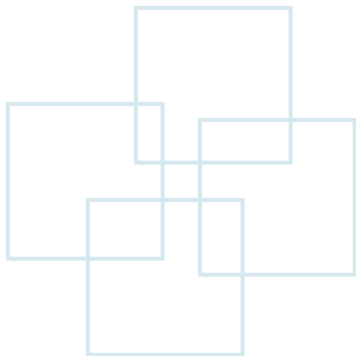


National University of Kaohsiung

Java Programming Variables

*Based on slides from the authors
Revised by Ya-Ju Yu 2017*

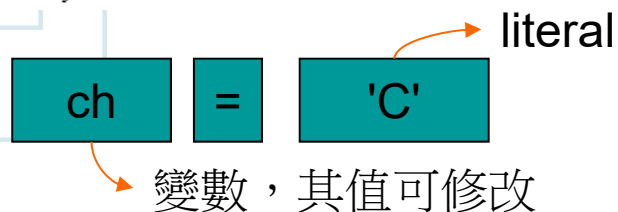
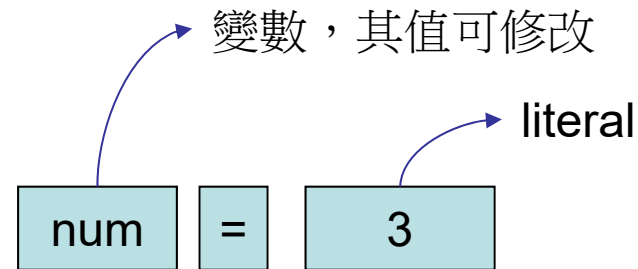




變數的使用

□ 下面是變數使用的範例：

```
01 // app3_1, 簡單的實例
02 public class app3_1
03 {
04     public static void main(String args[])
05     {
06         int num=3;        // 宣告 num 為整數，並設值為 3
07         char ch='C';      // 宣告 ch 為字元變數，並設值為 'C'
08         System.out.println(num+" is an integer");
09         System.out.println(ch+" is a character");
10     }
11 }
```



```
/* app3_1 OUTPUT----
3 is an integer
C is a character
-----*/
```



National University of Kaohsiung

變數宣告成final的格式

□ 若是變數值不會變動，則可將該變數宣告成final：

final 資料型態 變數名稱 = 字面值;

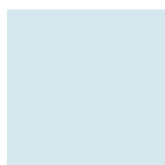
格式 3.1.1

變數宣告成 final
的格式

利用final宣告的變數，其值不能再被更改，如
final double PI=3.1415926;

□ In C

Static 資料型態 變數名稱

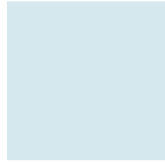


National University of Kaohsiung

基本資料型態

▣ 各種基本資料型態所佔的記憶體空間及範圍：

資料型態	位元組	表示範圍	初始值
long (長整數)	8	-9223372036854775808~9223372036854775807	0L
int (整數)	4	-2147483648~2147483647	0
short (短整數)	2	-32768~32767	0
byte (位元)	1	-128~127	0
char (字元)	2	0~65535 ('\u0000'~'\uFFFF')	'\u0000'
boolean (布林)	1	布林值只能使用 true 或 false	false
float (浮點數)	4	$-3.4\text{E}38 (-3.4 \times 10^{38}) \sim 3.4\text{E}38 (3.4 \times 10^{38})$	0.0F
double (倍精度)	8	$-1.7\text{E}308 (-1.7 \times 10^{308}) \sim 1.7\text{E}308 (1.7 \times 10^{308})$	0.0D



National University of Kaohsiung

整數型態 int

□ 整數型態可分為

- 長整數 (long int)
- 整數 (int)
- 短整數 (short int)
- 位元 (byte)

□ 下面為短整數型態宣告的範例：



```
short sum;
```

```
// 宣告sum為短整數
```



常數的資料型態

▣ **Java**把整數數值的型態視為**int**，超過範圍時會發生錯誤

```
01 // app3_2, 整數常數的使用--錯誤的範例
02 public class app3_2
03 {
04     public static void main(String args[])
05     {
06         long num=32967359818;
07         System.out.println("num= "+num);
08     }
09 }
```

敘述改成 `long num=32967359818L;`

編譯上面的程式碼，將會得到下列的錯誤訊息：

```
integer number too large: 32967359818
long num=32967359818;
```



簡單易記的代碼 (1/2)

□ 整數之最大值與最小值的識別字及代碼：

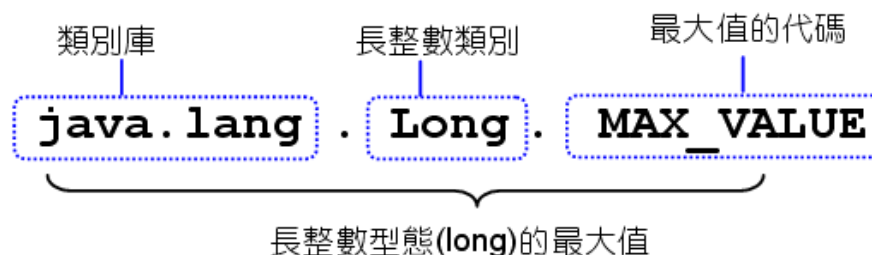


表 3.2.2 整數常數的特殊值代碼

	long	int
所屬類別	java.lang.Long	java.lang.Integer
最大值代碼	MAX_VALUE	MAX_VALUE
最大值常數	9223372036854775807	2147483647
最小值代碼	MIN_VALUE	MIN_VALUE
最小值常數	-9223372036854775808	-2147483648

	short	byte
所屬類別	java.lang.Short	java.lang.Byte
最大值代碼	MAX_VALUE	MAX_VALUE
最大值常數	32767	127
最小值代碼	MIN_VALUE	MIN_VALUE
最小值常數	-32768	-128



National University of Kaohsiung

簡單易記的代碼 (2/2)

□ 利用整數常數的代碼列印資料

```
01 // app3_3, 印出 Java 定義的整數常數之最大值
02 public class app3_3
03 {
04     public static void main(String args[])
05     {
06         long lmax=java.lang.Long.MAX_VALUE;
07         int imax=java.lang.Integer.MAX_VALUE;
08         short smax=Short.MAX_VALUE; // 省略類別庫 java.lang
09         byte bmax=Byte.MAX_VALUE;   // 省略類別庫 java.lang
10
11         System.out.println("Max value of long : "+lmax);
12         System.out.println("Max value of int : "+imax);
13         System.out.println("Max value of short : "+smax);
14         System.out.println("Max value of byte : "+bmax);
15     }
16 }
```

```
/* app3_3 OUTPUT-----
Max value of long : 9223372036854775807
Max value of int : 2147483647
Max value of short : 32767
Max value of byte : 127
-----
```




National University of Kaohsiung

溢位 (overflow) 的發生 (1/3)

□ 溢位：當儲存的數值超出容許範圍時

/ app3_4 OUTPUT---*

01 // app3_4, 整數資料型態的溢位

02 public class app3_4

03 {

04 public static void main(String args[])

05 {

06 int i=java.lang.Integer.MAX_VALUE; // 將 i 設為整數的最大值

07 int sum;

08

09 System.out.println("i="+i); // 印出 i 的值

10

11 sum=i+1;

12 System.out.println("i+1="+sum); // 印出 i+1 的值

13

14 sum=i+2;

15 System.out.println("i+2="+sum); // 印出 i+2 的值

16 }

17 }

i=2147483647

i+1=-2147483648

i+2=-2147483647

-----/*



National University of Kaohsiung

溢位 (overflow) 的發生 (2/3)

□ 下圖說明溢位的發生：

```
i=2147483647;  
sum=i+1;
```

計算 $\text{sum}=\text{i}+2$ 之後， sum 的值

計算 $\text{sum}=\text{i}+1$ 之後， sum 的值

i 的值

-2147483646

-2147483647

-2147483648

2147483647

2147483646

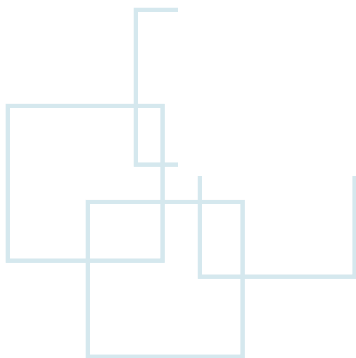
-2

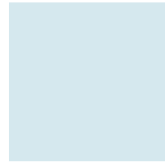
-1

0

1

2





National University of Kaohsiung

溢位 (overflow) 的發生 (3/3)

□int型態的溢位處理範例

```
01 // app3_5, int 型態的溢位處理
02 public class app3_5
03 {
04     public static void main(String args[])
05     {
06         int i=java.lang.Integer.MAX_VALUE; // 將 i 設為整數的最大值
07
08         System.out.println("i="+i);
09         System.out.println("i+1="+ (i+1)); // 會發生溢位
10         System.out.println("i+2="+ (i+2L)); // 自動型態轉換
11         System.out.println("i+3="+ ((long)i+3)); // 強制型態轉換
12     }
13 }
```

/ app3_5 OUTPUT---*

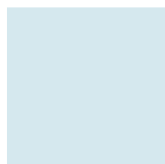
i=2147483647

i+1=-2147483648

i+2=2147483649

i+3=2147483650

-----/*



字元型態 (1/2)

- 字元型態佔 2 個位元組，用來儲存字元
- Java使用的編碼系統為Unicode(標準萬國碼)
- 宣告字元變數，並設值給它：

```
char ch;           // 宣告字元變數ch  
ch= 'A' ;          // 將字元literal 'A' 設值給字元變數ch
```

在宣告的同時便設定初值

```
char ch1='A' ;      // 宣告字元變數ch1，並將字元literal 'A' 設值給它  
char ch2=97 ;       // 將ch2設值為ASCII碼為97的字元  
char ch3='7' ;      // 將ch3設值為字元literal '7'
```



字元型態 (1/2)

▣ 下面的程式以不同的格式列印字元變數：

```
01 // app3_6, 字元型態的列印
02 public class app3_6
03 {
04     public static void main(String args[])
05     {
06         char ch1=71;           // 設定字元變數 ch1 等於編碼為 71 的字元
07         char ch2='G';          // 設定字元變數 ch2 等於 'G'
08         char ch3='\u0047';     // 以 16 進位值設定字元變數 ch3
09
10         System.out.println("ch1="+ch1);
11         System.out.println("ch2="+ch2);
12         System.out.println("ch3="+ch3);
13     }
14 }
```

/* app3_6 OUTPUT---

ch1=G
ch2=G
ch3=G

-----*/



跳脫字元 (1/2)

- 反斜線「\」稱為跳脫字元
- 反斜線「\」加上控制碼，稱為跳脫序列

表 3.2.3 常用的跳脫序列

跳脫序列	所代表的意義	跳脫序列	所代表的意義
\f	換頁 (Form feed)	\\	反斜線 (Backslash)
\b	倒退一格 (Backspace)	\'	單引號 (Single quote)
\n	換行 (New line)	\"	雙引號 (Double quote)
\r	歸位 (Carriage return)	\uxxxx	十六進位的 unicode 字元
\t	跳格 (Tab)	\ddd	八進位 Unicode 字元，範圍在八進位的 000~377 之間



跳脫字元 (2/2)

□ 利用跳脫序列列印字串：

```
01 // app3_7, 列印跳脫序列
```

```
02 public class app3_7
```

```
03 {
```

```
04     public static void main(String args[])
```

```
05     {
```

```
06         char ch1='\"';
```

```
07         char ch2='\"74';
```

```
08         char ch3='\"u003e';
```

```
09
```

```
10         System.out.println(ch1+"Time flies."+ch1);
```

```
11         System.out.println("\"Time is money!\");
```

```
12         System.out.println(ch2+"Tomorrow never comes"+ch3);
```

```
13     }
```

```
14 }
```

```
/* app3_7 OUTPUT-----
```

```
"Time flies."
```

```
"Time is money!"
```

```
<Tomorrow never comes>
```

```
-----*/
```

可改成char ch2=074;

// 將 ch1 設值為\"

// 以八進位值設定字元變數 ch2

// 以 16 進位值設定字元變數 ch3

可改成char ch3=0x3e;



National University of Kaohsiung

浮點數與倍精度浮點數型態 (1/4)

□ 浮點數 (float) 長度為4個位元組

有效範圍為 $-3.4 \times 10^{38} \sim 3.4 \times 10^{38}$

□ 倍精度 (double) 浮點數的長度為8個位元組

有效範圍為 $-1.7 \times 10^{308} \sim 1.7 \times 10^{308}$

□ float與double型態的變數宣告範例如下：

```
double num;           // 宣告num為倍精度浮點數變數  
float sum=2.0f;        // 宣告sum為浮點數變數，並設其初值為2.0
```

一定要加上f才是float，要不然會以為是double



浮點數與倍精度浮點數型態 (2/4)

float與double宣告與設值時注意事項

- `double num1=-5.6e64;` // 宣告num1為double，其值為 -5.6×10^{64}
- `double num2=-6.32E16;` // e 也可以用大寫的 E 來取代
- `float num3=2.478f;` // 宣告 num3 為 float，並設初值為 2.478
- `float num4=2.63e64f;` // 錯誤， 2.63×10^{64} 超過float可表示的範圍
- `float num4=2.63e64;` **//cannot convert from double to float**

01 // app3_8, 浮點數的使用

02 public class app3_8

03 {

04 public static void main(String args[])

05 {

06 float num=5.0f;

07 System.out.println(num+"*"+num+"="+ (num*num)); // 印出num*num的結果

08 }

09 }

/* app3_8 OUTPUT---

5.0*5.0=25.0

-----*/



National University of Kaohsiung

浮點數與倍精度浮點數型態 (3/4)

□ 浮點數型態的最大值與最小值的代碼

表 3.2.4 浮點數常數的特殊值代碼

	float	double
所屬類別	java.lang.Float	java.lang.Double
最大值	MAX_VALUE	MAX_VALUE
最大值常數	3.4028235E38	1.7976931348623157E308
最小值	MIN_VALUE	MIN_VALUE
最小值常數	1.4E-45	4.9E-324



National University of Kaohsiung

浮點數與倍精度浮點數型態 (4/4)

▣ 下面的範例是印出 float 與 double 的最大與最小值：

```
01 // app3_9, 印出 Java 定義的浮點數常數值
02 public class app3_9
03 {
04     public static void main(String args[])
05     {
06         System.out.println("f_max="+Float.MAX_VALUE);
07         System.out.println("f_min="+Float.MIN_VALUE);
08         System.out.println("d_max="+Double.MAX_VALUE);
09         System.out.println("d_min="+Double.MIN_VALUE);
10     }
11 }
```

/* app3_9 OUTPUT-----

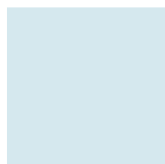
f max=3.4028235E38

f min=1.4E-45

d max=1.7976931348623157E308

d min=4.9E-324

-----*/



布林型態

□ 宣告布林變數的範例：

```
boolean status=true;           // 宣告布林變數status，並設值為true
```

在程式中印出布林值：

```
01 // app3 10, 印出布林值
02 public class app3 10
03 {
04     public static void main(String args[])
05     {
06         boolean status=false;    // 設定 status 布林變數的值為 false
07         System.out.println("status="+status);
08     }
09 }
```

```
/* app3_10 OUTPUT----
status=false
-----*/
```



自動型態的轉換 (1/2)

▣ Java會在下列條件皆成立時，自動做資料型態的轉換：

- (1) 轉換前的資料型態與轉換後的型態相容
- (2) 轉換後的資料型態之表示範圍比轉換前的型態大
 - ✓ 例如：int 和 float 相加，int 會被轉成 float
 - char 和 int 相加，char 會被轉成 int

型態轉換

發生在運算子左右兩邊的運算元型態不同時

自動資料型態的轉換只限該行敘述

透過自動型態的轉換，可以保證資料的精確度

這種轉換也稱為擴大轉換（augmented conversion）



National University of Kaohsiung

自動型態的轉換 (2/2)

□ 浮點數與整數作運算的結果：

```
01 // app3_11, 型態自動轉換
02 public class app3_11
03 {
04     public static void main(String args[])
05     {
06         int a=45;           // 宣告 a 為整數
07         float b=2.3f;       // 宣告 b 為浮點數
08
09         System.out.println("a="+a+", b="+b); // 印出 a、b 的值
10         System.out.println("a/b="+ (a/b));  // 印出 a/b 的值
11     }
12 }
```

/* app3_11 OUTPUT---

a=45,b=2.3

a/b=19.565218

-----*/

先將a轉換成float，再除以b



National University of Kaohsiung

強制型態轉換 (1/2)

□ 將資料型態強制轉換成另一種型態的語法：

資料型態的強制性轉換

(欲轉換的資料型態) 變數名稱;

強制型態轉換也稱為顯性轉換 (explicit cast)

強制型態轉換的注意事項

變數強制轉換成另一種型態，原先的型態不會被改變

縮小轉換 (narrowing conversion) 可能會漏失資料的精確度

Java不會主動做縮小轉換



強制型態轉換 (2/2)

□ 整數與浮點數進行強制轉換的範例：

```
01 // app3_12, 強制轉換
02 public class app3_12
03 {
04     public static void main(String args[])
05     {
06         int a=25;
07         int b=9;
08
09         System.out.println("a="+a+",b="+b); // 印出 a、b 的值
10         System.out.println("a/b="+a/b);      // 印出 a/b 的值
11         System.out.println("(float)a/b="+((float)a/b);
12     }
13 }
```

```
/* app3_12 OUTPUT-----
a=25,b=9
a/b=2
(float)a/b=2.777777
-----*/
```

a/(float)b

也可寫成

(float)a/(float)b

將a轉換成浮點數之後，再除以b



Screen Out

- Different approaches to formatting output

```
double price = 19.8;  
System.out.print("$");  
System.out.printf("%6.2f\n", price);  
System.out.println(price);
```

指換行

- The code outputs the following line:

\$ 19.80

19.8

- Try %5.1f, %5.2f, %7.2f



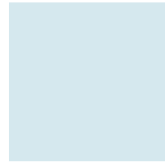
National University of Kaohsiung

Format Specifiers

□ For System.out.printf

Conversion Character	Type of output	Examples
d	Decimal integer	%5d
f	Floating point	%6.2f
e	E-notation floating point	%8.3e
g	General floating point (Java decides whether to use E-notation or not)	%8.3g
s	String	%12s
c	Character	%2c





Money Formats

□ Using the class NumberFormat

```
1 import java.text.NumberFormat;
2 import java.util.Locale;
3 import java.util.Scanner;
4 public class Hey
5 {
6     public static void main(String[] argv)
7     {
8
9         System.out.println("Default location:");
10        NumberFormat moneyFormatter = NumberFormat.getCurrencyInstance();
11        System.out.println(moneyFormatter.format(19.81111)); // output: NT$19.81
12        System.out.println(moneyFormatter.format(19.89999)); // output: NT$19.90
13
14        System.out.println("US as location:");
15        NumberFormat moneyFormatterUS = NumberFormat.getCurrencyInstance(Locale.US);
16        System.out.println(moneyFormatterUS.format(19.81111)); // output: $19.81
17        System.out.println(moneyFormatterUS.format(19.89999)); // output: $19.90
```



Decimal Format

```
1 import java.text.DecimalFormat;
2 import java.util.Scanner;
3 public class Hey
4 {
5     public static void main(String[] argv)
6     {
7         DecimalFormat pattern00dot000 = new DecimalFormat("00.000");
8         DecimalFormat pattern0dot00 = new DecimalFormat("0.00");
9
10        double d = 12.3456789;
11        System.out.println(pattern00dot000.format(d)); // output: 12.346
12        System.out.println(pattern0dot00.format(d)); // output: 12.35
13
14        DecimalFormat percent = new DecimalFormat("0.00%");
15        System.out.println(percent.format(0.308)); // output: 30.80%
```



Decimal Format (con't)

- ❑ '#' is used to indicate an optional digit.
- ❑ 使用E-notation表示時，會盡量按照格式和要顯示的數字個數，且會讓較“重要”的位數顯示較多（例如#號較多的那邊，且），但通常無法精準的知道重要位數的個數。

```
DecimalFormat formattingObject = new DecimalFormat("#0.0##");  
System.out.println(formattingObject.format(12.3456789)); // output: 12.346  
System.out.println(formattingObject.format(1.23456789)); // output: 1.2346
```

```
DecimalFormat eNotation1 = new DecimalFormat("#0.###E0");  
DecimalFormat eNotation2 = new DecimalFormat("00.###E0");  
System.out.println(eNotation1.format(123.987)); // output: 1.2399E2  
System.out.println(eNotation2.format(123.987)); // output: 12.399E2  
System.out.println(eNotation1.format(0.0000123456)); // output: 12.346E-6  
System.out.println(eNotation2.format(0.0000123456)); // output: 12.346E-6
```



National University of Kaohsiung

Decimal Format (con't)

□ E-notation

– More examples

1. 整數位個數小於小數位個數

```
DecimalFormat eNotation1 = new DecimalFormat("#0.####E0");  
System.out.println(eNotation1.format(0.0000123456)); // output: 1.23456E-6
```

2. 整數位和小數位個數差不多

```
DecimalFormat eNotation1 = new DecimalFormat("##0.####E0");  
DecimalFormat eNotation1 = new DecimalFormat("##0.##E0");  
System.out.println(eNotation1.format(0.0000123456)); // output: 12.3456E-6
```

3. 整數位個數大於小數位

```
DecimalFormat eNotation1 = new DecimalFormat("###0.####E0");  
System.out.println(eNotation1.format(0.0000123456)); // output: 1234.56E-8
```



National University of Kaohsiung

輸入資料的基本架構 (1/3)

□ 資料輸入的格式：

輸入資料的基本格式

```
import java.util.Scanner;
public class class name          // 類別名稱
{
    public static void main(String args[])
    {
        Scanner 物件名稱=new Scanner(System.in);    // 宣告 Scanner 類別的變數
        資料型態 變數名稱 a;                          // 宣告變數
        ... ..
        變數名稱=物件名稱.相對應型態的 Scanner 類別函數;    // 讀入資料至變數
        ... ..
    }
}
```



National University of Kaohsiung

輸入資料的基本架構 (2/3)

▣ 輸入資料時Scanner類別提供的函數：

<i>Method</i>	<i>Description</i>
<code>nextByte()</code>	reads an integer of the byte type.
<code>nextShort()</code>	reads an integer of the short type.
<code>nextInt()</code>	reads an integer of the int type.
<code>nextLong()</code>	reads an integer of the long type.
<code>nextFloat()</code>	reads a number of the float type.
<code>nextDouble()</code>	reads a number of the double type.
<code>next()</code>	reads a string that ends before a whitespace character.
<code>nextLine()</code>	reads a line of text (i.e., a string ending with the <i>Enter</i> key pressed).



National University of Kaohsiung

輸入資料的基本架構 (3/3)

□ 由鍵盤輸入字串的範例

```
01 // app3_13, 由鍵盤輸入資料
02 import java.util.Scanner;           // 載入 Scanner 類別
03 public class app3_13
04 {
05     public static void main(String args[])
06     {
07         Scanner scn=new Scanner(System.in); // 宣告 Scanner 類別的物件
08         String name;
09         int age;
10
11         System.out.print("What's your name? ");
12         name=scn.nextLine();           // 輸入字串
13         System.out.print("How old are you? ");
14         age=scn.nextInt();             // 輸入整數
15
16         System.out.print("Hi, "+name+", you're ");
17         System.out.println(age+" years old.");
18     }
19 }
```

```
/* app3_13 OUTPUT-----
What's your name? Junie Hong
How old are you? 6
Hi, Junie Hong, you're 6 years old.
-----*/
```



National University of Kaohsiung

輸入數值--不合型態的輸入

- 若是需要輸入數值，卻輸入字元 'k'，則會出現類似下列的錯誤訊息：

```
What's your name? Junie Hong
```

```
How old are you? k
```

```
Exception in thread "main" java.util.InputMismatchException
```

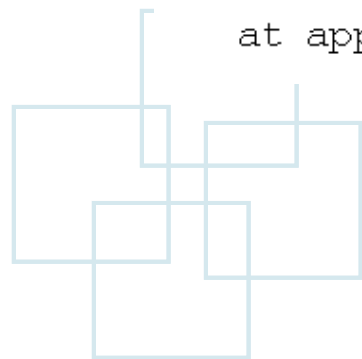
```
    at java.util.Scanner.throwFor(Scanner.java:909)
```

```
    at java.util.Scanner.next(Scanner.java:1530)
```

```
    at java.util.Scanner.nextInt(Scanner.java:2160)
```

```
    at java.util.Scanner.nextInt(Scanner.java:2119)
```

```
    at app3_13.main(app3_13.java:14)
```



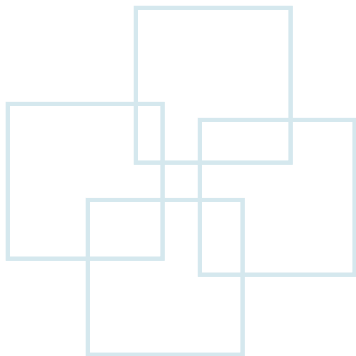


National University of Kaohsiung

要以Scanner類別輸入字元

▣ 用next() 取得字串後，再利用charAt(0) 函數取出字串中第0個字元即可：

```
01 Scanner scn=new Scanner(System.in);  
02 String str;  
03 char ch;  
04 str=scn.next();           // 輸入字串  
05 ch=str.charAt(0);        // 取出字串第 0 個字元
```





National University of Kaohsiung

Exercises

❑ Use two variables to compute the rectangle area and then show it on the screen

❑ Now, you can key in the two variables via keyboard.

❑ Please write a Java code to show the Unicode of char 'x' on the screen.

❑ Please convert the result of 352×32 to floating point

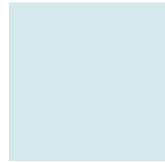


Exercises

- ❑ Body Mass Index (BMI) helps in specifying the weight category a person belongs to, depending on their body weight. BMI is estimated using the following formula:

$$\text{BMI} = \frac{\text{Weight in kilograms}}{(\text{Height in meters})^2}$$

- ❑ Write a program that calculates and outputs the BMI. Assume various input values wherever required.



Try It

```
1 import java.util.Scanner;
2 //Problems in using nextLine().
3 public class Hey
4 {
5     public static void main(String[] argv)
6     {
7         Scanner KB = new Scanner(System.in);
8         int age = 0;
9         String Address = "";
10
11         System.out.println("Enter your age");
12         age = KB.nextInt();
13         System.out.println("Enter you address");
14         Address = KB.nextLine();
15         System.out.println("I am "+age+" years old");
16         System.out.println("My address: "+Address);
17     }
18 }
```



National University of Kaohsiung

ASCII (7-bit version)

Hex	Dec	Char	Hex	Dec	Char	Hex	Dec	Char	Hex	Dec	Char
0x00	0	NULL null	0x20	32	Space	0x40	64	@	0x60	96	~
0x01	1	SOH Start of heading	0x21	33	!	0x41	65	A	0x61	97	a
0x02	2	STX Start of text	0x22	34	"	0x42	66	B	0x62	98	b
0x03	3	ETX End of text	0x23	35	#	0x43	67	C	0x63	99	c
0x04	4	EOT End of transmission	0x24	36	\$	0x44	68	D	0x64	100	d
0x05	5	ENQ Enquiry	0x25	37	%	0x45	69	E	0x65	101	e
0x06	6	ACK Acknowledge	0x26	38	&	0x46	70	F	0x66	102	f
0x07	7	BELL Bell	0x27	39	'	0x47	71	G	0x67	103	g
0x08	8	BS Backspace	0x28	40	(0x48	72	H	0x68	104	h
0x09	9	TAB Horizontal tab	0x29	41)	0x49	73	I	0x69	105	i
0x0A	10	LF New line	0x2A	42	*	0x4A	74	J	0x6A	106	j
0x0B	11	VT Vertical tab	0x2B	43	+	0x4B	75	K	0x6B	107	k
0x0C	12	FF Form Feed	0x2C	44	,	0x4C	76	L	0x6C	108	l
0x0D	13	CR Carriage return	0x2D	45	-	0x4D	77	M	0x6D	109	m
0x0E	14	SO Shift out	0x2E	46	.	0x4E	78	N	0x6E	110	n
0x0F	15	SI Shift in	0x2F	47	/	0x4F	79	O	0x6F	111	o
0x10	16	DLE Data link escape	0x30	48	0	0x50	80	P	0x70	112	p
0x11	17	DC1 Device control 1	0x31	49	1	0x51	81	Q	0x71	113	q
0x12	18	DC2 Device control 2	0x32	50	2	0x52	82	R	0x72	114	r
0x13	19	DC3 Device control 3	0x33	51	3	0x53	83	S	0x73	115	s
0x14	20	DC4 Device control 4	0x34	52	4	0x54	84	T	0x74	116	t
0x15	21	NAK Negative ack	0x35	53	5	0x55	85	U	0x75	117	u
0x16	22	SYN Synchronous idle	0x36	54	6	0x56	86	V	0x76	118	v
0x17	23	ETB End transmission block	0x37	55	7	0x57	87	W	0x77	119	w
0x18	24	CAN Cancel	0x38	56	8	0x58	88	X	0x78	120	x
0x19	25	EM End of medium	0x39	57	9	0x59	89	Y	0x79	121	y
0x1A	26	SUB Substitute	0x3A	58	:	0x5A	90	Z	0x7A	122	z
0x1B	27	FSC Escape	0x3B	59	;	0x5B	91	[0x7B	123	{
0x1C	28	FS File separator	0x3C	60	<	0x5C	92	\	0x7C	124	
0x1D	29	GS Group separator	0x3D	61	=	0x5D	93]	0x7D	125	}
0x1E	30	RS Record separator	0x3E	62	>	0x5E	94	^	0x7E	126	~
0x1F	31	US Unit separator	0x3F	63	?	0x5F	95	_	0x7F	127	DEL



The Class String

□ You can google java api

```
String blessing = "Live long and prosper";  
System.out.println(blessing);
```

```
System.out.println("100" + 42); // 10042
```

```
System.out.println(100 + 42); // 142
```

```
boolean equals(Other_String)
```

```
String greeting = "Hello"
```

```
Greeting.equals("Hello") returns true
```

```
Greeting.equals("hello") returns false
```




National University of Kaohsiung

The Class String

`boolean equals(Other_String)`

`String greeting = "Hello"`

`Greeting.equalsIgnoreCase("Hello")` returns true

`Greeting.equalsIgnoreCase("hello")` returns true

`String substring(Start, End)`

Example

After program executes `String sample = "AbcdefG"`

`sample.substring(2,5)` returns "cde"

從第2個位置開始，第5個位置結束

Positions are counted 0, 1, 2, etc.



National University of Kachsiung

Variable as Box





Variable Declaration

- ❑ Variable declaration tells the compiler to allocate appropriate memory space for the variable based on its data type.
- ❑ You give an identifier with a specific type for the variable.
- ❑ It is worth to mention that, the data type determines the size, which is measured in bytes.



int x; // x is declared as an integer data type variable



National University of Kaohsiung

Variable as Alias of Memory Address

- ❑ The number 0x000abc26 stands for one memory address in hexadecimal (0-9, and A-F)
- ❑ The variable x itself refers to 0x000abc26 in the program after compilation.

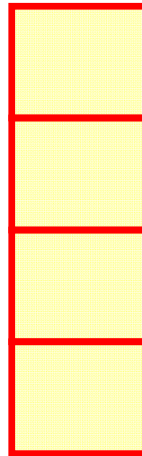


0x000abc26

0x000abc27

0x000abc28

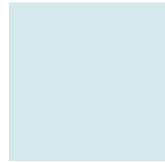
0x000abc29





Floating Points

- ❑ Floating points are used when evaluating expressions that require fractional precision.
 - All transcendental math functions, such as `sin()`, `cos()`, and `sqrt()`, return **double** values.
- ❑ The performance for the **double** values is actually faster than that for **float** values on modern processors that have been optimized for high-speed mathematical calculations.
- ❑ Be aware that floating-point arithmetic can only **approximate** real arithmetic. (Why?)

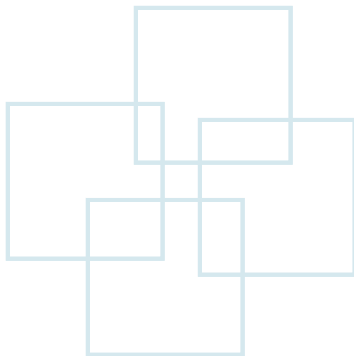


National University of Kachsiung

Example:

```
1 public class floatingPointsDemo {  
2     public static void main(String args[]) {  
3         System.out.println(0.5 - 0.1 - 0.1 - 0.1 - 0.1 - 0.1);  
4     }  
5 }
```

❑ The result is 0 ? Why ?





National University of Kaohsiung

IEEE Floating-Point Representation

$$x = (-1)^s \times M \times 2^e$$

$$7.0 = 7 \times 2^0$$

$$\Rightarrow 111 \times 2^0$$

$$\Rightarrow 0.111 \times 2^3$$

$$e = 3, M = 0.111$$

- ❑ The sign s determines whether the number is negative ($s = 1$) or positive ($s = 0$).
- ❑ The significant M is a fractional binary number that ranges either between 1 and $2-\varepsilon$, or between 0 and $1-\varepsilon$.
- ❑ The exponent e weights the value by a (possibly negative) power of 2.



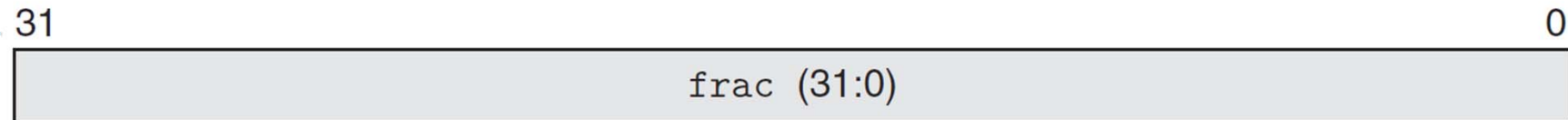
National University of Kachsiung

An Illustration

Single precision



Double precision





Tricky Pitfalls

❑ Can you explain this result?

– Output is 0. Why?

```
...  
double x = 1 / 2;  
System.out.println(x); // output?  
...
```

❑ Try it

```
int num1 = 7;  
int num2 = 5;  
float result = num1 / num2;  
System.out.println(result);
```



National University of Kaohsiung

Exercises

- Write a program that inputs the name, quantity, and price of three items. The name may contain spaces. Output a bill with a tax rate of 6.25%. All prices should be output to two decimal places. The bill should be formatted in columns with 30 characters for the name, 10 characters for the quantity, 10 characters for the price, and 10 characters for the total. Sample input and output are shown as follows:

Input name of item 1:

lollipops

Input quantity of item 1:

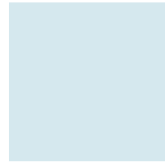
10

Input price of item 1:

0.50

Input name of item 2:

diet soda



National University of Kachsiung

Exercises

Input quantity of item 2:

3

Input price of item 2:

1.25

Input name of item 3:

chocolate bar

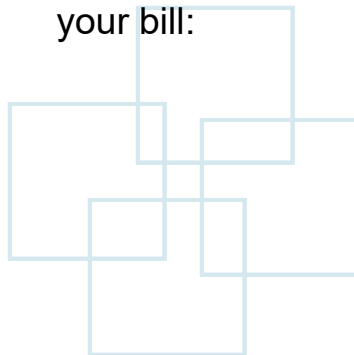
Input quantity of item 3:

20

Input price of item 3:

0.75

your bill:



Item	Quantity	Price	Total
lollipops	10	0.50	5.00
diet soda	3	1.25	3.75
chocolate bar	20	0.75	15.00
Subtotal			23.75
6.25% sales tax			1.48
Total			25.23