**Compound Conditional Statements**

So far we have two places where we use ***conditional*** ***statements***: while loops and if statements. Here we look at *compound conditional statements*.

A ***compound*** ***condition*** involves two or more conditions put together. A compound conditional statement can be created using any of these three ***Boolean*** ***operators***:

***and or not***

Here are some examples:

x = 10

y = 0

if x > 3 and y >3: # x and y are both greater than 3

print ("both true")

else:

print ("not both true")

x = 1

y = 5

if x > 3 or y >3: # either x or y are greater than 3

print ("one or the other")

else:

print ("not one or the other (i.e. neither)")

The **not** statement isn’t used as often. This is the opposite of the above code:

x = 1

y = 2

if not(x > 3 or y >3): # neither x nor y are greater than 3

print ("neither")

**Use of Brackets in Compound Statements**

Suppose you hear this conversation

Should I marry him?

Yes, if he is kind and well-dressed or good-looking

Brackets help clarify exactly what we are intending. A human might interpret the sentence like this:

“If he is kind and (well-dressed or good-looking).”

So he has to be kind and either well-dressed or good-looking (or both).

A computer would interpret it like this:

“If he is (kind and well-dressed) or good-looking.”

In this case, he has to be kind AND well-dressed OR he could be just good-looking. For example, he could be unkind and not well-dressed, but good looking, and he would be marriage material. Not recommended in real life.

If you don’t put any brackets, the computer will look at any AND statements first, followed by OR statements. Like BEDMAS in math, logical operations have an order:

(High Priority) NOT… AND… OR (Low Priority)

For example:

You work on Mondays and Tuesdays or Thursdays. What does this mean to you?

I work on Mondays and (Tues or Thurs)? - I work two days a week.

* OR -

I work on (Mondays and Tuesdays) or Thursdays? - I work two days or I work one day. Without brackets, a computer would use this interpretation and not the first interpretation.

**Exercises**:

6.4 Copy the code below to your IDE and change the values of A B and C to match those given in the table. Write the resulting output in the third column. Make sure you understand why it makes sense.

| A | B | C | Output |
| --- | --- | --- | --- |
| False | True | False |  |
| True | False | False |  |
| False | True | True |  |

A = False

B = False

C = False

if A or B or C:

print ("At least one of either A or B or C is true")

if A and B and C:

print ("A and B and C are all true")

if A and (B or C):

print ("A is true and at least one of B or C is true")

if (A and B) or C:

print ("Either both A and B are true, OR C is true")

if A or (B and C):

print ("Either A is true or B and C are both true")

if A and not (B or C):

print ("A is true and neither B nor C are true")

6.5 Predict the output before trying this out:

a = 5

b = 0

c = 20

if a > 10 and (b > 10 or c > 10):

print ("Hello!")

else:

print ("Goodbye!")

Now try this:

a = 5

b = 0

c = 20

if a > 10 and b > 10 or c > 10:

print ("Hello!")

else:

print ("Goodbye!")

Explain the output in each case.

6.6 Write a program that asks the user their percentage mark and gives their grade according to the simplified chart below.

To keep the marks exclusively for one grade, **use compound if statements (using AND and OR)**.

| Mark | Grade |
| --- | --- |
| [0,50[ | R |
| [50,60[ | 1 |
| [60,70[ | 2 |
| [70,80[ | 3 |
| [80,100] | 4 |
| Any other entry | “Invalid entry” |

Enter your mark: <73>

You earned a grade of: 3

Enter your mark: -900

Invalid entry

Note: The square brackets mean “inclusive” and “exclusive”. For example, the first mark range is [0,50[ which means “between 0 and 50, *including* 0 but *not including* 50”. DO NOT USE THESE SYMBOLS IN YOUR PROGRAM. They will not be recognized by Python.

The last mark range is 80 to 100, including 80 *and* 100.

6.7 Rewrite the Grades program you just wrote above using no compound statements, and only IF-ELIF-ELSE statements.

6.8

The game of blackjack involves two players. The rules (simplified a bit) are:

* each player adds up their card values
* The player with the higher total value wins.
* If either player gets over 21 in total, that player loses (“busts”).

1. In English, write a single sentence that describes how player 1 can win at Blackjack.

*Player One wins if:*

1. Write a Python program. Given two integer variables: **player1value** and **player2value**, write ***a single compound if statement*** that produces the output “player 1 wins!”.

You are only testing player 1. Do not produce any other output. (for example “player 2 wins”).

To make sure it works, **test your program** with the following situations:

| Player1value | Player2value | Output | Explanation |
| --- | --- | --- | --- |
| 18 | 10 | “Player 1 wins!” | P1 had a higher score |
| 10 | 18 | No output | P2 had a higher score |
| 12 | 25 | “player 1 wins!” | P2 busted |
| 25 | 23 | No output | Both players busted |

Keywords: ***conditional statement, compound condition, Boolean operator, and, or, not.***