資料輸入與選擇性邏輯

羅孟彥
myluo@kuas.edu.tw
高雄應用科技大學資訊工程系

以程式習作引導思考

- 假設我們現在要寫一個簡單的自我健康管理程式，程式需求如下:
- 程式輸入:讓使用者輸入他的姓名、身分證字號、年齡、身高、體重。
- 資料處理:計算身體質量指數(BMI) ：
  - ✓ 體重 (公斤) / 身高^2(公尺^2)
- 程式輸出:顯示使用者的個人資料
  - ✓ 例:某某某 先生(或小姐) 您的體重是正常
- 如何判斷以及告知體重是否標準?
  - ✓ BMI值大於18.5以及小於22.9則體重屬於正常

Expression and Statement

- 程式的最主要目的為處理資料，因此，程式設計即是將所需要做的資料處理工作以敘述句描述出來。
- 將常數、變數或函數以各種運算子聯結起來的組合稱之為運算式（expression）：

```
x < 15.8
x = y + sin( z )
( x + y ) * 0.8 – z
```

- 敘述句（statement）是指完整的運算式(以分號做結尾)，為程式裡面最小的可執行單元，如:

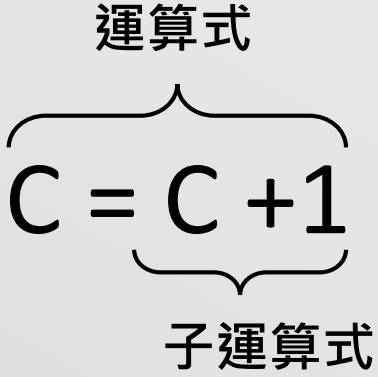
```
x = a + b * sin( c );
b++;
y = ( b > 10.0 );
x = c + 5.0;
```

運算式的組成

- 運算子：程式中用以運算的符號
- 運算元：被運算的常數或變數
- 運算式：運算子和運算元組成用以運算的式子，例如：

```
a+1
7+8
5+9*3-b
x=6
```

- 運算式的結果也是一個值，所以運算式可以視為另一個運算式的運算元



敘述 (Statement)

- 一個敘述（statement）為一運算式後面加一個分號（;），代表一行完整的指令。

- 不是每個運算式都可以加上分號而變成敘述

```
C + 1;              //錯誤
```

敘述句的規範

1. 以分號當結尾
2. 可以任意放置。例如：

```
x = a + b * ( c + 6.5 ) +
c * 105.8 - 4.0;
```

相同於

```
x = a + b * ( c + 6.5 ) + c * 105.8 – 4.0;
```

3. 各個數值或變數名稱與運算子 (operator) 之間的**空隔可有可無**。例如：

```
x=a+b;
```

相同於

```
x = a + b ;
```

運算子

- Java語言有**四十餘種**運算子，提供各種處理資料的功能。
- 要瞭解Java語言的運算子可由三個面向：
  - ✓ 功能
  - ✓ 結合序，有些運算是**由左而右計**算，這稱為**左結合**的(left associative)運算子，有些則**由右而左**計算，這稱為**右結合**的(right associative)運算子。
  - ✓ 運算優先順序(precedence)

指定運算子

- 指定運算子的功用：指定值給某一變數
- **設定常數值給變數**

```
a = 9;
```

- **將變數的內容指定給另一個變數**

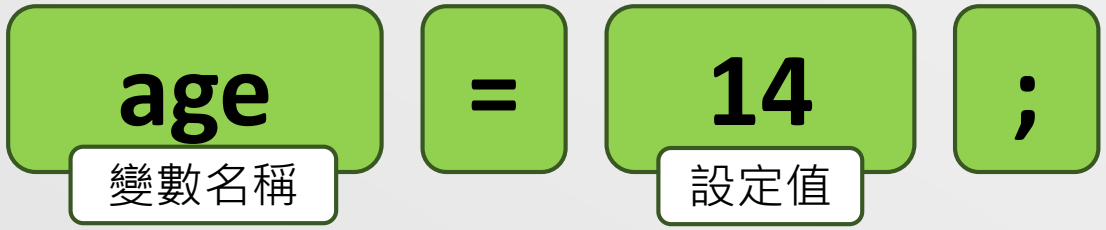
```
a = b;
```

- **將運算式的結果指定給變數**

```
a = b + 7 * 5;
```

指定運算子

- = 的左邊一定是變數，不能是數值。



- 結合性是由右至左：

```
a = b = c = d = 5;    //多重指定
```

程式範例：指定運算子

- 下面的程式碼，是設定運算子的範例：

```
1  // app4_1,設定運算子「=」
2  public class app4_1
3  {
4      public static void main(String args[])
5      {
6          int age = 18; //宣告整數變數age,並設值為18
7
8          System.out.println("before compute , age =" + age);// 印出 age 的值
9          age = age + 1 ; // 將 age 加 1 後再勝定給 age 存放
10         System.out.println("after compute, age =" + age); // 印出計算後 age 的值
11     }
12 }
```

app4_1 OUTPUT

before compute , age =18
after compute, age =19

複合運算子

- 若「=」符號右邊的第一個運算元和「=」左邊的變數名稱一樣時，則可改為「複合指定運算子」，以達簡化運算式的功能，比較常用的複合指定運算子如下：

運算子	名稱	用法	代表意義
=	指定	x = y	將y值指定給x
+=	加法指定	x += y	x = x – y
-=	減法指定	x -= y	x = x – y
*=	乘法指定	x *= y	x = x * y
/=	除法指定	X /= y	x = x / y
%=	餘數指定	x %= y	x = x % y

算術運算子

運算子	語法	說明	a=5, b=2的運算結果
+	a + b	a 加 b	7
-	a – b	a 減 b	3
*	a * b	a乘b	10
/	a / b	a除以b	2.5
%	a % b	a除以b的餘數	1

- 算術運算子的結合性是由左至右
- 算術運算子的優先序事先乘除%後加減

1+2*3-4/5%6 → **結果為7;**

算術運算子

減法運算子「-」

- 下面的敘述為減法運算子的範例：

```
age =age-1; // 計算age-1 之後，再將其結果設定給 age 存放
c=a-b;      // 計算 a-b 之後，再設定給c存放
54-12;      // 計算 54-12 的值
```

乘法運算子「*」

- 下面的敘述為乘法運算子的範例：

```
b=c*3;      // 計算 c*3 之後，再將其結果定給b存取
a=a*a;      // 計算 a*a 之後，在設定給a存放
17*5;       // 計算 17*5 的值
```

算術運算子

除法運算子「/」

- 下面的敘述為除法運算子的範例：

```
b =a/6;     // 計算a/6 之後，再將其結果設定給 b 存放
d=c/d;      // 計算 c/d 之後，再設定給d存放
3/8;        // 計算 3/8 的值
```

餘數運算子「%」

- 下面的敘述是使用餘數運算子的範例：

```
age=age%5;       // 計算 age/5 的餘數，再將計算的結果設定給 age 存取
c=a%b;           // 計算 a/b 的餘數，再將計算的結果設定給 c 存放
48%7;            // 計算 48%7 的值
```

遞增遞減運算子（1/2）

- **++為遞增運算子(Incrementing Operator)**
  - ✓ ++運算乃將運算式的值加1。
- **––為遞減運算子(Decrementing Operator)**
  - ✓ ––運算則是將運算式的值減1。
- **前置(Prefix)運算**
  - ✓ 將++(或––)放在運算式的前面，表示先做++(或––)運算之後才做運算式的計算。
- **後置(Postfix)運算**
  - ✓ 將++(或––)放在運算式的後面，表示先做完運算式的計算之後才做++(或––)運算。

遞增遞減運算子（2/2）

- i++;相當於
  - ✓ i = i + 1;
- i--;相當於
  - ✓ i = i - 1;
- J = j + (++i);相當於
  - ✓ i = i + 1; 和 j = j + i;
  - ✓ 先將i本身的值加1，然後再將 j+i 的結果指定給 j

移位運算子(Shift Operators)

- **>>(向右移位)**
  - ✓ 向右移1位，表示除以2的1次方，向右移2位， 表示除以2的2次方。
  - ✓ 24 >> 1；結果為12，亦即24/2

- **<<(向左移位)**
  - ✓ 左移1位，表示乘以2的1次方，向左移2位，表 示乘以2的2次方。
  - ✓ 1 << 12；結果為24，亦即12*2

優先序	運算子	運算子的功能	由左而右	由右而左
1	( )	小括號	Yes	
	[ ]	中括號	Yes	
2	++	遞增		Yes
	--	遞減		Yes
	!	否定(一元運算子)		Yes
	+	正(一元運算子)		Yes
	-	負(一元運算子)		Yes
	~	位元邏輯非		Yes
3	*	乘	Yes	
	/	除	Yes	
	%	取餘數	Yes	
4	+	加	Yes	
	-	減	Yes	
5	>>	向右移位	Yes	
	<<	向左移位	Yes	
	>>>	去負號向右移位	Yes	

運算子的優先順序

優先序	運算子	運算子的功能	由左而右	由右而左
6	>	大於	Yes	
	>=	大於等於	Yes	
	<	小於	Yes	
	<=	小於等於	Yes	
7	==	是否等於	Yes	
	!=	是否不等於	Yes	
8	&	且(位元邏輯)	Yes	
9	^	互斥或(位元邏輯)	Yes	
10	|	或(位元邏輯)	Yes	
11	&&	且(邏輯)	Yes	
12	||	或(邏輯)	Yes	
13	?:	條件		
14	=	指定		Yes
	op =	複合指定		Yes

運算子的優先順序

算術運算時的自動型別轉換

- 對於整數而言，經由算術運算的結果都會自動轉型為int，example：

```
Public class test{
        public static void main(String[] args){
                byte b = 3;
                byte c = 8;
                byte d = b + c;
                System.out.println(" d = " + d);
        }
}
```

- 結果：0005：possible loss of precision

  1 error

- 應該改為 d = (byte) (b + c)

算術運算時的自動型別轉換

- 對於浮點數而言，如果其中一個運算元是double 的話，運算的結果就會自動轉型為double;
- 如果其中一個運算元是float的話，運算的結果就會自動轉型為float。

```
int i = 1;
int sum;
float f = 1.0f;
sum = i +f ;
```

- 結果： 0005：possible loss of precision
  1 error
- 應該改為 **sum = (int) (i + f)**

算術運算時的 overflow 與 underflow

```
public class test{
        public static void main(String[] args){
                int a = 12345, b = 234567 , c, d;
                c = a*b/b;
                d = a/b*b;
                System.out.println(" a is " + a + " b is " + b + " c is " + c + " d is " + d);
        }
}
```

- 結果： a is 12345 b is 234567 c is -5965 d is 0

以程式習作引導思考 (課堂練習)

- **程式輸入：讓使用者輸入他的姓名、身分證 字號、年齡、身高、體重。**
- **資料處理:計算身體質量指數(BMI) ：**
  - ✓ 體重 (公斤) / 身高^2(公尺^2)
- **程式輸出:顯示使用者的個人資料**
  - ✓ 例:某某某 先生(或小姐) 您的體重是正常
- **如何判斷以及告知體重是否標準?**
  - ✓ BMI值大於18.5以及小於22.9則體重屬於正常

程式範例：如何跟使用者互動

- 寫個程式能讓使用者從鍵盤輸入一串文字 。
- 我們首先需要一個能夠取得使用者輸入的「程式零件」:
  - ✓ BufferedReader buf;
    - //宣告程式需要一個輸入資料的物件
    - //將這個物件命名為buf
  - ✓ buf =new BufferedReader (new InputStreamReader (System.in));
    - //實際產生一個程式零件，也就是建造出一個物件

如何跟使用者互動 (Cont.)

- BufferedReader類別是用做「緩衝字元讀入資料流」，buf是BufferedReader類別的物件變數名稱，它將資料暫時儲存到緩衝區，InputStreamReader用來把輸入的字元資料編碼，使用System.in傳遞參數，表示從鍵盤輸入字元的 資料流。

- 也可寫成:
  - ✓ **BufferedReader buf = new BufferedReader(new InputStreamReader (System.in));**
    - buf是由你自行命名的物件變數名稱

如何跟使用者互動：輸入資料

- 每一個程式零件都有它的作用(專屬功能)，用專業術語來說就是物件的**方法(method)**
- 啟動(呼叫)一個物件的語法: **物件名稱.方法**
  - ✓ String str=buf.readLine();
    - // 啟動buf這個物件的一個名為readline的方法
    - // 這個方法的作用是能夠讀取使用者在鍵盤的輸入
    - // 所以這行程式的作用是藉由呼叫buf這個物件的readline這個方法，來捕捉使用者在鍵盤輸入的字元，再將這些字元存在一 個被命名為str的變數中
    - // 如果readline()傳回有錯誤時，此時會丟出IOException的例外。所以我們在**「main()方法後面加上throws IOException」**，才可以讓程式順利完成編譯與執行

如何跟使用者互動：輸出資料

- 最後，要將結果顯示出來，嗯，還需要另 一個物件

```
System.out.println()
```

System物件　out變數　println方法

System：java 提供的一個類別，其中定義了所有與標準輸出入的相關動作。

out： System 類別中的一個變數。其型態為 PrintStream 類別 。

println： PrintStream 類別內的一個方法，由於 out 變數本身指向一個 PrintStream 物件，所以我們得以透過 out 呼叫 println 方法。

程式範例：跟使用者互動

```
import java.io.*;
Public class test1
{
        public static void main (String[] args) throws IOException
        {
                BufferedReader buf;
                String str;
                buf = new BufferedReader(new InputStreamReader(System.in));
                System.out.println("請問你對Java的感覺如何：");
                str = buf.readLine();
                System.out.println(str);
        }
}
```

Piece of cake！那せ架乾單！

等一下！

- import java.io.*; 這是什麼？？？
- 這行指令敘述稱為import敘述，當我們要使用一個內建的類別時，必須用import這個敘述宣告「我將要在程式中使用這一個類別」。
- 藉由這行import敘述，電腦才會將這個類別的相關資料事先匯入，(import即是匯入、載入的意思)。

如何取得數字？

- BufferedReader類別所讀到的資料型態為字串，必須經過轉換才可變成數值型態。
- **程式輸入**：讓使用者輸入一個數字
- **程式輸出**：顯示使用者輸入的數字
- 可是...buf讀取的結果是一個字串，可是要變成整數...，嗯，需要另一個物件
- num=Integer.parseInt(str);
  - ✓ //引用Integer這個物件
  - ✓ //呼叫Integer這個物件裡的一個功能—parseInt()將所得到的字串轉成整數
  - ✓ //再將這個整數指定給num這個變數

如何取得數字

- **範例程式**：讓使用者輸入一個數字，然後回應「你輸入的數字是*」

```
1  import java.io.BufferedReader;
2  import java.io.IOException;
3  import java.io.InputStreamReader;
4  public class app4_2 {
5      public static void main ( String[] args ) throws IOException
6      {
7        BufferedReader buf = new BufferedReader(new InputStreamReader(System.in));
8        System.out.println("請輸入一個整數：");
9        String str = buf.readLine();
10       int num = Integer.parseInt(str);
11       System.out.println("你剛剛輸入的是 " + num);
12     }
13 }
```

字串轉換

- 其他幾種常用的轉換方法

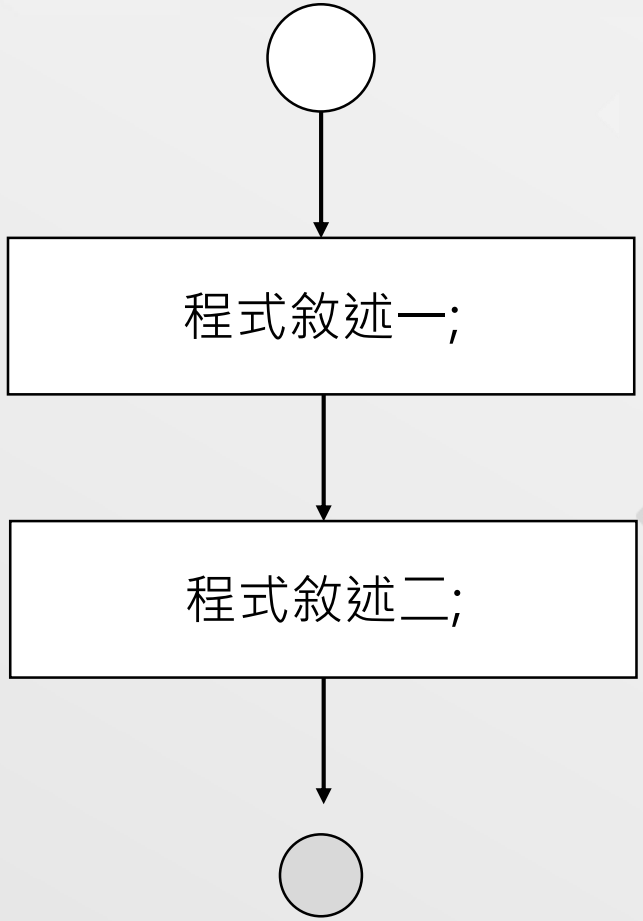
資料型態	轉換的method()
Long	Long.parseLong()
int	Integer.parseInt()
short	Short.parseShort()
byte	Byte.parseByte()
double	Double.parseDouble()
float	Float.parseFloat()

- ✓ int num=Integer.parseInt(字串變數);
- ✓ long num=Long.parseLong(字串變數);
- ✓ float num=Float.parseFloat(字串變數);
- ✓ double num=Double.parseDouble(字串變數);

選擇性敘述

陳述式的種類

- 一般陳述式
  - ✓ 註解
    - 指定陳述式
    - 物件的使用
- 宣告陳述式
  - ✓ 基本型態的宣告
  - ✓ 物件的宣告



- 以上均為**循序執行**
- 可以利用**選擇敘述**或**迴圈敘述**改變流程方向

流程控制敘述

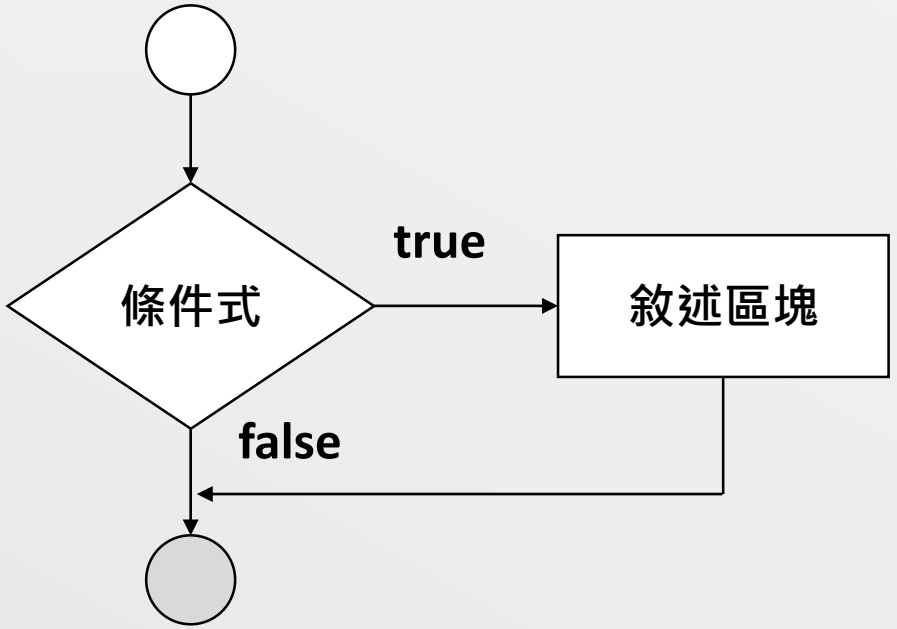
- 五種控制敘述：
  - if/else
  - switch
  - for
  - while
  - do/while

if/else選擇敘述

流程控制 --選擇敘述

- 選擇敘述

```
if(條件式)
{
    程式敘述
}
```



- 條件式為一運算式，而其結果必須為布林值。
- { } 包起來的程式敘述為敘述區塊 (block)。
- 若條件式為true，執行敘述區塊。
- 若條件式為false，則跳過敘述區塊。

if 敘述常需搭配關係（比較）運算子

關係運算子	說明	範例	範例結果
>	是否大於	5 > 2	true
>=	是否大於等於	5 >= 2	true
<	是否小於	5 < 2	false
<=	是否小於等於	5 <= 2	false
==	是否等於	5 == 2	false
!=	是否不等於	5 != 2	true

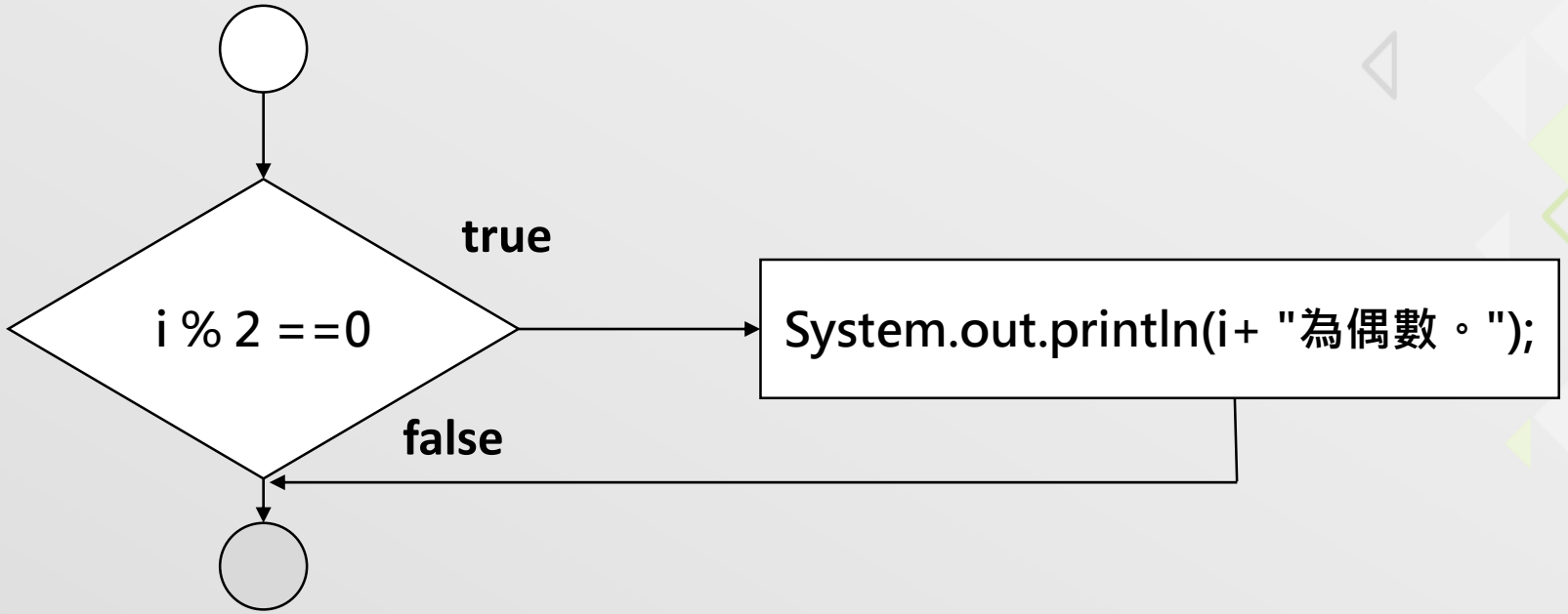
- 布林值之間的關係運算只能使用 == 和 != 兩個運算子，其餘的關係運算子都不能使用

if 選擇敘述

```
if( x > 3 )
        y = 2;
        z += 8;
        a= y + x;
```

- 如果敘述區塊中只有一個陳述要被執行則可省略{ }
- 左例中不管 x 的結果為何末兩行都會執行

- 例：I 若為偶數則輸出訊息



程式範例：if 選擇敘述

- 程式輸入：讓使用者輸入學生的姓名、期中成績、平時成績、期末成績。
- 資料處理:計算學期成績：
  - ✓ 期中*30%+平時*30%+期末成績*40%
- 程式輸出：顯示使用者的學期成績以及判斷是否及格?

程式實作：if 選擇敘述

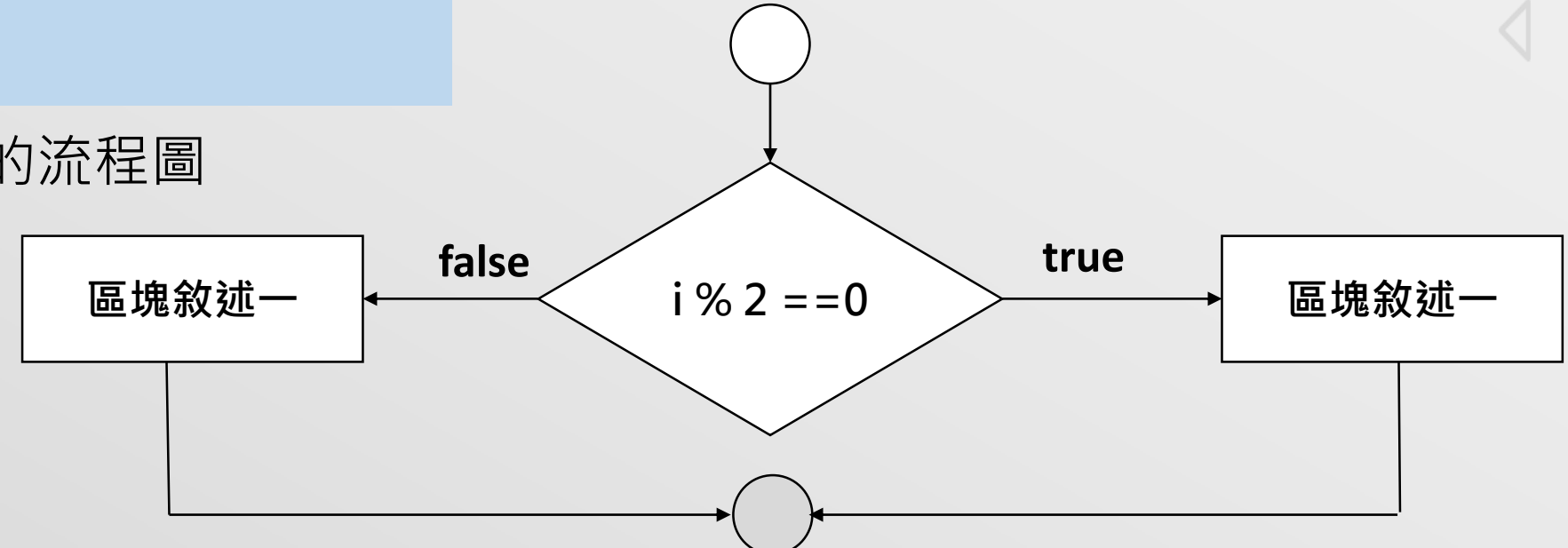
```
1  import java.io.*;
2  public class app4_3 {
3      public static void main (String[] args) throws IOException
4      {
5          BufferedReader buf;
6          String name;
7          int ord,mid,fin;
8          float score;
9          buf=new BufferedReader (new InputStreamReader (System.in));
10         System.out.print("請輸學生姓名:" );
11         name=buf.readLine();
12         System.out.print("請輸期中成績:" );
13         mid=Integer.parseInt(buf.readLine());
14         System.out.print("請輸平時成績:");
15         ord=Integer.parseInt(buf.readLine());
16         System.out.print("請輸末成績:");
17         fin=Integer.parseInt(buf.readLine());
18         score=(float)(mid*0.3+ord*0.3+fin*0.4);
19         System.out.println(name +"同學的學期成績為:"+score);
20         if(score<60)
21             System.out.println(name +"同學很抱歉，你被當了！");
22     }
23 }
```

if/else 敘述基本用法

```
if(條件式){
        //區塊敘述一
}
else{
        //區塊敘述二
}
```

- 若**條件式成立**時，執行區塊敘述一。
- 若**條件式不成立**時，執行區塊敘述二。

- if/else 的流程圖



程式範例：if/else 敘述基本用法

- 下面是if-else的範例：

```
1  public class app4_4 {
2      public static void main (String args[])
3      {
4          int a = 15;
5          if( a % 2 == 0 ) //如果可被 2 整除
6              System.out.println(a + " is an even number "); //印出 a為偶數
7          else
8              System.out.println(a + " is an odd number ");  //印出 a為奇數
9      }
10 }
```

app4_4 OUTPUT

15 is an odd number

如果有兩個條件以上怎麼辦？

- 邏輯運算子語法

運算子	意義	語法
&（ &&）	且，and	a & b
|（ ||）	或，or	a | b
!	非，not	! a

- & 的例子：如果學期成績低於60分而且缺課次數大於3次即為不及格：

```
if (Score < 60 & absent > 3)
        System.out.println("同學，很抱歉，你被當了！");
```

如果有兩個條件以上怎麼辦？

- | 的例子：如果學期成績低於60分或缺課次數大於3次即為不及格：

```
if (Score < 60 | absent > 3)
        System.out.println("同學，很抱歉，你被當了！");
```

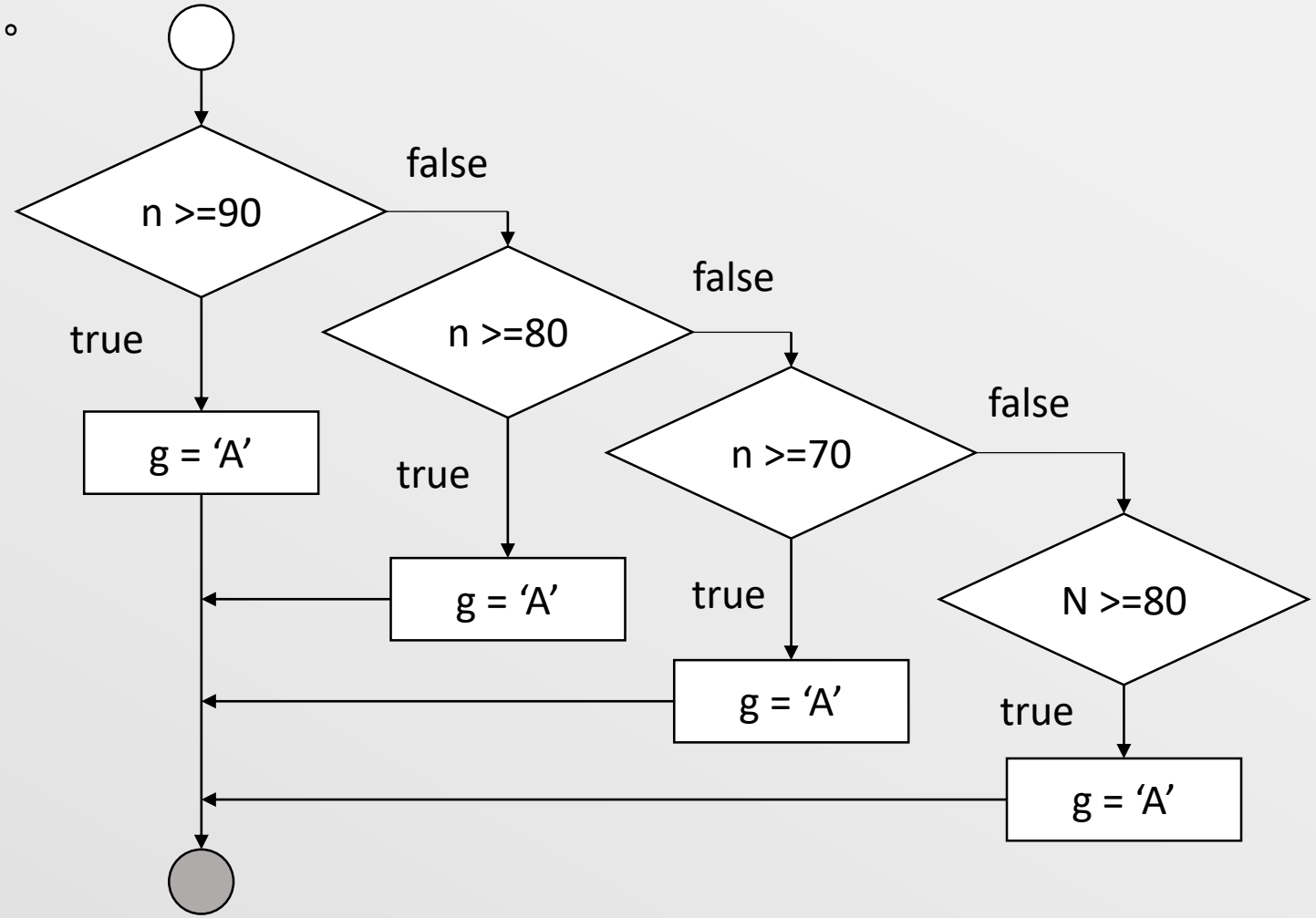
- 邏輯運算子的運算元與結果都是boolean值

變數A	變數B	A & B	A | B	! A
True	True	True	True	False
True	False	False	True	False
False	True	False	True	True
False	False	False	False	True

if/else 的巢狀結構

- 巢狀if-else敘述指的是if-else敘述區塊中還包含有if-else 敘述
- 例：判斷學生分數等級。

```
if(n >= 90) {
        g = ‘A‘ ;
}
else if(n >= 80) {
        g = ‘B’ ;
}
else if(n >= 70) {
        g = ‘C’ ;
}
else if (n >= 60) {
         g = ‘D’ ;
}
else
        g = ‘F’ ;
```



(補充教材)?：與if-else敘述有類似功能

- ?：是唯一的三元運算子，其語法如下

```
判斷值 ? 真時選擇值 : 假時選擇值
```

- Example：

```
成績 >= 60 ? "及格" : "不及格"
```

- **真時選擇值**和**假時選擇值**的型別應該相同，或是兩者都可以自動型別轉換指定給變數。

```
ifPass = score >= 60 ? 'Y' : 'N';
```

範例程式：?：三元運算

- 試著練習用條件運算子撰寫一程式：

```
1 public class app4_5 {
2     public static void main (String args[])
3     {
4         int a=8, b=3, max;
5         max = (a > b) ? a : b ; //a>b時,max=a,否則max=b
6         System.out.println("a= " + a + ", b= " + b);
7         System.out.println(max + "是較大的數");
8     }
9 }
```

app4_5 OUTPUT

a= 8, b= 3
8是較大的數

後記

- 運算式是由**運算元(Operand)**和**運算子 (Operator)**所組成。

- 運算式的求值規則：
  - ✓ 由左而右；
  - ✓ 小括號最優先運算；
  - ✓ 依照運算子的優先順序來計算。

- 程式中的運算式不可以使用中括號和大括號，只能使用小括號。