# 3.2P: Answer Sheet

Recall task *2.2P Counter Class* and answer the following questions.

1. How many *Counter* objects were created?

2 Counter objects were created.

2. Variables declared without the ***new*** keyword are different to the objects created using ***new***. In the ***Main*** function, what is the relationship between the variables initialized with and without the ***new*** keyword?

In C#, when objects are created using the new keyword, memory is being allocated on the heap for those objects and obtaining a reference to that memory location. When one variable is assigned to another without using the new keyword, the reference is copied which means both variables point to the same object in memory.

3. In the ***Main*** function, explain why the statement ***myCounters***[2].***Reset***(); also changes the value of ***myCounters***[0].

myCounters[2].Reset() changes the value of myCounters[0] because both myCounters[2] and myCounters[0] are references to the same Counter object in memory (myCounters[2] = myCounters[0]).

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

"Dynamically allocated memory" refers to memory that is allocated at runtime
Stack: Used for static memory allocation with fixed size, temporary lifetime and automatic memory management.
Heap: Used for dynamic memory allocation where memory size can be determined at runtime, potentially longer lifetime and requiring explicit management.

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

Objects are allocated on the heap and local variables are allocated on the stack.

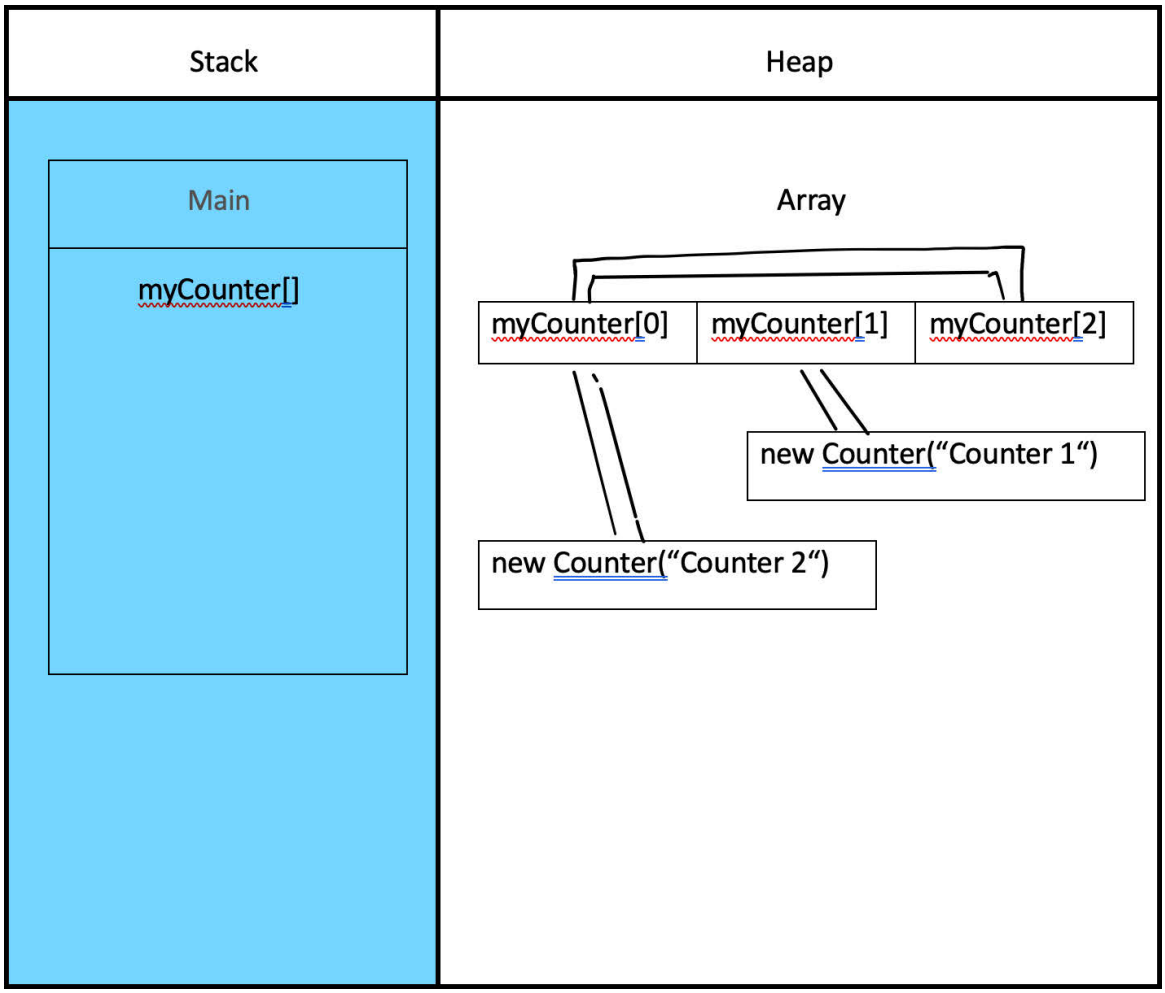
6. What is the meaning of the expression ***new*** *ClassName*(), where *ClassName* refers a class in your application? What is the value of this expression?

new ClassName() means creating a new variables that contains identical reponsilities and requirements compares to the original ClassName() class. new ClassName() is used to assigned to a variable.

7. Consider the statement "*Counter* ***myCounter***;". What is the value of ***myCounter*** after this statement? Why?

The value of Counter myCounter is null because there is no reference being assigned to it.

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




9. **If the variable myCounters is assigned to null, then you want to change the value of myCounters[X], where X is the last digit of your student ID, what will happen? Please provide your observation with screenshots and explaination.**

In the screenshot, `myCounters` is set to null, so it doesn't reference an object. Attempting to modify `myCounters[0]` triggers a "NullReferenceException" because there's no accessible array. Null references can cause runtime errors and affect program behavior.