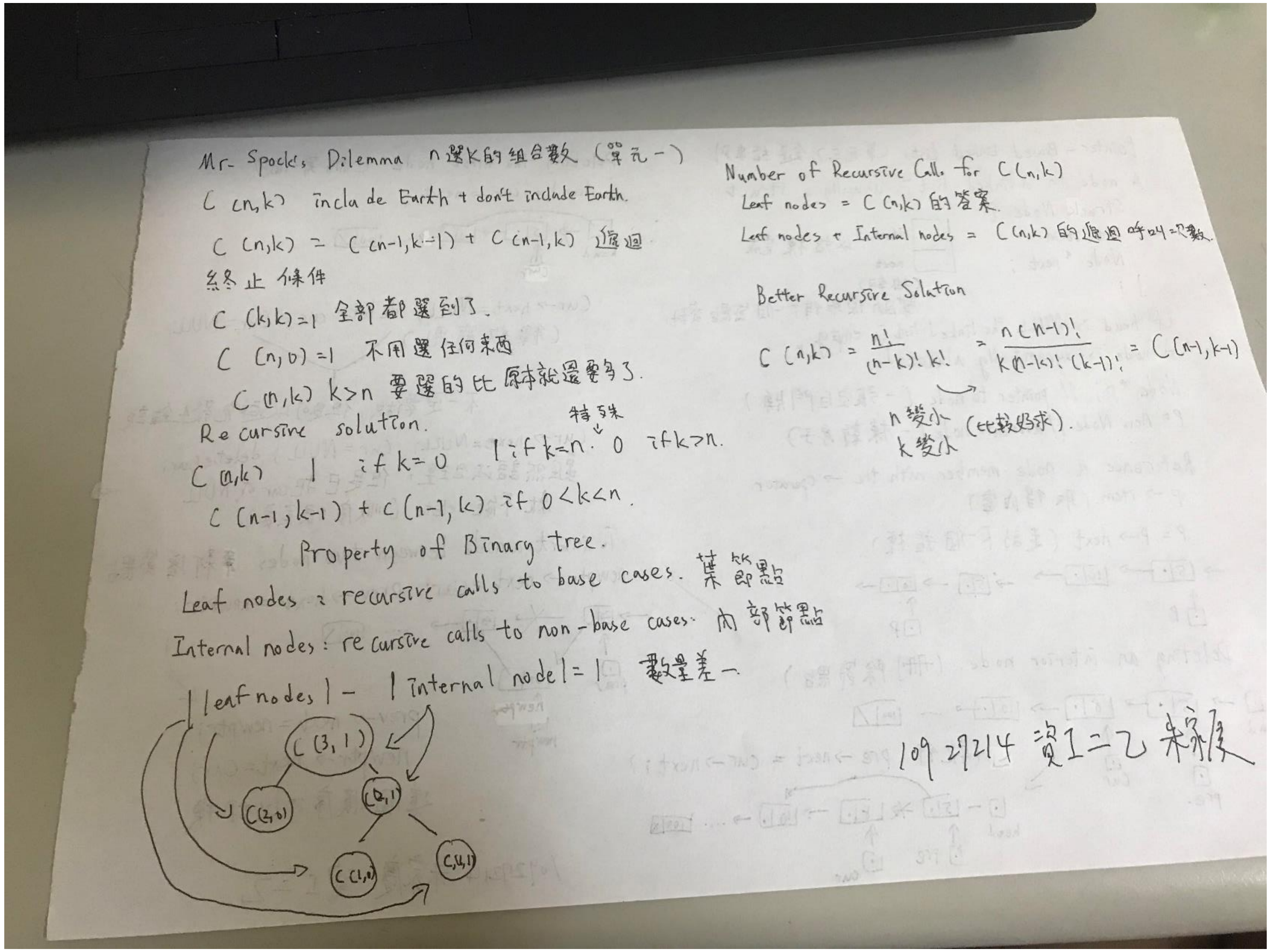
10927214 資工二乙 朱家慶



Mr. Spock's Dilemma n選k的組合數（單元一）

C(n,k) include Earth + don't include Earth.

C(n,k) = C(n-1,k-1) + C(n-1,k) 遞迴

終止條件

C(k,k)=1 全部都選到了.

C(n,0)=1 不用選任何東西

C(n,k) k>n 要選的比原本就還要多了.

Recursive solution.

$$C(n,k) \quad 1 \text{ if } k=0 \quad 1 \text{ if } k=n \overset{特殊}{\downarrow} \quad 0 \text{ if } k>n.$$

$$C(n-1,k-1) + C(n-1,k) \text{ if } 0<k<n.$$

Property of Binary tree.

Leaf nodes : recursive calls to base cases. 葉節點

Internal nodes : recursive calls to non-base cases. 內部節點

|leaf nodes| − |internal node| = 1 數量差一

Number of Recursive Calls for C(n,k)

Leaf nodes = C(n,k) 的答案

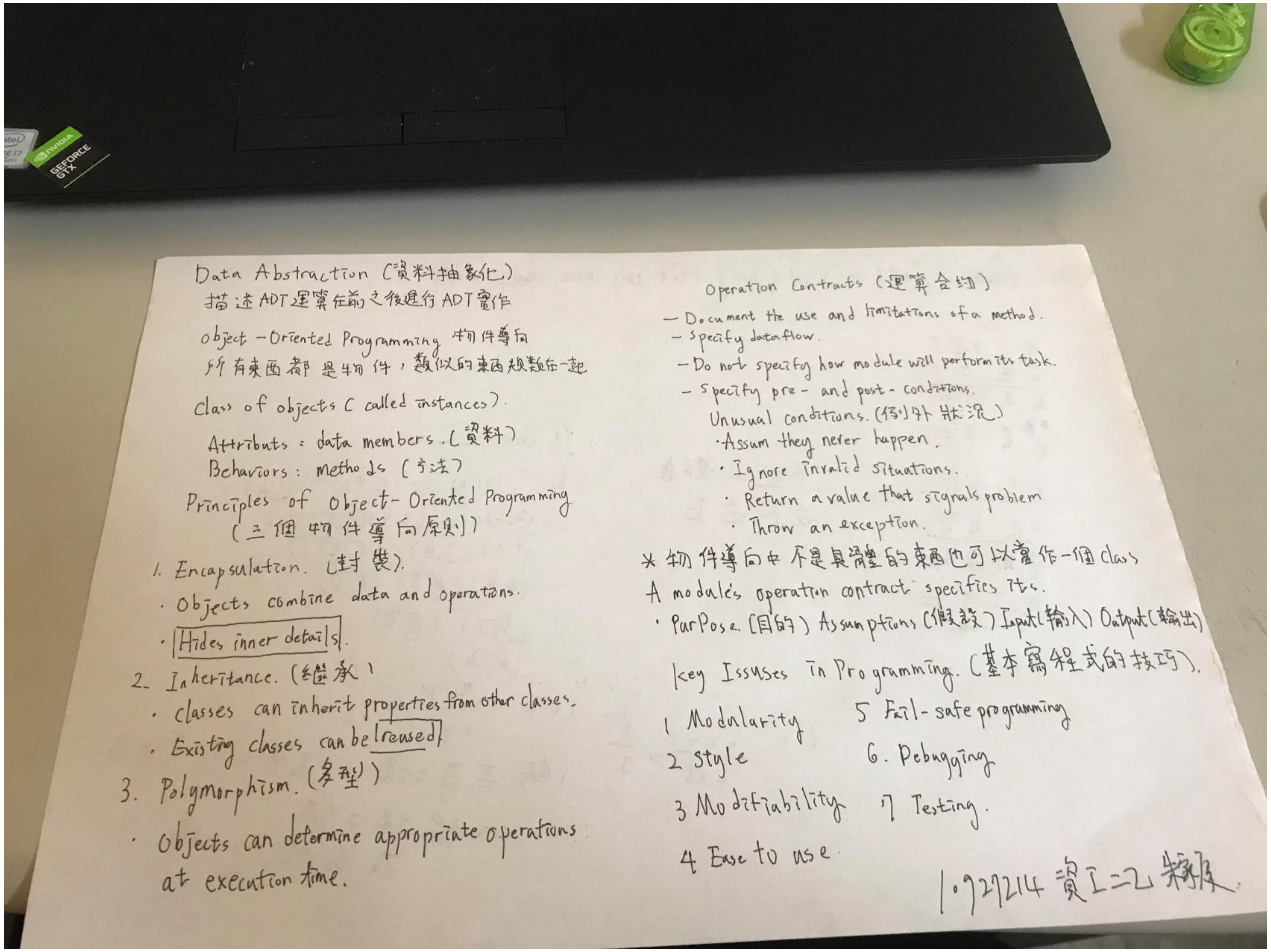
Leaf nodes + Internal nodes = C(n,k) 的遞迴呼叫次數.

Better Recursive Solution

$$C(n,k) = \frac{n!}{(n-k)!\,k!} = \frac{n\,(n-1)!}{k\,(n-k)!\,(k-1)!} = C(n-1,k-1)$$

n 變小
k 變小 (比較好求).

109 27214 資工二乙 朱家慶



Data Abstraction (資料抽象化)
描述ADT運算在前之後進行ADT實作

object -Oriented Programming 物件導向
所有東西都是物件，類似的東西規類在一起

Class of objects (called instances)

Attributs : data members . (資料)
Behaviors : methods (方法)

Principles of Object- Oriented Programming
(三個物件導向原則)

1. Encapsulation. (封裝).
· Objects combine data and operations.
· Hides inner details.

2. Inheritance. (繼承)
· classes can inherit properties from other classes.
· Existing classes can be reused.

3. Polymorphism. (多型)
· Objects can determine appropriate operations at execution time.

Operation Contracts (運算合約)

— Document the use and limitations of a method.
— Specify dataflow.
— Do not specify how module will perform its task.
— Specify pre - and post - conditions.
Unusual conditions. (例外狀況)
· Assum they never happen.
· Ignore invalid situations.
· Return a value that signals problem
· Throw an exception.

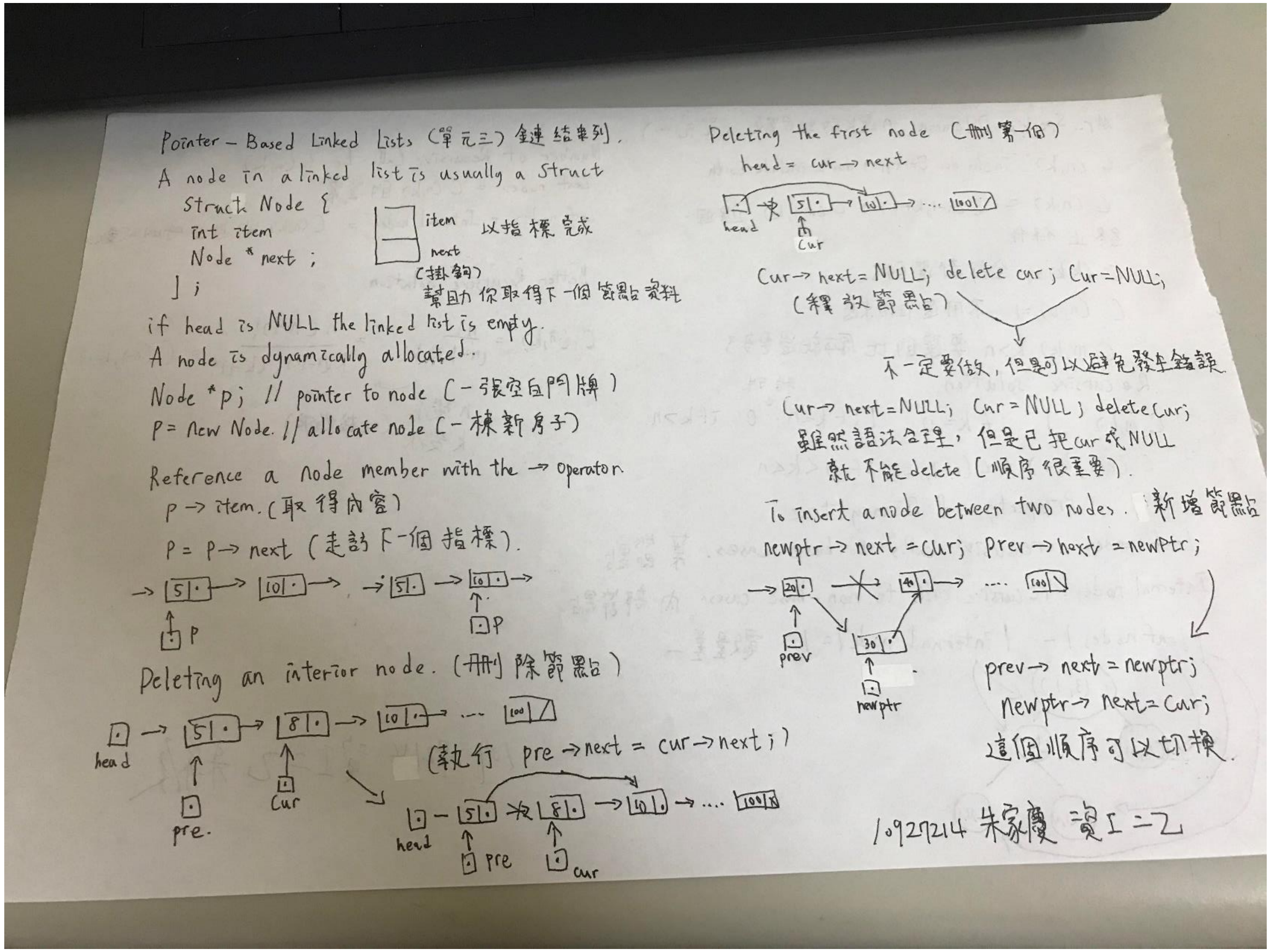
* 物件導向中不是具體的東西也可以當作一個 class

A module's operation contract specifies its.
· PurPose (目的) Assumptions (假設) Input (輸入) Output (輸出)

key Issuses in Programming. (基本寫程式的技巧).

1 Modularity
2 Style
3 Modifiability
4 Ease to use
5 Fail-safe programming
6. Debugging
7 Testing.

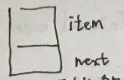
1.927214 資工二乙 [illegible]



Pointer - Based Linked Lists (單元三) 鏈結串列.

A node in a linked list is usually a Struct

```
Struct Node {
  int item
  Node * next ;
} ;
```



以指標完成
(掛鉤)
幫助你取得下一個節點資料

if head is NULL the linked list is empty.

A node is dynamically allocated.

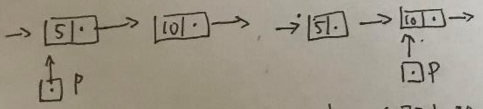
Node *p; // pointer to node (一張空白門牌)

p = new Node. // allocate node (一棟新房子)

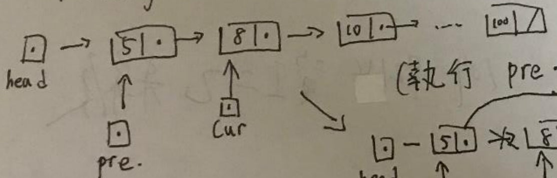
Reference a node member with the → operator

p → item. (取得內容)

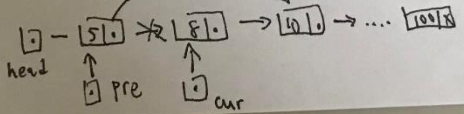
p = p → next (走訪下一個指標).



Deleting an interior node. (刪除節點)

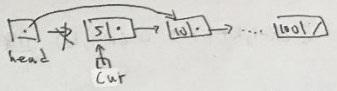


(執行 pre → next = cur → next ;)



Deleting the first node (刪第一個)

head = cur → next



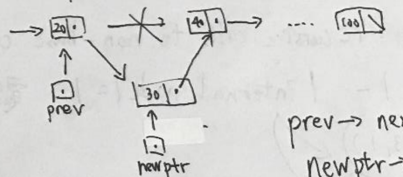
Cur → next = NULL; delete cur ; Cur = NULL;
(釋放節點)

不一定要做，但是可以避免發生錯誤.

Cur → next = NULL ; Cur = NULL ; delete cur;
雖然語法合理，但是已把cur成NULL
就不能delete (順序很重要).

To insert a node between two nodes. 新增節點

newptr → next = cur; prev → next = newptr ;



prev → next = newptr;
newptr → next = cur;
這個順序可以切換.

10927214 朱家慶 資工二乙