

Computing

Lesson 4: XOR

Programming Part 4: Subroutines

Ben Garside



Task 1 - Truth tables - AND

Copy the code below into your development environment.

```
1 num1 = 1
2 num2 = 2
3 if num1 == 1 and num2 == 2:
4     print("This is true")
5 else:
6     print("This is false")
```

Modify the code above to help you complete the table.

For example, in order to test the output of the 2nd row you could change the condition to:

```
num1 == 2 and num2 == 2.
```

A num1 == 1	B num2 == 2	A AND B
False	False	False
False	True	
True	False	
True	True	



Task 1 - Truth tables - OR

Copy the code below into your development environment.

```
1 num1 = 1
2 num2 = 2
3 if num1 == 1 or num2 == 2:
4     print("This is true")
5 else:
6     print("This is false")
```

Modify the code above to help you complete the table.

For example, in order to test the output of the 2nd row you could change the condition to:

num1 == 2 and num2 == 3.

A num1 == 1	B num2 == 2	A OR B
False	False	False
False	True	
True	False	
True	True	



Task 2 - Make an XOR function - worked example 1

This is a function for an **AND** operator

```
1 def and_function(a, b):  
2     if a == True and b == True:  
3         c = True  
4     else:  
5         c = False  
6     return c  
7  
8 one = 4 == 4  
9 two = 2 == 2  
10  
11 print(and_function(one, two))
```



Task 2 - Make an XOR function - worked example 1

This is a function for an **OR** operator

```
1 def or_function(a, b):
2     if a == True or b == True:
3         c = True
4     else:
5         c = False
6     return c
7
8 one = 4 == 4
9 two = 2 == 2
10
11 print(or_function(one, two))
```



Task 2 - Make an XOR function - XOR Information

How does an XOR operator work?

- The condition is evaluated as False if both expressions are True.
- The condition is evaluated as False if both expressions are False.
- The condition is evaluated as True if one expression is True.

A expression1	B expression2	A XOR B
False	False	False
False	True	True
True1	False	True
True	True	False



Task 2 - Make an XOR function - activity 1

Design the pseudocode for an XOR function. Use the pseudocode (oaknat.uk/comp-ks4-pseudocode) resource to help you with this.

Tip: Look at the worked examples on slides 5 and 6 to help you with this.

```
1
2
3
4
5
6
7
8
9
10
11
1
2
13
14
15
```



Task 2 - Make an XOR function - activity 2

- Now write the algorithm in program code in your development environment.
- Use the truth table to help you test if your program is working correctly.

A expression1	B expression2	A XOR B	Did it output correctly?
False	False	False	
False	True	True	
True1	False	True	
True	True	False	

