Task 3.2P Answer Sheet

Name: Le Gia Hoang An
Student ID: 104789808

1. In 2.2P, how many Counter objects were created?
   There were 3 Counter objects created and nested within an array.

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?
   Both types can store data, but they are still different from each other. When storing data into a variable initialised without the "new" keyword, it will act like a normal variable for only storing one data type. On the other hand, a variable initialised with the "new" keyword will become an object that can store various data types, depending on the Class's constructor that the object originated from.

3. In 2.2P, explain why resetting the counter in myCounters[2] also changed the value of the counter in myCounters[0].
   myCounters[2] also changed after the reset of myCounters[0]'s constructor parameter, counter, because myCounters[2] share the same name value as myCounters[0]. This led to both of the objects registered with the same data set and acknowledged by the program. Upon new changes to the data set, both objects will be affected

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.
   Dynamically allocated memory is when the data is assigned to memory space during the program's run time. This helps with better resource management and flexibility with data size. For example, features such as List and its functions can help with dynamically adjusting the size of a data collection, which can also directly impact the size of allocated memory space. The life span of these allocations will last until the data has been removed from that memory space.

5. Are objects allocated on the heap or the stack? What about local variables?
   Objects are allocated on the heap while the 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() method will initiate an object or class instances. This will be used as a reference back to the class that is being called and from there, class functions and properties can be accessed.

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?

myCounter would simply be an empty object or an object that contains a value of null. This is because there was no data assigned to that object.

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.

Stack	Heap

Stack

PrintCounters
Counter[]
counters,
Counter c

Main
Counter[]
myCounters, int
i, Increment(),
Reset(),
PrintCounters()

Heap

myCounters[0]
_name =
"Counter 1",
_count = 0

myCounters[1]
_name =
"Counter 2",
count = 0

myCounters[2]
_name =
"Counter 3",
_count = 0

Class Counter
string _name,
int _count,
string Name
<<property>>,
int Ticks <<readonly
property>>