2.3 參數型別檢驗（Parameter Type-Checking）	336
7.3 引數的傳遞（Argument Passing）	338
7.3.1 Reference 參數	340
7.3.2 Reference 參數和 Pointer（指標）參數之間的關係	343
7.3.3 陣列參數（Array Parameters）	346
7.3.4 以抽象的 Container 型別做為參數	349
7.3.5 預設引數（Default Arguments）	351
7.3.6 簡略符號（Ellipses）	353
7.4 傳回一個數值	356
7.4.1 參數與傳回值 v.s. Global Objects	359
7.5 遞迴（Recursion）	361
7.6 Inline 函式	363
7.7 聯結指示（Linkage Directives）：extern "C" A	364
7.8 main() 如何處理命令列選項（Command Line Options） A	367
7.8.1 一個與命令列（Command Line）相關的 Class	376
7.9 函式指標	378
7.9.1 「函式指標」型別	378
7.9.2 初始化動作（Initialization）與指派動作（Assignment）	380
7.9.3 函式指標的喚起（Invocation）	380
7.9.4 以函式指標構成的陣列	381
7.9.5 參數與傳回值的型別	383
7.9.6 指向 extern "C" 函式	386
第 8 章 生存空間與生命週期（Scope and Lifetime）	389
8.1 Scope（生存空間）	389
8.1.1 Local Scope（區域性生存空間）	391
8.2 全域物件（Global Objects）和全域函式（Global Functions）	395
8.2.1 宣告（Declarations）和定義（Definitions）	395
8.2.2 檔案間的宣告契合	397
8.2.3 談談表頭檔	398
8.3 Local Objects（區域性物件）	402
8.3.1 Automatic Objects（自動配置與釋放的物件）	402
8.3.2 Register Automatic Objects（放在暫存器裡的物件）	403
8.3.3 Static Local Objects（靜態的區域性物件）	404
8.4 動態配置的物件（Dynamically Allocated Objects）	405
8.4.1 單一物件的動態配置與釋放	406
8.4.2 auto_ptr A	409
8.4.3 陣列的動態配置與釋放	414
8.4.4 常數物件（const Objects）的動態配置與釋放	416

8.4.5 Placement New 算式 A.....	417
8.5 Namespace (命名空間) 的定義.....	420
8.5.1 Namespace 的定義.....	421
8.5.2 Scope 運算子 (Scope Operator, ::)	424
8.5.3 巢狀 (Nested) Namespaces.....	426
8.5.4 Namespace 成員的定義.....	428
8.5.5 ODR (一次定義規則) 和 Namespace 成員.....	430
8.5.6 無具名的 Namespaces.....	431
8.6 使用 Namespace 的成員.....	434
8.6.1 Namespace Aliases.....	434
8.6.2 Using Declarations.....	435
8.6.3 Using Directives.....	437
8.6.4 標準的 Namespace std.....	440
第 9 章 多載化函式 (Overloaded Functions)	443
9.1 多載化函式 (Overloaded Function) 的宣告.....	443
9.1.1 為什麼要將函式名稱多載化?	444
9.1.2 如何將一個函式名稱多載化.....	444
9.1.3 什麼時候不需要將函式多載化?	447
9.1.4 多載化與 Scope 的關係 A.....	448
9.1.5 extern "C" 和多載化函式 A.....	452
9.1.6 多載化函式的指標 A.....	453
9.1.7 Type-Safe Linkage (保證型別安全的聯結方式) A.....	454
9.2 多載決議程序 (Overload Resolution) 的三個步驟.....	456
9.3 引數型別轉換 A.....	458
9.3.1 「完全吻合 (Exact Match)」細節探討.....	460
9.3.2 晉升轉換 (Promotion) 的細節.....	465
9.3.3 標準轉換 (Standard Conversion) 的細節.....	467
9.3.4 References 的影響.....	471
9.4 函式多載決議程序 (Function Overload Resolution) 細節 A.....	474
9.4.1 候選函式 (Candidate Functions)	474
9.4.2 可行函式 (Viable Functions)	479
9.4.3 最佳可行函式 (Best Viable Function)	481
9.4.4 預設引數 (Default Arguments)	486
第 10 章 Function Templates (函式範本)	489
10.1 Function Template 的定義.....	489
10.2 Function Template 具現化 (Instantiation)	497
10.3 Template 的引數推導 (Argument Deduction) A.....	500
10.4 明白指定的 Template 引數 (Explicit Template Arguments) A.....	505
10.5 Template 編譯模型 (Compilation Models) A.....	509
10.5.1 Inclusion (含入式) 編譯模型.....	509
10.5.2 Separation (分離式) 編譯模型.....	510
10.5.3 明白的具現化宣告 (Explicit Instantiation Declarations)	512
10.6 Template Explicit Specialization (Template 明白特製體) A.....	514

10.7 將 Function Templates 多載化 A.....	520
10.8 多載決議程序 (Overload Resolution) 與 template 具現體 (Instantiations) A.....	522
10.9 Template 定義式中的名稱決議過程 (Name Resolution) A.....	531
10.10 Namespaces (命名空間) 和 Function Templates A.....	538
10.11 Function Template 實例.....	542
第 11 章 異常處理 (Exception Handling)	547
11.1 丟出一個異常 (Throwing an Exception)	547
11.2 Try Block.....	551
11.3 捕捉 (Catching) Exception.....	555
11.3.1 Exception Objects.....	556
11.3.2 Stack Unwinding (堆疊輾轉開解)	559
11.3.3 重新丟出 (Rethrow) exception.....	560
11.3.4 Catch-All 處理常式.....	562
11.4 Exception Specifications (異常規格)	564
11.4.1 Exception Specifications (異常規格) 和函式指標.....	567
11.5 Exceptions (異常) 與設計上的論點.....	568
第 12 章 泛型演算法 (The Generic Algorithms)	571
12.1 概觀.....	571
12.2 使用泛型演算法 (Generic Algorithms)	575
12.3 Function Objects.....	586
12.3.1 預先定義的 Function Objects.....	588
12.3.2 算術類 (Arithmetic) Function Objects.....	590
12.3.3 相對關係類 (Relational) Function Objects.....	591
12.3.4 邏輯類 (Logical) Function Objects.....	592
12.3.5 作用於 Function Objects 身上的所謂 Function Adaptors.....	592
12.3.6 實作一個 Function Object.....	593
12.4 重訪 Iterators (泛型指標)	594
12.4.1 Insert Iterators.....	595
12.4.2 Reverse Iterators.....	597
12.4.3 iostream Iterators.....	598
12.4.4 istream iterator.....	599
12.4.5 ostream iterator.....	600
12.4.6 五種 Iterators.....	601
12.5 泛型演算法 (The Generic Algorithms)	603
12.5.1 搜尋 (Search) 演算法.....	604
12.5.2 排序 (Sorting) 和次序整理 (Ordering) 演算法.....	604
12.5.3 刪除 (Deletion) 及取代 (Substitution) 演算法.....	605
12.5.4 排列組合 (Permutation) 演算法.....	605
12.5.5 數值 (Numeric) 演算法.....	605
12.5.6 生成 (Generation) 和質變 (Mutation) 演算法.....	606
12.5.7 相對關係 (Relational) 演算法.....	606
12.5.8 集合 (Set) 演算法.....	606
12.5.9 堆積 (Heap) 演算法.....	606

12.6 什麼時候不適合使用泛型演算法？	606
12.6.1 list::merge()	607
12.6.2 list::remove()	608
12.6.3 list::remove_if()	608
12.6.4 list::reverse()	608
12.6.5 list::sort()	608
12.6.6 list::splice()	609
12.6.7 list::unique()	609
第四篇：以物件為基礎的程式設計（Object-Based Programming）	611
第 13 章 Classes（類別）	613
13.1 Class 定義式（Definition）	614
13.1.1 Data Members（資料成員）	614
13.1.2 Member Functions（成員函式）	616
13.1.3 Member Access（成員的存取）	617
13.1.4 Friends	619
13.1.5 Class 的宣告（declarations）與定義 definitions）	620
13.2 Class Objects	621
13.3 Class Member Functions（類別的成員函式）	624
13.3.1 Inline / Non-Inline Member Functions	625
13.3.2 存取 Class Members	626
13.3.3 Private / Public Member Functions	628
13.3.4 特殊的 Member Functions	630
13.3.5 const/volatile Member Functions	631
13.3.6 Mutable（易變的）Data Members	634
13.4 隱含的 this 指標	636
13.4.1 何時使用 this 指標	638
13.5 Static Class Members	641
13.5.1 Static Member Functions	646
13.6 指標，指向 Class Member	649
13.6.1 Class Member 的型別	651
13.6.2 使用 Class Member 指標	654
13.6.3 Static Class Members 的指標	656
13.7 Union：一種節省空間的 Class	658
13.8 位元欄位（Bit-field）：一種節省空間的 Member	663
13.9 Class Scope A	665
13.9.1 Class Scope 內的名稱決議過程（Name Resolution）	669
13.10 Nested Classes（巢狀類別） A	672
13.10.1 Nested Class Scope 內的名稱決議程序	679
13.11 當 Classes 成為 Namespace Members A	683
13.12 Local Classes（區域性類別） A	687
第 14 章 Class 的初始化、內容指派、解構	689
14.1 Class 初始化（Initialization）	689
14.2 Class Constructor（建構式）	691

14.2.1 Default Constructor (預設建構式)	698
14.2.2 限制 Object 的誕生	700
14.2.3 Copy Constructor (拷貝建構式)	700
14.3 Class Destructor (解構式)	703
14.3.1 明白喚起 Destructor	706
14.3.2 程式碼膨脹	707
14.4 由 Class Object 組成的 Arrays 和 Vectors	709
14.4.1 Heap 陣列的初始化 A	711
14.4.2 由 Class Objects 組成的 vector	714
14.5 Member Initialization List (成員初值列)	716
14.6 Memberwise (以成員為單元的) 初始化動作 A	723
14.6.1 Member Class Object 的初始化	726
14.7 Memberwise (以成員為單元的) 指派動作 A	729
14.8 效率考量 A	732
第 15 章 多載化運算子 (Overloaded Operators) 與使用者自定的轉換	737
15.1 運算子的多載化 (Operator Overloading)	737
15.1.1 Class Member 與 Nonmember	741
15.1.2 多載化運算子的名稱	744
15.1.3 多載化運算子的設計	746
15.2 Friends	748
15.3 Operator = (assignment 運算子)	751
15.4 Operator [] (subscript 運算子)	754
15.5 Operator () (function call 運算子)	755
15.6 Operator -> (member access arrow 運算子)	756
15.7 Operator++ (遞增運算子) 和 Operator-- (遞減運算子)	759
15.8 Operators new 和 Operator delete	763
15.8.1 陣列的 Operator new[] 和 Operator delete[]	767
15.8.2 Placement Operator new() 和 Placement Operator delete()	769
15.9 使用者自定的轉換動作 (User-Defined Conversions)	772
15.9.1 轉換函式 (conversion functions)	776
15.9.2 Constructor (建構式) 就像個轉換函式	780
15.10 選擇一個轉換動作 A	782
15.10.1 再次討論函式多載決議程序 (Function Overload Resolution)	786
15.10.2 候選函式 (Candidate Functions)	786
15.10.3 Class Scope 內之函式呼叫動作的候選函式	789
15.10.4 使用者自定轉換程序 (User-Defined Conversion Sequences) 的分等	790
15.11 多載化決議程序 (Overload Resolution) 與 Member Functions 的關係 A	795
15.11.1 多載化 Member Functions 的宣告	795
15.11.2 候選函式 (Candidate Functions)	797
15.11.3 可行函式 (Viable Functions)	798
15.12 多載化決議程序 (Overload Resolution) 與運算子的關係 A	801
15.12.1 候選之運算子函式 (Candidate Operator Functions)	802
15.12.2 可行函式 (Viable Functions)	806
15.12.3 模稜兩可 (Ambiguity)	807

第 16 章 Class Templates.....	811
16.1 Class Template 定義式.....	812
16.1.1 Queue 和 QueueItem 的 "Class Template 定義式".....	818
16.2 Class Template 的具現體 (Instantiation)	820
16.2.1 為非型別參數 (Nontype Parameters) 指定 Template 引數.....	825
16.3 Class Templates 的 Member Functions.....	829
16.3.1 Queue 和 QueueItem 的 Template Member Functions.....	831
16.4 Class Templates 中的 Friend 宣告.....	833
16.4.1 Queue 和 QueueItem 的 Friend 宣告.....	835
16.5 Class Templates 的 Static Data Members.....	839
16.6 Class Templates 內的巢狀型別 (Nested Types)	841
16.7 Member Templates A.....	844
16.8 Class Templates 及其編譯模型 (Compilation Model) A.....	849
16.8.1 Inclusion (含入式) 編譯模型.....	851
16.8.2 Separation (分離式) 編譯模型.....	851
16.8.3 Explicit Instantiation Declarations (明白的具現宣告)	854
16.9 Class Template Specializations (特製體、特殊化) A.....	856
16.10 Class Template Partial Specializations (局部特殊化) A.....	860
16.11 Class Templates 內的名稱決議方式 (Name Resolution) A.....	862
16.12 Namespaces (命名空間) 與 Class Templates A.....	865
16.13 一個 Template Array Class.....	867
第五篇：物件導向程式設計 (Object-Oriented Programming)	877
第 17 章 Class 的繼承 (Inheritance) 與子型別化 (Subtyping)	879
17.1 定義一個 Class 階層體系 (Class Hierarchy)	882
17.1.1 物件導向 (object-oriented) 設計概念.....	885
17.2 確認階層體系中的成員.....	890
17.2.1 定義 Base Class (基礎類別)	891
17.2.2 定義 Derived Classes (衍生類別)	895
17.2.3 摘要.....	898
17.3 存取 Base Class 的 Member.....	900
17.4 Base Class (基礎類別) 和 Derived Class (衍生類別) 的建構.....	908
17.4.1 Base Class Constructor (基礎類別建構式)	910
17.4.2 Derived Class Constructor (衍生類別建構式)	911
17.4.3 另一個 Class 階層體系.....	912
17.4.4 緩式錯誤偵測 (Lazy Error Detection)	915
17.4.5 Destructors (解構式)	916
17.5 Base Class (基礎類別) 和 Derived Class (衍生類別) 的虛擬函式.....	919
17.5.1 Virtual Input/Output.....	920
17.5.2 純虛擬函式 (Pure Virtual Functions)	926
17.5.3 虛擬函式的靜態喚起行為 (Static Invocation)	927
17.5.4 虛擬函式和預設引數 (Default Arguments)	929
17.5.5 Virtual Destructors (虛擬解構式)	932
17.5.6 eval() 虛擬函式.....	933

17.5.7 「Virtual new 運算子」的一個替代品.....	939
17.5.8 在建構式 (Constructors) 與解構式 (Destructors) 中呼叫虛擬函式.....	941
17.6 Memberwise 初始化動作和 Memberwise 指派動作 A.....	943
17.7 一個 UserQuery Manager Class.....	948
17.7.1 定義 UserQuery Class.....	953
17.8 把所有的努力集合起來.....	958
第 18 章 多重繼承與虛擬繼承 (Multiple and Virtual Inheritance)	965
18.1 做好準備.....	965
18.2 多重繼承 (Multiple Inheritance)	970
18.3 三種繼承型式：Public, Private, 和 Protected.....	977
18.3.1 繼承 (Inheritance) 與組合 (Composition)	980
18.3.2 針對個別 Members，免除「Private 繼承」所造成的限制.....	981
18.3.3 Protected 繼承型式.....	982
18.3.4 Object Composition (物件組合)	983
18.4 繼承機制下的 Class Scope.....	985
18.4.1 多重繼承下的 Class Scope.....	988
18.5 虛擬繼承 (Virtual Inheritance) A.....	993
18.5.1 Virtual Base Class (虛擬基礎類別) 的宣告.....	996
18.5.2 特殊的初始化語意.....	998
18.5.3 Constructor (建構式) 和 Destructor (解構式) 的喚起次序.....	1000
18.5.4 Virtual Base Class Members 的可見度.....	1002
18.6 多重、虛擬繼承的實例 A.....	1005
18.6.1 一個具備範圍檢驗能力的 Array Class 衍生體.....	1008
18.6.2 一個具備排序能力的 Array Class 衍生體.....	1011
18.6.3 一個多層衍化的 Array Class.....	1016
第 19 章 在 C++ 中使用繼承機制.....	1021
19.1 Run-Time Type Identification (執行時期型別鑑識)	1021
19.1.1 dynamic_cast 運算子.....	1022
19.1.2 typeid 運算子.....	1027
19.1.3 type_info Class.....	1030
19.2 Exceptions (異常情況) 與 Inheritance (繼承)	1033
19.2.1 將 Exceptions 定義為 Class 階層體系.....	1033
19.2.2 丟出一個「Class 型別」的 Exception.....	1035
19.2.3 處理一個「Class 型別」的 Exception.....	1036
19.2.4 Exception Objects 與虛擬函式.....	1039
19.2.5 Stack Unwinding (堆疊輾轉開解) 與 Destructor (解構式) 的關係.....	1041
19.2.6 Exception Specifications (異常規格)	1042
19.2.7 Constructors (建構式) 和 Function try Blocks 的關係.....	1045
19.2.8 C++ Standard Library 中的 Exception Class 階層體系.....	1047
19.3 多載決議程序 (Overload Resolution) 與繼承 (Inheritance) A.....	1051
19.3.1 候選函式 (Candidate Functions)	1052
19.3.2 可行函式 (Viable Functions) 與使用者自定轉換序列.....	1056
19.3.3 最佳可行函式 (Best Viable Function)	1057

第 20 章 iostream 程式庫.....	1063
20.1 輸出用的運算子 Operator<<.....	1067
20.2 輸入.....	1072
20.2.1 字串的輸入.....	1077
20.3 其他的 Input/Output 運算子.....	1083
20.4 將輸出運算子 Operator << 多載化.....	1090
20.5 將輸入運算子 Operator >> 多載化.....	1095
20.6 檔案輸入和輸出.....	1097
20.7 Stream（資料流）的狀態.....	1107
20.8 String Streams（字串資料流）.....	1109
20.9 格式狀態（Format State）.....	1112
20.10 一個對型別有強烈要求（Strongly Typed）的程式庫.....	1121
附錄：泛型演算法（Generic Algorithms，按字母排序）.....	1123
accumulate() 元素累加.....	1125
adjacent_difference() 相鄰元素的差額.....	1126
adjacent_find() 搜尋相鄰的重複元素.....	1127
binary_search() 二元搜尋.....	1128
copy() 複製.....	1129
copy_backward() 逆向複製.....	1130
count() 計數.....	1131
count_if() 在特定條件下計數.....	1133
equal() 判斷相等與否.....	1134
equal_range() 判斷相等與否（傳回一個上下限區間範圍）.....	1136
fill() 改填元素值.....	1138
fill_n() 改填元素值，n 次.....	1139
find() 搜尋.....	1140
find_if() 在特定條件下搜尋.....	1141
find_end() 搜尋某個子序列的最後一次出現地點.....	1143
find_first_of() 搜尋某些元素的首次出現地點.....	1144
for_each() 對範圍內的每一個元素施行某動作.....	1145
generate() 以指定動作的運算結果充填特定範圍內的元素.....	1146
generate_n() 以指定動作的運算結果充填 n 個元素內容.....	1147
includes() 涵蓋於.....	1148
inner_product() 內積.....	1149
inplace_merge() 合併並取代（覆寫）.....	1150
iter_swap() 元素互換.....	1152
lexicographical_compare() 以字典排列方式做比較.....	1153
lower_bound() 下限.....	1154
max() 最大值.....	1156
max_element() 最大值所在位置.....	1156
min() 最小值.....	1156
min_element() 最小值所在位置.....	1157
merge() 合併兩個序列.....	1158
mismatch() 找出不吻合點.....	1159

<code>next_permutation()</code>	獲得下一個排列組合.....	1161
<code>nth_element()</code>	重新安排序列中第 <i>n</i> 個元素的左右兩端.....	1162
<code>partial_sort()</code>	局部排序.....	1163
<code>partial_sort_copy()</code>	局部排序並複製到它處.....	1164
<code>partial_sum()</code>	局部總和.....	1165
<code>partition()</code>	切割.....	1167
<code>prev_permutation()</code>	獲得前一個排列組合.....	1168
<code>random_shuffle()</code>	隨機重排.....	1169
<code>remove()</code>	移除某種元素（但不刪除）.....	1170
<code>remove_copy()</code>	移除某種元素並將結果複製到另一個 <code>container</code>	1170
<code>remove_if()</code>	有條件地移除某種元素.....	1171
<code>remove_copy_if()</code>	有條件地移除某種元素並將結果複製到另一個 <code>container</code>	1172
<code>replace()</code>	取代某種元素.....	1173
<code>replace_copy()</code>	取代某種元素，並將結果複製到另一個 <code>container</code>	1173
<code>replace_if()</code>	有條件地取代.....	1174
<code>replace_copy_if()</code>	有條件地取代，並將結果複製到另一個 <code>container</code>	1174
<code>reverse()</code>	顛倒元素次序.....	1175
<code>reverse_copy()</code>	顛倒元素次序並將結果複製到另一個 <code>container</code>	1176
<code>rotate()</code>	旋轉.....	1177
<code>rotate_copy()</code>	旋轉，並將結果複製到另一個 <code>container</code>	1177
<code>search()</code>	搜尋某個子序列.....	1178
<code>search_n()</code>	搜尋「連續發生 <i>n</i> 次」的子序列.....	1180
<code>set_difference()</code>	差集.....	1181
<code>set_intersection()</code>	交集.....	1181
<code>set_symmetric_difference()</code>	對稱差集.....	1182
<code>set_union()</code>	聯集.....	1182
<code>sort()</code>	排序.....	1184
<code>stable_partition()</code>	切割並保持元素相對次序.....	1185
<code>stable_sort()</code>	排序並保持等值元素的相對次序.....	1186
<code>swap()</code>	置換（對調）.....	1187
<code>swap_range()</code>	置換（指定範圍）.....	1188
<code>transform()</code>	以兩個序列為基礎，交互作用產生第三個序列.....	1189
<code>unique()</code>	將重複的元素摺疊縮編，使成唯一	1190
<code>unique_copy()</code>	將重複的元素摺疊縮編，使成唯一，並複製到他處.....	1191
<code>upper_bound()</code>	上限.....	1193
Heap（堆積）演算法.....		1194
<code>make_heap()</code>	製造一個 <code>heap</code>	1194
<code>pop_heap()</code>	從 <code>heap</code> 內取出一個元素.....	1195
<code>push_heap()</code>	將一個元素推進 <code>heap</code> 內.....	1195
<code>sort_heap()</code>	對 <code>heap</code> 排序.....	1195
索引（Index）.....		1199