if elif else statements (class slides)

CSc 110 if elif else statements

if elif else

While the if condition is required, the elif and else statements are not

```
if conditionA:
    statements
elif conditionB:
    statements
elif conditionC:
    statements
else:
    statements
```

Rewrite the code

Rewrite the code below to use `elif` and `else`

```
def polarity(x):
  if x > 0:
    return "positive"
  if x < 0:
    return "negative"
  if x == 0:
    return "zero"
```

Write a function

Write a function that does the following:

1. Its name is `max_of_two`
2. It takes two numeric arguments
3. It returns the highest value

Test cases

```
print( max_of_two(-1, 3) ) # 3
print( max_of_two(-1, -3) ) # -1
print( max_of_two(5, 5) ) # 5
```

Write a function – solution

```
def max_of_two(x, y):
  if x >= y:
    return x
  else:
    return y

def main():
  print( max_of_two(-1, 3) ) # 3
  print( max_of_two(-1, -3) ) # -1
  print( max_of_two(5, 5) ) # 5

main()
```

```
3
-1
5
```

Submit code for attendance

One submission per group, add your TA's name in a comment

Submit your `max_of_two` function to Gradescope for attendance.

Name your file `max_of_two.py`

Write a function

Write a function that does the following:

1. Its name is `max_of_three`
2. It takes three numeric arguments
3. It returns the highest value

Test cases

```
print( max_of_three(-1, 3, 3) ) # 3
print( max_of_three(-1, -3, 0) ) # 0
print( max_of_three(5, 5, 10) ) # 10
```

Write a function – solution 1

```
def max_of_three(x, y, z):
  if x >= y and x >= z:
    return x
  elif y >= x and y >= z:
    return y
  else:
    return z

def main():
  print( max_of_three(-1, 3, 3) ) # 3
  print( max_of_three(-1, -3, 0) ) # 0
  print( max_of_three(5, 5, 10) ) # 10

main()
```

```
3
0
10
```

Write a function – solution 2

```
def max_of_three(x, y, z):
  max_value = x

  if y >= max_value:
    max_value = y

  if z >= max_value:
    max_value = z

  return max_value

def main():
  print( max_of_three(-1, 3, 3) ) # 3
  print( max_of_three(-1, -3, 0) ) # 0
  print( max_of_three(5, 5, 10) ) # 10

main()
```

```
3
0
10
```

Write a function – solution 3

```
def max_of_two(x, y):
  if x >= y:
    return x
  else:
    return y

def max_of_three(x, y, z):
  max_x_y = max_of_two(x, y)
  return max_of_two(max_x_y, z)

def main():
  print( max_of_three(-1, 3, 3) ) # 3
  print( max_of_three(-1, -3, 0) ) # 0
  print( max_of_three(5, 5, 10) ) # 10
```

```
main()
```

```
3
0
10
```

Write a function

Write a function that does the following:

1. Its name is `average_of_highest`
2. It has three numeric parameters: `x`, `y` and `z`
3. It returns the average of the two highest of the three arguments
4. Test cases:
    1. arguments `1`, `3`, `4` should return `3.5`
    2. arguments `6`, `4`, `2` should return `5.0`
    3. arguments `4`, `2`, `1` should return `3.0`

Write a function - solution

```
def average_of_highest(x, y, z):
  if x >= z and y >= z:
    return (x + y) / 2
  elif y >= x and z >= x:
    return (y + z) / 2
  else:
    return (x + z) / 2

def main():
  print( average_of_highest(1, 3, 5) ) # should print 4.0
  print( average_of_highest(6, 4, 2) ) # should print 5.0
  print( average_of_highest(4, 2, 1) ) # should print 3.0
  print( average_of_highest(2, 2, 1) ) # should print 2.0
  print( average_of_highest(2, 1, 2) ) # should print 2.0
  print( average_of_highest(1, 2, 1) ) # should print 1.5

main()
```

```
4.0
5.0
3.0
2.0
2.0
1.5
```

Write a function

1. Its name is `triangle_type`, and it takes three arguments: `x`, `y` and `z` representing the lengths of the three sides of a triangle
2. It returns:
   - `"Equilateral"` if all three sides are the same
   - `"Isosceles"` if two of the sides are of equal length
   - `"Obtuse"` when all three sides are of different lengths
3. Remember that `equality` is transitive (as long as `x` equals `y`, and `y` equals `z`, then `z` equals `y`) – btw, `greater than` is also a transitive relation

Write a function - solution

```
def triangle_type(x, y, z):
  '''
  This function labels a triangle of side lengths x, y, z into three categories:
    * Equilateral if all three sides are the same length
    * Isosceles if two of the sides are of equal length
    * Obtuse if all three sides are of different lengths
  Args:
    x, y, z: numeric (integer or float) representing tiangle side lengths
  Returns:
    A string: "Equilateral", "Isosceles", or "Obtuse"
  '''
  if x == y and y == z: # if expression is true, then we know x == z
    return "Equilateral"
  elif x == y or y == z or z == x: # are any two sides equal?
    return "Isosceles"
  else: # all three sides are different
    return "Obtuse"

def main():
```

```
  print( triangle_type(3, 3, 3) ) # "Equilateral"
  print( triangle_type(3, 2, 3) ) # "Isosceles"
  print( triangle_type(4, 5, 6) ) # "Obtuse"

main()
```

```
Equilateral
Isosceles
Obtuse
```

Write a function

1. Its name is `triangle_inequality`, and it takes three arguments: `x`, `y` and `z` representing the lengths of the three sides of a triangle
2. For any triangle, the sum of the lengths of any two sides must be greater than the length of the remaining side (learn more about it) – when inequality is violated, the triangle does not exist
3. It returns:
   - `True` if the triangle inequality holds true
   - `False` otherwise

Write a function - solution

```
def triangle_inequality(x, y, z):
  '''
  This function checks for the triangle inequality given three numbers
  representing the side lengths of a triangle
  Args:
    x, y, z: numeric (integer or float) representing tiangle side lengths
  Returns:
    True if inequality is held, False if it is violated
  '''
  if x >= y + z or y >= x + z or z >= y + x:
    return False
  else:
    return True

def main():
  print( triangle_inequality(3, 3, 3) ) # True
```

```
  print( triangle_inequality(3, 2, 3) ) # True
  print( triangle_inequality(4, 5, 6) ) # True
  print( triangle_inequality(1, 2, 3) ) # False

main()
```

```
True
True
True
False
```

Modify `triangle_type`

Modify your `triangle_type` function to check if the triangle exists before checking if the triangle is equilateral, isosceles, or obtuse.

Solution

```
def triangle_inequality(x, y, z):
  if x >= y + z or y >= x + z or z >= y + x:
    return False
  else:
    return True

def triangle_type(x, y, z):
  '''
  This function labels a triangle of side lengths x, y, z into four categories:
    * Non-existing if side lengths violate inequality
    * Equilateral if all three sides are the same length
    * Isosceles if two of the sides are of equal length
    * Obtuse if all three sides are of different lengths
  Args:
    x, y, z: numeric (integer or float) representing tiangle side lengths
  Returns:
    A string: "Equilateral", "Isosceles", "Obtuse",
    or "The triangle does not exist"
  '''
  if triangle_inequality(x, y, z):
    if x == y and y == z: # if expression is true, then we know x == z
```

```
      return "Equilateral"
    elif x == y or y == z or z == x: # are any two sides equal?
      return "Isosceles"
    else: # all three sides are different
      return "Obtuse"
  else:
    return "The triangle does not exist"

def main():
  print( triangle_type(3, 3, 3) ) # "Equilateral"
  print( triangle_type(3, 2, 3) ) # "Isosceles"
  print( triangle_type(4, 5, 6) ) # "Obtuse"
  print( triangle_type(1, 2, 3) ) # "The triangle does not exist"

main()
```

```
Equilateral
Isosceles
Obtuse
The triangle does not exist
```

Organize your code

Split your functions and your code testing into two files: one called `triangles.py` and another called `test_triangles.py` – remember to use `from triangles import *` in your `test_triangles.py` script.