chapter 1.

遞迴：可以由大問題分解為小問題去解
程式內容較精簡
呼叫同一個程式重複解相同的問題

Recursive Functions

- Factorial 階乘
- Greatest Common Divisor 最大公因數
- Search in Array 搜尋
- Fibonacci series 費式數列
- Combinatorial numbers 組合數
- Towers of Hanoi 河內塔.

遞迴過程

1. 遞迴定義
2. 問題簡化
3. 終止條件
4. 保證終止

Greatest Common Divisor

① $gcd1(x,y) = x$ if $y = 0$
$= gcd1(x, y \bmod x)$ if $y > x$
$= gcd1(y, x \bmod y)$ otherwise

② $gcd2(x,y) = y$ if $x \bmod y = 0$
$= gcd(y, x \bmod y)$ otherwise

base cases
gcd2 saves one recursive call
when x divides y.

overall :
gcd2 is more efficient
whenever $x \geq y$.

special case
gcd2 spends one extra call
if initially $x < y$.

chapter 2

Data Abstraction 資料抽象化

- Principles of Object - Oriented Programming
  - object - oriented languages enable us to build classes of objects (called instances)
  - A class combines
    - Attributes (characteristics) of objects of a single type
      - Typically data
      - Called data members.
    - Behaviors (operations)
      - Typically operate on the data
      - Called methods or member functions.
  - Three characteristics
    - Encapsulation 封裝
      - object combine data and operations.
      - Hides inner details.
    - Inheritance 繼承
      - Classes can inherit properties from other classes
      - Existing classes can be reused.
    - Polymorphism 多型
      - Object can determine appropriate operations at execution time.
- Operation Contracts 運算合約
  - Document the use and limitations of a method
  - Specify data flow
  - Do not specify how module will perform its task
  - Specify pre- and post- conditions.
  - Unusual conditions. 例外狀況
    - Assume they never happen
    - Ignore invalid situations
    - return a value that signals a problem
    - Throw an exception

chapter 2

- A module's operation contract specifies its
  - purpose
  - Assumptions
  - Input
  - Output

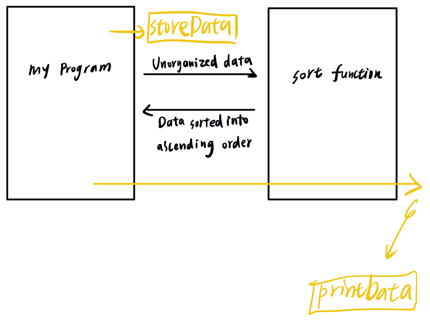
* Begin the contract during analysis, finish during design.

* Use to document code, paticularly in header files.

- Modularity 模組化
  - Cohesion – modules perform single well-defined tasks.

    highly cohesive modules desired 高內聚
  - Coupling – measure of dependence among modules.

    Loosely coupled modules desired 低耦合

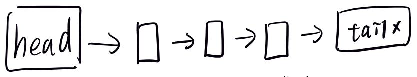


- Information hiding 資訊隱藏（類似封裝）
- Typical operations on data
  - Add data to a data collection
  - Remove data from a data collection
  - Ask questions about the data in a data collection.

Abstract Data Type (ADT)

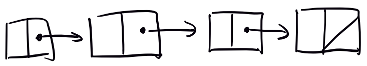
- An ADT is composed of $\left\{ \begin{array}{l} \text{A collection of data} \\ \text{A set of operations on that data} \end{array} \right.$
- Specifications of an ADT indicate 描述
- Implementation of an ADT 實作

chapter 3

- 

  link list 鏈結串列

  - array ↔ Link list
  - preliminaries



- Pointer

  p = & x

  p = new int

  delete p

  p = NULL.

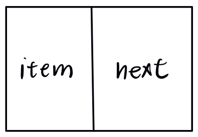
- 動態陣列

  int arraysize = 50;

  AnArray = new double [arraysize];

  delete [] AnArray (歸還)

- struct



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

- Array vs. pointer
  - Size : Linked List grow and shrink as necessary.
  - storage : array requires less memory.
  - retieval : Array 常數時間快, Link List 線性時間慢
  - Insert / delete : 若資料多, Array 會比較慢.

- Restoring Linked List by using file :
  - ofstream outFile

    outFile << item 寫檔

    outFile close; ✲✲
  - ifstream inFile

    inFile >> nextitem 讀檔

    inFile close;

chapter 4

- The Basic of grammer → recognition algorithm 辨識演算法

  is Id 遞迴 　 終止條件：單一數 / 遞迴呼叫：排除最後一個字元

  isId suffix

- Palindromes 回文

  ex：38+83 =121

  $<pal> =$ empty string $\overset{\downarrow \text{base}}{< ch >}$ | a<pal>a | b<pal>b | …… | z<pal>z

  $<ch> =$ a|b|c…|z

- 描述到下一個字母所組成a字串：
  (第1個字母必須為大寫，而其餘為小寫)

  Algebraic Expression

  - infix 　 中序運算式 　 ex：a+b
  - prefix 　 ex：+ab 　 前序
  - postfix 　 ex ab+ 　 後序

  Advantages：
  - No precedence rules 優先權
  - No association 結合率
  - No parenthese 括弧

  - prefix
    - Grammar

      <prefix> = < identifier> | < operator> < prefix><prefix>

    - recursive recognition
      - Base case ： one lowercase letter is a prefix exp.
      - recursive ： < operator> < prefix><prefix>

  * Important ： 一個前序式後面接著非空字串，一定不是前序式

- 遞迴 and 數學歸納法關係
  - both use base case
  - both solve smaller problems to derive a solution.
  - 工作量

    * 河內塔