C++資料結構與程式設計

C++ 樣板(Template)

NTU CSIE

樣板（Template）

C++ 的 Template 是種將資料型態參數化的功能。將資料型態資訊自程式碼中抽離，代之以簡化的符號 (T, T1, T2, ...)。再由編譯器透過類似巨集代換的方式，根據樣板內容產生實際的程式碼

- Function Template (函式樣板)
- Class Template (類別樣板)。含部份特殊化

大綱

函式樣板

類別樣板

C++函式樣板 (Function Template)

思考:

- 如果今天我們想要設計一些函式來列印出不同型態的陣列內容，這些函式的參數列定義的資料型態必須設為不同的型態，該如何設計這些函式？

描述上述函式的三種方式：

- 一般函式 (Function)：
  - 描述各種可能用到的型態，每個函式都取不同的名稱
- 函式重載 (Function Overloading)：
  - 描述各種可能用到的型態，每個函式都取相同的名稱
- 函式樣板 (Function Template)：
  - 描述一個型態，一個函式名稱

一般函式 (Function)

作法:

- 描述各個函式，每個函式都取不同的名稱，將可能用到的型態對應到各個函式中

優點：

- 一般程式語言皆支援這種語法

缺點：

- 程式設計師不容易記住所有函式名稱，造成開發程式的複雜度
- 容易因為不小心函式使用錯誤，還造成程式錯誤

範例

使用C++寫一個泡沫排序法(BubbleSort)，可對整數、浮點數、字元、字串資料做排序，並將排序結果印出 (c++_bubblesort.cpp)

函式重載(Function Overloading)

作法:

- 描述各個函式，每個函式都取相同的名稱，將可能用到的型態對應到各個函式中。

優點：

- 程式設計師不需記一堆函式名稱。編譯器自動會把呼叫的函式對應到同參數型態的函式中。

缺點：

- 雖比起一般函式的做法方便，但差不多的程式碼還是要寫很多遍。

範例

使用C++函式重載寫一個泡沫排序法(BubbleSort)，可對整數、浮點數、字元、字串資料做排序，並將排序結果印出 (c++_bubblesort_overloading.cpp)

函式樣板 (Function Template)

作法:

- 描述一個函式，只需一個名稱，使用樣板參數取替資料型態。

優點：

- 程式設計師除了不需記一堆函式名稱之外， 也不需要寫一堆類似的程式碼。編譯器自動產生需要的程式碼(重載函式)，然後把呼叫的函式對應到同參數型態的函式中。
- 程式碼一般較小 (用到的型態才會產生程式)

缺點：

- 程式功能缺乏彈性(只能描述類似的程式行為)。

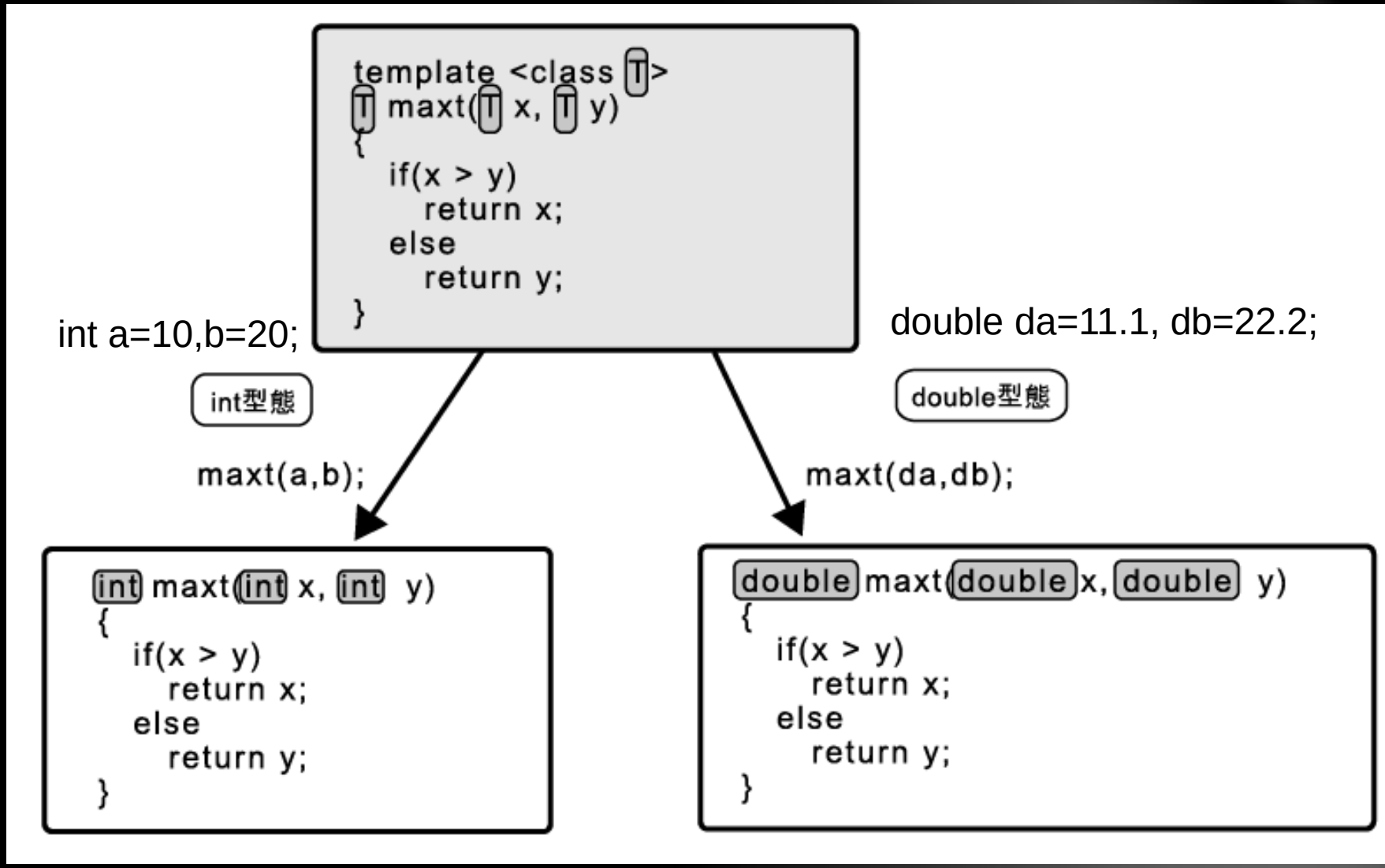
函式樣板

以關鍵字 *template* 起始一個樣板宣告，後接參數。C/C++ 的程式語法，是以 ( ) 括起參數。而 Template 語法，則以 < > 括起參數。

樣板所用參數之完整稱呼是「樣板參數(template parameters)」，慣例以 T, T1, T2 等作為樣板參數名稱。

接著再寫上函式的程式碼樣板，又稱原型(prototype)。函式原型的函式名稱即為函式樣板的名稱。

```
template < 樣板參數型態 樣板參數名 , ...其他樣板參數 >
原型回傳型態 函式名稱(參數型態 原型參數名, ...)
{
   程式碼;
}
```


template <class T>
T maxt(T x, T y)
{
if(x > y)
return x;
else
return y;
}
int a=10,b=20;
double da=11.1, db=22.2;
int型態
double型態
maxt(a,b);
maxt(da,db);
int maxt(int x, int y)
{
if(x > y)
return x;
else
return y;
}
double maxt(double x, double y)
{
if(x > y)
return x;
else
return y;
}

函式樣板

參數型態可用關鍵字 *class* 或 *typename* 表示泛用型態 (即任何型態)；或是一個已宣告的資料型態，如 *int* 與自定類。

原型中的參數型態若是已宣告的資料型態，則是一種特殊化的函式樣板。

```
#include <iostream>
using namespace std;
int Add(int a, int b)
{
	return a + b;
}

template <class T>
T Add(T a, T b)
{
	return a + b;
}
```

```
int main()
{
	int c1;
	double c2;
	c1 = Add(10, 20);
	c2 = Add(10.3, 20.4);
	return 0;
}
```

範例

使用C++函式樣板寫一個泡沫排序法(BubbleSort)，可對整數、浮點數、字元、字串資料做排序，並將排序結果印出

(c++_bubblesort_template.cpp)

小練習

請將之前 矩陣相乘-函式間傳遞陣列(list) 的練習改成使用函式樣版

```
void Input2dAry(int [][3]);
void Print2dAry(int [][3]);
void MatrixMul2dAry(int [][3],int [][3],int [][3]);
```

大綱

函式樣板

類別樣板

類別樣板

語法與函式樣板相同，差別在其原型為類別。此外，樣板參數可以設定預設值。

套用類別樣板的語法則是以樣板名稱括起參數值，括號用 < >。

```
template < 樣板參數型態 樣板參數名 , ... >
class 類別名稱<部份特殊化型態,...>
{
   原型回傳型態 函式名稱(參數型態 原型參數名, ...) ;
   參數型態 變數名稱;
}

template < 樣板參數型態 樣板參數名 , ... >
原型回傳型態 類別名稱< 樣板參數型態 樣板參數名 , ... > ::函式名稱<部份特殊化型態,...>
{
   程式碼;
}

類別名稱
```

```
template<class T>
class LinkedList
{
  void printList();
};
```

```
template<class T>
void LinkedList<T>::printList(){
//.....
}
```

類別樣板

範例

```
#include <iostream>
#include <string>
using namespace std;

class myClass
{
public:
   void display(int n)
   {
      a = n;
      cout << a << endl;
   }
   int a;
};
```

```
template <class T>
class myTClass
{
public:
   void display(T n)
   {
      a = n;
      cout << a << endl;
   }
   T a;
};

int main()
{
   myTClass<int> a;
   myTClass<double> x;
   a.display(10);
   x.display(10.4);
   return 0;
}
```

類別樣板

範例

```
#include <iostream>
#include <string>
using namespace std;

class myClass
{
public:
    void display(int n, char ch)
    {
        a = n;
        b = ch;
        cout << a << " " << b << endl;
    }
    int a;
    char b;
};
```

```
template <class T1, class T2>
class myTClass
{
public:
    void display(T1 n, T2 ch)
    {
        a = n;
        b = ch;
        cout << a << " " << b << endl;
    }
    T1 a;
    T2 b;
};

int main()
{
    myTClass<int, char> a;
    myTClass<double, double> x;
    a.display(10, 'A');
    x.display(10.4, 20.5);
    return 0;
}
```

練習

- 使用鏈結串列範例為例, 製作鏈結串列樣板 (c++_linked_list_template.cpp)
- 請試用 string 型態輸入與輸出資料
- 請將此範例加入「新增節點在鏈結串列的前面」

```
template <class T>
class node
{
        private:
                T data;
                node<T> *next;
        template <class T2> friend class LinkedList;
        //friend class LinkedList<double>;
        //friend class LinkedList<int>;
};
```

https://jgirl.ddns.net/problem/0/1046

進階練習

使用鏈結串列範例為例, 製作鏈結串列樣板

可對整數、浮點數、字元、字串資料做處理

功能

- 輸入'1'選擇使用整數鏈結串列
- 輸入'2'選擇使用浮點數鏈結串列
- 輸入'3'選擇使用字元鏈結串列
- 輸入'4'選擇使用字串鏈結串列

提示：使用函式樣板選擇串列型態

https://jgirl.ddns.net/problem/0/2133

(回家)練習

- 使用二元搜尋樹範例，製作二元搜尋樹樣板
- 可參見批改系統

  https://jgirl.ddns.net/problem/0/2134