

SWINBURNE UNIVERSITY OF TECHNOLOGY

COS20007 OBJECT ORIENTED PROGRAMMING

The Stack and Heap

PDF generated at 10:51 on Sunday 20th August, 2023

Task 3.2P Answer Sheet

Name:

Student ID:

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

Two Counter objects were created: one through `myCounters[0]` and another through `myCounters[1]`. (`myCounters[2]` is a reference to the same object created using `myCounters[0]`).

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?

Variables declared with the "new" keyword create new instances on the heap. For instance, `myCounters[0]` and `myCounters[1]` were initialized with the "new" keyword, which created separate these Counter objects.

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[2]` was not initialized with "new" (assigned to `myCounters[0]`), which make `myCounters[2]` points to the same object as `myCounters[0]`.

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.

Memory on the heap means that memory is allocated at runtime based on the needs of the program. The memory allocated on the heap has a longer lifetime and persists beyond the scope of the function or block that allocated it.

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

Objects are allocated on the heap. Local variables are usually 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 is used to create an instance of a class. It invokes the class's constructor to initialize the object and returns a reference to the (newly) created object. This allows memory allocation for an object on the heap and initialize its members.

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?

If you wrote '`Counter myCounter;`' without initializing it with an instance using the "new" keyword, '`myCounter`' would have a default value of null because it's a reference type. It can't refer to any actual instance of the Counter class unless you explicitly use i.e. `myCounter = new Counter();`.

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.

