Answers to Questions from P1.2

Name: Ujjwol Basnet
Student ID: 103056475

How many Counter objects were created?

A total of 2 objects

Variables declared in main() are different to the objects created when we call new. What is the relationship between the declared variables in main and the objects created?

Variables created when we call new are references to objects.

Resetting the counter in myCounters[2] also changes the value of the counter in myCounters[0]. Why does this happen?

myCounter[2] and myCounter[0] [are references to the same object created]

The key difference between memory on the heap compared to the stack and the heap is that the heap holds dynamically allocated memory. What does this mean ?

Dynamic memory allocation in the heap means that memory is allocated without a set order.
Stack has a proper order whereby its memory is accessed in the order it was added in.

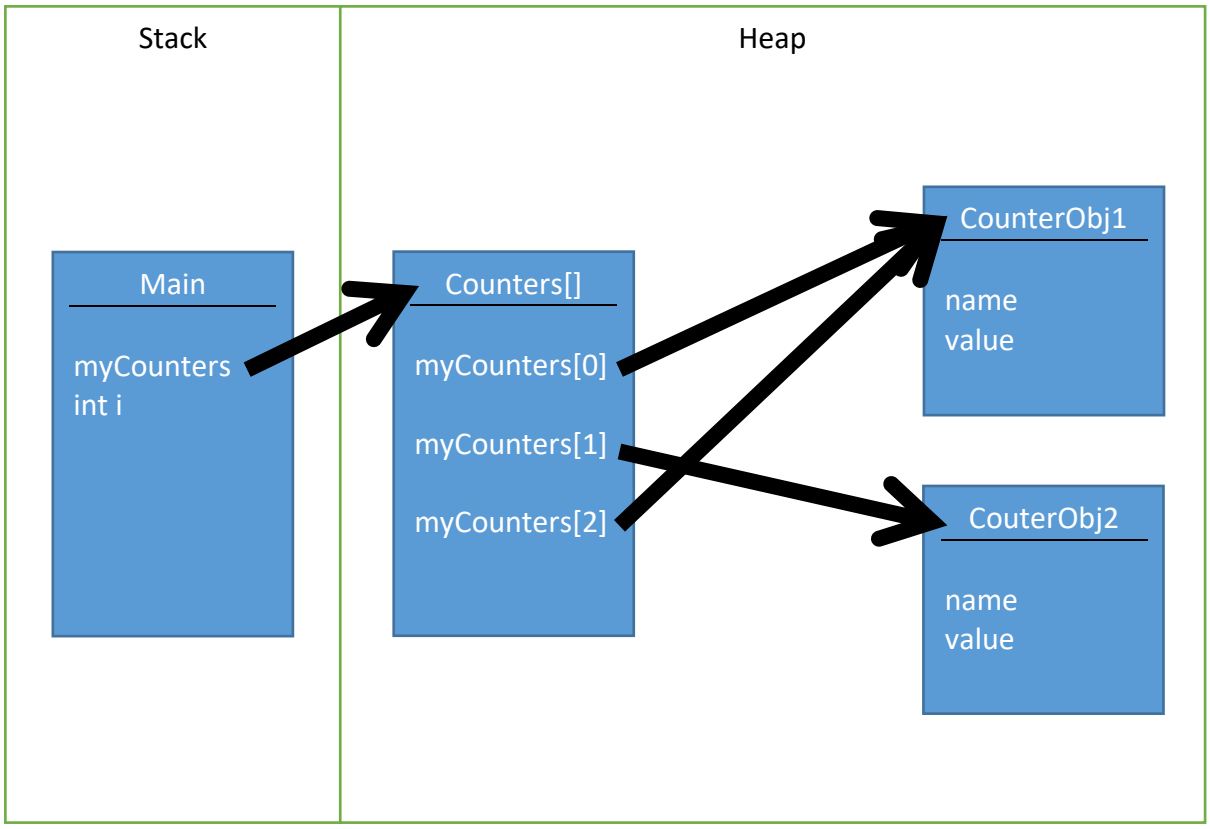
On which are objects allocated (heap or stack) ? On which are local variables allocated (heap or stack) ?

Objects are allocated on the Heap.
Local variables are allocated on the Stack which can contain references to objects in the heap.

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

When new is called on a class it *allocates memory for that object in the heap and creates it* then it returns *a reference to that object's memory location.*

Draw a diagram showing the locations of the variables and objects in main.




```
/*
 static is a method that belongs to the class itself
 So you don't have to create a object of this program to call it
 you can just call it. For example you can't just call c.Name without
first creating an object of Counter first, but if it was static you
could (also has to be public though to use it from this file).
*/

// variables will always be a stack that may point to class objects
which are stored in the heap, however structs are stored in stacks.
So for above example if you change the class file counters with the word
struct instead of class, there are three objects created in main and
myCounter[2] contains reference to its own object.
```