cd

rm

mkdir

mv

ls

touch

man

grep

find

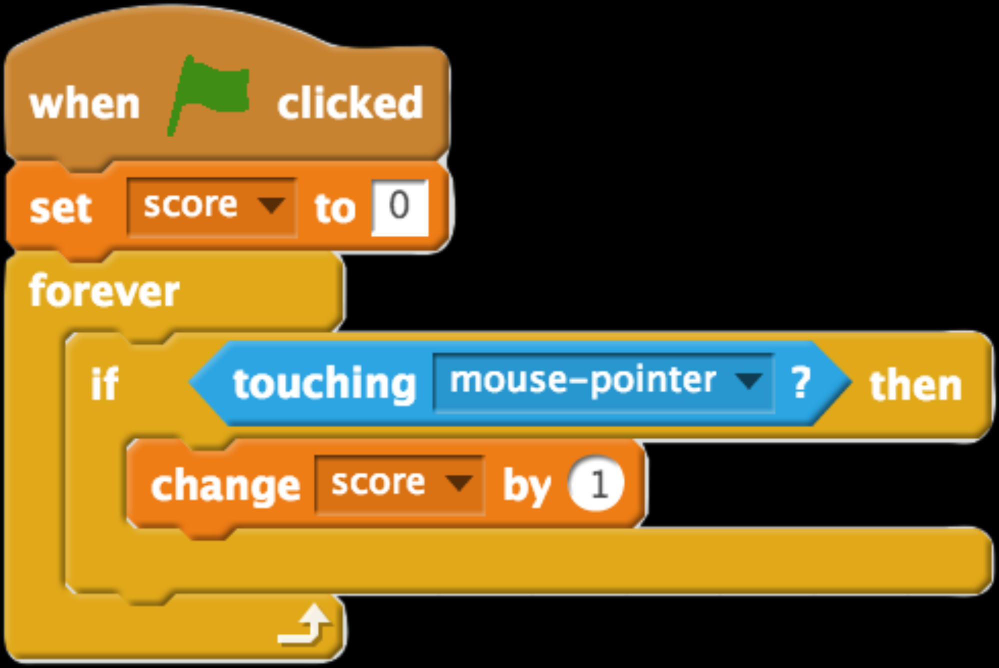
<

>

|

…

## Variables



```
when green flag clicked
set score to 0
forever
    if touching mouse-pointer ? then
        change score by 1
```

## Defining Variables

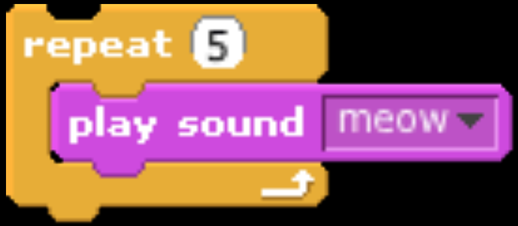
```
type variable_list;

char grade = 'A';

float x, y, z;

int score = 7, num_of_teams = 4;
```

# Conventions

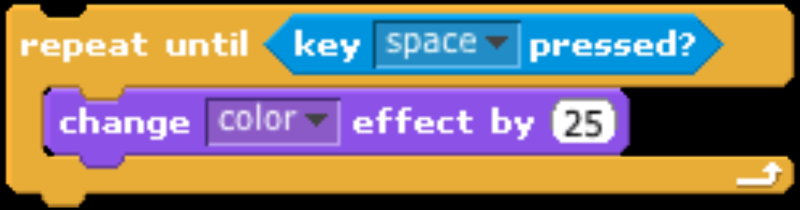
- Meaningful names
  - For loops, single character variables are fine
- Consistent initialization

```
int quarters, dimes = 0, pennies;
```

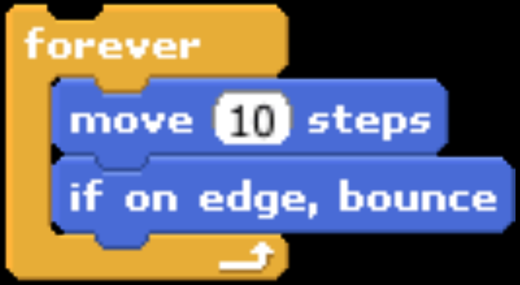
# Loops



```
repeat 5
    play sound meow
```



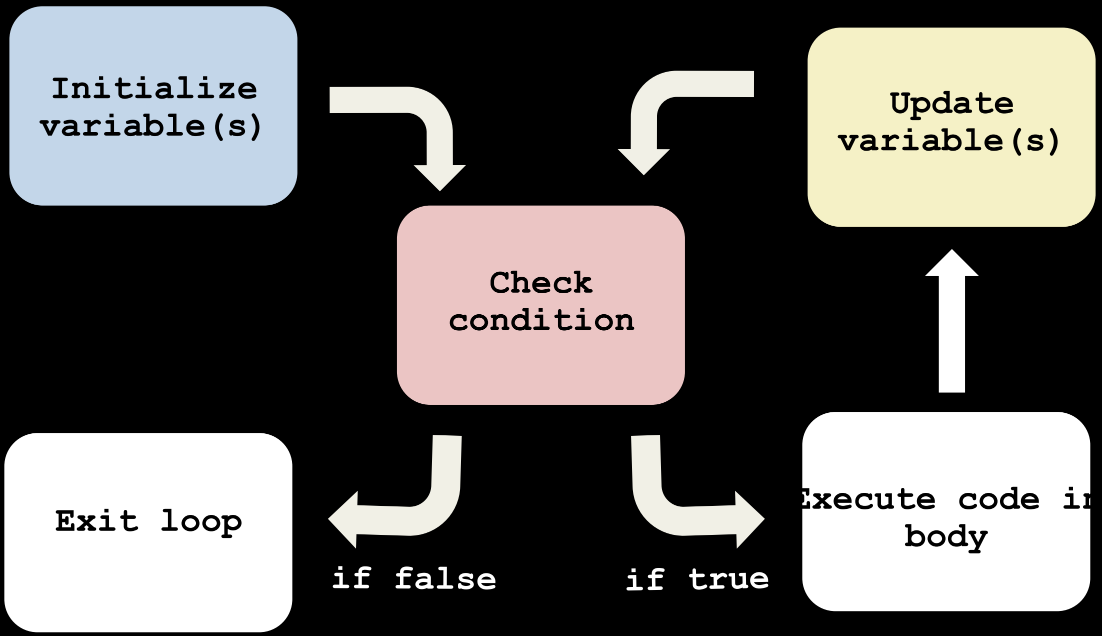
```
repeat until key space pressed?
    change color effect by 25
```



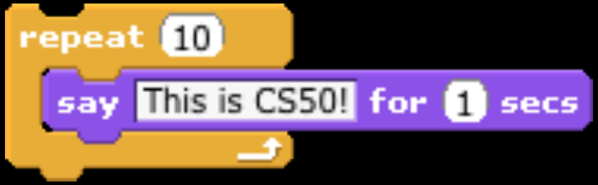
```
forever
    move 10 steps
    if on edge, bounce
```

# For Loops

```
for (initialization; condition; update)
{
      execute this code
}
```



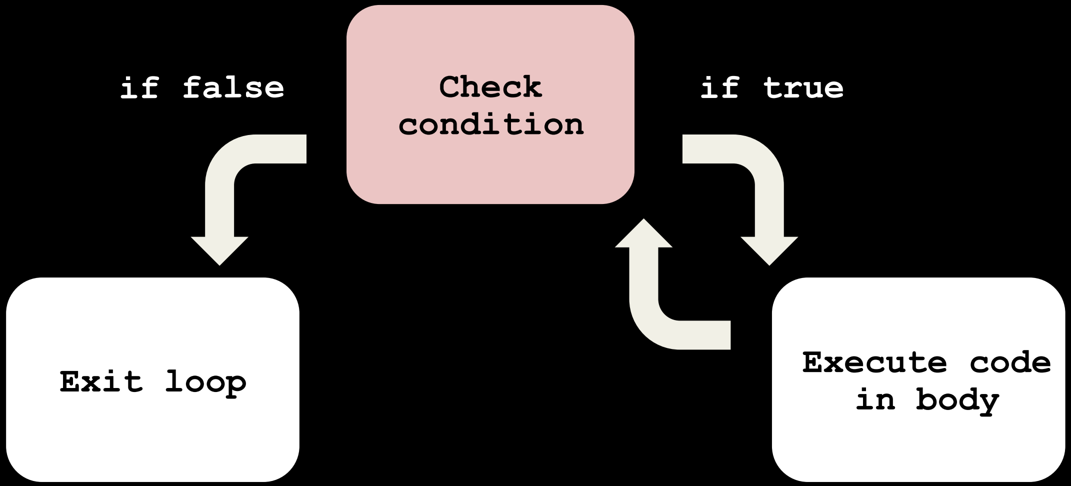
## Example
## Prints “This is CS50!” ten times



```
for (int i = 0; i < 10; i++)
{
    printf("This is CS50!\n");
}
```

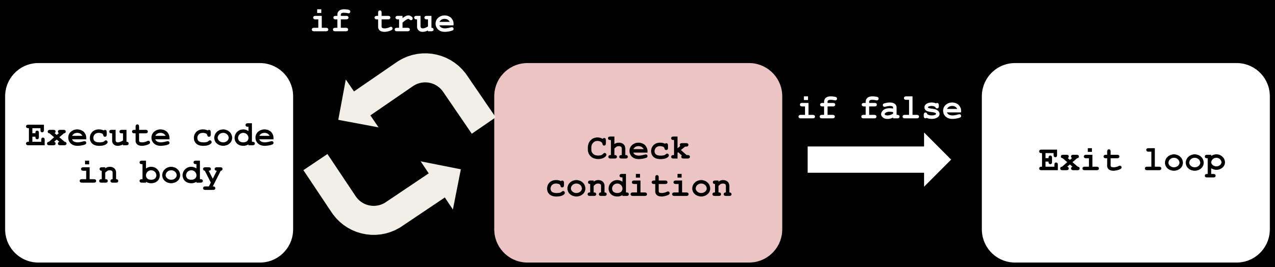
# While Loops

```
while (condition)
{
      execute this code
}
```



# Do While Loops

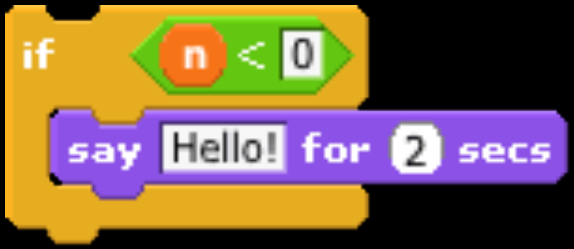
```
do
{
      execute this code
}
while (condition);
```



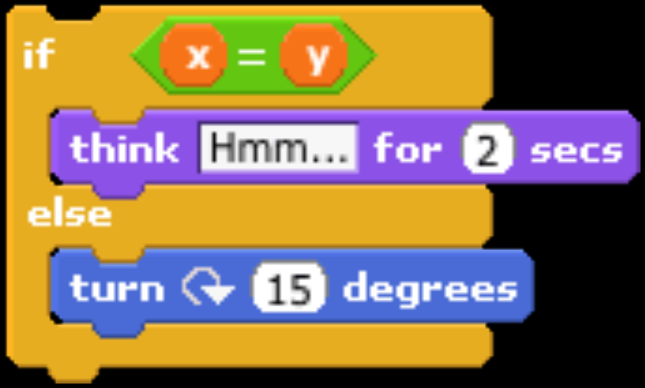
# Example
## Reprompts until user enters a positive number

```
int input;
do
{
    printf("Enter a positive number: ");
    input = GetInt();
}
while (input < 1);
```

# Conditions and Boolean Expressions



```
if <(n) < [0]>
    say [Hello!] for (2) secs
```



```
if <(x) = (y)>
    think [Hmm...] for (2) secs
else
    turn ↻ (15) degrees
```

## If

```
#include <cs50.h>
#include <stdio.h>

int main(void)
{
    printf("Give me an integer: ");
    int n = GetInt();

    if (n > 0)
    {
        printf("You picked a positive number!\n");
    }
}
```

## If... Else

```c
#include <cs50.h>
#include <stdio.h>

int main(void)
{
    printf("Give me an integer: ");
    int n = GetInt();

    if (n > 0)
    {
        printf("You picked a positive number!\n");
    }
    else
    {
        printf("You picked a negative number!\n");
    }
}
```

# If... Else if... Else

```c
#include <cs50.h>
#include <stdio.h>

int main(void)
{
    int n = GetInt();

    if (n > 0)
    {
        printf("You picked a positive number!\n");
    }
    else if (n < 0)
    {
        printf("You picked a negative number!\n");
    }
    else
    {
        printf("You picked 0!\n");
    }
}
```

```c
#include <cs50.h>
#include <stdio.h>

int main(void)
{
    printf("Enter your grade: ");
    int n = GetInt();

    if (n >= 90)
    {
        printf("You got an A!\n");
    }
    if (n >= 80)
    {
        printf("You got a B!\n");
    }
    if (n >= 70)
    {
        printf("You got a C!\n");
    }
    if (n >= 60)
    {
        printf("You got a D!\n");
    }
}
```

## Switch Statements

```c
#include <cs50.h>
#include <stdio.h>

int main(void)
{
    printf("Give me an integer between 1 and 3: ");
    int n = GetInt();

    switch (n)
    {
        case 1:
            printf("You picked a low number.\n");
            break;
        case 2:
            printf("You picked a medium number.\n");
            break;
        case 3:
            printf("You picked a high number.\n");
            break;
        default:
            printf("Invalid.\n");
            break;
    }
}
```

# Ternary Operator

```c
#include <cs50.h>
#include <stdio.h>

int main(void)
{
    printf("Give me an integer: ");
    int n = GetInt();

    string s = (n > 100) ? "high" : "low";

    printf("You picked a %s number!\n", s);
}
```

## But how does all of this get compiled?

## Compilation Process

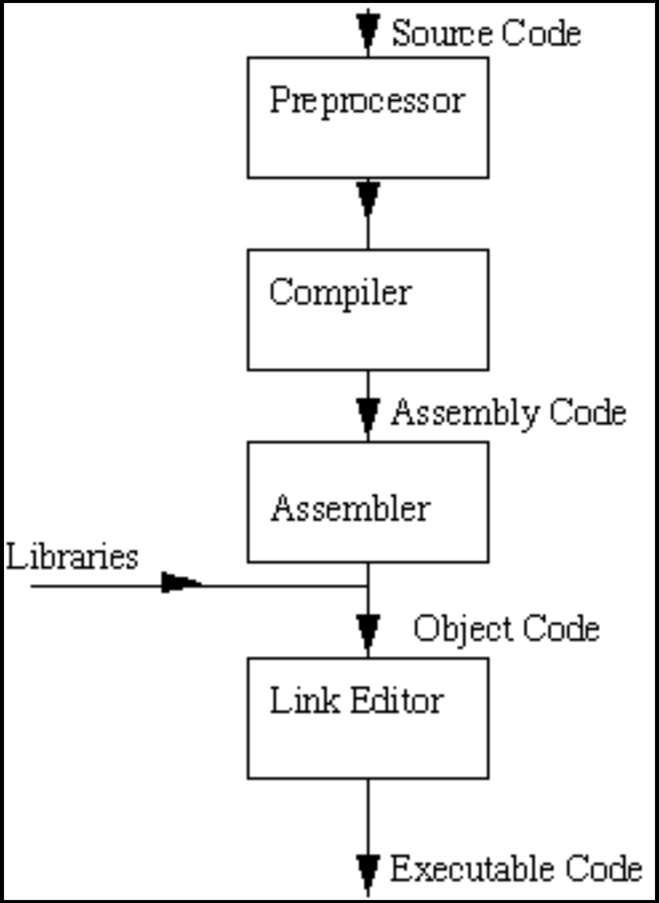
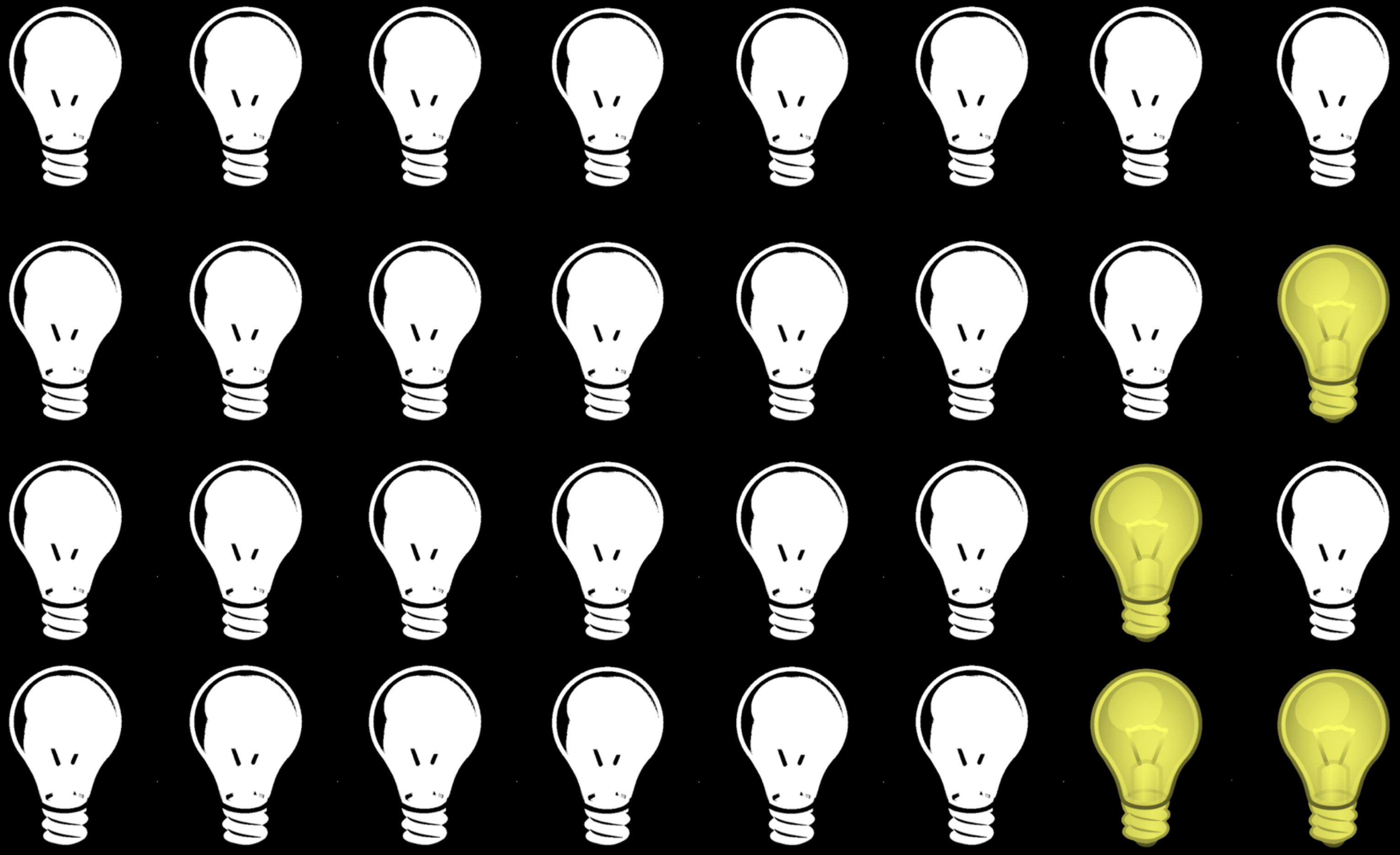


Image from http://www.cs.cf.ac.uk/Dave/C/node3.html



# Addition and Subtraction (Don’t forget to carry your 1s)

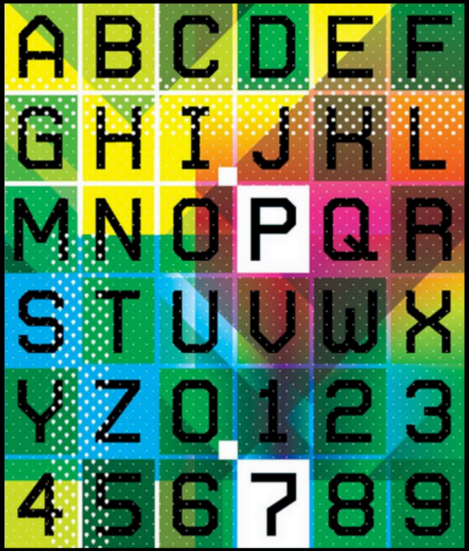
$$\begin{array}{rcccccc}
 & 1 & 0 & 1 & 0^{1} & 1^{1} & 1 \\
+ & 0 & 1 & 0 & 0 & 0 & 1 \\
\hline
 & 1 & 1 & 1 & 1 & 0 & 0
\end{array}$$

$$\begin{array}{rccccc}
 & 1 & 1 & \cancel{1}^{0} & \cancel{0}^{10} & 0 \\
- & 0 & 0 & 0 & 1 & 0 \\
\hline
 & 1 & 1 & 0 & 1 & 0
\end{array}$$

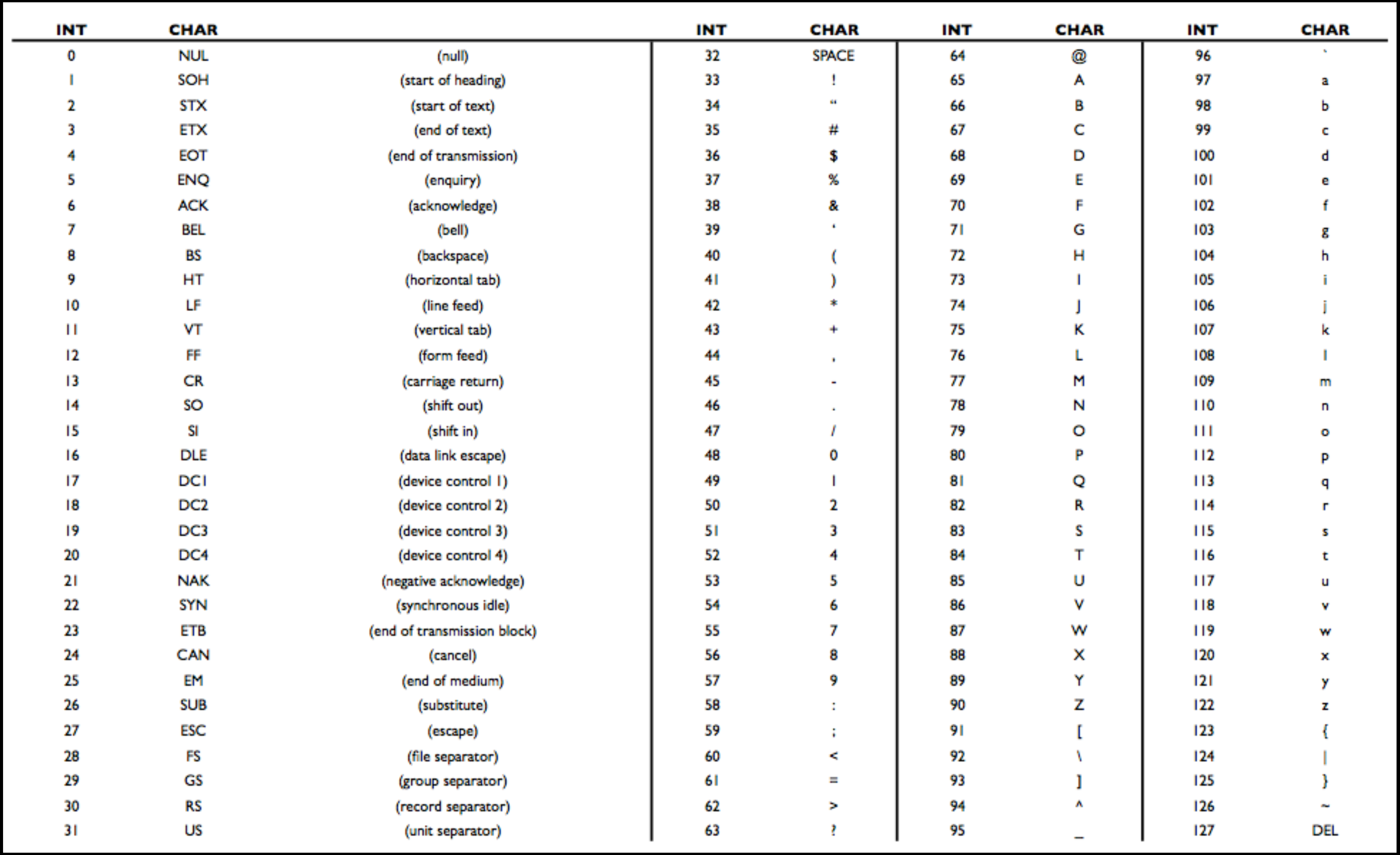
## How can we represent -1?

## two’s complement

## Characters must also be encoded in binary



# ASCII maps characters to numbers



| INT | CHAR | | INT | CHAR | INT | CHAR | INT | CHAR |
|---|---|---|---|---|---|---|---|---|
| 0 | NUL | (null) | 32 | SPACE | 64 | @ | 96 | ` |
| 1 | SOH | (start of heading) | 33 | ! | 65 | A | 97 | a |
| 2 | STX | (start of text) | 34 | " | 66 | B | 98 | b |
| 3 | ETX | (end of text) | 35 | # | 67 | C | 99 | c |
| 4 | EOT | (end of transmission) | 36 | $ | 68 | D | 100 | d |
| 5 | ENQ | (enquiry) | 37 | % | 69 | E | 101 | e |
| 6 | ACK | (acknowledge) | 38 | & | 70 | F | 102 | f |
| 7 | BEL | (bell) | 39 | ' | 71 | G | 103 | g |
| 8 | BS | (backspace) | 40 | ( | 72 | H | 104 | h |
| 9 | HT | (horizontal tab) | 41 | ) | 73 | I | 105 | i |
| 10 | LF | (line feed) | 42 | * | 74 | J | 106 | j |
| 11 | VT | (vertical tab) | 43 | + | 75 | K | 107 | k |
| 12 | FF | (form feed) | 44 | , | 76 | L | 108 | l |
| 13 | CR | (carriage return) | 45 | - | 77 | M | 109 | m |
| 14 | SO | (shift out) | 46 | . | 78 | N | 110 | n |
| 15 | SI | (shift in) | 47 | / | 79 | O | 111 | o |
| 16 | DLE | (data link escape) | 48 | 0 | 80 | P | 112 | p |
| 17 | DC1 | (device control 1) | 49 | 1 | 81 | Q | 113 | q |
| 18 | DC2 | (device control 2) | 50 | 2 | 82 | R | 114 | r |
| 19 | DC3 | (device control 3) | 51 | 3 | 83 | S | 115 | s |
| 20 | DC4 | (device control 4) | 52 | 4 | 84 | T | 116 | t |
| 21 | NAK | (negative acknowledge) | 53 | 5 | 85 | U | 117 | u |
| 22 | SYN | (synchronous idle) | 54 | 6 | 86 | V | 118 | v |
| 23 | ETB | (end of transmission block) | 55 | 7 | 87 | W | 119 | w |
| 24 | CAN | (cancel) | 56 | 8 | 88 | X | 120 | x |
| 25 | EM | (end of medium) | 57 | 9 | 89 | Y | 121 | y |
| 26 | SUB | (substitute) | 58 | : | 90 | Z | 122 | z |
| 27 | ESC | (escape) | 59 | ; | 91 | [ | 123 | { |
| 28 | FS | (file separator) | 60 | < | 92 | \ | 124 | \| |
| 29 | GS | (group separator) | 61 | = | 93 | ] | 125 | } |
| 30 | RS | (record separator) | 62 | > | 94 | ^ | 126 | ~ |
| 31 | US | (unit separator) | 63 | ? | 95 | _ | 127 | DEL |

# ASCII Math

**What will print?**

```
printf("%d\n", 'A' + 1);

printf("%c\n", 65 + ('a' - 'A'));
```

## Numerical Variables

```
int
float
double
long long
```

## Floating Point Imprecision

```
int main(void)
{
    // initialize answer
    float answer = 1.0 / 10.0;

    // print answer to twenty decimal places
    printf("%.20f\n", answer);
}
```

## How are floats stored?

$$1.2345 = \underbrace{12345}_{\text{mantissa}} \times 10^{\overbrace{-4}^{\text{exponent}}}$$

Java™

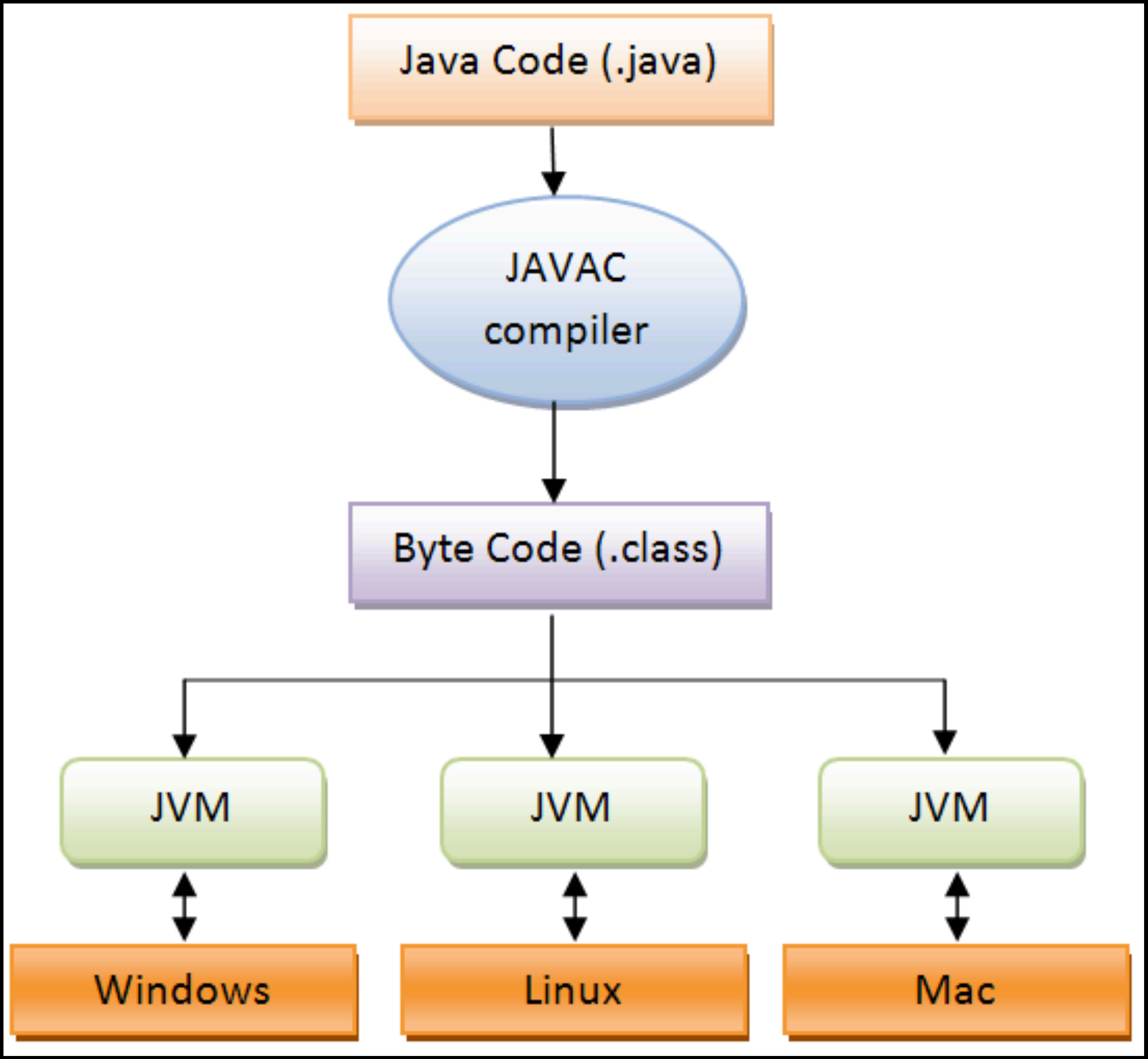


Image from http://javapapers.wordpress.com/2011/11/28/java-virtual-machine-jvm/

# Object-oriented programming

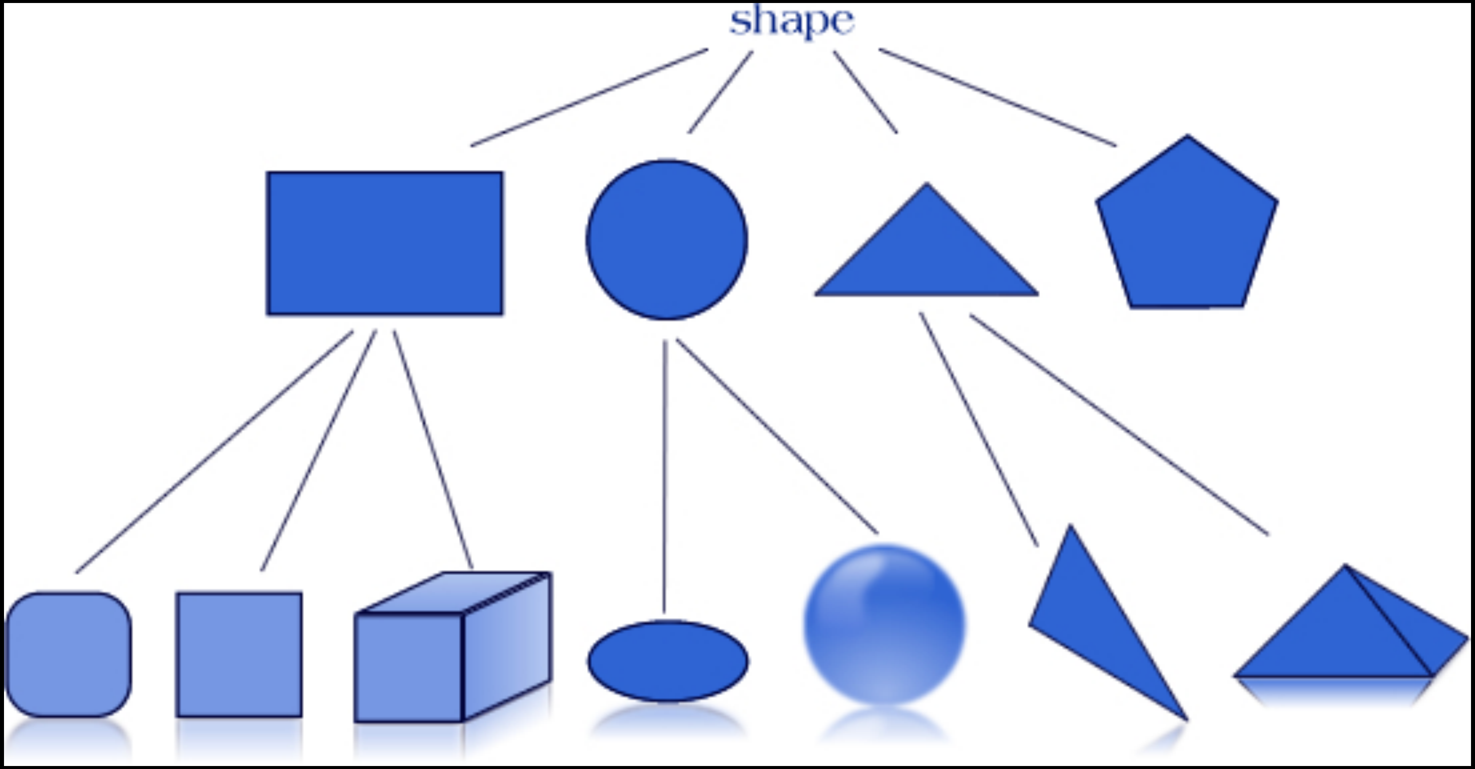


Image from http://pascal-programming.info/articles/oop_part1.php