

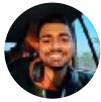


Get unlimited access

Open in app



Published in Dev Genius



Sanket Manmode

Follow

Aug 1, 2020 · 3 min read · Listen



Save



Memory Allocation in C++

Static Memory Allocation and Dynamic Memory Allocation in C++.



Photo by [Anna Dyakova](#) on [Unsplash](#)

Each variable uses space on computer's memory to store its value.

When we use the term **allocate**, we indicate that the variable has been given a space





Get unlimited access

Open in app

This article explains how this memory is allocated to variables in C++ programming language.

Memory is divided into two parts. First is called **Stack memory** and other is **Heap memory**.

Stack memory store variables declared inside function call and is generally smaller than heap memory.

Heap memory is used in dynamic memory allocation and is generally larger than stack memory.

Memory allocation in C++ is done by two methods.

One of them is **Static Memory Allocation** which is also called as Compile Time Allocation.

And the other one is called as **Dynamic Memory Allocation** which is also know as Run Time Allocation.

Static Memory Allocation

In static memory allocation, size and location where variable will be stored is fixed during compile time.

Let us see how static memory allocation takes place using following code.





Get unlimited access

Open in app

```
6     char ch;  
7     int a[100];  
8     return 0;  
9 }
```

static.cpp hosted with ❤ by GitHub

[view raw](#)

In the above code there are three variables, **x** is a integer variable, **ch** is a character variable and **a** is a integer array.

In static memory allocation, compiler calculates how much memory these variables will need and fix a location where these variables will be stored.

Imagine a symbol table like following,

a — -> 0

ch ---> 401

x — ->402

Using table like this, locations of where variables will be stored is saved. But, actual physical memory is **not** allocated to the variable at compile time.

Actual physical memory is allocated only at runtime.

Some other important point about static memory allocation are as follows:

1. Static memory allocation is **slightly faster than dynamic memory allocation**.
2. This type of memory allocation is **less flexible**.
3. Allocation and deallocation of memory is done by compiler itself.

Dynamic Memory Allocation





Get unlimited access

Open in app

memory requirement during execution of the program.

Dynamic memory allocation uses special type of memory called as **heap memory**. Do not confuse this with heap data structure.

In dynamic memory allocation, **new** keyword is used to **allocate** memory and **delete** keyword is used to **deallocate** memory.

Unlike static memory allocation, allocation and deallocation of memory should be done by the programmer. That means programmer need to write code for allocation and deallocation the memory.

The *new* keyword returns the address that has been allocated to the variable on the *heap memory*. This happens during execution of the program.

Let us see how memory is allocated dynamically during runtime using following code.

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int n;
6      cin >> n;
7      //allocate memory
8      int *arr = new int[n];
9      cout << sizeof(arr) << endl;
10     for(int i = 0; i < n; i++)
11     {
12         cin >> arr[i];
13     }
14     //Free up space
15     delete []arr;
16     return 0;
17 }
```

dynamic.cpp hosted with ❤ by GitHub

[view raw](#)



Get unlimited access

Open in app

memory on the heap.

Please read the above statement carefully to understand the working of the program.

***arr** points to the address where array is stored. The address in question here is the one returned by **new** keyword.*

Some other important points to notice about dynamic memory allocation are as follows:



84



1. The advantage of dynamic memory allocation is that *reuse* of memory is possible.
2. Memory leak is possible. Programmer needs to be careful while re-assigning memory to another variable.
3. Compiler does not help with allocation and deallocation. Programmer needs to both allocate and deallocate the memory.

This was memory allocation in C++. Thank you for reading and I hope you had a good read. Have a nice one!

Thanks to Yenson Lau and Domenico Nicoli

Enjoy the read? Reward the writer. ^{Beta}

Your tip will go to Sanket Manmode through a third-party platform of their choice, letting them know you appreciate



[Get unlimited access](#)[Open in app](#)

Sign up for DevGenius Updates

By Dev Genius

Get the latest news and update from DevGenius publication [Take a look.](#)

Emails will be sent to umangbagadi77@gmail.com. [Not you?](#)



Get this newsletter

