C++ Onderzoek

Inhoud

# Hoofdvraag

## Deelvraag 1

**Wat zijn stack en heap memory allocation en wat is het verschil?**

In programming, stack and heap are two areas of memory used to store data. The main difference between them is how they allocate and deallocate memory.

Stack memory is a contiguous block of memory used for storing function call frames, local variables, and function arguments. The stack is an automatically managed memory region where data is stored in a last-in-first-out (LIFO) order. This means that the most recently added item to the stack is the first one to be removed. Memory allocation and deallocation in the stack are handled automatically by the compiler, and the size of the stack is predetermined at the start of the program.

Heap memory, on the other hand, is a larger, more flexible pool of memory used for storing data that needs to be dynamically allocated and deallocated at runtime. Heap memory is allocated using dynamic memory allocation functions, such as malloc() or new, and deallocated using free() or delete. Unlike stack memory, heap memory is not automatically managed by the compiler, and it is the programmer's responsibility to manage the allocation and deallocation of memory in the heap.

In summary, stack memory is a limited, automatically managed memory region used for storing function call frames, local variables, and function arguments, while heap memory is a larger, dynamically allocated memory region used for storing data that requires flexible allocation and deallocation.

## Deelvraag 2

**Wat zijn shallow en deep copies en wat is het verschil?**

In C++, when an object is copied, a copy constructor is invoked to create a new object with the same values as the original object. There are two types of copying methods in C++: shallow copy and deep copy.

Shallow copy creates a new object, but it copies only the address of the memory location of the original object's data members. As a result, the new object and the original object share the same memory location. This means that if any changes are made to one object's data members, the other object's data members are also affected. Shallow copying is the default copying mechanism in C++.

Deep copy creates a new object and copies all the data members of the original object into the new object. This means that the new object has its own separate memory location and changes made to one object's data members do not affect the other object's data members. Deep copying requires that a copy constructor be defined for the class.

For example, let's say we have a class Person with a name attribute and a pointer to a Car object. When a shallow copy is made, the new object will have a new memory location but its pointer to the Car object will point to the same memory location as the original object's pointer. When a deep copy is made, the new object will have a new memory location and a new Car object with its own memory location will be created and the new object's pointer will point to that new memory location.

In summary, the main difference between shallow copy and deep copy is that shallow copy copies only the address of the memory location of the original object's data members, while deep copy copies the values of the original object's data members into a new memory location.

## Deelvraag 3

**Wat is een assignment operator en wat is het verschil tussen hem en een copy constructor?**

The copy constructor is used to create a new object as a copy of an existing object. It is called automatically when an object is initialized with another object of the same type. The copy constructor takes a const reference to an object of the same class, and creates a new object with the same values as the original object. The copy constructor is only called once during the lifetime of an object.

On the other hand, the assignment operator is used to assign the value of one object to another object of the same class. It is called when an object is assigned a new value. The assignment operator takes a reference to an object of the same class, and copies the values of its member variables into the current object. The assignment operator can be called multiple times during the lifetime of an object.

In summary, the copy constructor creates a new object with the same values as the original object, while the assignment operator copies the values of an existing object into another object of the same class.

Voorbeeld:

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Afbeelding met tekst

Automatisch gegenereerde beschrijving

# Bronnenlijst

<https://www.geeksforgeeks.org/stack-vs-heap-memory-allocation/>

<https://www.mylearningmania.com/2021/02/deep-and-shallow-copy-in-cpp.html>

<https://chat.openai.com>