C++ 入門指南 – **樣版 (template)**

<http://pydoing.blogspot.tw/2013/02/cpp-function-template.html>

以下C++檔案，請參考

**函數樣版**

函數樣版 (function template) 可定義適用多種型態 (type) 的函數 (function) ，而不需要多載 (overload) 適用的各種情況。

|  |  |
| --- | --- |
|  | #include <iostream>    **template** <typename T1, typename T2>  T1 do\_something(const T1 &a, const T2 &b) {      return a + b;  }    int main(void) {      std::cout << do\_something(1, 10) << std::endl;      std::cout << do\_something(1, 10.5) << std::endl;      std::cout << do\_something(2.5, 6.32) << std::endl;        return 0;  } |

定義函數樣版要先用關鍵字 (keyword) **template** 宣告是個樣版定義，然後是角括號圍起來的樣版參數列 (parameter list) ，使用 **typename** 宣告所用的型態名稱，底下函數的參數 (parameter) 要用 **const** 參考 (reference)

|  |  |
| --- | --- |
|  | template <typename T1, typename T2>  T1 do\_something(const T1 &a, const T2 &b) {      return a + b;  } |

**類別樣版**

<http://pydoing.blogspot.tw/2013/02/cpp-class-template.html>

以下C++檔案，請參考 子目錄 t15\_8.cpp

類別樣版 (class template) 可定義適用多種型態 (type) 的類別 (class) ，而不需要多載 (overload) 適用各種情況的成員函數 (member function) 。

|  |  |
| --- | --- |
|  | #include <iostream>    **template** <typename T> class Demo {      public:          Demo(const T &a, const T &b) {              this->a = a;              this->b = b;          }            T do\_something() {              return a + b;          }        private:          T a;          T b;  };    int main(void) {      Demo<int> d1(20, 25);      std::cout << d1.do\_something() << std::endl;      Demo<double> d2(5.2, 3.95);      std::cout << d2.do\_something() << std::endl;      Demo<char> d3('1', '2');      std::cout << d3.do\_something() << std::endl;        return 0;  } |

定義函數樣版要先用關鍵字 (keyword) **template** 宣告是個樣版定義，然後是角括號圍起來的樣版參數列 (parameter list) ，使用 **typename** 宣告所用的型態名稱，底下成員函數的參數 (parameter) 要用 **const** 參考 (reference)

|  |  |
| --- | --- |
|  | template <typename T> class Demo {      public:          Demo(const T &a, const T &b) {              this->a = a;              this->b = b;          }            T do\_something() {              return a + b;          }        private:          T a;          T b;  }; |

**typename** 用 **class** 替代也可以。

使用類別樣版宣告變數，也要用角括弧宣告 *T* 的實際型態

|  |  |
| --- | --- |
|  | Demo<int> d1(20, 25);  std::cout << d1.do\_something() << std::endl;  Demo<double> d2(5.2, 3.95);  std::cout << d2.do\_something() << std::endl;  Demo<char> d3('1', '2');  std::cout << d3.do\_something() << std::endl; |