C++ 入門指南 - 封裝

節錄自

<http://pydoing.blogspot.tw/2012/10/cpp-encapsulation.html>

以下C++檔案，請參考 t15\_3.cpp

開發新類別 (class) 的目的不外是提供給其他人使用，然而用的人往往只需要知道類別的規格就可以了，至於怎麼樣實作出類別的細節，其實不是那麼重要，這是程式設計中頗為重要的概念 --- 資訊隱藏 (information hiding)

最基本的就是把成員變數 (variable member) 封裝 (encapsulation) 在類別之中，這時候就要用到 **private** 存取標籤 (accsee label) 了

|  |
| --- |
| class Demo {public:    void setA(int n);    void setB(int n);    int getA();    int getB();    int do\_something(); private:    int a;    int b; }; |

這裡 *a* 與 *b* 已經改放到 **private** 之後，也由於 *a* 與 *b* 都是 **private** 的，因此另外宣告 **public** 的 *setA()* 與 *setB()* 設定 *a* 與 *b* 之值，*getA()* 與 *getB()* 取得 *a* 與 *b* 之值。

*setA()* 與 *setB()* 就是俗稱的 ***setter*** ，至於 *getA()* 與 *getB()* 是俗稱的 ***getter*** 。

因此 *setA* 、 *setB* 、 *getA* 、 *getB* 的實作很簡單，如下

|  |
| --- |
| void Demo::setA(int n) {    a = n;} void Demo::setB(int n) {    b = n;} int Demo::getA() {    return a;} int Demo::getB() {    return b;} |

*do\_something*也要稍作改變

|  |
| --- |
| int Demo::do\_something() {    return a + b;} |

我們寫成一個完整範例，如下

|  |  |
| --- | --- |
|  | #include <iostream> using namespace std; class Demo {public:    void setA(int n);    void setB(int n);    int getA();    int getB();    int do\_something(); private:    int a;    int b; }; int Demo::do\_something() {    return a + b;} void Demo::setA(int n) {    a = n;} void Demo::setB(int n) {    b = n;} int Demo::getA() {    return a;} int Demo::getB() {    return b;} int main(void) {    Demo t;    t.setA(11);    t.setB(22);      cout << endl;    cout << t.do\_something() << endl;    cout << endl;         return 0;}  |