

Java 程式設計入門

資料型態與變數

鄭安翔

ansel_cheng@hotmail.com

課程大綱

- 1) 變數宣告及使用
- 2) Java 基本資料型別
- 3) 命名規則及慣例
- 4) 資料型別的晉升與轉型

Variable Declaration 變數宣告

<type> <identifier> [=initial_values];

→ Primitive or reference type

→ Name of the variable

→ Initialize literal values

- 資料型態
 - 適當記憶體儲存空間
- 變數名稱
 - 以該名稱來取得儲存值
- = 指定運算子
 - 右邊的值存到左邊的記憶體



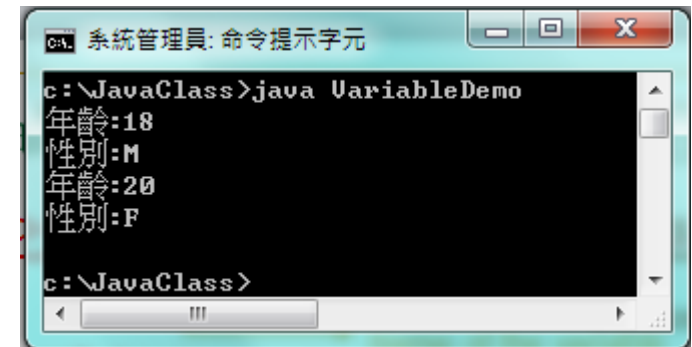
Variable Declaration 變數宣告

```
01 public class VariableDemo {  
02     public static void main(String[] args) {  
03         int age = 18;  
04         char gender = 'M';  
05         System.out.print("年齡:");  
06         System.out.println(age);  
07         System.out.print("性別:");  
08         System.out.println(gender);  
09  
10         age = 20;  
11         gender = 'F';  
12         System.out.print("年齡:");  
13         System.out.println(age);  
14         System.out.print("性別:");  
15         System.out.println(gender);  
16     }  
17 }
```

宣告變數並賦值

變更變數值

取得變數值



```
c:\JavaClass>java VariableDemo  
年齡:18  
性別:M  
年齡:20  
性別:F  
c:\JavaClass>
```

課程大綱

- 1) 變數宣告及使用
 - 2) **Java 基本資料型別**
 - 邏輯型態
 - 整數型態
 - 浮點數型態
 - 字元型態
 - 3) 命名規則及慣例
 - 4) 資料型別的晉升與轉型
-

Java 基本資料型別

■ 基本資料型別之資料型態及範圍：

資料類別	資料型態	位元數(bits)	資料範圍 (Range)	範例	初始值
邏輯布林值	boolean	1	只能有 true 或 false	true, false	false
整數	byte	8	-128 ~ 127	2, -114	0
	short	16	-32,768 ~ 32,767	2, -32699	0
	int	32	$-2^{31} \sim 2^{31}-1$	2, -147344778	0
	long	64	$-2^{63} \sim 2^{63}-1$	2, -2036854708L	0L
浮點數	float	32	$-3.4\text{E}+38 \sim 3.4\text{E}+38$	99F, -32699.01F	0.0F
	double	64	$-1.7\text{E}+308 \sim 1.7\text{E}+308$	-111, 21E12	0.0D
字元	char (Unicode)	16	'\u0000' ~ '\uFFFF' 0 ~ 65535	'A', 'x', '3', '中', '\n', '\u0063'	'\u0000'

布林值 boolean

■ 布林值 boolean

- 只有 true 和 false 兩種字面值(Literal Value)

- 直接在程式裡寫下一個數值,稱為字面值

- 不能與數字轉換

```
boolean b1 = true;
```

```
boolean b2 = false;
```

整數 Integer

■ 整數

- 在電腦中以二進位儲存
- 三種字面值表示法:

int i = 45; //十進位的變數值設定

int j = **0b**101101; //二進位的變數值設定

int k = **0**55; //八進位的變數值設定

int l = **0x**2D; //十六進位的變數值設定


- 兩種字面值型態: int (預設) 與 long (字尾加上L)
- 可使用_增加數值可讀性

int m = **0b**0010_1101;

十進位與二進位轉換

■ 十進位與二進位轉換

45 \longrightarrow 101101

$$\begin{array}{r} 2 \overline{) 45} \\ 2 \overline{) 22} \dots 1 \\ 2 \overline{) 11} \dots 0 \\ 2 \overline{) 5} \dots 1 \\ 2 \overline{) 2} \dots 1 \\ 1 \dots 0 \end{array}$$


101101 \longrightarrow 45

$$\begin{array}{r} 1 \ 0 \ 1 \ 1 \ 0 \ 1 \\ * \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0 \\ \hline 32 + 0 + 8 + 4 + 0 + 1 = 45 \end{array}$$

八進位與十六進位轉換

八進位

1 0 1 1 0 1 → 0 5 5

0 5 5 → 45

$$\begin{array}{r} 0 \ 5 \ 5 \\ * \quad 8^1 \ 8^0 \\ \hline 40 + 5 = 45 \end{array}$$

000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

十六進位

0 0 1 0 1 1 0 1 → 0x 2 D

0x 2 D → 45

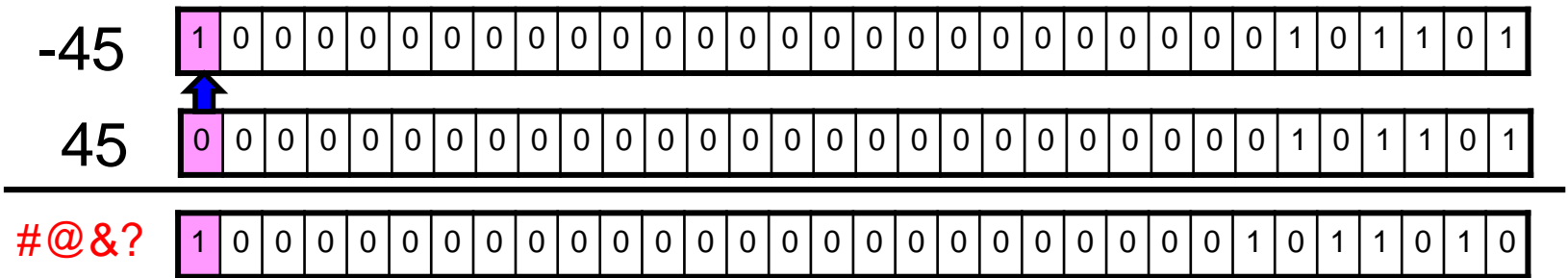
$$\begin{array}{r} 0x \ 2 \ D \\ * \quad 16^1 \ 16^0 \\ \hline 32 + 13 = 45 \end{array}$$

0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

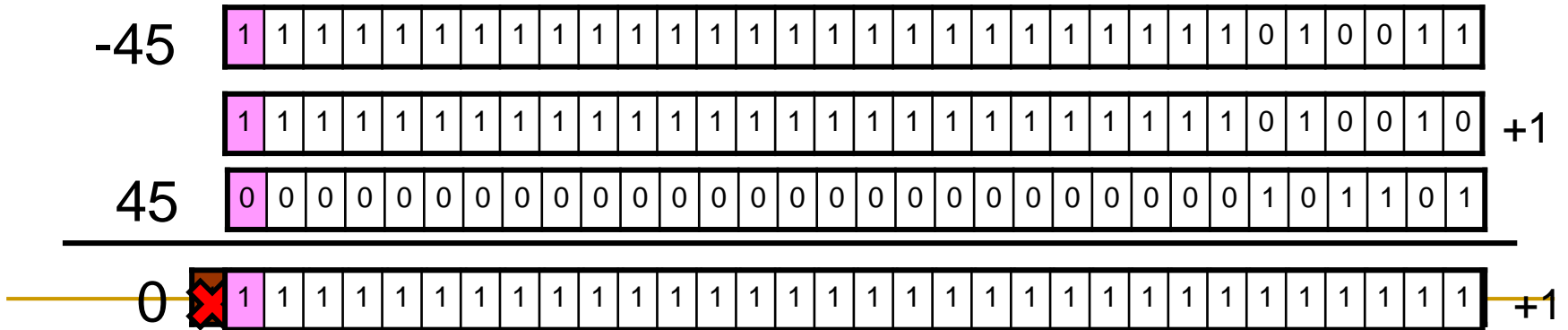
負數二進位表達 : 2's complement

■ 負數二進位表達

- 最高位元(Most Significant Bit, MSB)用來表示正負號, 0為正, 1為負



- ## 二的補數



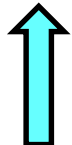
負數二進位表達：2's complement

- 找到2的補數
 1. 將所有1→0, 0→1
 2. 將Step1結果+1

十進位負數之二進位表示

$$\begin{array}{r} 5 \quad 00000101 \\ \quad 11111010 \\ + \quad \quad \quad 1 \\ \hline -5 \quad 11111011 \end{array}$$

負數二進位求十進位值

$$\begin{array}{r} 11101101 = -19 \\ 00010010 \\ + \quad \quad \quad 1 \\ \hline 00010011 = 19 \end{array}$$


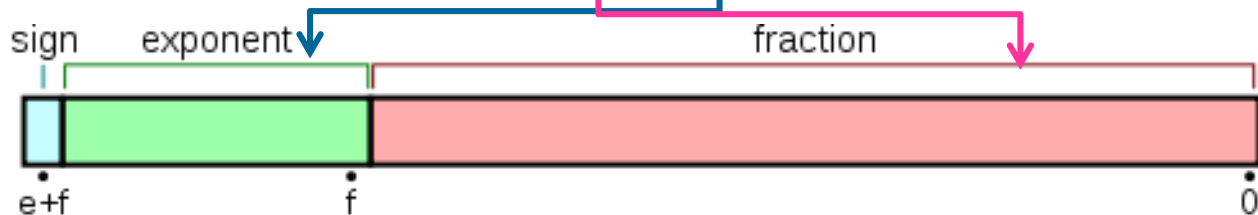
浮點數

■ 浮點數

- 包含有小數點的數字及科學記數符號表示的數字
 $125.625 = 1.25625 * 10^2 = 1.25625E2$

- IEEE 754二進位浮點數標準

$$125.625 = 1111101.101 = 1.111101101 * 2^{00000110}$$



- 兩種型別: float (字尾加上F)與double (預設, 字尾加上D)



字元

- 字元: 使用單引號 (') 括住
 - 16 bits Uni-Code: 'X', '3', '中'
 - Uni-Code編碼: '\u0041'
 - 跳脫字元: '\\'

```
char gender = 'M';
```

- 字串: 使用雙引號 (")括住
 - 參考資料型別,不是基本資料型別
 - 可以用 **literal value**字面值方式賦值

```
String name = "Sean Cheng";
```

Char 跳脫字元

- 某字元直接撰寫時,被Java當作特定意義的語法,使用跳脫字元來表示

跳脫字元	意義
\b	退一格（backspace）。
\f	跳頁（form feed，使用於印表機）。
\n	換行（newline）。
\r	返回（carriage return），游標移至行首。
\t	水平跳格，相當於按鍵盤的Tab鍵。
\\	表示 \ 字元。
\'	表示 ' 字元。
\"	表示 " 字元。
\uXXXX	以十六進位指定Unicode字元輸出
\XXX	以八進位指定Unicode字元輸出

課程大綱

- 1) 變數宣告及使用
- 2) **Java** 基本資料型別
- 3) **命名規則及慣例**
- 4) 資料型別的晉升與轉型

Identifier 識別字命名規則

- 下列四種字元組成
 - Unicode letters A-Z, a-z, 中文, ...
 - Underscore _
 - Dollar sign \$, ¥, £...
 - Digits 0,1,...9 (不能出現在起始字元)
- 不可為 **keywords / Reserved words**
- 大小寫有別 **Case sensitive**

Java 關鍵字及保留字

Java的關鍵字

abstract	assert	boolean	break	byte
case	catch	char	class	continue
default	do	double	else	enum
extends	final	finally	float	for
if	implements	import	instanceof	int
interface	long	native	new	package
private	protected	public	return	short
static	strictfp	super	switch	synchronized
this	throw	throws	transient	try
void	volatile	while		

Java的保留字

const	goto	true	false	null
-------	------	------	-------	------

Identifier 識別字命名慣例

■ 類別(class)名稱

- 名詞
- 第一個英文字母大寫, 複合字用大寫區隔
- Ex: SavingsAccount

■ 屬性(attribute)名稱

- 名詞
- 第一個英文字母小寫, 複合字用大寫區隔
- Ex: userName

Identifier 識別字命名慣例

- 方法(method)名稱

- 動詞
- 第一個英文字母小寫, 複合字用大寫區隔
- Ex: calculateInterest()

- 常數名稱

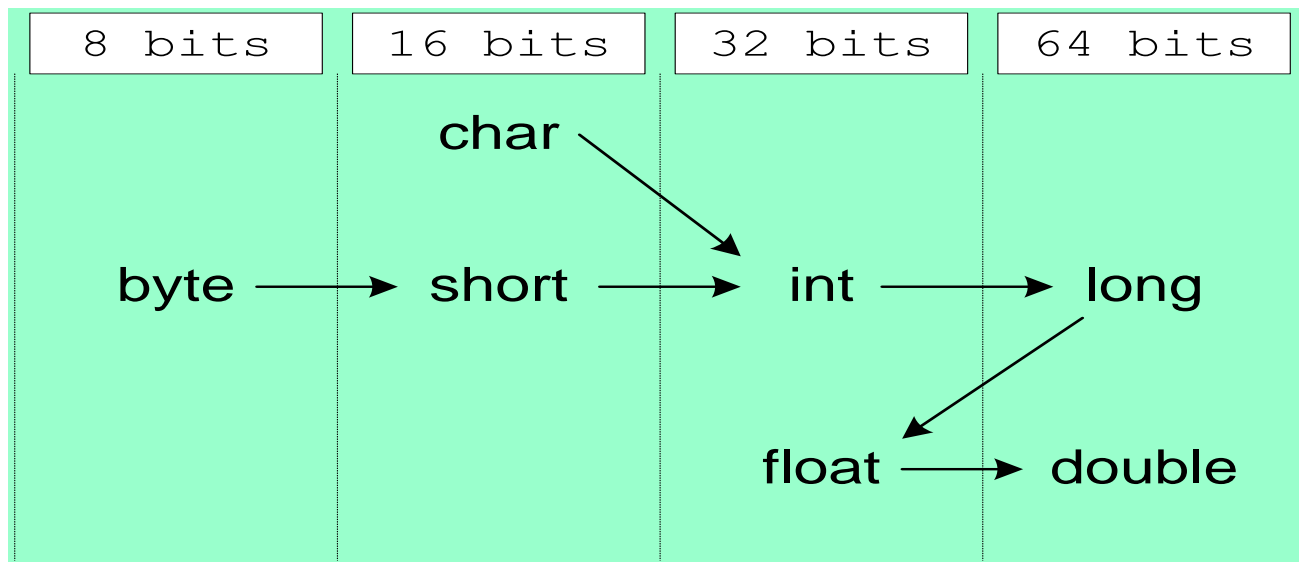
- 利用 **final** 來修飾屬性或有常數
- 全大寫
- Ex: final double PI = 3.1415926;

課程大綱

- 1) 變數宣告及使用
- 2) **Java** 基本資料型別
- 3) 命名規則及慣例
- 4) **資料型別的晉升與轉型**

Java資料型別的晉升及轉型

- Java 語言中針對基本與參考資料型別皆提供了 2 種資料型別轉換：
 - 隱含式的轉換(Implicit Casting), 晉升(Promotion)
 - 強制式的轉換(Explicit Casting), 強制轉型(Casting)



基本資料型別的晉升

- 隱含式的轉換(Implicit Casting)：
 - 以較小的資料型別轉成較大的資料型別。
 - 系統會根據程式的需要自動且適時地做轉型。

```
01 class TestPromotion {  
02     public static void main(String[] args) {  
03         int i = 1; double d; float f;  
04  
05         d = i;  
06  
07         i = d;  
08  
09         f = 2.5;  
10     }  
11 }
```

將 i 變數的內容值指派給 d。合法，implicit Casting。

將 d 變數的內容值指派給 i。不合法，程式需要作 Explicit Casting。如：i = (int)d;

將 2.5 指派給 f。不合法，
在 Java 語言中 literal 浮點數預設的資料型別是 double
所以必須改寫成
f = (float)2.5; 或
f = 2.5f;

基本資料型別的強制轉型

- 強制式的轉換(Explicit Casting)：
 - 以較大的資料型別切割成較小的資料型別，例如：將 32 bit 的資料放到 16 bit 的資料中。
 - 必須在程式中給予明確的指令，系統並不自動轉型。
 - 基本資料型別中的強制轉換機制，原則上就是切割資料。

Ex: byte b = **(byte)**257;

