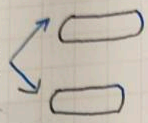
My Notes

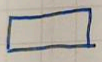
Important Concepts worth keeping

Today : / /

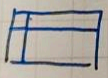
基本符號

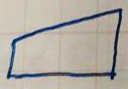
□ 起止符號 

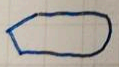
□ 流程符號 ↓

□ 程序(Process) 

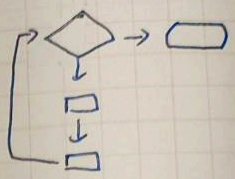
常用符號

□ 宣告符號 

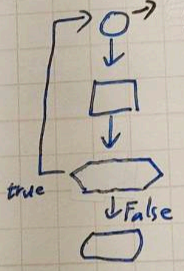
□ 人工輸入 

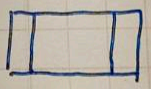
□ 顯示符號 

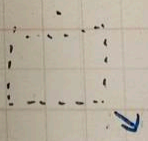
□ 決策符號 

重複 while 

do - while ↓ 先做一次 (連接用)



已定義流程 

註解 

My Questions
Problems & Difficulties needing exploration

n個選k個

有包含 不包含

$C(n,k) = C(n-1, k-1) + C(n-1, k)$

Base case

$C(k,k) = 1$ 一個即全部

$C(n,0) = 1$ 都不選

$k > n = 0$

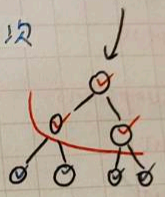
Leaf nodes 葉節點

Internal nodes 內部節點

|Leaf nodes| - |Internal nodes| = 1

4 3

呼叫次數，只要是 base case 都有有兩次



2 · C(n,k) - 1 呼叫次數

My Opinions
Thoughts, inspirations, and suggestions

尾端遞迴 → 稱為修改可改為迴圈

遞迴式後程式結束後無程式碼

Summary

1、遞迴定義

2、問題簡化

3、終止條件

4、保證終止

My Notes

Important Concepts worth keeping

Today: , ,

Data Abstraction 資料抽象化

ADT 抽象資料型別

物件導向
- 封裝 Encapsulation
- 繼承 Inheritance
- 多型 Polymorphism

Operation Contracts 運算合約

目的 Purpose

假設 Assumptions

輸入 Input

輸出 Output

Abstract Data Types 抽象化資料型態

Modularity 模組化

好的程式：1、高內聚 high cohesive (Cohesion) moduels desired

2、低耦合 Lossely coupled (coupling) moduels desired

函式間的參數越少越好

盡量簡化

Today :

型別

My Questions
Problems & Difficulties needing exploration

Functional abstraction 功能性的抽象化

Specifications 描述

Implementation 實作

資訊隱藏 Information hiding

The ADT Sort List

Inserts and Deletes by value 依值排序

My Opinions
Thoughts, inspirations, and suggestions

Implementing ADTS

限使用提供的 Functions

C++ classes

封裝 encapsulated

- 成員 member function
- 私密 private
- 公開 public

魔鬼躲在細縫裡，天使更是這樣。

My Notes

Important Concepts worth keeping

Today: , ,

C++ Classes Constructors

when declare an instance of the class:
have same name as the class

no return type, not void

default constructor 預設建構

if don't have any arguments

scope resolution operator ::

Set initializers

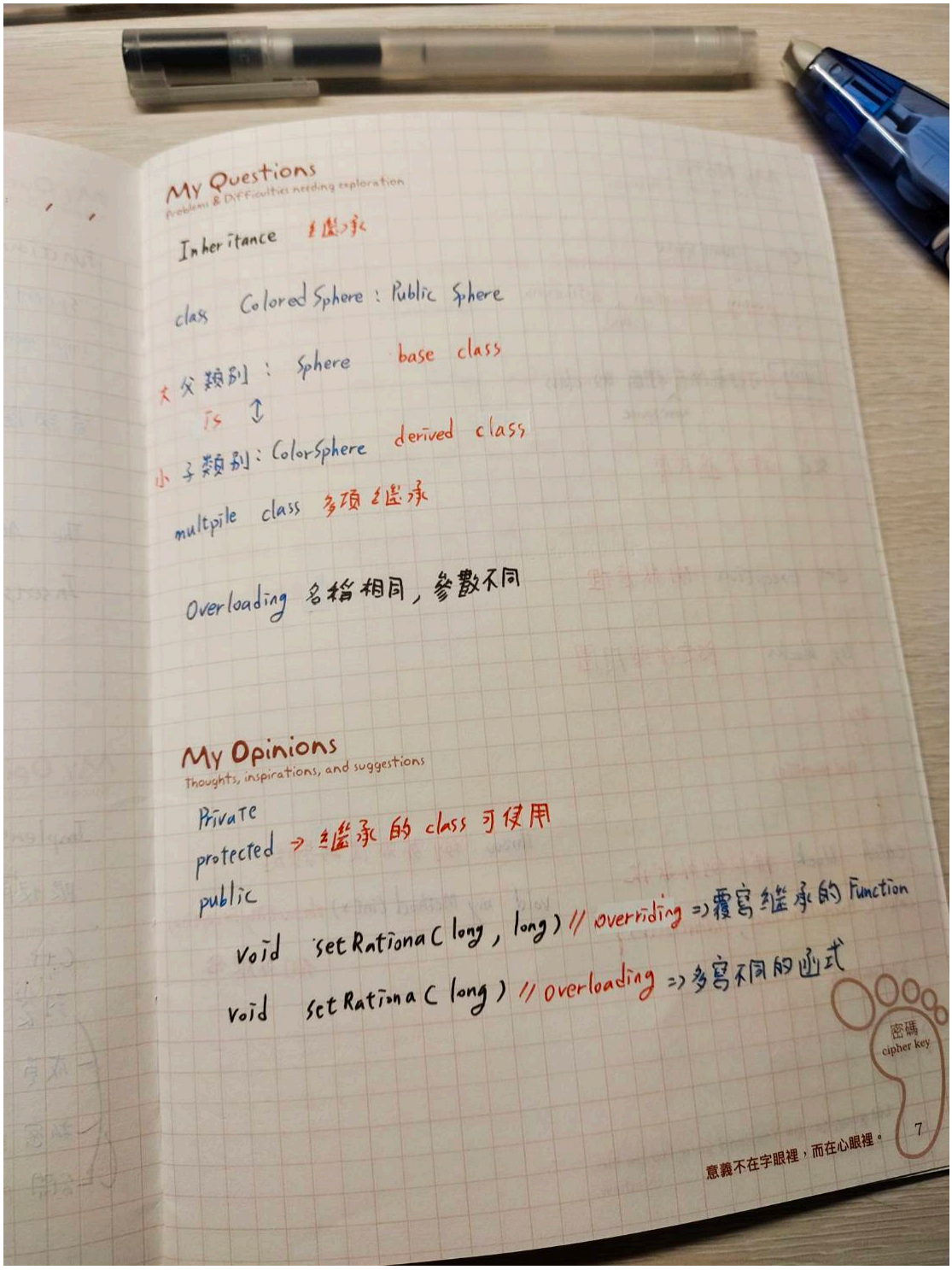
Destructor 解構

each class has one destructor 預設解構

Using Class Sphere

Sphere unitSphere 宣告變數：預設建構

Sphere mySphere(3,1) 執行運算



My Questions
Problems & Difficulties needing exploration

Inheritance 繼承

class ColoredSphere : Public Sphere

父類別 : Sphere base class

is ↕

子類別 : ColorSphere derived class

multpile class 多項繼承

Overloading 名稱相同，參數不同

My Opinions
Thoughts, inspirations, and suggestions

Private

protected => 繼承的 class 可使用

public

void setRational(long, long) // overriding => 覆寫繼承的 Function

void setRational(long) // overloading => 多寫不同的函式

My Notes

Important Concepts worth keeping

Today: , ,

C++ name space

grouping declarations and definitions

using 可使用所有裡面的 class

namespace

std 標準函式庫

C++ Exception 例外處理

try blocks 設定保護範圍

```
try
{
  statement(s)
}
```

catch block 捕捉例外狀況

```
catch ( ExceptionClass, identifier) {
  statements
}
```

throw 例外狀況的類別

```
void my Method (int x) throw (MyException)
```

例外規格

Let your Yes be Yes, and your No be No

Today :

My Questions
Problems & Difficulties needing exploration

Summary

1. Data abstraction controls the interaction between program and data structures

2. Define ADT fully before making any decision about implement.

3. Hide implementation of an ADT by defining as a C++ class

4. Define and implement a class within header and implementation files

My Opinions
Thoughts, inspirations, and suggestions

類別

throw (My Exception)

例外規格

密碼
cipher k

你們的話，是，就說是；不是，就說不是。
《馬太福音》

My Notes
Important Concepts worth keeping

Today: , ,

鏈結串列 Link List

Pointers 指標

Pointers - based Linked List 指標鏈結串列

Circular Linked Lists 環狀鏈接串列
Doubly Linked Lists 雙向鏈接串列

陣列 需移動資料
鏈接串列 不需移動資料

(int *)p 一般變數：直接配給
↓
資料位置(門牌號碼)

p = (&x);
↓
x的資料位置(x的門牌號碼)

p = new int 緊急配置
↓ ↘ 申請新的房子
裡面是空的

std: bad_alloc
↓
沒有空的空間(沒有新房子)

My Questions
Problems & Difficulties needing exploration

delet p 清理空間

p=NULL 忘記位置（以免使用到）

memory Leak (leakage)

dangling reference 輸出到已經 delete 的指標
(illegal access)

My Opinions
Thoughts, inspirations, and suggestions

動態陣列

int arraySize = 30 動態(配置)陣列

double *anArray = new double[arraySize]

anArray[2] ≡ *(anArray+2) 陣列名稱＝指標

double *oldArray = anArray; 配置更大空間

anArray = new double[3*arraySize] → double array size

密碼
cipher key

實話直說很重要，優雅說來更巧妙。

My Notes

Important Concepts worth keeping

Today: , ,

```
for ( int index =0 ;  index < armySize ;  index ++)
    anArray [index] = oldArray [index]   一個一個搬
```

```
FILE *outline = NULL ;
                ⇒檔案位置
outFile = fopen( fileName.c_str(), "a");
```

Linked list is empty

```
Node *p ;  // pointer to node 節點
p=new Node // allocate node
```

Remarks

pointer

new , delete, next 動態配置

special case

Insert a node at beginning

Delet the first node

A word fitly spoken is like apples of gold in baskets of silver.

《Proverbs》

My Questions
Problems & Difficulties needing exploration

Public methods
- isEmpty
- getLength
- insert
- remove
- retreve

private method
find

Private data member
head , size 節點數

Local variables to methods
cur 現
prev 前

```
struct
{
ListItemType  item;
ListNode  *next;
}
```

My Opinions
Thoughts, inspirations, and suggestions

清空陣列 destructor

```
List::~List()
{
  while (!isEmpty()) 是否為空
      remove(1); 刪除
}
```

淺層複製 只複製頂
shallow copy

深層複製 全部複製

複製 List (deep copy)

```
List :: List(const List &aList)
    :size(aList.size)
{
  ...
```

copy 部分:

```
head = new ListNode
head->item = aList.head->item;
ListNode *newPtr = head
for (ListNode *origPtr = aList.head->next;
     origPtr != NULL; origPtr = origPtr->next)
{
    newPtr->next = new ListNode;
    newPtr = newPtr->next;
    newPtr->item = origPtr->item;
}
```


If you don't know where you're going it doesn't matter what path you take.
– Lewis Carroll

My Notes

Important Concepts worth keeping

Today : , ,

Defining Languages

A language

A set of strings of symbols

algebraic expression 代數運算式

| or 或

• followed by 緊接著

Infix expressions 中序運算式 ((a+b)*c) ab+c*

Prefix expressions 前序運算式 *+abc · +a*bc

Postfix expressions 後序運算式 *+abc ab+c*

My Questions
Problems & Difficulties needing exploration

Advantage of prefix and postfix

No 優先權, 結合律, 括弧

Back tracking

八皇后問題 64格, 8個方向

40億種放法

由右→右 每欄放一個 40000種 (碰到不行就回1格)

終止：8欄項滿

返回呼叫：續項其他欄位

My Opinions
Thoughts, inspirations, and suggestions