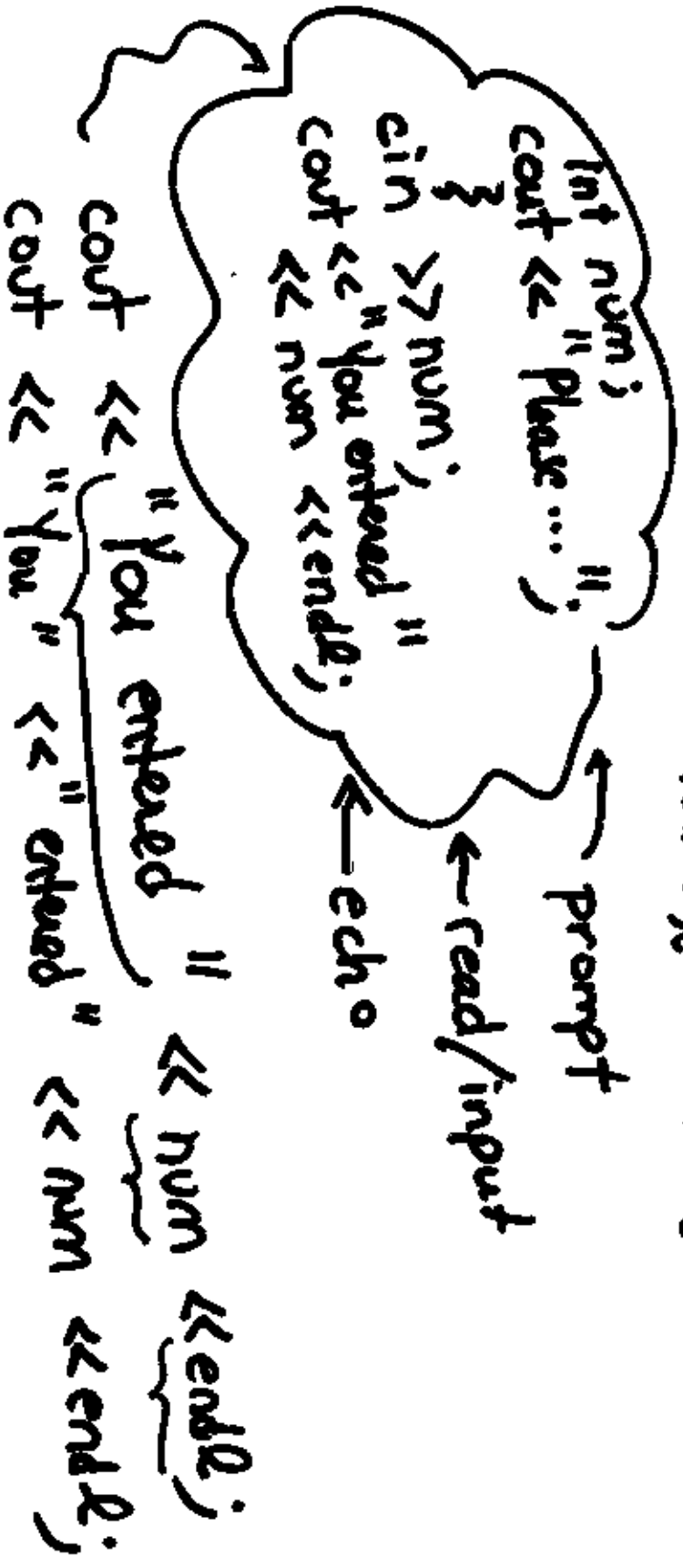


C#

- Data Types: char, int, float, bool
- Input: long, double

```
cin >> variable;
```

letter, (letters, digits, underscore)



```
int num;
cout << "Please ...";
cin >> num;
cout << "You entered";
cout << num << endl;
```

```
cout << "You entered" << num << endl;
cout << "You" << " entered" << num << endl;
```

```
if (num < 0)
  cout << "Error";
```

```
if (num < 0)
  cout << "Error";
}
Block
```



```

if (cost < 0)
    cout << "Error";
else if (cost > 1000.0)
    cout << "Too Expensive";
else
    cout << "We are on track";

```

Relational

< <= > >=

```
if (age >= 20)
```

// age is NOT less than 20

Equality

== !=

```
if (age != 20)
```

Boolean

&& || !
(AND) (OR) (NOT)

4.

```
char yesno = 'y';  
if (yesno == 'y')  
{  
    // ...  
}  
else if (yesno != 'n')  
{  
    cout << "Gee... should be y or n";  
}  
else // yesno is an 'n'  
{  
    // ...  
}
```

↑ char variable ↑ single quotes

IF age NOT less IS

if (age >= 15)

```
char yesno;
cout << "Would you like to continue?"
    << "y or n" << endl;
```

```
cin >> yesno;
```

(A)

```
if (yesno == 'y' || yesno == 'n')
    cout << "Good job!";
else
    cout << "What did you type in";
```

(B)

```
if (yesno != 'y' && yesno != 'n')
    cout << "Error";
```

NO if (yesno == 'y' || 'n')

TRUE Always NO

NO = NO

6. // Find out if age is between 10-30 (inclusive)

if (age >= 10 && age <= 30)
operand True True

if (10 <= age <= 30)

True 100% Time

$\emptyset \rightarrow$ False
Not zero \rightarrow True

count = 100;

↑
assignment operator

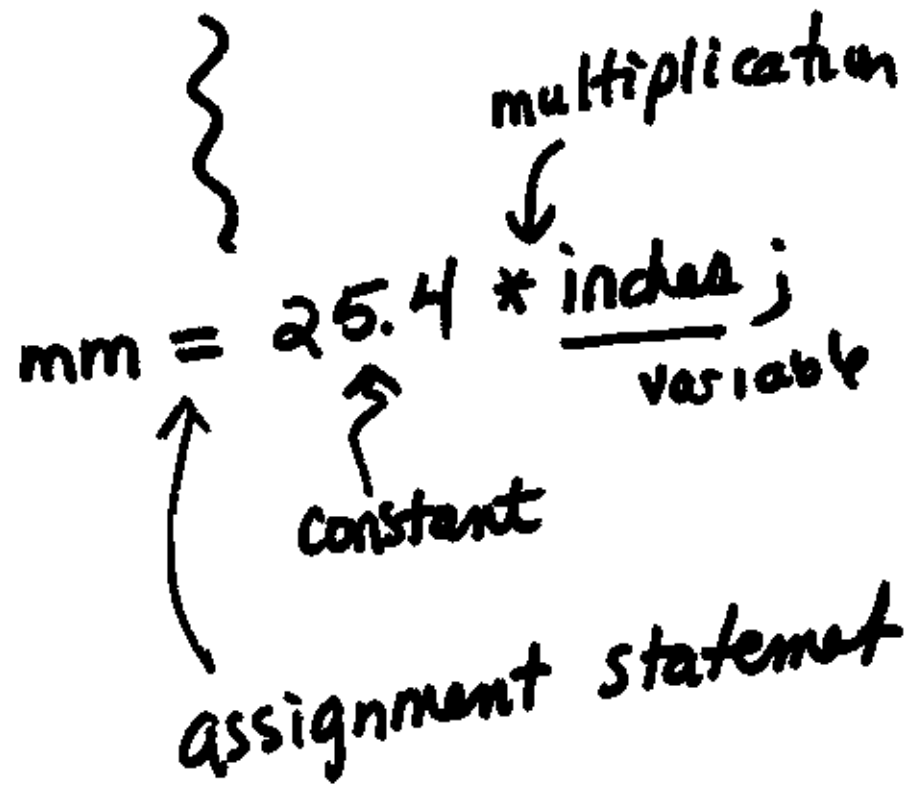
+, -, *, /, %

if (count == 100)

if (count = 100)

if (100 == count)

float inches, mm;



int count;

// Add 1

count = count + 1;

count += 1;

compound arithmetic

+= -= *= /=

%=

Prefix Increment
++ count;

Adds 1.



cout << ++count;



Postfix Increment 8.
count++;



② Adds 1, to count

③ Residual value (Result) is the temp value

"Pre-increment number"

cout << count++;




```
float total = 0.0; float num;
cout << "please enter" << count << "numbers"
<< endl;
```

```
for ( int i = 0; i < count; ++i )
{
    cin >> num;
    total += num;
}
```

Annotations:
 ① under int i = 0
 ② under i < count
 ③ under ++i
 ④ next to ++i
 ⑤ next to ++i
 ⑥ next to ++i
 ⑦ next to ++i
 ⑧ next to ++i
 ⑨ next to ++i
 ⑩ next to ++i
 Prefix

③

```
① int i = 0;
while ( i < count )
{
    cin >> num;
    total += num;
}
++i;
```

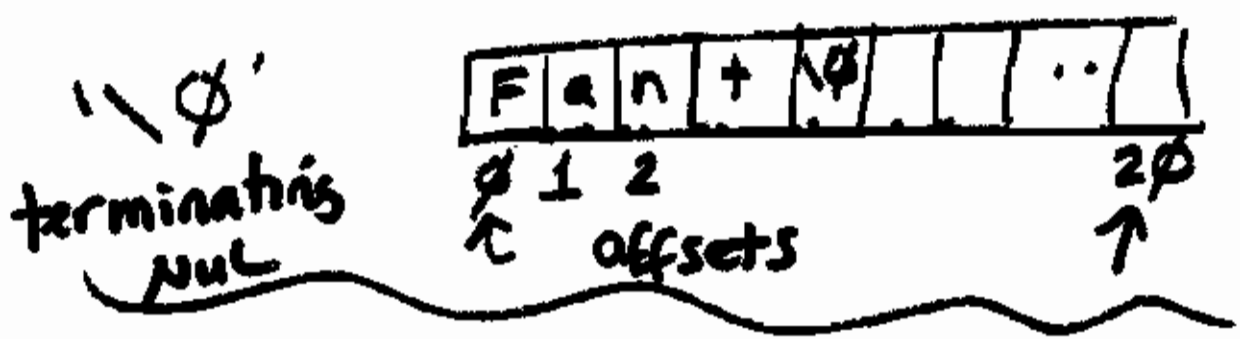
total = total + num;

Annotations:
 ① next to int i = 0
 ② next to i < count
 ③ next to cin >> num;
 ④ next to total += num;
 ⑤ next to ++i;
 ⑥ next to total = total + num;

Arrays

Constant size
↓

```
char name [21];
```



```
#include <ctype.h>
```

⋮

```
int main()
```

```

{
  char name [21];
  name[0] = toupper(name[0]);
}

```

Annotations: local points to the `main` function scope. An arrow points from `name[0]` to the `name` array. Another arrow points from `name[0]` to the `toupper` function call. A third arrow points from `name[0]` to the `name[0]` argument of `toupper`.

Reading in an Array of characters

1) `cin.width(21);`
`cin >> name;`

SKIP leading whitespace, read in characters until whitespace is encountered but not read

Reads in 1 word

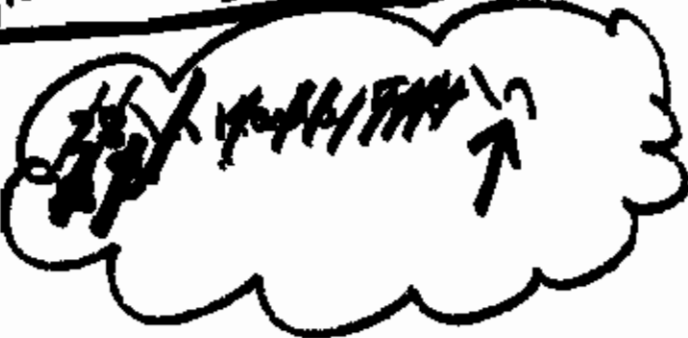
2) `cin.get(name, 21, '\n');`

↑ ↑ ↑
 array name size of array stop at

Reads 20 characters or stops when encounters '\n'

```
cout << " please enter your age ";
cin >> age; cin.ignore();
cin << " please enter your full name ";
```

```
cin.get(name, 21, '\n');
cin.ignore(100, '\n');
```

Input Buffer → 

```
cin.getline(name, 21, '\n');
```