Allocation of Dynamic Memory

Learn how memory can be dynamically allocated to variables in C++.

We'll cover the following

Introduction#

For dynamic allocation, we have to do the following two steps:

- First, allocate the dynamic space.
- Then, store the starting address of the dynamic space in the pointer.

`new` operator:#

> *The unary operator **new** allocates memory in bytes during the run time from the free store.*

If memory is available on the free store, the new operator will reserve that memory and return its starting address.

Syntax#

The basic syntax for getting memory during the run-time is given below:

```
new datatype ;
```

See the code given below!

```
#include <iostream>
using namespace std;

int main() {
  // Reserve 4 bytes in free store
  new int;
  return 0;
}
```

In the above program, we are requesting **4 bytes** of memory from the free store. The `new` operator reserves the **4 bytes** of memory in the free store and returns the starting address of the allocated space. To access the dynamically allocated space, we must store the returned address somewhere. Here, pointers come in handy!

Pointers#

To store the starting address returned by the new operator, we use pointers. The basic syntax is given below:

```
datatype *pointer = new datatype
```

```
#include <iostream>
using namespace std;
```

```
2  using namespace std;
3
4  int main() {
5    // Declare pointer ptr
6    int * ptr = nullptr;
7    // Store the starting address of dynamically reserved 4 bytes in ptr
8    ptr = new int;
9    return 0;
10 }
```

Line No. 8: In the above program, we are storing the starting address of the allocated **4 bytes** in the pointer `ptr`.

Accessing dynamic space#

How can we store something in a reserved dynamic space?

Here, we use the dereferencing operator.

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5    // Declare pointer ptr
6    int * ptr = nullptr;
7    // Store the starting address of dynamically reserved 4 bytes in ptr
8    ptr = new int;
9    // Store 100 in dynamic space
10   *ptr = 100;
11   // Print value pointed by ptr
12   cout << *ptr;
13   return 0;
14 }
```

```
Output                                                    1.69s

 100
```

We can access the content of the dynamic space pointed by the pointer `ptr` using the dereference operator `*` before the pointer, and then store **100** in it.

Quiz

Q What would the value of `*ptr1` at the end of this code be?

```
int main(){
  int * ptr1 , *ptr2;
  ptr1 = new int;
  ptr2 = new int;
  *ptr1 = 100;
  *ptr2 = 200;
  ptr1 = ptr2;
}
```

A) 100

Your Answer

B) 200

Explanation

`ptr1` is now pointing to the value pointed by the `ptr2` which is 200

ptr2, which is 200.

C) Address of `ptr1`

D) Address of `ptr2`

Submit Answer

Reset Quiz

In the upcoming lesson, you will learn how to free dynamically allocated space.

Stay tuned!

Back

Types of Allocation

Next

Deallocation of Dynamic Memory

Mark as Completed

Report an Issue