

FORM 4 PASCAL PROGRAMMING**Unit 2: Input and Output Statements**

March, 01

2.1. OUTPUT STATEMENTS: WRITELN AND WRITE

- **Purpose (目的):** to output (print) information from the computer to the user
- **Syntax (語法):**

```
Write(Item1, Item2, ... ItemN);
Writeln(Item1, Item2, ... ItemN);
```

- Writeln (pronounced as "write-line") statement displays the data on screen and then *moves the cursor to the next new line* (開新行) while write statement does not.

Examples: (writeln)

Statement	Output
Writeln(123);	
Writeln(-123);	
** writeln(123.0);	
** writeln(0.000123);	
Writeln('a');	
Writeln('Hello');	
Writeln('How are you ?');	
Writeln;	
Writeln('a', 'b', 'c');	
Writeln('program', 213);	
Writeln(1+2+3*4);	

Examples: (write)

Statement	Output
Write('Hello');	
Write('Welcome to Pascal programming');	
Write('I have a ');	
Write('Macintosh ');	
Write('Computer.');	

2.2. FORMATTED (格式化) OUTPUT

- **Purpose:** to add certain format features within the write and writeln statements
- **Syntax:**

Write(num : W);	{for integers}
Writeln(num : W);	{for integers}
Write (num : W : D);	{for real numbers}
Writeln (num : W : D);	{for real numbers}

Where W = field width (min. Number of character spaces needed)
 D = number of decimal places

- **Examples:**

	Statement	Output
Integers	<pre>Writeln(1234:7); Writeln(1234:6); Writeln(1234:4); Writeln(1234:1); Writeln(1234:0);</pre>	
Real numbers	<pre>Writeln(12.35:10:4); Writeln(12.35:10:3); Writeln(12.35:8:1); Writeln(12.35:8:0); Writeln(-12.35:10:1); Writeln(12.35:9); Writeln(12.35:7); Writeln(12.35:0:2);</pre>	
Strings	<pre>Writeln('a':6); Writeln('abc':6); Writeln('abc':9);</pre>	

2.3. INPUT STATEMENT: READ AND READLN

- **Purpose (目的):** to read data from the keyboard or input data file to designated variables

- **Syntax:**

```
Read(variable1, variable2,...VariableN);
Readln(variable1, variable2,...VariableN);
```

- Readln statement is very similar to the read statement with one exception: readln statement will cause the computer to *advance (move) the cursor to the beginning of the next line* after execution while read statement will not.
- **Example #1:** a Pascal program reads four data items into four variables and then echo their values.

Program statement	Output
<pre>Program ReadData; Var Student : string; Classnum : integer; Testscore : real; Grade : char; Begin Readln (student); Readln(classnum); Readln(testscore); Readln(grade); Writeln; Writeln ('student name: ', student); Writeln('class number: ',classnum); Writeln('test score: ',testscore); Writeln('grade: ',grade) End.</pre>	<pre>Wong Ka Wai 37 65.5 A</pre>

- **Example #2:** A Pascal program contains the following three readln statements:

```
Readln(a,b);
Readln(c,d);
Readln(e,f);
```

Suppose that the variables are all of the type integer and the input lines contains the numbers

```
1 2 3 ↵
4 5 6 ↵
7 8 9 ↵
```

Write down the values assigned to all variables listed below:

Variable	Value assigned
a	
b	
c	
d	
e	
f	

- **Example #3:** A Pascal program contains both READLN and READ statements:

```
read(a,b);
readln(c);
readln(d,e);
readln(e,f,g);
```

Suppose that the variables are all of the type INTEGER and the input lines contains the numbers

```
1 2 3 ↵
4 5 6 ↵
7 8 9 ↵
```

Write down the values assigned to all variables listed below:

Variable	Value assigned
a	
b	
c	
d	
e	
f	
g	

end of unit 2