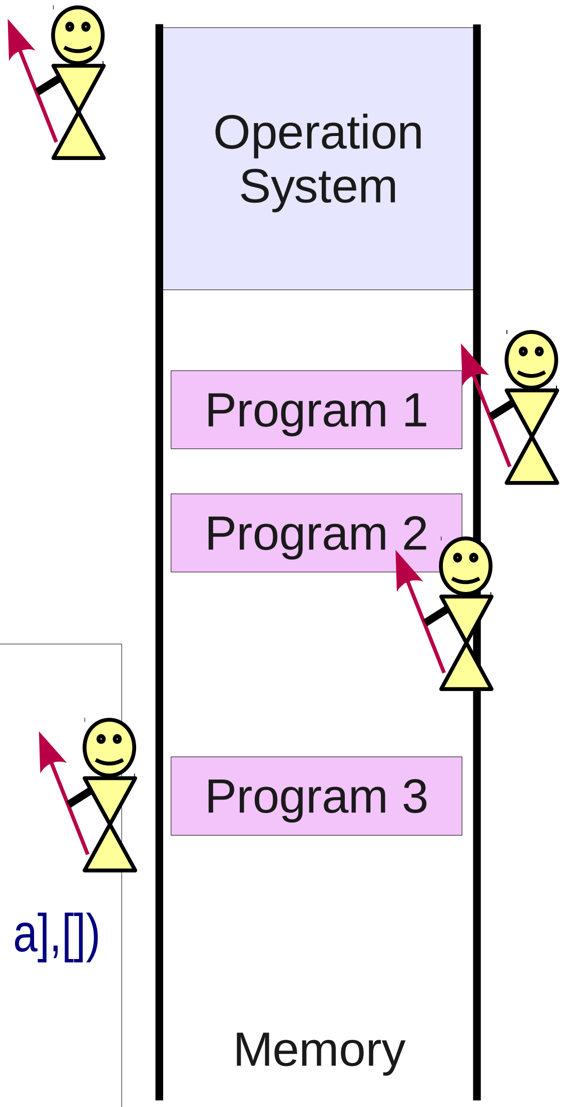
命令式 (imperative) 程式設計

對比： 宣告式 (declarative) 程式設計

```
>>> def fac(n):
...     if n: return n * fac(n - 1)
...     else: return 1
...
>>> fac(10)
3628800
```

```
>>> def exc(a, k): return [x for x in a if x != k]
...
>>> def per(a):
...         if len(a) == 1: return [a]
...         else: return sum([map(lambda l: [k] + l, per(exc(a, k))) for k in a],[])
...
>>>  per(['a','b','c'])
[['a', 'b', 'c'], ['a', 'c', 'b'], ['b', 'a', 'c'], ['b', 'c', 'a'], ['c', 'a', 'b'], ['c', 'b', 'a']]
```



C++ 程式語言

檔案:　ex1.cc

```
#include <iostream>
using namespace std;

double func(double a)
{
     return a * a;
}

int main()
{
     double val;
     cout << "Please enter a value: ";
     cin >> val;
     double sq;
     sq = func(val);
     cout << "The square of " << val << " is " << sq << "\n";
     return 0;
}
```

函數

變數宣告

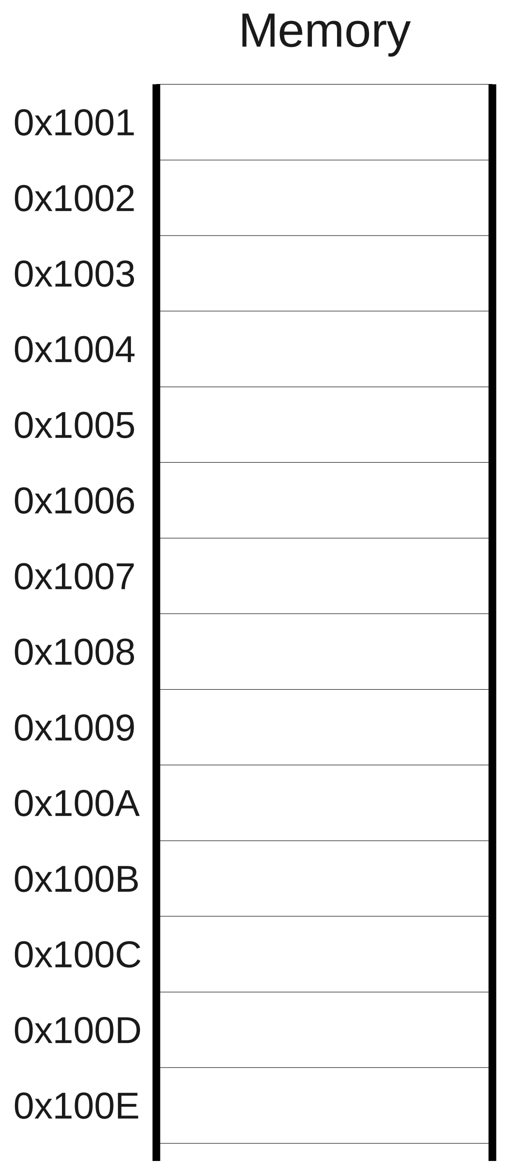
輸入

函數呼叫

輸出

主程式

變數、位址與陣列



```
int a = 10;
char s[4] = "hi!";
char * d = s + 1;
cout << d[1] << '\n';
int b = * (s + 1);
cout << b << '\n';
```

型別	位元	位元組	例子	名稱
bool	1	1	true, false	布林
char	8	1	'A', '\n'	字元
short	16	2	123, 0234	短整數
int	32	4	100000, 0x1ADE23	整數
long	64	8	102L, 0x123L	長整數
float	32	4	1220.125	單精度浮點數
double	64	8	1e-200	雙精度浮點數
long double	80	12	9.32e+600	長雙精浮點數

C++ 運算元

precedence	operator	description	overridable	associativity
1	::	scope resolution	n	LR
2	()	function call	y	LR
	[]	array access	y	
	->	member access	y	
	.		n	
	++ --	postfix	y	
	dynamic_cast static_cast reinterpret_cast const_cast	type conversion	n	
	typeid	type information	n	
3	!	logical negation	y	RL
	~	bitwise negation	y	
	++ --	prefix	y	
	+ -	sign operations	y	
	* &	indirect & ref.	y	
	sizeof	size in bytes	n	
	new new[] delete delete[]	memory	y	
	(type)	cast to type	y	
4	->*	pointer selector	y	LR
	.*	object selector	n	
5	* / %	arithmetic operations	y	LR
6	+ -			
7	<< >>	shift operations	y	LR
8	< <= > >=	relational operations	y	LR
9	== !=			
10	&	bitwise AND	y	LR
11	^	bitwise XOR	y	LR
12	|	bitwise OR	y	LR
13	&&	logical AND	y	LR
14	||	logical OR	y	LR
15	?:	ternary conditional	n	RL
16	= += -= *= /= %= &= ^= |= <<= >>=	assignment	y	RL
17	,	sequential evaluation	y	LR

C++ 字串 class 及 I/O Stream

基本的輸出輸入

```
cout << "Hello World!";
string var;
cin >> var;
```

字串的使用

```
#include <string>
       。。。
string str;
str = "Hello";
str.append(' ');
str += "World!";
cout << str.length() << '\n';
cout << str.substr(3,5) << '\n';
```

字串的轉換

```
#include <sstream>
       。。。
ostringstream oss;
oss << "a = " << a;
cout << oss.str();
```

函數與程序

call by reference

call by value

```
#include <iostream>
using namespace std;

int func(int a, int * b, int & c)
{
	a = 10;
	* b = 20;
	c = 30;
	return a + * b + c;
}

int main()
{
	int x = 1;
	int y = 2;
	int z = 3;
	int r = func(x, & y, z);
	cout << x << ' ' << y << ' ' << z << ' ' << r << '\n';
	return 0;
}
```

```
Output:
1 20 30 60
```

return reference

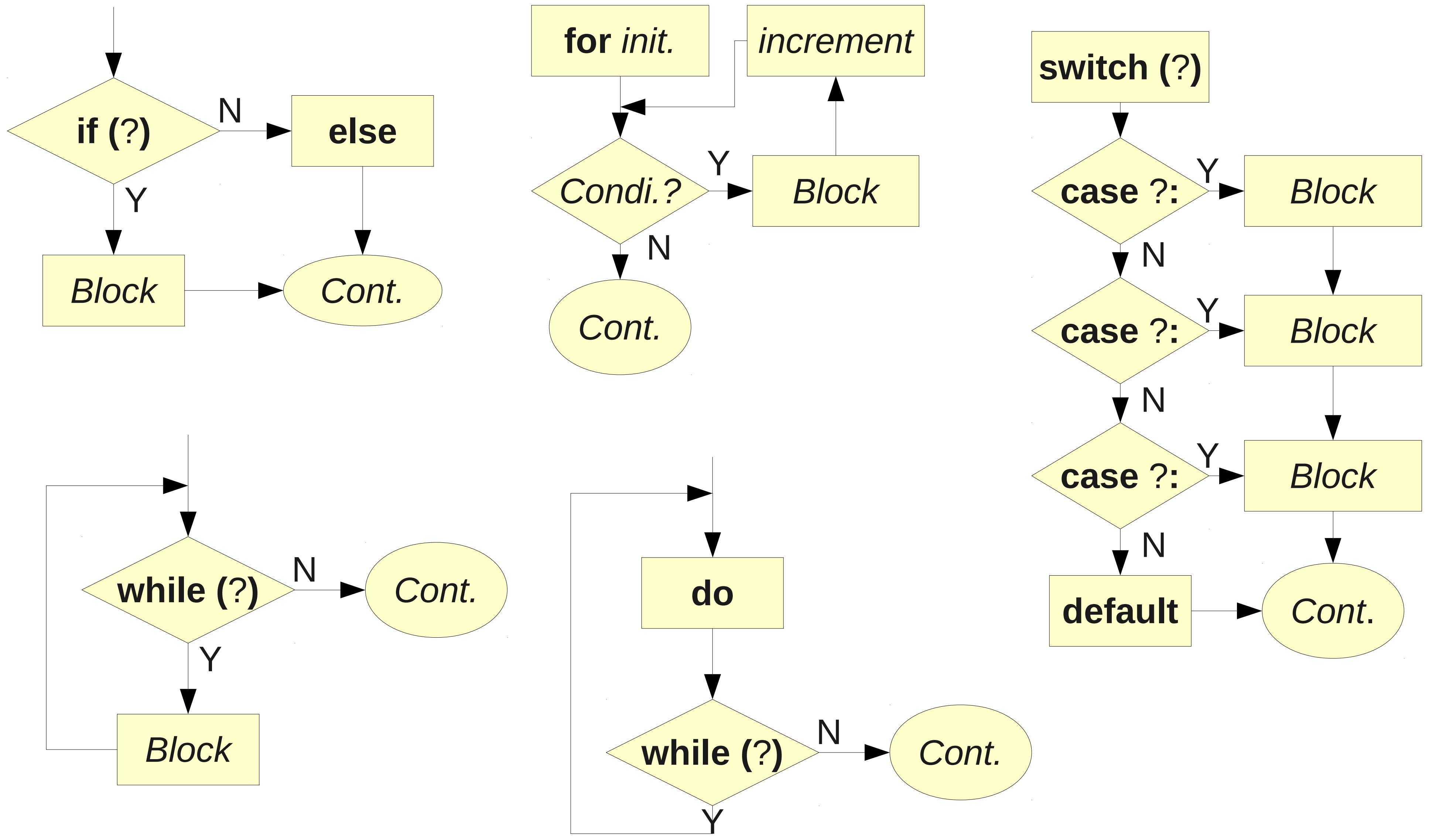
```
#include <iostream>
using namespace std;

int & func(int i)
{
	static int d = 0;
	d += i;
	return d;
}

int main()
{
	cout << func(2) << '\n';
	cout << func(3) << '\n';
	func(4) += 11;
	cout << func(5) << '\n';
	return 0;
}
```

```
Output:
2
5
25
```

流程控制


if (?)
N
else
Y
Block
Cont.
for init.
increment
Condi.?
Y
Block
N
Cont.
switch (?)
case ?:
Y
Block
N
case ?:
Y
Block
N
case ?:
Y
Block
N
default
Cont.
while (?)
N
Cont.
Y
Block
do
while (?)
N
Cont.
Y

Python 解譯器

- 無預設變數型別
- 串列 list
- 强制性縮排
- 函數及匿名函式

本週作業

1. 在 C++ 的迴圈（for, while, do while）中要如何直接跳下一輪或是結束迴圈?
2. 何謂遞迴 (recursion)？ 請並舉出一個程式範例。
3. 程式寫作 (用 C++ string class)： 輸入一 string， 輸出其中字元 'a' 出現的次數。

   ［隨意題］ 輸出所出現各種字元的次數。
4. 程式寫作： 輸入一整數 n
   - a) 輸出 Fibonacci 數列的前 n 項
   - b) 輸出其 16 進位的表示字串
   - c) 輸出以 '#' 字元繪製有 n 字元為底的等腰三角形
5. 閱讀課程網站上的相關連結

程式執行範例

＊＊＊ 可以自行連結到 CompPhys SSH 伺服器上測試

```
cp1@area:~$ hw3-2
input: lsqaoaqqaa
the number of 'a's is 4
cp1@area:~$ hw3-2x
input: lsqaoaqqaa
the number of 'a's is 4
the number of 'l's is 1
the number of 'o's is 1
the number of 'q's is 3
the number of 's's is 1
cp1@area:~$
```

程式執行範例

＊＊＊ 可以自行連結到 CompPhys SSH 伺服器上測試

```
cp1@area:~$ hw3-3a
Input n: 8
The Fibonacci Seq.: 1, 1, 2, 3, 5, 8, 13, 21
cp1@area:~$ hw3-3b
Input n: 54321
54321 = 0xE431
cp1@area:~$ hw3-3c
Input n: 6
      #
     # #
    # # #
   # # # #
  # # # # #
 # # # # # #
cp1@area:~$
```