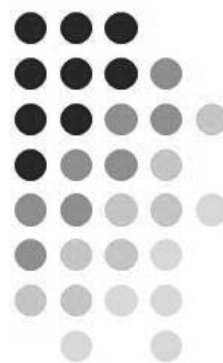


# 第二章

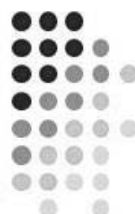
## C語言基本概述

學習C語言的基本語法  
認識關鍵字與識別字的不同  
學習程式碼偵錯的流程  
學習如何提高程式的可讀性



1

### 2.1 簡單的例子



## 簡單的 C 程式

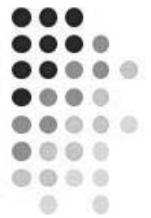
- 下面的程式碼可印出兩行字串：

```
01  /* prog2_1, 簡單的C語言 */
02  #include <stdio.h>          /* 把 stdio.h 這個檔案含括進來 */
03  #include <stdlib.h>         /* 把 stdlib.h 這個檔案含括進來 */
04  int main(void)              /* 主函數 main() 從這兒開始 */
05  {
06      int num;                 /* 宣告整數變數 num*/
07      num=2;                   /* 把 num 的值設為 2 */
08      printf("I have %d cats.\n",num); /* 呼叫 printf()函數 */
09      printf("You have %d cats, too.\n",num); /* 呼叫 printf()函數 */
10      system("pause"); /* 呼叫 dos 裡的 pause 指令,用來暫停程式的執行 */
11      return 0;
12  }
```

/\* prog2\_1 OUTPUT--

```
I have 2 cats.
You have 2 cats, too.
-----*/
```

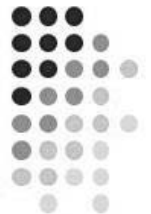
2



## 含括指令與標頭檔 (1/4)

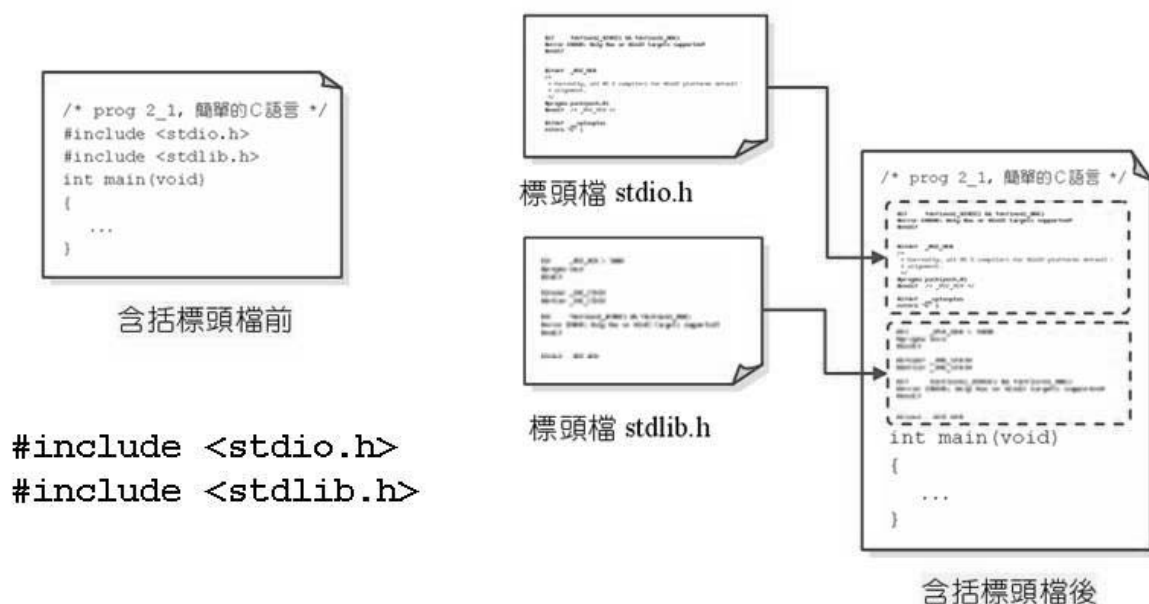
- `#include` 是一種前置處理器指令
  - `#include` 稱為含括指令
  - 語法為 `#include<標頭檔>`
  - 標頭檔 (header file) 會被含括在程式碼的起頭處
  - 因為是在編譯前執行，所以稱為"前置" 處理器

3

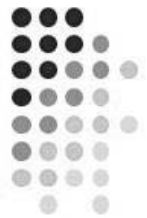


## 含括指令與標頭檔 (2/4)

- 含括動作前後的比較：



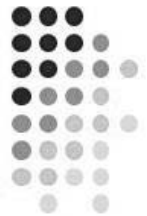
4



## 含括指令與標頭檔 (3/4)

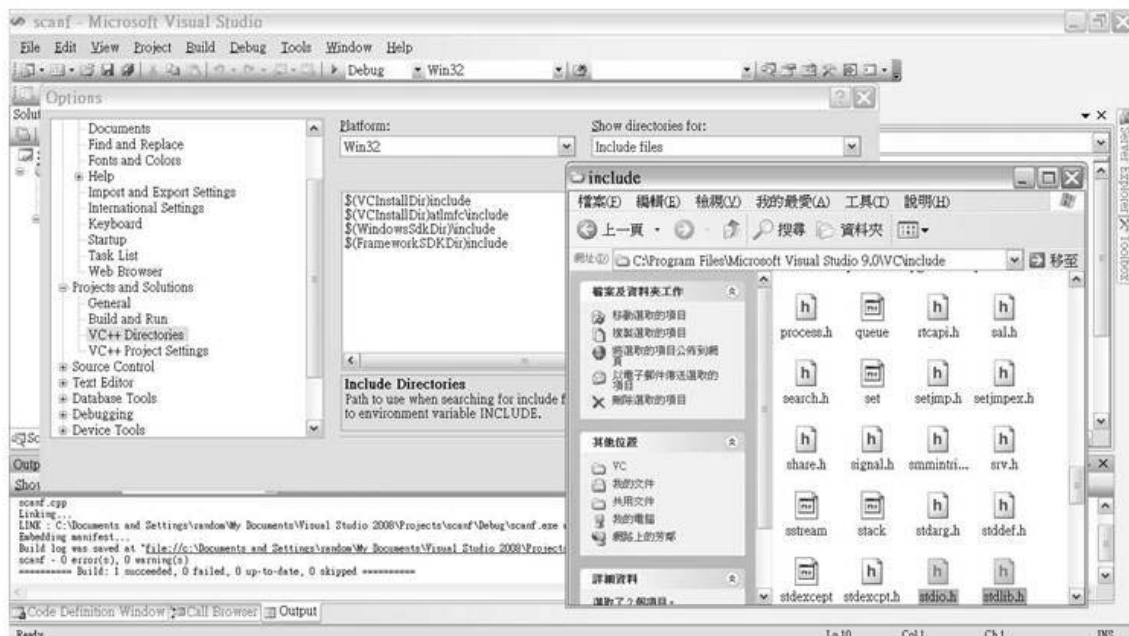
- 不包括stdio.h或stdlib.h標頭檔也可以編譯？
  - 某些編譯器會將常用的標頭檔自動載入
  - 有些編譯器發現沒含括標頭檔之後，會出現警告訊息，並自動含括該有的標頭檔
- 對語法檢查較嚴的編譯程式，如果沒有含括該有的標頭檔，即無法編譯

5

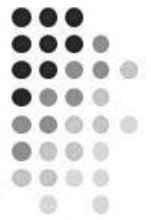


## 含括指令與標頭檔 (4/4)

- 標頭檔的內容：



6



# 主函數main()

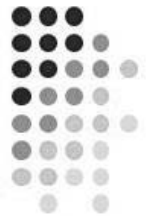
- main() 是程式執行的開端
- 每個 C 程式必須有一個 main() 函數，而且只能有一個

傳回型態為整數      main()函數不需傳入引數

```

    ↑           ↑
int main( void )
{
    程式敘述;
    return 0;  → main()函數執行完畢，傳回整數 0
}
  
```

7



# 程式區塊及本體

- 程式區塊與本體的範圍：

```

01  /* prog2_2, main()函數的本體與程式區塊 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06      int i=2; /* 宣告整數 i，並設值為 2 */
07      if(i<5) /* if 區塊由此開始 */
08      {
09          printf("變數 i 的值小於 5");
10          printf("\n"); /* 換行 */
11      } /* if 區塊結束 */
12      system("pause");
13      return 0;
14  }
  
```

/\* prog2\_2 OUTPUT--  
變數 i 的值小於 5  
-----\*/

main() 函數的本體

if 敘述的程式區塊

8

表 3.2.1 C 語言所提供的本資料型態

資料型態		型態說明	位元組	表示範圍
整數 類型	long int	長整數	4	-2147483648 到 2147483647
	int	整數	4	-2147483648 到 2147483647
	short int	短整數	2	-32768 到 32767
	char	字元	1	0 到 255 (256個字元)
浮點數 類型	float	浮點數	4	1.2e-38 到 3.4e38
	double	倍精度浮點數	8	2.2e-308 到 1.8e308

# 變數的使用

## ● 宣告方式：

- `int num;`                      `/* 宣告 num 為整數變數 */`
- `int a,b,c;`                  `/* 同時宣告 a,b 與 c 為整數變數 */`
- `float sum=0.0;`            `/* 宣告浮點數變數sum，並設值為0.0 */`

## ● 變數的資料型態：

- `char`    字元，如 'A'、'2' 與 '&' 等
  - `int`      整數
  - `long`    長整數
  - `short`   短整數
  - `float`   單精度浮點數
  - `double` 倍精度浮點數
- 如 12、-27 等
- 如 12.762、-37.483 等

9

## 2.2 解析 C 語言

# 變數的命名規則

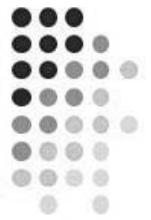
## ● 變數名稱可以是英文字母、數字或底線

- 名稱中不能有空白字元
- 第一個字元不能是數字
- 不能使用到關鍵字

```

intel_4x            /* 正確 */
AMD                /* 正確，變數的第一個字母可以是底線 */
2dos                /* 錯誤，變數的第一個字母不能是數字 */
my dogs            /* 錯誤，變數不能有空格 */
goto                /* 錯誤，變數不能是C語言的關鍵字 */

```



## 變數的設值方式

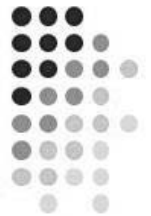
- 宣告的時候設值

```
int num=2;      /* 宣告變數，並直接設值 */
```

- 宣告後再設值

```
int num1,num2;      /* 宣告變數 */
char ch;
num1=2;             /* 將整數變數num1的值設為2 */
num2=30;            /* 將整數變數num2的值設為30 */
ch = 'm';           /* 將字元變數ch的值設為'm' */
```

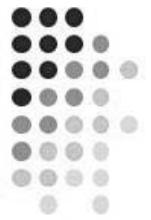
11



## 宣告變數的好處

- 宣告變數有許多好處，諸如：
  - 方便編譯器找到錯誤的變數名稱
  - 避免變數名稱打錯（如數字0與英文字母O）
  - 除錯容易
  - 增加程式的可讀性
  - 便於程式碼的維護

12



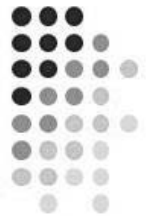
# 格式化輸出函數printf()

- 利用printf() 函數印出字串：

```
01  /* prog2_3, printf()函數的練習 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06      int num=2;    /* 定義變數 num，並設值為 2 */
07      printf("I have %d cats.\n",num);  /* 呼叫printf()函數 */
08      system("pause");
09      return 0;
10  }

/* prog2_3 OUTPUT---
I have 2 cats.
-----*/
```

13



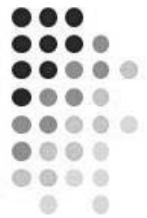
# 識別字 (identifier)

- 識別字是用來命名變數或函數的文字

```
01  /* prog2_3, printf()函數的練習 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06      int num=2;    /* 定義變數 num，並設值為 2 */
07      printf("I have %d cats.\n",num);  /* 呼叫printf()函數 */
08      system("pause");
09      return 0;
10  }
```

識別字

14



## 關鍵字（keyword）(1/2)

- 關鍵字是編譯程式本身所使用的識別字

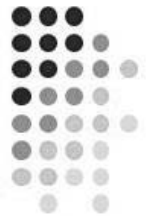
```

01  /* prog2_3, printf() 函數的練習 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06      int num=2;      /* 定義變數 num，並設值為 2 */
07      printf("I have %d cats.\n",num);  /* 呼叫 printf() 函數 */
08      system("pause");
09      return 0;
10  }

```

關鍵字

15



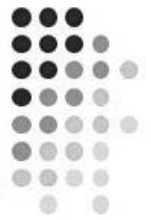
## 關鍵字（keyword）(2/2)

- 下表為 C 語言的關鍵字

auto	break	case	char	const
continue	default	defined	do	double
else	enum	extern	float	for
goto	if	int	long	register
return	short	signed	sizeof	static
struct	switch	typedef	union	unsigned
void	while	volatile		

16

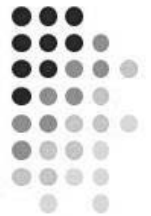




# 程式錯誤的分類

- 語法錯誤 (syntax error)
  - 程式含有不合語法的敘述，它無法被編譯程式翻譯
- 語意錯誤 (semantic error)
  - 語意錯誤(又稱邏輯錯誤)，就是程式的執行結果非我們所願

17



# 語法錯誤

- 下面是有語法錯誤的程式：

```

01  /* prog2_4, 有錯誤的程式 */
02  #include <stdio.h>
03  #include <stdlib.h>
04
05  int main(void)
06  {
07      int num;          /* 宣告整數 num */
08      num=2;            /* 將 num 設值為 2 */
09      printf("I have %d dogs. \n",num);
10      printf("You have %d dogs, too. \n,num);
11      system("pause")
12      return 0;
13  )

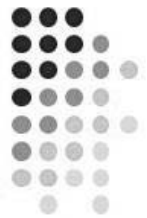
```

```

/* prog2_4 OUTPUT 除錯後的結果 --
I have 2 dogs.
You have 2 dogs, too.
-----*/

```

18



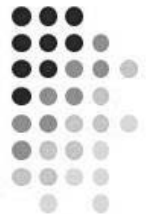
## 語意錯誤

- 下面是語意錯誤的程式：

```
01  /* prog2_5, 語意錯誤的程式 */
02  #include <stdio.h>
03  #include <stdlib.h>
04
05  int main(void)
06  {
07      int num=-2;          /* 宣告整數變數 num，並設值為-2 */
08
09      printf("I have %d dogs.\n", num);
10      system("pause");
11      return 0;
12  }
```

/\* prog2\_5 OUTPUT---  
I have -2 dogs.  
-----\*/

19

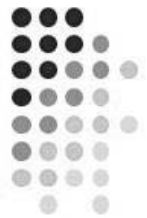


## 提高程式的可讀性 (1/4)

- 程式碼請用固定字距

```
/* 使用固定字距的程式碼，字型為 Courier New */
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    printf("We all love C. \n");
    system("pause");
    return 0;
}
```

20

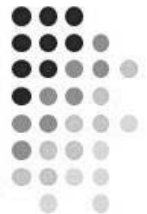


## 提高程式的可讀性 (2/4)

- 使用非固定字距，且斜體字的程式碼，較難閱讀

```
/* 使用非固定字距，且斜體字的程式碼，字型為 Times New Roman*/
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    printf("We all love C. \n");;
    system("pause");
    return 0;
}
```

21



## 提高程式的可讀性 (3/4)

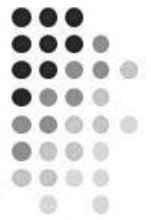
- 程式碼縮排，可提高可讀性

```
/* prog2_6, prog2_7 OUTPUT-----
cats are running, dogs are chasing.
-----*/
```

```
01  /* prog2 6, 有縮排的程式碼 */
02  #include <stdio.h>
03  #include <stdlib.h>
04
05  int main(void)
06  {
07      int i;
08      for(i=1;i<=2;i++)
09      {
10          printf("Cats are running, ");
11          printf("dogs are chasing.\n");
12      }
13      system("pause");
14      return 0;
15  }
```

```
01  /* prog2 7, 沒有縮排的程式碼 */
02  #include <stdio.h>
03  #include <stdlib.h>
04
05  int main(void)
06  {
07      int i;
08      for(i=1;i<=2;i++)
09      {
10          printf("Cats are running, ");
11          printf("dogs are chasing.\n");
12      }
13      system("pause");
14      return 0;
15  }
```

22



## 提高程式的可讀性 (4/4)

- 註解有助於程式的閱讀與偵錯

```
/* prog2_7, examples */  
/* created by Wien Hong */
```

 } 以註解符號對每一行做註解

```
/*  
    This paragraph demonstrates the  
    capability of comments used by C  
    November 06 2003  
*/
```

 } 於「/\*」和「\*/」符號之間的文字均是註解