1.3 - Conditionals and Loops

Sunday, August 24, 2025 2:44 PM

Precedence and Associativity of operators

→ Mathematical operators

$\underbrace{5 \times 3}/2 + 5$

$15/2 \times 7 + 5 \Rightarrow \underline{12}$

$\underline{\%}$ $\quad \boxed{DM} \quad \boxed{AS}$ ←

$15 \times 2 + \boxed{40 \% 3}/2$

$\downarrow$ 1

$\underbrace{15 \times 2}_{30} + \underbrace{1/2}_{0} \Rightarrow \boxed{30}$ ✓

⇒ Logical Operators

$$\boxed{! > \&\& > ||}$$

Q) $\underbrace{(5 > 3)}_{T} \; \&\& \; ! \underbrace{(4 >= 8)}_{F} \; || \; \underbrace{(15 < 16)}_{T}$

$T \; \&\& \; \boxed{!F} \; || \; T \Rightarrow \underbrace{T \; \&\& \; T}_{T} \; || \; T$

$T \; || \; T = \underline{True}$

Short circuiting

$$\underbrace{(5>3)}_{T} \;||\; \underbrace{(10<=11)}_{T} \Rightarrow F$$

$$\underbrace{(5>3)}_{T} \;||\; \underbrace{(13>20)}_{F}$$

$$\underbrace{T \; \boxed{||}}_{\boxed{T}} \qquad \Rightarrow F$$

$$\rightarrow \quad \underbrace{(10>20)}_{F} \;\&\&\; \underbrace{(13<12)}_{\text{Not evaluated}}$$

$$\overset{\&\&}{\boxed{F}}$$

Conditionals

If-Else

```
if (condition) {
    ////
}
else {
    ////
}
```

If-else ladder

[illegible]

If-else ladder

```
if ( condition1 ){
    ((
}
else if (cond 2) {
  ( ((
}
else if ( cond3 ) {
  }
}
else
```

Nested if-else

```
if ( cond1 ) {
    if (we 2) {
    }
    else
}
else {

}
```

* else cannot exist without if
* if can exist without else

Character input

```
char c = sc.next().charAt(0);
```

ASCII value

$\boxed{'A'} \rightarrow \boxed{65}$

'Z' → 90

'a' → 97

'z' - 122

#Loops

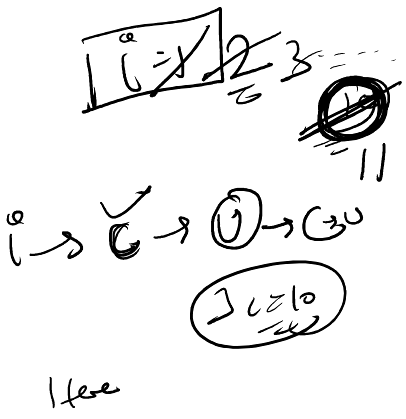
↳ used to repeat a programming statement

→ for
→ while
→ do-while

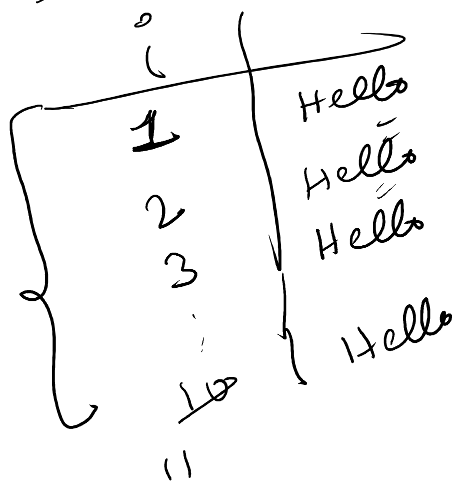
→ for loop

```
for( initialization ; condition ; updation){
        ///
}
```

```
for(int i = 1 ; i<=10 ; i++){
    sout("Hello");
}
```



for (int i = 1 ; i <= 10 ; i = i+1) { body
    sout ("Hello")



→ for (; ;) → infinite loop

→ loop variable can be used inside the loop

⇒ Reverse loop
    for (i = 10 ; i >= 1 ; i--) {

    }

i = 10 ✓
10 >= 1
9 >= 1
1 >= 1
0 >= 1

Sum of N natural nos

$\frac{N(N+1)}{2}$ $\frac{5 \times 6}{2} = 15$

→ Factorial

5! ⇒ 1 × 2 × 3 × 4 × 5
= 120

$1 \times 2 \times \ldots \times N$

while loop

```
initialization;
while ( condition) {
    // 
    updation
}
```

⊛ We generally use for loop when we know exact no. of iterations.
But when we know only condition but not the no. of iteration, then we use while.

⇒ do- while

```
init;
do {
    // logic
    updation
}
while ( condition)
```

⊛ One iteration will definitely happen in do - while no matter what.

(A) One iteration [illegible]
do - while no matter what.