



# 程式語言 (Programming Language)

## Lec. 00 課程規劃介紹

---



楊 吳 泉

義守大學資訊工程學系

web: <http://elearning.isu.edu.tw>

<http://audtm.net/~wcyang>

mail: [wcyang@isu.edu.tw](mailto:wcyang@isu.edu.tw)



# 綱要 (Outline)

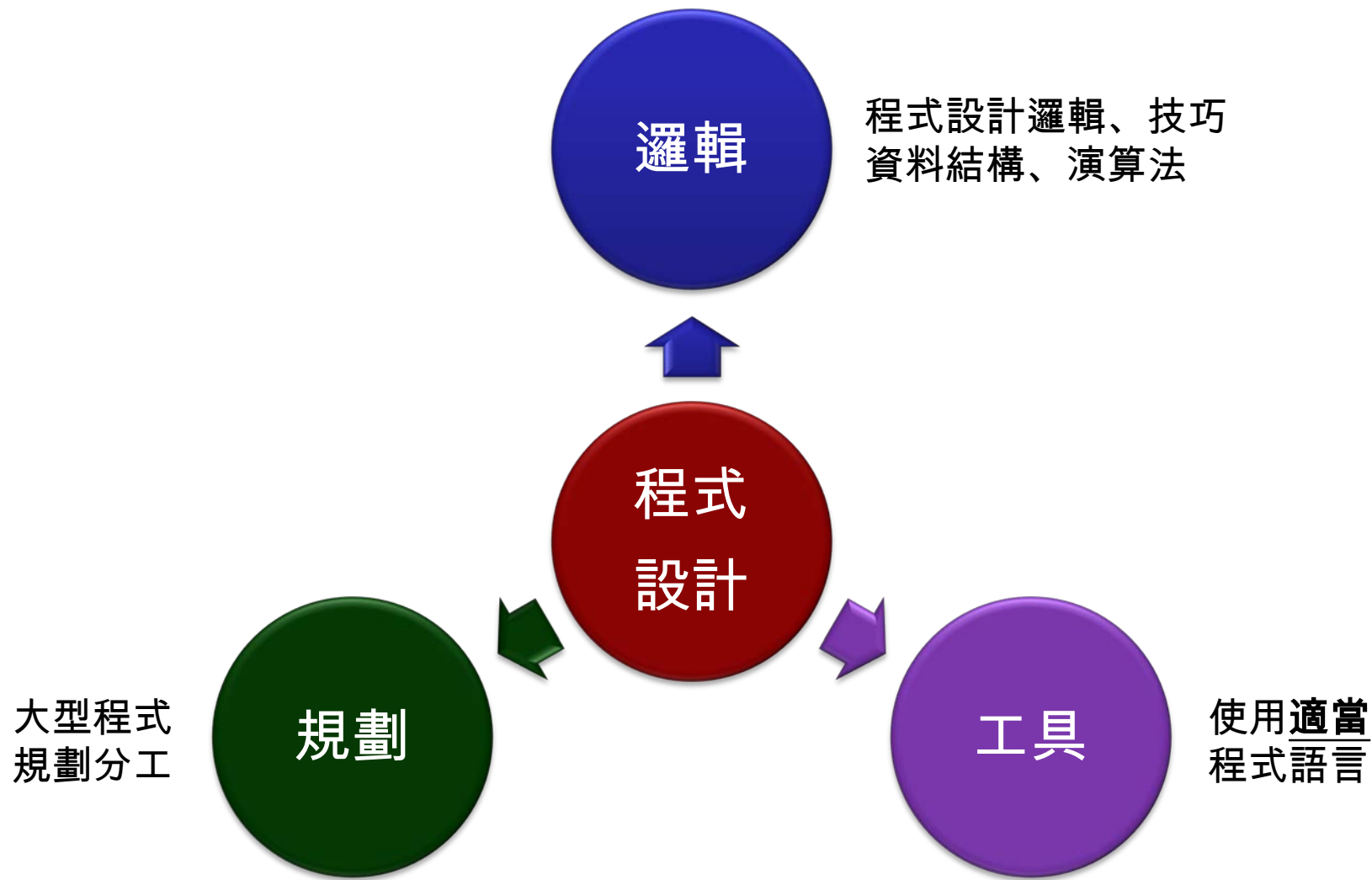
---

1. 背景介紹
2. 參考書籍
3. 課程進度
4. 範例介紹





# 1. 背景介紹





# 程式語言共通特性

## ➤ 程式語言共通特性

此為具有方法、  
繼承、類別架  
構的Script語言  
範例

```
1 sub Calculator()  
2   sub addition(self, other)  
3     return self + other  
4   end  
5  
6   /** Create a list of 2 numbers. */  
7   sub makeFraction(numerator, denominator)  
8     return [numerator, denominator]  
9   end  
10  
11   /** Warning: Destroys original fraction! */  
12   sub multiplyFrac(frac, otherFrac)  
13     frac[0] *= otherFrac[0]  
14     frac[1] *= otherFrac[1]  
15     return frac  
16   end  
17 end  
18  
19 sub InfinityCalculator()  
20   inherit Calculator()  
21   /** Create a list of 2 numbers. */  
22   sub makeFraction(numerator, denominator)  
23     if denominator == 0  
24       /* The user is trying to divide by 0.  
25        * Use Java's way of handling this: */  
26       import math into mathematics  
27       return mathematics.INFINITY  
28     end  
29     return [numerator, denominator]  
30   end  
31 end
```

← Method

← Class

了解程式語言共通特性有助於  
學習各種不同的程式語言

From: [http://en.wikipedia.org/wiki/Programming\\_language](http://en.wikipedia.org/wiki/Programming_language)



# 常見程式語言類型

## ➤ 程式語言類型(參考:[http://www.oreilly.com.tw/column\\_sleepless.php?id=j006](http://www.oreilly.com.tw/column_sleepless.php?id=j006))

- ✓ Web Page Script Languages
  - 用來做網頁的語言, 可以對網頁做控制, ex: JavaScript
- ✓ Interpreted Languages ( 直譯式語言 )
  - 常被稱為 script 語言, 直接執行每一程式敘述執行
  - 通常比較高階, 程式比較好寫, ex: Perl, Python, Ruby
- ✓ Hybrid Languages ( 混合式語言 )
  - 介於直譯式語言和編譯式語言之間, ex: Java, C#, Visual Basic
- ✓ Compiling Languages ( 編譯式語言 )
  - 典型架構程式語言, 程式整體編譯後再執行, ex: C/C++, Delphi
- ✓ Assembly Languages ( 組合語言 )
  - 最接近硬體的語言, 可以對電腦的運作有相當程度了解



# 程式語言結構與目標

---

## ➤ 程式語言結構

- ✓ 程式架構與解析
  - 語法要素、語彙分析、語法剖析與語意分析
- ✓ 變數宣告與使用
- ✓ 資料型態
- ✓ 運算式
- ✓ 流程控制
- ✓ 副程式使用與撰寫
- ✓ 程式封裝與物件導向概念



# 課程目標

---

## ➤ 溫故知新

- ✓ 從較熟悉語言到較陌生語言之學習
  - 深入瞭解程式語言架構、容易學習新的程式語言並更了解原先之程式語言
  - **主要: From C/C++ to Java**
  - 也可對其他語言做了解，以Script語言為主



## 2. 參考書籍

### ➤ JAVA SE 7基礎必修課

- ✓ 蔡文龍、張志成、何玉安 編著 / 吳明哲 策劃
- ✓ 出版社： 碁峰資訊
- ✓ 出版日期：2012/10/12



### ➤ Thinking in Java (4<sup>th</sup> Ed)

- ✓ Bruce Eckel, Prentice-Hall Inc.
- ✓ Free e-Books(3<sup>rd</sup> Ed), <http://www.mindview.net/Books/TIJ/>
- ✓ Thinking in Java中文版(4<sup>th</sup> Ed.)
  - 王建興、王舜正譯, 碁峰出版, 2008.08

### ➤ Programming Languages (10<sup>th</sup> Ed.)

- ✓ Robert W. Sebesta, Addison Wesley





### 3. 課程進度

週次	內容與進度	週次	內容與進度
1	課程內容簡介	10	程式封裝概念
2	程式架構 – 語意與語法	11	物件概念 I
3	變數宣告與使用	12	物件概念 II
4	資料型態	13	Java程式語言 I
5	運算式	14	Java程式語言 II
6	流程控制	15	Case Study I
7	副程式使用	16	Case Study II
8	副程式撰寫	17	Case Study III
9	期中考	18	期末考



## 4. 範例介紹

---

### ➤ 猜數字題目

- ✓ 系統產生一個 1-100 的亂數
- ✓ 提示使用者輸入一個數字
- ✓ 猜中，印出 “Successful!” 並結束程式  
否則提示使用者答案較大或較小
- ✓ 如果5次沒猜中，則印出標準答案

### ➤ 猜數字作法提示

- ✓ 亂數產生可以以目前時間作亂數種子來產生亂數
  - e.g. `srand(time(NULL));`
- ✓ `rand()`產生0 到 `RAND_MAX` 之間的一個整數值，  
`RAND_MAX` 是 `cstdlib` 定義之常數，在不同系統有不同值
  - 產生1-100之亂數可用：`(rand() % 100) + 1`



# 設計構想

---

## ➤ 亂數產生方法提示

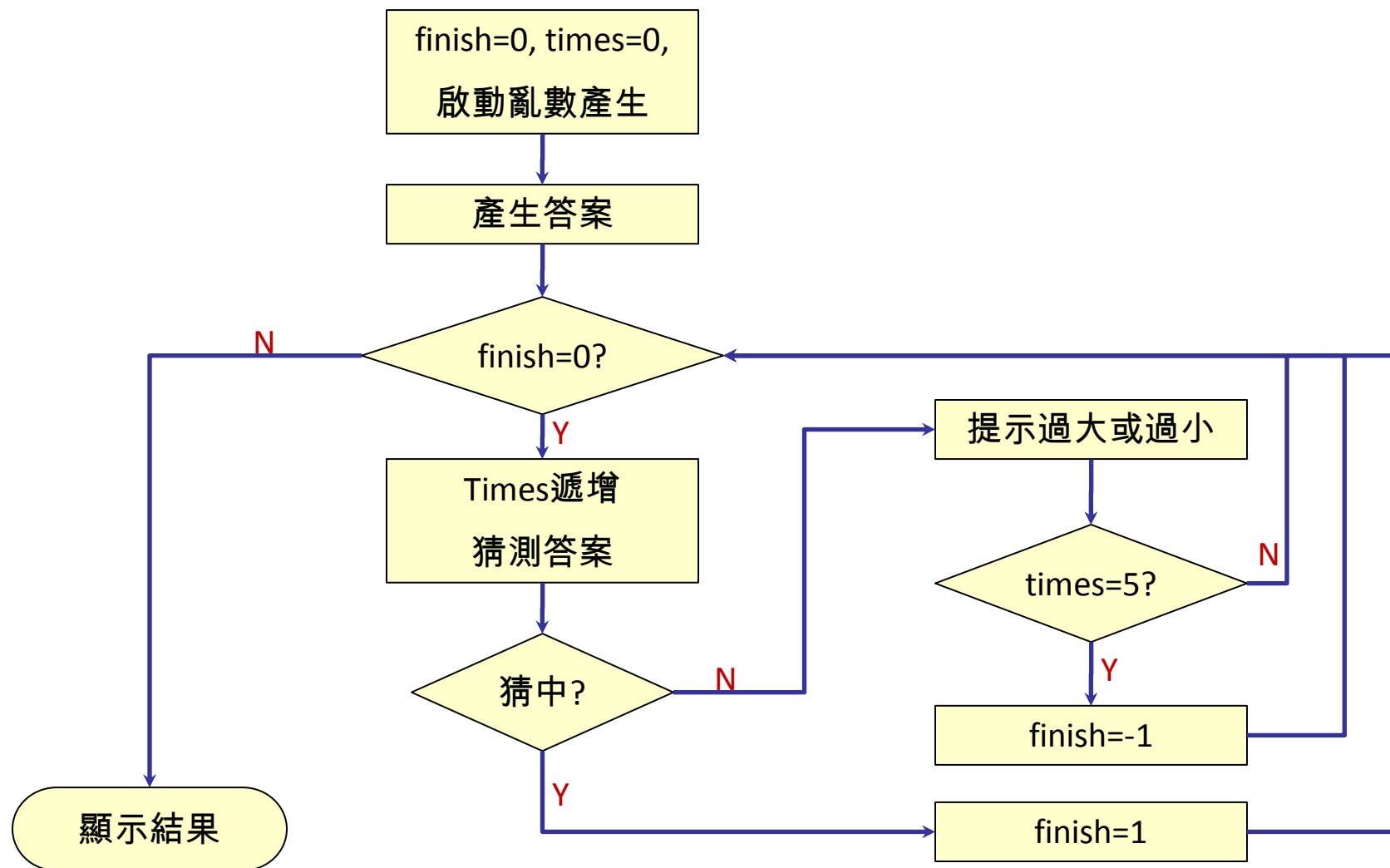
- ✓ 亂數產生可以以目前時間作亂數種子來產生亂數
  - e.g. `srand(time(NULL));`
- ✓ `rand()`產生0 到 `RAND_MAX` 之間的一個整數值，  
`RAND_MAX` 是 `cstdlib` 定義之常數，在不同系統有不同值
  - 產生1-100之亂數可用：`(rand() % 100) + 1`

## ➤ 設計構想

- ✓ 相關變數：`guess`(猜測數字), `ans`(正確答案), `times`(猜測數字)
- ✓ 旗號：`finish` - 次數達到5次或猜中數字時，遊戲終止
- ✓ 因為迴圈次數未能確定 – 使用`while`迴圈較適合



# 程式流程





# C++ 參考程式

```
#include <ctime>
#include <cstdlib>
#include <iostream>
using namespace std;
```

```
cout<<endl<<endl;
if(finish==1) cout<<"Successful! ";
else cout<<"Fail! The answer is "<<ans;
cout<<endl<<endl;
system("PAUSE");
return EXIT_SUCCESS;
}
```

```
int main(int argc, char *argv[])
{
    int ans,guess,times=0,finish=0;
    srand(time(NULL));
    ans = rand()%100+1;
    cout<<"Guess a number between 1~100: "<<endl;
    while(finish==0)
    {
        times++;
        cout<<" Guess "<<times<<" : ";
        cin>>guess;    cout<<"      ";
        if(guess==ans) finish=1;
        else if(guess>ans) cout<<"too big!"<<endl;
        else cout<<"too small!"<<endl;
        if(times==5 && finish==0) finish=-1;
    }
}
```



# Java程式架構

[**package** 定義; ]  
[**import** 相關類別;]

□ Java為純物件導向語言，語法與C非常類似。

[修飾字] **class** 類別名稱  
[**extends** 父類別]  
[**implements** 相關介面s]  
[**throws** 相關例外s] {

變數與方法定義;

```
[ //如果需要主程式(main method)
  public static void main(String args[]) {
    變數與敘述 ;
  }
]
```



# Java參考程式

```
import java.io.*;  
import java.util.Random;
```

```
public class PLex01 {
```

```
    public static void main(String[] args)  
        throws IOException {
```

(A)

```
}
```

```
}
```

(A)

```
int ans,guess,times=0,finish=0;
```

```
Random ran = new Random();
```

```
BufferedReader br =
```

```
    new BufferedReader(new InputStreamReader(System.in));
```

```
ans = ran.nextInt(100)+1;
```

```
System.out.println("Guess a number between 1~100: ");
```

```
while(finish==0)
```

```
{
```

```
    times++;
```

```
    System.out.print(" Guess "+times+" : ");
```

```
    guess = Integer.parseInt(br.readLine());
```

```
    System.out.print("      ");
```

```
    if(guess==ans) finish=1;
```

```
    else if(guess>ans) System.out.println("too big!");
```

```
    else System.out.println("too small!");
```

```
    if(times==5 && finish==0) finish=-1;
```

```
}
```

```
System.out.print("\n\n");
```

```
if(finish==1) System.out.println("Successful! ");
```

```
else System.out.println("Fail! The answer is "+ans+"\n\n");
```