

C++面向對象編程 (C++Object-Oriented Programming)

課程簡介

這是我的所有 C++技術課程中最基礎最根本的一門課。

C++可說是第一個高度普及的 Object-Oriented (面向對象) 編程語言。”第一個”或“最早的”並非重點，重點是經過多年的淬煉和考驗 C++的影響深入各層面，擁有眾多使用者和極其豐富的資源 (書籍、論文、期刊、視頻、第三方程序庫...)。

作為 Object-Oriented (面向對象) 技術的主流語言，C++ 其實還擁有另一支編程主線：模板與泛型編程 (Templates and Generic Programming)。

本課程涵蓋上述兩條主線：Object-Oriented (面向對象) 和泛型編程 (Generic Programming)。

由於視頻錄製時程的因素，本課程分為 Part I 和 Part II.

Part I 主要討論 OO (面向對象) 的基礎概念和編程手法。基礎最是重要，勿在浮沙築高台，我著重的是大器與大氣。課程首先探討 Class without pointer members 和 Class with pointer members 兩大類型，而後晉昇至 OOP/OOD，包括 classes 之間最重要的三種關係：繼承(Inheritance)、複合(Composition)、委託(Delegation)。Part II 繼續探討更多相關主題，並加上低階的對象模型 (Object Model)，以及高階的 Templates (模板) 編程。

本課程所談主題都隸屬 C++1.0 (C++98)；至於 C++ 2.0 (C++11/14) 帶來的嶄新內容則由我的另一門課程涵蓋。C++2.0 在語言和標準庫兩方面都帶來了很多新事物，份量足以形成另一門課程。

你將獲得整個 video 課程的完整講義 (也就是 video 呈現的每一張投影片畫面)，和完整的示例程序。你可以 (也必要) 在視聽過程中隨時停格思考和閱讀講義。

侯捷簡介：程序員，軟件工程師，臺灣工研院副研究員，教授，專欄主筆。曾任教於中壢元智大學、上海同濟大學、南京大學。著有《無責任書評》三卷、《深入淺出 MFC》、《STL 源碼剖析》...，譯有《Effective C++》《More Effective C++》《C++ Primer》《The C++ Standard Library》...。

以下這份不算太細緻的主題劃分，協助您認識整個課程內容。

C++面向對象編程 (C++Object-Oriented Programming)

Part I

Introduction of C++98, TR1, C++11, C++14

Bibliography

Data and Functions, from C to C++

Basic forms of C++ programs

About output

Guard declarations of header files

Layout of headers

Object Based

Class without pointer member

Class declarations

Class template, introductions and overviews

What is 'this'

Inline functions

Access levels

Constructor (ctor)

Const member functions

Parameters : pass by value vs. pass by reference

Return values : return by value vs. return by reference

Friend

Definitions outside class body

Operator overloading, as member function

Return by reference, again

Operator overloading, as non-member function

Temporary objects

Expertise

Code demonstration

class with pointer members

The "Big Three"

Copy Constructor

Copy Assignment Operator

Destructor

Ctor and Dtor, in our String class

"MUST HAVE" in our String class

Copy Constructor

Copy assignment operator

Deal with "self assignment"

Another way to deal with "self assignment" : Copy and Swap

Overloading output operator (<<)

Expertise

Code demonstration

Objects from stack vs. objects from heap

Objects lifetime

new expression : allocate memory and then invoke ctor

delete expression : invoke dtor and then free memory

Anatomy of memory blocks from heap

Allocate an array dynamically

new[] and delete[]

MORE ISSUES :

static

private ctors

cout

Class template

Function template

namespace

Standard Library : Introductions and Overviews

Object Oriented

Composition means "has-a"

Construction : from inside to outside

Destruction : from outside to inside

Delegation means "Composition by reference"

Inheritance means "is-a"

Construction : from inside to outside

Destruction : from outside to inside

Construction and Destruction, when Inheritance+Composition

Inheritance with virtual functions

Virtual functions typical usage 1 : Template Method

Virtual functions typical usage 2 : Polymorphism

Virtual functions inside out : vptr, vtbl, and dynamic binding

Delegation + Inheritance : Observer

Delegation + Inheritance : Composite

Delegation + Inheritance : Prototype

Part II

緒論

Conversion function (轉換函數)

Non-explicit one-argument constructor

Pointer-like classes, 關於智能指針

Pointer-like classes, 關於迭代器

Function-like classes, 所謂仿函數

標準庫中的仿函數的奇特模樣

namespace 經驗談

class template, 類模板

function template, 函數模板

member template, 成員模板

specialization, 模板特化

partial specialization, 模板偏特化 —— 個數的偏

partial specialization, 模板偏特化 —— 範圍的偏

template template parameter, 模板模板參數

variadic templates (since C++11)

auto (since C++11)

ranged-base for (since C++11)

reference

Composition (複合) 關係下的構造和析構

Inheritance (繼承) 關係下的構造和析構

Inheritance+Composition 關係下的構造和析構

對象模型 (Object Model) : 關於 `vptr` 和 `vtbl`

對象模型 (Object Model) : 關於 `this`

對象模型 (Object Model) : 關於 `Dynamic Binding`

談談 `const`

關於 `new`, `delete`

重載 `::operator new`, `::operator delete`

重載 `::operator new[]`, `::operator delete[]`

重載 member operator `new/delete`

重載 member operator `new[]/delete[]`

示例

重載 `new()`, `delete()`

示例

`basic_string` 使用 `new(extra)` 擴充申請量

-- the end