Hyperiondev

TASK

Logical Programming - Operators

Visit our website

Introduction

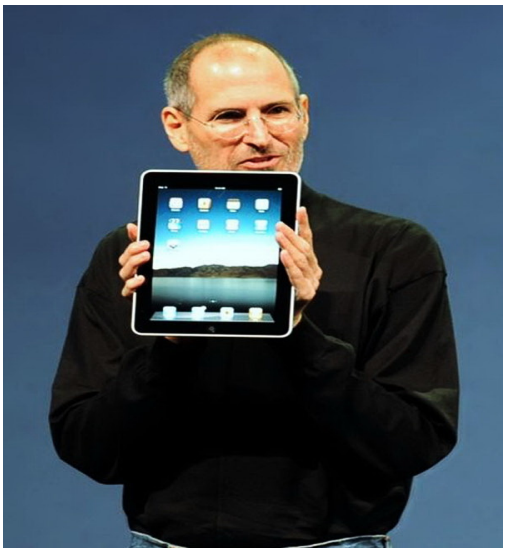
WELCOME TO THE LOGICAL PROGRAMMING - OPERATORS TASK!

In a programming language, an operator is a symbol that tells the compiler or interpreter to perform specific operations (whether it be mathematical, relational, or logical) and produce a final result. This task will introduce you to the different types of operators and show you how to use them.

A note from the
Hyperion Team

The iPad combines many of the popular capabilities of the iPhone, such as built-in high-definition camera, access to the iTunes Store, and audio-video capabilities, but with a bigger screen and without the phone.

Apps, games, and accessories helped spur the popularity of the iPad and led to its adoption in thousands of different applications from movie making, creating art, making music, inventory control, and point-of-sale systems, to name but a few.



WHAT ARE OPERATORS?

Operators are symbols that tell the computer which mathematical calculations to perform or which comparisons to make.

COMPARISON OPERATORS

Here is a quick look at four basic comparative operators:

- greater than >
- less than <
- equal to ==
- not !

We can combine the greater than, less than, and not operator with the equals operator and form three new operations.

- greater than or equal to >=
- less than or equal to <=
- not equal to !=

Comparing strings:

```
my_name = "Tom"
if my_name == "Tom":
    print("I was looking for you")
```

Comparing numbers:

```
num1 = 10
num2 = 20

if num1 >= num2: # The symbol for 'greater than or equal to' is >=
    print("It's not possible that 10 is bigger than or equal to 20.")

elif num1 <= num2: # The symbol for 'less than or equal to' is <=
    print("10 is less than or equal to 20.")

elif num1 != num2: # The symbol for 'not equal to' is !=
    print("This is also true since 10 isn't equal to 20, but the elif
statement before comes first and is true so Python will execute that!")
```

```
elif num1 == num2 : # The symbol for 'equal to' is ==
    print("Will never execute this print statement...")
```

The program will check the first part of the if statement (is *num1* bigger than or equal to *num2*?).

If it is not, then it goes into the first elif statement and checks if *num1* is less than or equal to *num2*. If it is not then it goes into the next elif statement, etc. In the example above, the first elif will execute.

LOGICAL OPERATORS

A *logical operator* is another type of operator that is used to control the flow of a program. Logical operators are usually found as part of a control statement such as an *if statement*. Logical operators basically allow a program to make a decision based on multiple conditions; for example, if you would like to test more than one condition in an *if statement*.

The three logical operations are:

Operator	**Explanation**
and	both conditions need to be true for the whole statement to be true (also called the conjunction operation)
or	at least one condition needs to be true for the whole statement to be true (also called the disjunction operation)
not	the statement is true if the condition is false (only requires one condition)

The 'and' and 'or' operators require two operands, while the 'not' operator requires one.

Let's take a real-life situation. When buying items at a store, two criteria need to be met. The item needs to be in stock and you need to have enough money to pay for the item. This is an example of a *conjunction* operation where both conditions need to be true for the whole statement to be true. This would be represented using the *and* operation.

A person could receive a good mark at school either because they are very bright or because they studied hard. In this instance, either one of the options can be true, or both can be true, but at least one needs to be true. This is a *disjunction* operation where at least one of the conditions needs to be met for the whole statement to be true. This would be represented using the *or* operation.

Example of an **AND** operation:

```
team1_score = 3
team2_score = 2
game = "Over"

if (team1_score > team2_score) and (game == "Over"):
    print("Congratulations you have won the match!")
```

Example of an **OR** operation:

```
speed = int(input("How many kilometres per hour are you travelling at?"))
belt = input("Are you wearing a safety belt?")

if (speed > 80) or (belt != "Yes"):
    print("Sorry Sir, but I have to give you a traffic fine.")
```

ARITHMETIC OPERATORS

The arithmetic operators in Python are as follows:

Arithmetic Operations	Symbol used in Python
Addition: adds values on either side of the operator	**+**
Subtraction: subtracts the value on the right of the operator from the value on the left	**–**
Multiplication: multiplies values on either side of the operator	*****
Division: divides the value on the left of the operator by the value on the right	**/**
Modulus: divides the value on the left of the operator by the value on the right and returns the remainder.	**%**

E.g. *4 % 2 == 0* but *5 % 2 == 1*.	
Exponent: performs exponential calculation, i.e. calculates the answer of the value on the left to the power of the value on the right.	****
Assignment Operations	**Symbol used in Python**
Equals: set the value of the thing on the left to that of the thing on the right. e.g. n = 7	**=**
Plus-equals: adds 1 to the variable for each iteration. e.g. *n += 1* is shorthand for *n = n + 1*. (This is particularly useful when using loops, as you will eventually see.)	**+=**
Minus-equals: minuses 1 from the variable for each iteration. e.g. *n -= 1* is shorthand for *n = n - 1*.	**-=**

Instructions

First, read and run the **example file** provided. Feel free to write and run your own example code before doing the compulsory task to become more comfortable with the concepts covered in this task.

Compulsory Task 1

Follow these steps:

- Create a new Python file in this folder called **award.py**.
- Design a program that determines the award a person competing in a triathlon will receive.
- Your program should read in the times (in minutes) for all three events of a triathlon, namely swimming, cycling, and running, and then **calculate** and **display** the **total** time taken to complete the triathlon.
- The award a participant receives is based on the **total time** taken to complete the triathlon. The qualifying time for awards is 100 minutes. Display the award that the participant will receive based on the following criteria:

Qualifying Criteria	**Time Range**	**Award**
Within the qualifying time.	0 - 100 minutes	Provincial Colours
Within 5 minutes of the qualifying time.	101 - 105 minutes	Provincial Half Colours
Within 10 minutes of the qualifying time.	106 - 110 minutes	Provincial Scroll
More than 10 minutes off from the qualifying time.	111+ minutes	No award

Thing(s) to look out for:

1. Make sure that you have installed and set up all programs correctly. You have set up **Dropbox** correctly if you are reading this, but **Python or your editor** may not be installed correctly.
2. If you are not using Windows, please ask a reviewer for alternative instructions.

Rate us

Share your thoughts

Hyperion strives to provide internationally-excellent course content that helps you achieve your learning outcomes.

Think that the content of this task, or this course as a whole, can be improved, or think we've done a good job?

Click here to share your thoughts anonymously.