

Short-circuit evaluation is a strategy used by many programming languages when evaluating **logical expressions** involving the operators `AND (&&)` and `OR (||)`:

- **For AND (&&):**

The second operand is evaluated **only if** the first operand is `true`.

If the first operand is `false`, the whole expression is immediately `false`, so the second operand is skipped.

- **For OR (||):**

The second operand is evaluated **only if** the first operand is `false`.

If the first operand is `true`, the whole expression is immediately `true`, so the second operand is skipped.

Why it matters:

- **Efficiency:** Avoids unnecessary computation.
- **Safety:** Prevents errors like null-pointer dereferencing or divide-by-zero.
- **Side effects:** Skipped expressions won't execute any side effects (e.g., function calls or increments).

Example:

```
int x = 0;
bool result = (x != 0) && (10 / x > 1);
```

- If `x == 0`, the second part `(10 / x > 1)` is **never evaluated**, preventing division by zero.