WK03\Assignement03_isSorted_&_bubble_sort.py

```python
x = [100,65,11,7,8,8,8,5,3,18,3,5,1,2,34,76,3,3,98]
y = sorted(x)
z = list(reversed(y))

# class example
# def isSorted(arr):
#     return arr == sorted(arr) or arr == list(reversed(sorted(arr)))

def isSorted(arr: list) -> tuple | bool:
    '''
    isSorted checks to see if the input list/array is in ascending or descending order.

    Parameters
    ----------
    arr : list
        Input array of values

    Returns
    -------
    tuple | bool
        If the input list/array is sorted, returns a tuple with two entries: boolean True and either "ascending" or "descending".\n
        Otherwise returns the boolean, False.
    '''
    for i in range(len(arr)-1):
        if arr[i] > arr[i+1]:
            for j in range(len(arr)-2):
                if arr[j+1] < arr[j+2]:
                    return False
                else:
                    continue
            return True, 'descending'
        else:
            continue
    return True, 'ascending'


def bubble_sort(arr: list, ascending: bool = True) -> list:
    '''
    bubble_sort takes in a list/array and, based on the optional bool parameter,\n
    sorts the array in ascending (default) or descending order.

    Parameters
    ----------
    arr : list
        Input array of values
    ascending : bool, optional
        Sort the list in ascending (True) or descending (False) order, by default True
```

```
48
49      Returns
50      -------
51      list
52          Returns a sorted list/array with respect to the input list/array
53      '''
54
55      if ascending == False:
56          for j in range(len(arr)):
57              for i in range(len(arr)-1, 0+j, -1):
58                  if arr[i] > arr[i-1]:
59                      arr[i], arr[i-1] = arr[i-1], arr[i]
60      else:
61          for j in range(len(arr)):
62              for i in range(len(arr)-1-j):
63                  if arr[i] > arr[i+1]:
64                      arr[i], arr[i+1] = arr[i+1], arr[i]
65      return arr
66
67  print('\nTesting "isSorted" Function\n')
68
69  print('x is not sorted.\t\t\tTesting: is x sorted?', isSorted(x))
70  print('y is sorted in ascending order.\t\tTesting: is y sorted?', isSorted(y))
71  print('z is sorted in descending order.\tTesting: is z sorted?', isSorted(z))
72
73  print("\n-----------------------------------------------------\n")
74
75  print('Testing "bubble_sort" Function\n')
76
77  print("Unsorted starting array:\n", x)
78  print("\nAfter bubble_sort, ascending:\n", bubble_sort(x, True))
79  print("Compared to Python sorting:\n", y)
80
81  print("\nAfter bubble_sort, descending:\n", bubble_sort(x, False))
82  print("Compared to Python reversed array:\n", z)
83  print()
```

```
PS G:\My Drive\School\01_Fall2024\CS210\WK03> py
'.\Assignement03_isSorted_&_bubble_sort.py'

Testing "isSorted" Function

x is not sorted.                        Testing: is x sorted? False
y is sorted in ascending order.         Testing: is y sorted? (True, 'ascending')
z is sorted in descending order.        Testing: is z sorted? (True, 'descending')

-----------------------------------------------------

Testing "bubble_sort" Function

Unsorted starting array:
 [100, 65, 11, 7, 8, 8, 8, 5, 3, 18, 3, 5, 1, 2, 34, 76, 3, 3, 98]

After bubble_sort, ascending:
 [1, 2, 3, 3, 3, 3, 5, 5, 7, 8, 8, 8, 11, 18, 34, 65, 76, 98, 100]
Compared to Python sorting:
 [1, 2, 3, 3, 3, 3, 5, 5, 7, 8, 8, 8, 11, 18, 34, 65, 76, 98, 100]

After bubble_sort, descending:
 [100, 98, 76, 65, 34, 18, 11, 8, 8, 8, 7, 5, 5, 3, 3, 3, 3, 2, 1]
Compared to Python reversed array:
 [100, 98, 76, 65, 34, 18, 11, 8, 8, 8, 7, 5, 5, 3, 3, 3, 3, 2, 1]

PS G:\My Drive\School\01_Fall2024\CS210\WK03>
```