

ENAPS 10

11-24-09



---

• Input & Output in C++

cin >> variable;

cout << variable;

cout << expression;

cout << literal value;

Ex. int a = 1, b = 2, c = 7;

cout << a;

cout << b;

cout << c;

output : 127□

cout << endl; prints newline  
char

Ex. cout << a << endl;  
 cout << b << endl;  
 cout << c << endl;

output: 1  
 2  
 7  
 □

Ex. cout << a << " " <<  
 << b << " " <<  
 << c << endl;

output: 1 2 7  
 □

Ex int hours = 6, minutes = 43;

```
cout << "The time is now \n"
      << hours << ':' << minutes
      << endl;
```

\n ← new line char

```
cout << "\n";
cout << endl;
cout << '\n';
```

→ output:

The time is now  
6:43

□ ← Position of cursor.

# Calculation $\rightarrow$ Assignment

4

Variable = expression;

Ex. int a=1, b=3, c=5, d;

d = a + b - c;

before: 

a	b	c	d
1	3	5	

after: 

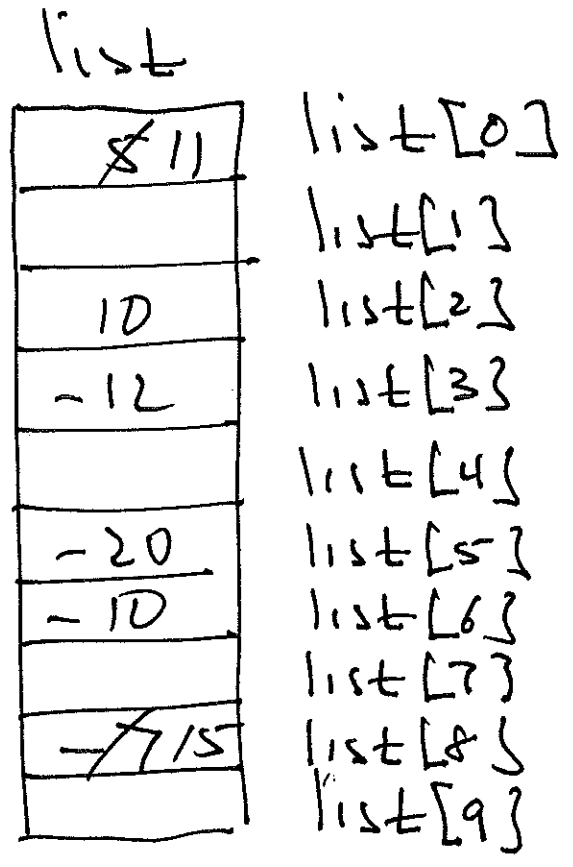
1	3	5	-1
---	---	---	----

Examples of: straight line algorithm.

# • Arrays

Declare :

```
int list[10];  
double weight[20];  
char word[30];
```



$$\text{list}[0] = 5 ;$$

$$\text{list}[3] = -12 ;$$

$$\text{list}[8] = \text{list}[0] + \text{list}[3] ;$$

$$\text{list}[0] = 11 ;$$

$$\text{int } i = 2, j = 5, k = 6 ;$$

$$\text{list}[i] = 10$$

$$\text{list}[j] = -20$$

$$\text{list}[k] = \text{list}[i] + \text{list}[j] ;$$

$$k = i + j ;$$

$$\text{list}[k] = 15$$

Ex.

int list[5] = {2, 4, 6, 8, 10};

list	
2	0
4	1
6	2
8	3
10	4

## Logical Expressions

→ Comparison operators

• logical operators

# Comparison OPS

<u>Symbol</u>	<u>Ex.</u>	<u>Value</u>
==	1 == 2	false
<	1 < 2	true
<=	1 <= 2	true
>	1 > 2	false
>=	1 >= 2	false
!=	1 != 2	true

## Compare for eq:

Pseudo code =  
 C++ ==

## Assignment

Pseudo code ←  
 C++ =



# Logical ops

19

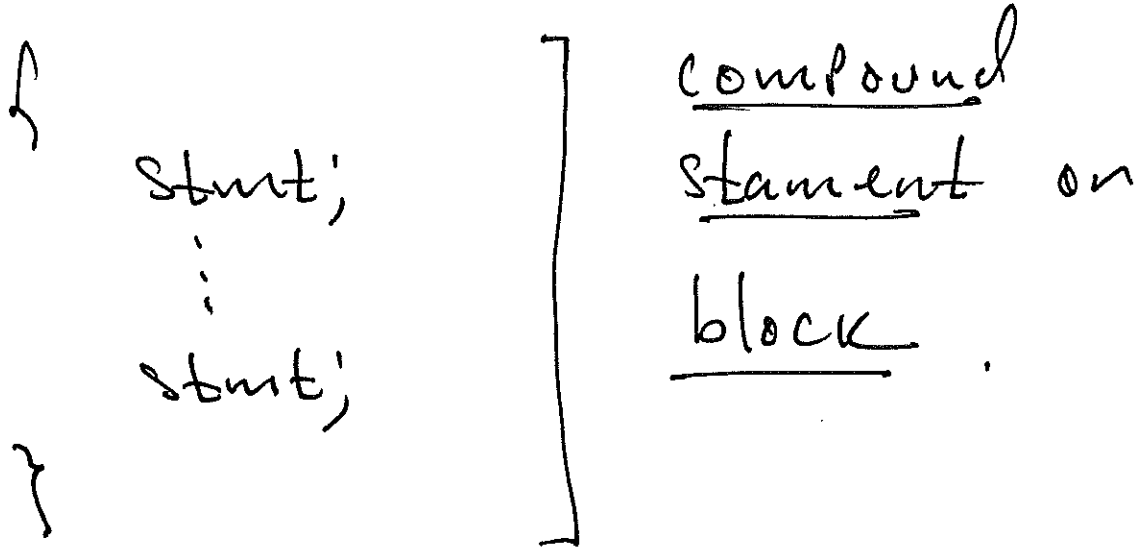
<u>Op.</u>	<u>Symbol</u>	<u>Ex.</u>	<u>Value</u>
and	&&	$(1 < 2) \&\& (3 > 4)$	false
or		$(1 < 2)    (3 > 4)$	true
not	!	$!(1 == 2)$	true

## Conditionals:

```
if (cond)
    stmt;
stmt;
```

```
if (cond) {
    stmt;
    :
    stmt;
}
stmt;
```

note :



```
if (cond)  
  stmt;  
else  
  stmt;  
stmt;
```

```
if (cond) {  
  stmt;  
  ⋮  
  stmt;  
} else {  
  stmt;  
  ⋮  
  stmt;  
}  
stmt;
```

loops :

- while
- do-while
- for

11

```
while (cond)
    stmt;
```

```
stmt;
```

```
while (cond) {
    stmt;
    :
    stmt;
}
stmt;
```

Ex. Pseudo-code

12

```
Sum ← 0
i ← 1
while i ≤ 10
    Sum ← Sum + i
    i ← i + 1
Print Sum
```

Ex. C++

```
int Sum, i;
Sum = 0;
i = 1;
while (i ≤ 10) {
    Sum = Sum + i;
    i = i + 1;
}
cout << Sum << endl;
```

```

do
    stmt;
while (cond);
stmt;

do {
    stmt;
    :
    stmt;
} while (cond);
stmt;

```

Ex.

```

int sum, i;
sum = 0;
i = 1;
do {
    sum = sum + i;
    i = i + 1;
} while (i <= 10);
cout << sum << endl;

```

```
for (init. LCV ; Last LRC ; increment LCV) {
```

```
    stmt;
```

```
    .
```

```
    stmt;
```

```
}
```

```
stmt;
```

Ex

```
int sum, i;
```

```
sum = 0;
```

```
for (i = 1; i <= 10; i = i + 1) {
```

```
    sum = sum + i;
```

```
}
```

```
cout << sum << endl;
```