

Exercise walk-through

Compound Boolean Expressions ■ 2.5

eXercise

You must be able to

- combine conditions with **AND** (&&) and **OR** (||)
- use **NOT** (!) to reverse a Boolean value
- understand **short-circuit evaluation** 短路求值

Method — Logical operators

- `&&` (**AND**) is true only when **both** sides are true.
- `||` (**OR**) is true when **either** side is true.
- `!` (**NOT**) reverses the value: `!true` is `false`.

Method — Short-circuit evaluation

`&&` stops as soon as the left side is `false`; `||` stops as soon as the left side is `true` — the right side is not even checked.

Practice 2.1 ■ 1 mark

State what the `&&` operator requires in order to be true.

Solution 2.1

Worked steps

both of its sides must be true **B1**.

Practice 2.2 ■ 1 mark

Evaluate $(5 > 3) \parallel (2 > 8)$.

Solution 2.2

Worked steps

true (the left side is true) **B1**.

Practice 2.3 ■ 1 mark

State what short-circuit evaluation means.

Solution 2.3

Method: Short-circuit evaluation

Worked steps

&& and || stop evaluating as soon as the result is known **B1**.

Exam-style 3.1 ■ 1 mark

The logical AND operator in Java is

A &

B &&

C AND

D +

Solution 3.1

A &

B &&



C AND

D +

Worked steps

B.

Exam-style 3.2 ■ 1 mark

`!(true)` evaluates to

A `true`

B `false`

C `1`

D an error

Solution 3.2

A true

B false



C 1

D an error

Worked steps

B.

Exam-style 3.3 ■ 1 mark

Given `int a = 4, b = 0;`:

- (a) evaluate `(a > 0) && (b > 0)`.
- (b) evaluate `(a > 0) || (b > 0)`.
- (c) State what the `!` operator does.

Solution 3.3

Worked steps

(a) false **B1**. (b) true **B1**. (c) it reverses the Boolean value **B1**.