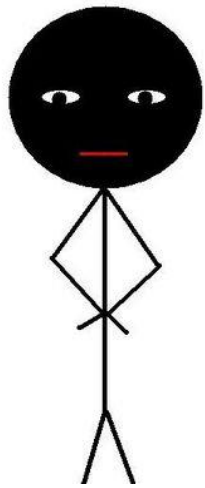


Notes 1

Name: Art lin

Class: C programming



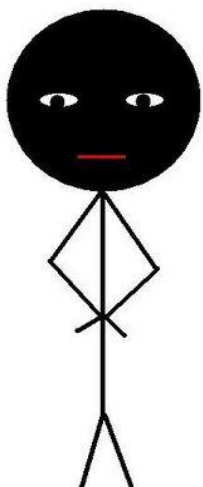
C language



Talk



C programming is composed of commands in terms of human



C language



翻譯員
Translator

compiler

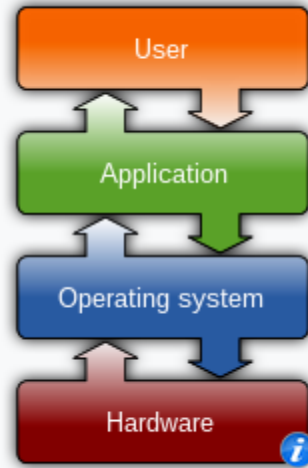


```
#include <stdio.h>

int main() {
    // printf() displays the string inside quotation
    printf("Hello, World!");
    return 0;
}
```

```
0100101101100010101
010001011001110011
10011011111010101011
11010110010001111110
0100101001100010100
0101100001110111100
```

Operating systems



Common features

C program

Window 11

CPU Time: 0.00 sec(s), Memory: 2076 kilobyte(s)

Sum of $x+y = 35$

```
int x=10;
```

Command terminator,end of command,called **semicolon**



Assing 10 to vaibles x
One direction only

```
1 #include<stdio.h>
2
3 int main() {
4     int x=10;
5     int y=25;
6     int z=x+y;
7     printf("Sum of x+y = %i", z);
8     return 0;
9 }
```

Curly baraces , 大括號

```
1  #include<stdio.h>
2
3  int main() {
4      int x=10;
5      int y=25;
6      int z=x+y;
7      printf("Sum of x+y = %i", z);
8      return 0;
9  }
```

Main function , the starting point of execution

Bring in library function

```
1 #include<stdio.h>
2
3 int main() {
4     int x=10;
5     int y=25;
6     int z=x+y;
7     printf("Sum of x+y = %i", z);
8     return 0;
9 }
```



```
int x=10;
```



Assing 10 to vaibles x
One direction only

(A) Home of a variable

(B) In the view of hardware it 's a memory location with
space to restore data.

Datatype

Integer-3,-2,-1,0,1,2,3,.....

Rule-before using a variable must define it's
data type.

Data type of ariable

A. Before using a variable.

We must declare it's data type.

B. The reason for example,our integer and character are same to the machines.

printf (control string, arg1,arg2,...);

A is the control statement control format for presentation.



Int B=66,C=66,

Printf(A, B, C, D)

A is the control statement control format for presentation.
B, C,.....are arguments; No control.



```
#include <    >
```

(1)

```
(2)Int  main {
```

```
    Printf("    ");    (3)  
    Return 0;
```

```
}
```

```
#include <    >
```

(1)

```
(2)Int  main {
```

```
(3)    for(.....){  
        (4)  
    }
```

```
}
```

```
#include <stdio.h>

int main() {
    // printf() displays the string inside quotation
    printf("Hello, World!");
    return 0;
}
```



Box1

```
#include <stdio.h>
```



Box2

```
int main() {
    // printf() displays the string inside quotation
    printf("Hello, World!");
    return 0;
}
```

Box3



```
int main()
```



Box3

```
// printf() displays the string inside quotation
printf("Hello, World!");
return 0;
```


Programming statements

Types(類型)

- A) execu. on statement like `printf`
- B) Logica on statement like `if...else`
- C) Looping on statement like `for()`
 - (1) scope by braces
 - (2) statements inside braces
 - (3) will be executed for each iteration indexing variable
 - (4) looping condition , if true then continue
 - (5) looping update operation , EX: `i++`
EX: `i++` meaning to increase 1 for each loop

```
#include <stdio.h>
int main(){

    int i;
    for(i=0;i<4;i++)
        printf(" %d",i);
}
```

A deeper concept about looping

(1) Why variable for looping

```
EX for(i=0;i<4;i++){  
    for(j=0;j<4;j++) 4 times  
}
```

(A) Looping expression (a standard format structure 結構)
(表達式)

$F(x), f(x, y), \dots$

Looping(i)=for(i=0;i<4;i++)

(1) init

(2) condition

(3) updates

(B) Real world (to see is to believe)

```
for(i=0;i<4;i++)      0123  
printf("*");          I   ****X
```

*
**



I\J	0	1	2	3
0	T	F		
1	T	T	F	
2	T	T	T	F
3	T	T	T	T



I\J	0	1	2	3
0	*			
1	*	*		
2	*	*	*	
3	*	*	*	*

(1)

i\j	0	1	2	3
-----	---	---	---	---

OTF

1 T T F

2 T T T F

3 T T T T

(2)

i\j	0	1	2	3
-----	---	---	---	---

0 T T T T

1 T T T F

2 T T F

3TF

(3)

i\j	0	1	2	3
-----	---	---	---	---

0 T T T T

1 F T T T

2 F F T T

3 F F F T

(4)

i\j	0	1	2	3
-----	---	---	---	---

O F F F T

1 F F T T

2 F T T T

3 TTTT