Task 3.2P Answer Sheet

Name: Duong Quang Thanh
Student ID: 104167828

1. In 2.2P, how many Counter objects were created?

   There are 2 Counter objects were created.

2. Variables declared without the “new” keyword are different to the objects created when we call “new”. Referring to the main method in task 2.2P, what is the relationship between the variables initialised with and without the “new” keyword?

   When we declare a variable without utilizing the "new" keyword, we're making a reference variable capable of holding the memory address of an object. This variable doesn't generate a new object in memory; instead, it points to an existing one.

   On the other hand, when we employ the "new" keyword, we're actually crafting a fresh object in memory. This keyword not only sets up the object's instance variables but also reserves memory for it. The variable is then given the memory location of this recently formed object.

3. In 2.2P, explain why resetting the counter in myCounters[2] also changed the value of the counter in myCounters[0].

   This is because myCounters[0] and myCounters[2] point to the same object.

4. The key difference between memory on the heap and memory on the stack is that the heap holds “dynamically allocated memory”. What does this mean? In your answer, focus on the size and lifetime of the allocations.

   The term "dynamically allocated memory" in the context of the heap means that the program can request memory at runtime and determine its size dynamically.

   Heap Memory:
   - Size: scalable, you can request a different amount of memory as needed.
   - Lifetime: It can last beyond the function that created it.

   Stack Memory:
   - Size: The size is typically fixed and known at compile time.
   - Lifetime: Memory on the stack is short-lived and tied to the scope of the function.

5. Are objects allocated on the heap or the stack? What about local variables?

   Objects are allocated on the Heap and Local variables are allocated on the Stack

6. What does the new() method do when called for a particular class, and what does it return?

   The new keyword is used to create a new instance of a class. When using the new

keyword with a class, it allocates memory for a new object of that class and returns a reference to that object

7. Assuming the class Counter exists in my project, if I wrote the code “Counter myCounter;” (note there is no “=”), what value would myCounter have? Why?

When we declare a variable like "Counter myCounter;" without giving it a specific value, it gets a default value based on its type, which is null. So, in this scenario, myCounter would be null because it's a reference to a Counter class instance that hasn't been assigned a real instance yet. This implies that myCounter doesn't currently refer to any valid object in the computer's memory.

8. Based on the code you wrote in task 2.2P, draw a diagram showing the locations of the variables and objects in main and their relationships to one another.