# Let’s solve it

11

# Operator precedence & Associativity

$$\begin{tikzcd}
 & 2+3*5 \arrow[dl, dash] \arrow[dr, dash, "2+15"] & \\
5*5 & & 2+15 \\
25 & & 17
\end{tikzcd}$$

* has higher precede than plus. With the rules

computer has to be deterministic
(Finite Automata)

Associativity :

when there are more than one operator from same level meaning having same priority/rank.

At that time computer checks associativity of that level.

i.e.

addition '+' has left to right associativity

L $\boxed{2+3}+9$ R

$\Rightarrow$ $\boxed{5+9}$

14

L $\boxed{2+3}-9$ R

5-9

-4

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

x=y=z;

printf("x is %d", x);
printf("y is %d\n", y);
printf("z is %d\n", z);
```

x=y;    x 20

y 2    y 30

z 30

Bcoz assignment operation level has associativity right to left.

$$\begin{tikzcd} y \arrow[r, phantom, "="] & z \arrow[l, bend left] & ; \end{tikzcd}$$

$z = 30$

$y = 30$

$$\begin{tikzcd} x & y \arrow[l, bend left] \end{tikzcd}$$

$x = 30$

---

x is 30

y is 30

z is 30

Precedence has to with priority which one first to do operation

//PUMAS REBL TAC
//P - Parenthesis
//U - Unary
//M - Multiplication
//Af - Addition
//S - Shift
//R - Relational
//E - Equal/Not equal comparison ==
//B - Bitwise except shift
//L - Logical
//T - Turnary/Conditional
//Al - Assignment
//C - Comma

Associativity has to do with direction

left to right

all others

right to left

Assignment

Unary

Ternary

$\phi$

$2 - 3 + 1$

$\longrightarrow$

$\boxed{-1} + 1$

Unary

~~A~~Addition binary

$(-1) + 1$

$0$

5 & 0

Bitwise $(101)_2 \Rightarrow (5)_{10}$

$$\begin{array}{r} 101 \\ 000 \\ \hline 000 \end{array}$$

5 || 0 $\Rightarrow$ True OR False

Logical OR

$\Downarrow$

True

Can you decide result ahead of time?

Logical AND

True && False && False && True

$\uparrow$

first false indicates overall result is going to be false. And hence, it does not do more.

Logical OR

False || True || False || False || False

if you find very first True, overall will be true. Rest no need to check.

write conditions/expressions in your program accordingly to make your program execute faster.

For &&

write conditions in the beginning which tend to or have more chances of False

For ||

write conditions in the beginning which have more chances of True

Type Casting

Data type conversion

int a;     float f = 30.50;

a = f;   // $\boxed{\text{downsizing}}$

30     Data loss

Because float is more bytes.

The ______ will be truncated.

~~float~~ f;   int a=10;

f=a; //Upcast e;

f = 10.00

10.random ?

garbage ?

Initialisation

~~default~~