

(程式)程式之構成：

1. 儲存資料之機制 – 變數(variables) / 陣列(arrays)
2. 處理資料之方法 – 函式(functions) / 方法(methods) / 副程式(subroutines)

何謂物件導向(Object Oriented)?

使程式之結構/構成，有如現實(或虛擬)之世界

現實(或虛擬)世界如何構成?

In real world, anything is an object, and object consists of object.

Characteristics of real world objects: (1) states (of attributes) (2) behaviors.

如何描述/模擬現實世界中的物件?

How to describe/model real-world object?

Terms:

1. 物件(object)
2. 成員(member)
3. 屬性(attribute)
4. 狀態(state)
5. 行為/能力(behaviors)
6. 介面(interface)

Definition of Object in Object-Oriented Programming:

In Object-Oriented Programming, an object is a software bundle of variables and related functions/methods to model real-world object. Everything that a software object knows (states) and can do (behaviors) is expressed by the variables and functions/methods with that object.

Features of Object-Oriented Programming:

1. Encapsulation(資料封裝):

Each object has a public interface that other objects can use, but the implementation of that interface is kept private.

功能：使程式模組化

例： Encap.cpp

2. Inheritance(繼承):

New classes are created from existing classes by absorbing their attributes and behaviors and add the capabilities the new classes require.

功能：讓程式可再使用

例： Inher.cpp

3. Polymorphism(多型): (*virtual function in C++*)

The ability for objects of different classes related by inheritance to response differently to the same member function call.

功能：讓物件的管理與使用通常化(一致化/抽象化)，但物件的反應特殊化

例： Poly.cpp