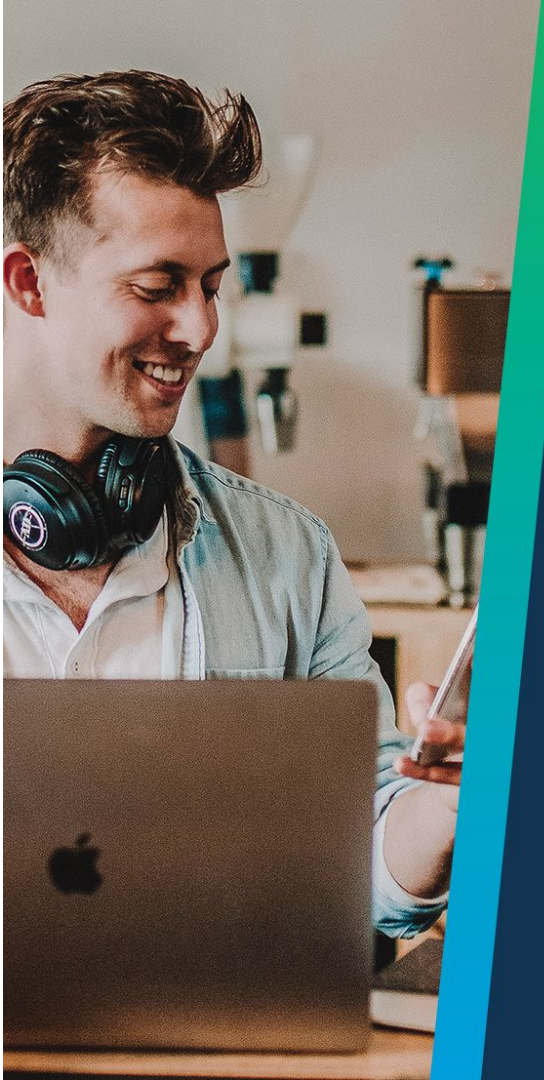

Cyber Security Bootcamp

Hyperiondev

Logical Programming - Operators

Welcome

Your Lecturer for this session

Liano Naidoo

Objectives

- Learn how to tell the compiler how to perform specific mathematical, relational, or logical operations using operators.

Operators

- So far, we have used a few operators, namely:
  - Assignment (=)
  - Equal to (==)
  - Greater than (>)
  - Less than (<)
- Here we will cover more operators available to us and how to utilise them.

Comparison Operators

x = 1 and **y** = 1

OPERATOR	OPERATION	EXAMPLE
== Equal to	True if **x** has the same value as **y**	x == y # True
!= Not equal to	True if **x** does **NOT** have the same value as **y**	x != y # False
>= greater than or equal to	True if **x** is greater than or equal to **y**	x >= y # True
<= Less than or equal to	True if **x** is less than or equal to **y**	x <= y # True

Logical Operators

OPERATOR	OPERATION	EXAMPLE
and	True if both **x AND y** are true (**logical conjunction**)	**If x and y:** **print(z)**
or	True if either **x OR y** are true (**logical disjunction**)	**If x or y:** **print(z)**
not	True if the **opposite** of **x** is true (**logical negation**)	**If not x:** **print(y)**

and Operator

- ★ Returns as True when both conditions specified are met.
- ★ Example:

```
if grade > 50 and grade > 75:
    print("conjunction")
    print("Both Conditions have been met :)")
```

or Operator

- ★ Returns True if either of the specified conditions are met.
- ★ Example:

```
elif grade > 50 or grade > 75:
    print("disjunction")
    print("At least one of these conditions have been met.")
```

not Operator

- ★ Changes the condition from True to False and vice versa.
- ★ Example:

```
if not grade > 50:
    print("Negation")
```

Hyperiondev

Q & A Section

Please use this time to ask any questions relating to the topic, should you have any.

Hyperiondev

Thank You for Joining Us