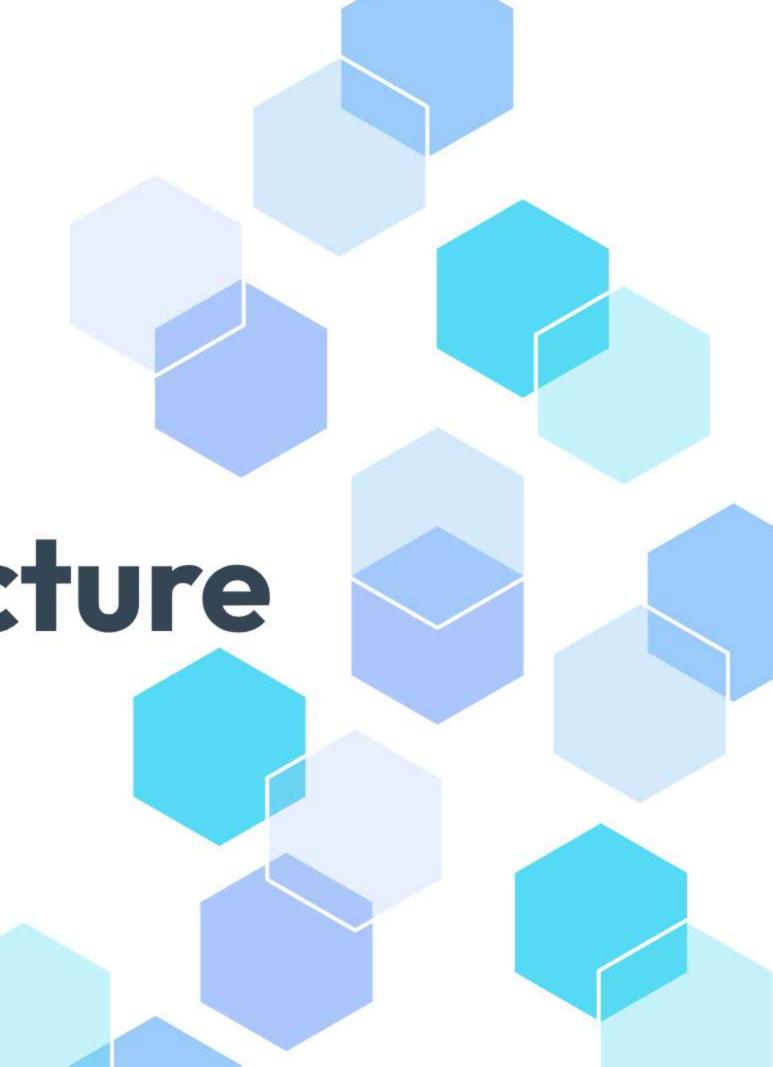
O1 Repetition Structure

Types of Loops





Repetition Structure

(Loops)

Allows the **execution of a block of code repeatedly** as long as a specified **condition is true**.

- 1. Entry Control Loop
 - while loop
 - for loop
- 2. Exit Control Loop
 - do while loop





Repetition Structure

In any type of loops, following three essential parts must have to be included:

Initialization



Condition



Iteration

Increment

Decrement







Initialization

Initialization in a loop refers to the process of setting an initial value to the loop control variable before the loop starts executing.

The <u>loop control variable</u> is a variable that determines the <u>number of iterations</u> or the condition for the loop to continue.

int
$$i = 1$$
;





Condition

It is a loop is a **boolean expression** that **determines** whether the loop should continue or terminate.

It is evaluated before each iteration of the loop.

If the condition is true, the loop body is executed; if it is false, the loop is terminated, and program control moves to the next statement after the loop.





Iteration

Increment and decrement in a loop refer to the operation of increasing or decreasing the loop control variable.

These operations are typically part of the loop control statement and are **executed after each iteration** of the loop.

They are **used to update the loop control variable** and play a crucial role in determining when the loop should terminate.





Iteration (Increment/Decrement)

Increment and decrement can be done using three different approaches:

Normal way

Shorthand syntax (using assignment operators) Using unary operator



Let's see types of loops in detail...



while loop

(Entry Controlled Loop)

```
Initialization
while (Condition)
{
    // Code to be executed if the condition is true
    // Increment / Decrement
}
```





do while loop

(Exit Controlled Loop)

```
Initialization
do
{
    // Code to be executed if the condition is true
    // Increment / Decrement
} while (Condition);
```





for loop

(Entry Controlled Loop)

```
for (Initialization; Condition; Iteration)
{
    // Code to be executed if the condition is true
}
```





TL;DR

Entry Controlled Loop

The condition is checked before the execution of the loop block.

The block of code is guaranteed to run only if the condition is true.

▶ I.e., while loop, and for loop

Exit Controlled Loop

The condition is checked after the execution of the loop block.

The block of code is guaranteed to run at least once.

I.e., do while loop

