Assignment on test-driven development using Python and the Circular Queue data structure:

Implement a Priority Queue Class using Test-Driven Development

Assignment: Individual work.

You have been assigned the task of implementing a Priority Queue class in Python using the principles of test-driven development (TDD).

The Priority Queue class should have the following functionalities:

1. `enqueue(item, priority)`: Adds an item to the priority queue with a specified priority.

2. `dequeue()`: Removes and returns the item with the highest priority from the priority queue. If the queue is empty, raise an exception.

3. `peek()`: Returns the item with the highest priority from the priority queue without removing it.
If the queue is empty, **raise an exception**.

4. `is_empty()`: Returns True if the priority queue is empty; otherwise, returns False.

5. 'is_full()' Returns True if the priority queue is full; otherwise, returns False.

6. `size()`: Returns the number of items in the priority queue.

7 change_priority(item, new_priority): Modifies the priority of a specific item in the priority queue. If the item is not found, raise an exception.

8 remove(item): Removes a specific item from the priority queue. If the item is not found, raise an exception.

9 clear(): Clears all items from the priority queue.

The priority queue should prioritize items based on their priority value.

Items with a higher priority value should be dequeued before items with a lower priority value.

If multiple items have the same priority, they should be dequeued in the order they were enqueued.

Your task is to write test cases for each of these methods and then implement the Priority Queue class to pass those tests.

You should follow the TDD process:

1. Write a failing test case.

2. Implement the minimum code required to make the test pass.

3. Refactor the code if necessary.

4. Repeat the process for the next test case.

Write a Python file `priority_queue_test.py` to create and run your test cases.

Here's an **example** structure for your test file:

```python
import unittest
# Import the PriorityQueue class here
class PriorityQueueTestCase(unittest.TestCase):
    def test_enqueue(self):
        # Write test cases for the enqueue() method here
    def test_dequeue(self):
        # Write test cases for the dequeue() method here
    def test_peek(self):
        # Write test cases for the peek() method here
    def test_is_empty(self):
        # Write test cases for the is_empty() method here
    def test_size(self):
        # Write test cases for the size() method here
if __name__ == '__main__':
    unittest.main()
```

Your task is to fill in the test cases for each method in the `PriorityQueueTestCase` class.

Be sure to cover different scenarios and edge cases, such as enqueuing items with different priorities, dequeuing items based on their priority, checking the queue size, and handling exceptions.

Once you have completed the test cases, implement the Priority Queue class in a separate file called `priority_queue.py`. Run the test file to verify that your Priority Queue class passes all the tests.

Remember to use the TDD process and ensure that each test fails before implementing the corresponding functionality.

Any Copy or using ChatGPT => -5

GOOD LUCK

Deadline

20/5/2023 – Saturday & Thursday Group

22/5/2023 – Monday Group