2 Truths and a Lie

Group: Omaima, Fatima, Bismah

With code on the right, which one is the lie?

Statement 1: If x = 3, the code will print 6

Statement 2: If x = 5, the code will print 0

Statement 3: If x = "4", the code will print 8.

```
if x < 5:
    if  x == 2:
      x = x / 2
    else:
      x = x * 2
else:
   if x > 10:
     x == x + 5
   else:
     x = x - x

print (x)
```

1. **Explain how your two truths and a lie meet the criteria from the rubric (3 points)**

For our code, I believe it meets the criteria of considerable for all aspects. This is because, we have clearly demonstrated what is a boolean, by using it in our code. A boolean is a statement in code, which returns the user of a value of true or false. We have shown this in many ways by using multiple if and else statements which can either result in a true or false. For the next part, we have also shown our knowledge on being able to trace which part of the code comes first. For example, let's take x = 3. We can input the x value at the first if outer statement. 3 is smaller than 5, so we go to the inner if statement. It says 3=2 which is false, so we do the else statement then. So, its 3 * 2. = 6, hence why the first statement is true. For the last part of the rubric we have explained that using if statements is very easy and simple to use and understand while still using python symbols/ thinking. Signs such as (* , = , > , < , print , else , if , /) etc. help show our understanding of what is going on in our code. By following the rubric, it is very easy to tell that our code is easily understandable and predictable on what the output will supposedly look like.

2. **Read through your code and create a variety of example test cases such that testing the program with your samples will result in all of the lines of code running.**

TEST CASES; EXAMPLES & EXPLANATION

True or False?	Explanation
If x = 3 , code will print 6	First we look at the outer if statement, which indicates that 3 < 5, which is true. Leads us to the first inner if statement, which says that if x == 2, should be true. But it is not, x = 2. Which leads us to the inner else statement right after. Since it has not been applied yet, we have to use the x value on the else statement. So, 3 * 2 = 6. If x = 3, the will will print 6 —-> **TRUE**
If x = 5, the code will print 0	To test this case out, we need to check the first if statement, since 5 is not smaller than 5, rather that it is equal, we will use the next statement of else, which is the outer side. Now, we use the inner if statement, x > 10. However, 5 is smaller than 10, so that statement is false, leading us to the next inner else statement. Since there are no other statements after, we can use the else, to determine if the statement of x is true. x = x - x. So, 5 - 5 = 0 Therefore, if x = 5, the code will print 0 –--> **TRUE**
If x = "4", the code will print 8	So just by even looking at this statement, we can see that the value of "4 " has quotation marks around it. This indicates that 4 = a string. However, we did not put into our code that strings are allowed. Since python does not allow strings with integers unless called upon, the code will not work make the statement: if x = "4", the code will print 8 —-> **FALSE**

if x < 5:
if x == 2:
x = x / 2
else:
x = x * 2
else:
if x > 10:
x == x + 5
else:
x = x - x
 	
print (x)
	(For the part that is bolded in code):
	
	For this part of the code to work, the first outer and first inner if statements must be true. Which means that by process of elimination the value must be x = 2. So the code will produce 1.

if x < 5:
  if x == 2:
  x = x / 2
  else:
  x = x * 2
else:
  if x > 10:
  x == x + 5
  else:
  x = x - x
 	
print (x)
	(For the part of the code that is bolded):
	
	For thai part of the code to work, the first if statement must be true. A number should be equal to or less than 4 and should not be the number 2. Then, the else statement will take place resulting in a number 1,2,4 being * by 2.

if x < 5:
  if x == 2:
  x = x / 2
  else:
  x = x * 2
else:
  if x > 10:
  x == x + 5
  else:
  x = x - x
 	
print (x)
	(For the part of the code that is bolded):
	
	For this part of the code to work, the x value should not fit the description of the first if statement, so the x value should not be less than 5 or a number between 0-10. When x is greater than 10, the value of x will equal x because the double equal sign indicates that the + 5 is not in the equation which makes it cancel out..

if x < 5:
  if x == 2:
  x = x / 2
  else:
  x = x * 2
else:
  if x > 10:
  x == x + 5
  else:
  x = x - x
 	
print (x)
	(For the part of the code that is bolded):
	
	When the value of x does fit the description of the outer if statement, and is not a number that is greater than 10, so this number can be 5-10. The x value will use the statement of x - x. When this happens, the value will always be equal to 0.

Test Cases; Examples of x in 4 Different Situations

x = 2	By using the highlighted part,
	if x < 5:
	if x == 2:
	x = x / 2
	else:
	x = x * 2
	else:
	if x > 10:
	x == x + 5
	else:
	x = x - x
	
	print (x)
	The code will return 1, because we know that whent he value is 2, the number will be divided by 2, which is 1.

x = 4	By using the highlighted part.
	if x < 5:
	if x == 2:
	x = x / 2
	else:
	x = x * 2
	else:
	if x > 10:
	x == x + 5
	else:
	x = x - x
	
	print (x)
	This code will return 8. Because we know that a number 0,1,3,4, will be multiplied by 2.

x = 18	By using the highlighted part.
	if x < 5:
	if x == 2:
	x = x / 2
	else:
	x = x * 2
	else:
	if x > 10:
	x == x + 5
	else:
	x = x - x
	
	print (x)
	Because of the double equal signs, the code will return the value of x, which is 18.

x = 9	By using the highlighted part.
	if x < 5:
	  if x == 2:
	  x = x / 2
	  else:
	  x = x * 2
	else:
	  if x > 10:
	  x == x + 5
	  else:
	  x = x - x
	
	print (x)
	The code will return 0 because a number that is 5-10 will be used in the last else statement return x - x = 0