

Arithmetic  $\leftarrow$  unary  
binary

$++$   $--$   $+$   $-$   $*$   $/$   $\%$

Operator

Bitwise

$\&$   $|$   $\wedge$   $\sim$   $\ll$   $\gg$

Relational

$<$   $>$   $<=$   $>=$   $==$   $!=$

Assignment

$=$   $=$   $=$   $=$   $=$   $=$   $=$   $=$

$+=$   $-=$   $*=$   $/=$   $\%=$

logical operator

$\&\&$   $||$   $!$

Other Operator

$\rightarrow$  size of

$\rightarrow$   $?:$

$\rightarrow$   $\&$   $\rightarrow$  address

$\rightarrow$   $*$   $\rightarrow$  pointer

$\rightarrow$   $[]$   $\rightarrow$  array

$\sim$   $\ll$   $\gg$   
 $\> =$   $==$   $!=$   
 $\sim =$   $\ll =$   $\gg =$   
 $\% :$

### Operations

$\text{int } x = 10;$   
 $\rightarrow x++;$   
 $\text{printf}(\text{"\%d"}, x); \rightarrow 11$

$\text{int } x = 10;$   
 $++x;$   
 $\text{printf}(\text{"\%d"}, x); \rightarrow 11$

$\text{int } x = 10, y;$   
 $y = x++;$   
 $\text{printf}(\text{"\%d"}, x); \rightarrow 11$   
 $\text{printf}(\text{"\%d"}, y); \rightarrow 10$

$\text{int } x = 10, y;$   
 $y = ++x;$   
 $x \rightarrow 11$   
 $y \rightarrow 11$



$+$   $-$   $*$   $/$   $\%$   
 $\sim$   $\ll$   $\gg$   
 $<=$   $>=$   $==$   $!=$   
 $=$   $!=$   $\wedge$   $\sim$   $\ll$   $\gg$   
 $==$   $!=$   $/$   $\%$   
 $||$   $!$

### Operations

$5 \rightarrow 0101$   
 $6 \rightarrow 0110$

586

$4 \rightarrow 0100$   
 $2421$

$10 \gg 2$

$0001010$

$00000010$

$32 \gg 5$

$0000000010000000$

$1 \ll 2$

Number shifts

$20 \gg 2$

$0000010100$

5

$3 \gg 1$

$7 \gg 2$

$11 \gg 5$

$00000001 \ll 2$

$00000100$

$5 \ll 2$

$0000101 \ll 2$

$00010100$

$6 \ll 1$

$7 \ll 6$

$1 \ll 1$

$\gg$   
 Number shifts

$5 \gg 2$      $5 \gg 3$

$000000101$      $0000001010000000$

# Operations

int x=10, y=10;

$x < y \rightarrow F$

$x > y \rightarrow F$

$x <= y \rightarrow T$

$x >= y \rightarrow T$

$x == y \rightarrow T$

$x != y \rightarrow F$

int x=10;  
int y=20;

$(x > y) \&\& (x < y)$   
 $F \quad \quad T$

$(x == 10) \&\& (y > 10)$   
 $T \quad \quad T$

$(x > y) || (x == y)$   
 $F \quad \quad F$



if

↳ Syntax

if ( Condition )  
    {  
        // Code  
    }

// Code

3 → close

if ( Condition )

if ( Condition )

Conditions

if  
else

switch  
case

ternary

?

Global

ex. int x=10;  
if ( X == 100 )  
    printf("Done");

3

x = 100

if (

int

if (x

if



int

Group of Condition in one line

if (

↳ logical operator  
↳ Relational operator

int x=10, y=20, z=30;

if (x > z) {

F

if (x == 10)

T

done

int x=2;  
int y=20;

if (x > y) {

F

else if (x < y) {

T

if (x == 10) {  
Print

else {



int x;

switch(x)

Variable  
Value

Value only

case 11:  
printf("Reem");

break;

case 10:  
printf("IMT");  
break;

char y = 'A';

switch(y)

case 'a':  
printf("a");

break;

case 'b':  
printf("b");  
break;

default:  
printf("ERROR");  
break;

switch  
int x

```
switch ( )  
{  
    case [ ] :  
        // Code  
        break;  
}
```

Decimal value → 10  
A  
10 X  
10 X

Constan value

```
int x;  
int y;  
switch (x) {  
    case 11 :  
        printf("Reem");  
        break;  
    case 10 :  
        printf("IMT");  
        break;  
}
```

Variable  
Value  
Value only

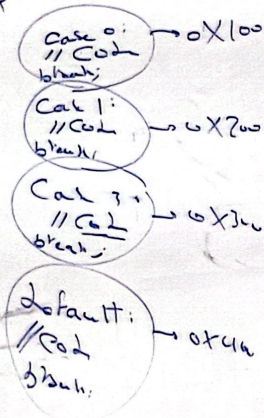
```
char y = 'A';  
switch (y) {  
    case 'a':  
        printf("a");  
        break;  
    case 'b':  
        printf("b");  
        break;  
    default :  
        printf("ERROR");  
        break;  
}
```



X 2

| Case    | Address |
|---------|---------|
| 0       | 0x1a    |
| 1       | 0x2a    |
| 3       | 0x3a    |
| default | 0x4a    |

X  
switch ( X )



Speed

Memory  
Space

if  
else

switch

Condition

?

True

False

int x = 10;  
int y = 20;

$x == y$

?

Print("Doh") ; Print("Not") ;

Ternary



```
int x=6  
int y=10  
>  
<  
==  
!=
```

$x > y$  ? `printf("x > y")` :  $x < y$  ? `printf("x < y")` :  $(x == y)$  ?  
F X F T