

CNAPS 10

11-18-10

L

function main:

```
int main(void) {
```

```
    // var declarations ←
```

```
    // executable stmts
```

```
    return 0;
```

```
}
```

Data types:

int : signed integer

double : floating point number

char : single ascii character

bool : true or false.

Literal values:

int : 1, 2, 57, -379

double : 1.0, 2.0, 57.0, -379.0

char : 'a', 'b', 'c'

bool : true, false

Can initialize & declare vars at same time

Ex. int count = 0; Count
or 0

int count;
count = 0; Count
0

Ex. double sum = 0.0, average;



Ex. char first = 'P';



Ex bool found = false;
 ~~found~~
 false

4

Symbolic constants:

Ex const double pi = 3.14;

pi
3.14

Similar to: Constant Macro

#define PI 3.14

Executable Statements:

- Sequential
 - input
 - output
 - calculation/assignment
- Conditional
- Iterative

Input & Output:

Data streams:

- | | | |
|---------------------------|------------|-------------|
| | <u>C++</u> | <u>unix</u> |
| • standard input stream: | cin | stdin |
| • standard output stream: | cout | stdout |

cin & cout are defined in the
 iostream library: #include <iostream>

Ex.

int count = 5;

⋮

cout << count;

↗
extraction operator

count
[5]

[6]

Ex.

int count;

⋮

cin >> count;

↖
insertion operator

count
[]

count
[6]

↖
if user enters 6

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

cout << a;

cout << b;

cout << c;

output

127

note: cout << endl;

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

cout << a << endl;

cout << b << endl;

cout << c << endl;

output:

```
1
2
7
□
```

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

output:

```
1 2 7
□
```



```

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

```

Same output.

```

Ex. int hours = 6, minutes = 43;

```

```

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

```

output

The time is
6:43
□

String literal
new line char

Calculation & Assignment:

form: Variable = expression;

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

before:

a	b	c	d
1	3	5	

after

a	b	c	d
1	3	5	-1

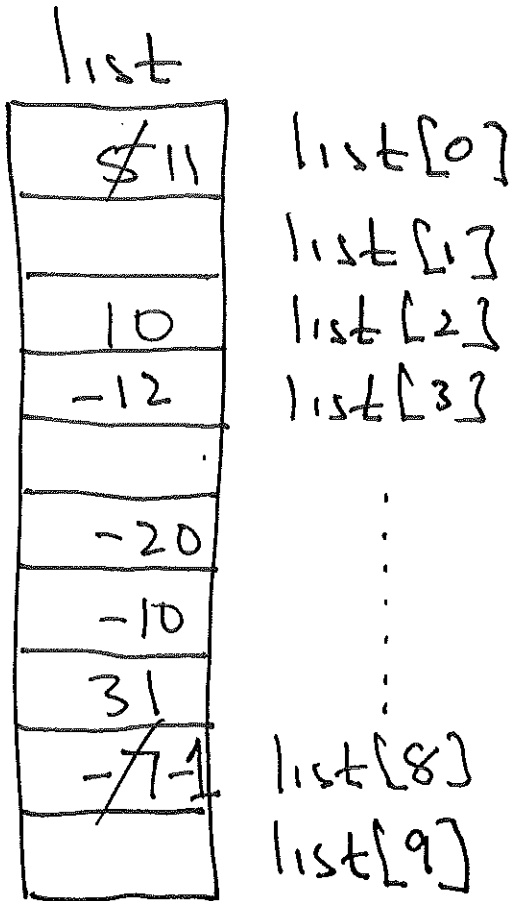
Arrays:

Declaration:

```
int list[10];
```

```
double weight[20];
```

```
char word[30];
```



Continue example:

(12)

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

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

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

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

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

more declare stuffs

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

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

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

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

$$\text{list}[i+j] = 31;$$