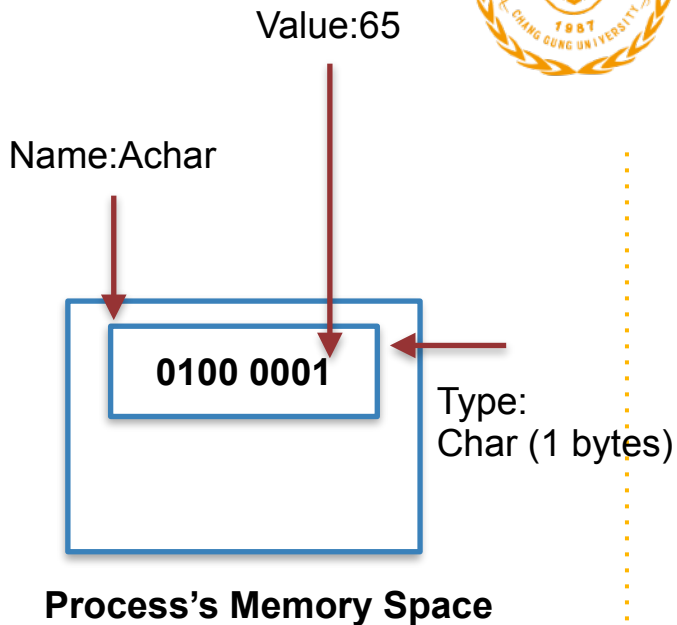


Introduction to Computer Science Fall 2022 #18 Chi-Jen Wu

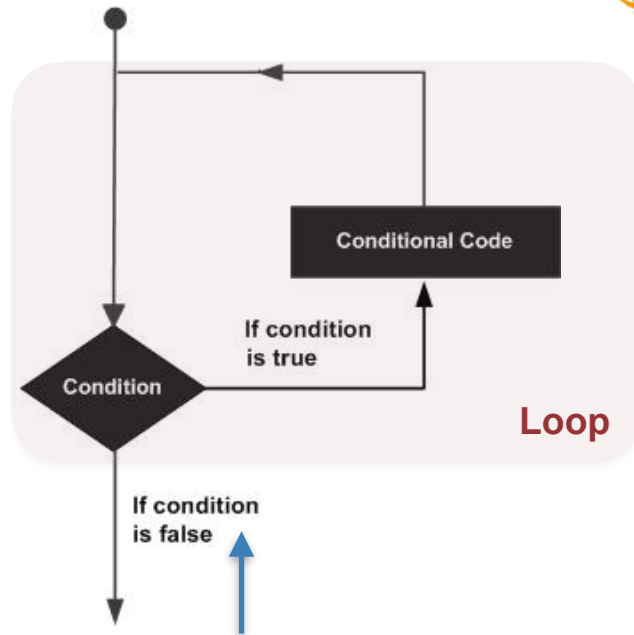
Variables

- 值 (Value)
 - 在記憶體實際的二進位值
- 名字 (Name)
 - 在程式裡的代號
- 型別 (Type)
 - 在記憶體所佔的空間



Flow of Control

- if
- If else
- If else if else
- Switch case
- While
- For
- Do while loop
- For loop
- Do While loop
- Nested loops
- Break
- Continue

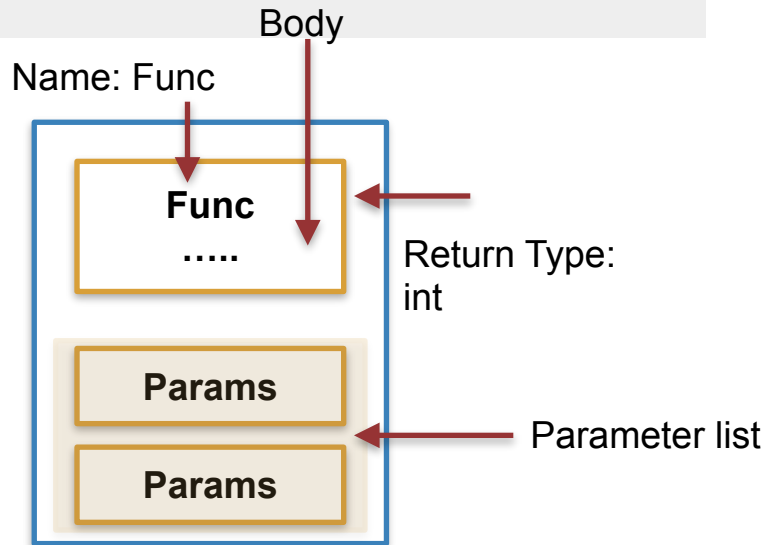


Condition 決定要不要轉彎

Function Basics

```
return_type function_name( parameter list ) {  
    body of the function  
}
```

- Return Type
- Function Name
- Parameters
- Function Body



Process's Memory Space

Topics


- Problem Solving with Programming Language
- C Programming
 - C Basics
 - Variables
 - Flow of Control
 - Function Basics ←
 - Programming with Arrays
 - Strings
 - Structures
 - Streams and File I/O
- Cloud Platform/Cloud Shell Editor (gcc/g++/Makefile)
- Cloud Platform/Cloud Source Repositories (git)



Variables 轉型

- 在C語言裡，常常會在多種變數轉換型別
- int to float, int to double, float to int 等等
- C語言允許強制轉型！
- 滿常用！但須小心使用！

```
1 #include <stdio.h>
2
3 int main() {
4     printf("  %d\n", (int)1.4);
5     printf("  %.2f\n", (float)2);
6 }
7
```



```
1
2.00

...Program finished with exit code 0
Press ENTER to exit console.
```

If & Loop & Function

- 到目前你應該要知道，目前都是這些的排列組合
 - IF
 - Loop
 - Function
- 要可以默寫出他們的語法
- 如果還不能，代表練習還不夠，請花時間練習！
- 要練習到可以直接翻譯你的語言成為C語言

練習的方式就是把投影片的
程式碼抄打一次，
然後自己打一次。
過兩天再打一次。

If (condition)

- 用來邏輯判斷
- 如果是
 - 就做A
- 如果不是
 - 就做B
- 兩者都不是
 - 就做C

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int a = 1;
6     int b = 10;
7     if (a > b) {
8         A printf("a > b\n");
9     } else if (a < b) {
10        B printf("a < b");
11    } else {
12        C printf("a = b");
13    }
14
15    return 0;
16 }
```

a < b

...Program finished with exit code 0
Press ENTER to exit console.



for (;;)

- 無窮迴圈
 - while (1)
 - for (;;)
- 寫的時候
 - 先打好語法
 - 再去填判斷

```
3 int main()
4 {
5     if () {} ←
6     for () {}
7     while () {} ←
8     return 0;
9 }
```

```
3 int main()
4 {
5     for (int i=0; i<=10; i++) {
6         printf(); ←
7     }
8     return 0;
9 }
```

for (int i=0; i<=10; i++)

- 第一個初始設定
- 第二個是限制條件
- 第三個是每做完一個 Loop 後要做的事
 - 可以是空的

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int i=0, j =0;
6     for (; i<=10; i++, j+=i) {
7         printf("i = %d, j = %d\n", i, j);
8     }
9     return 0;
10 }
```

```
i = 0, j = 0
i = 1, j = 1
i = 2, j = 3
i = 3, j = 6
i = 4, j = 10
i = 5, j = 15
i = 6, j = 21
i = 7, j = 28
i = 8, j = 36
i = 9, j = 45
i = 10, j = 55
```

請注意開始是0
條件為 <=
總共做了 11次



for 一行， 只做一件事

- 不建議這種寫法

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int i=0, j =0;
6     for (; i<=10; i++, j+=i) ←
7         printf("i = %d, j = %d\n", i, j);
8
9     return 0;
10 }
```

- 很容易發生低級錯誤

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int i=0, j =0;
6     for (; i<=10; i++, j+=i) ←
7         printf("XXXX"); ←
8         printf("i = %d, j = %d\n", i, j); ←
9
10    return 0;
11 }
```



```
1 #include <stdio.h>
2
3 int main() {
4     double pi = 4.0;
5     int flag = 1;
6     long ipi = 0;
7     int x = 0;
8     for (int i = 3; i <= 1000000; i+=2) {
9         if (flag == 0) {
10             pi += (4.0/i);
11             flag++;
12         } else {
13             pi -= (4.0/i);
14             flag--;
15         }
16         ipi = pi * 100000; ←
17         if (ipi == 314159) {
18             x = i;
19             break;
20         }
21     }
22     printf("%d %.5f", x, pi);
23     // 272241 3.14159
24     return 0;
25 }
```

272241 3.14160

...Program finished with exit code 0
Press ENTER to exit console.

- 計算PI by Gregory-Leibniz Series
- $PI = (4/1) - (4/3) + (4/5) - (4/7) + (4/9) - \dots$
- 請利用 for loop 來計算 pi
- double PI = 4.0f;
- 找到 x 計算出來 = 3.14159 (精準五位數)



```
1 #include <stdio.h>
2
3 int main() {
4     for (int i=1; i<=9; i++) {
5         for (int j=1; j<=9; j++) {
6             printf("%d*%d=%d\t", i, j, i*j);
7         }
8         printf("\n");
9     }
10 }
```

- 九九乘法表

- 兩個loop

- 請改成一個loop

input								
1*1=1	1*2=2	1*3=3	1*4=4	1*5=5	1*6=6	1*7=7	1*8=8	1*9=9
2*1=2	2*2=4	2*3=6	2*4=8	2*5=10	2*6=12	2*7=14	2*8=16	2*9=18
3*1=3	3*2=6	3*3=9	3*4=12	3*5=15	3*6=18	3*7=21	3*8=24	3*9=27
4*1=4	4*2=8	4*3=12	4*4=16	4*5=20	4*6=24	4*7=28	4*8=32	4*9=36
5*1=5	5*2=10	5*3=15	5*4=20	5*5=25	5*6=30	5*7=35	5*8=40	5*9=45
6*1=6	6*2=12	6*3=18	6*4=24	6*5=30	6*6=36	6*7=42	6*8=48	6*9=54
7*1=7	7*2=14	7*3=21	7*4=28	7*5=35	7*6=42	7*7=49	7*8=56	7*9=63
8*1=8	8*2=16	8*3=24	8*4=32	8*5=40	8*6=48	8*7=56	8*8=64	8*9=72
9*1=9	9*2=18	9*3=27	9*4=36	9*5=45	9*6=54	9*7=63	9*8=72	9*9=81

...Program finished with exit code 0
Press ENTER to exit console.□



```
1 #include <stdio.h>
2
3 int main() {
4     for (int i=1, j=0; i <= 9; ) {
5         printf("%d*%d=%d\t", i, j, i*++j);
6         if (j == 9) {
7             i++;
8             j = 0;
9             printf("\n");
10        }
11    }
12 }
```

- 九九乘法表
 - 兩個loop
 - 請改成一一個loop

input

1*1=1	1*2=2	1*3=3	1*4=4	1*5=5	1*6=6	1*7=7	1*8=8	1*9=9
2*1=2	2*2=4	2*3=6	2*4=8	2*5=10	2*6=12	2*7=14	2*8=16	2*9=18
3*1=3	3*2=6	3*3=9	3*4=12	3*5=15	3*6=18	3*7=21	3*8=24	3*9=27
4*1=4	4*2=8	4*3=12	4*4=16	4*5=20	4*6=24	4*7=28	4*8=32	4*9=36
5*1=5	5*2=10	5*3=15	5*4=20	5*5=25	5*6=30	5*7=35	5*8=40	5*9=45
6*1=6	6*2=12	6*3=18	6*4=24	6*5=30	6*6=36	6*7=42	6*8=48	6*9=54
7*1=7	7*2=14	7*3=21	7*4=28	7*5=35	7*6=42	7*7=49	7*8=56	7*9=63
8*1=8	8*2=16	8*3=24	8*4=32	8*5=40	8*6=48	8*7=56	8*8=64	8*9=72
9*1=9	9*2=18	9*3=27	9*4=36	9*5=45	9*6=54	9*7=63	9*8=72	9*9=81

...Program finished with exit code 0
Press ENTER to exit console.



```
1 #include <stdio.h>
2
3 int main() {
4     for (int i=1, j=0; i<=9 ; ) {
5         printf("%d*%d=%d\t", i, j, i*++j);
6         (j==9) ? (i++, j=0, printf("\n")) : i ;
7     }
8     return 0;
9 }
```

● 九九乘法表

- 兩個loop
- 請改成一一個loop
- 三元運算子

input

1*1=1	1*2=2	1*3=3	1*4=4	1*5=5	1*6=6	1*7=7	1*8=8	1*9=9
2*1=2	2*2=4	2*3=6	2*4=8	2*5=10	2*6=12	2*7=14	2*8=16	2*9=18
3*1=3	3*2=6	3*3=9	3*4=12	3*5=15	3*6=18	3*7=21	3*8=24	3*9=27
4*1=4	4*2=8	4*3=12	4*4=16	4*5=20	4*6=24	4*7=28	4*8=32	4*9=36
5*1=5	5*2=10	5*3=15	5*4=20	5*5=25	5*6=30	5*7=35	5*8=40	5*9=45
6*1=6	6*2=12	6*3=18	6*4=24	6*5=30	6*6=36	6*7=42	6*8=48	6*9=54
7*1=7	7*2=14	7*3=21	7*4=28	7*5=35	7*6=42	7*7=49	7*8=56	7*9=63
8*1=8	8*2=16	8*3=24	8*4=32	8*5=40	8*6=48	8*7=56	8*8=64	8*9=72
9*1=9	9*2=18	9*3=27	9*4=36	9*5=45	9*6=54	9*7=63	9*8=72	9*9=81

...Program finished with exit code 0
Press ENTER to exit console.



```
4 int main() {  
5     int i = 12345;  
6     int t = 0;  
7     int d = 0;  
8  
9     if (i >= 1000) {  
10         t = i % 10000 / 1000;  
11     }  
12     d = i % 10;  
13     i -= t*1000;  
14     i -= d;  
15     i += d*1000;  
16     i += t;  
17     printf("i = %d\n", i);  
18     return 0;  
19 }  
20
```



- 把一個大於零的數的千位數字和個位數字調換
 - `int i = 12345;`
 - 12345 → 15342
 - 123 → 3120
 - 12 → 2010
 - 1 → 1000



Inverted pyramid of *, rows = 5

```
* * * * *
 * * * *
  * * *
   * *
    *
```

```
...Program finished with exit code 0
Press ENTER to exit console.
```



```
1 #include <stdio.h>
2 /* global variable declaration */
3 int rows = 5;
4 void print_spaces(int r) {
5     /* local variable declaration */
6     int i;
7 }
8
9 void print_stars(int r) {
10    /* local variable declaration */
11    int i;
12 }
13
14 int main() {
15    /* local variable declaration */
16    int j;
17    for (int i = rows; i >= 1; --i) {
18        print_spaces(i);
19        print_stars(i);
20    }
21
22    return 0;
23 }
```

Function Scope Rules
在C語言裡，很嚴謹的方式
規範變數的參照範圍 (scope)

QZ#2 , 今天12/29 17:00PM Commit to your GitHub

- Inverted pyramid of *, rows = 5



- 請注意！作業沒有輸入！
- 如果程式需要輸入零分計算！



Thanks!

Open for any questions

CJ Wu

cjwu@mail.cgu.edu.tw