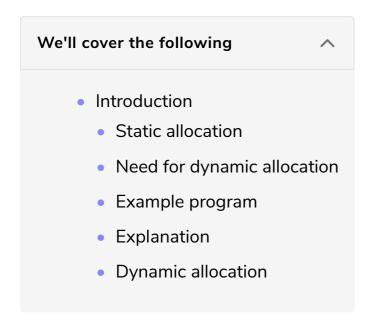
Types of Allocation

In this lesson, you will study the types of allocation.



Introduction

In C++, we can allocate memory in two ways:

- Static allocation
- Dynamic allocation

Static allocation

In **static allocation**, a fixed amount of memory is allocated to the variables or arrays before the execution of the program (during compile-time), and we cannot request more memory during the running program.

In static allocation:

- We must know the size of an array or variable during the compile time.
- Memory is allocated and deallocated to the variables by the compiler.

So far, we have seen the static allocation.

Need for dynamic allocation

However, sometimes you will encounter a situation where you don't know in advance how much memory is needed to store the data.

Example program

Suppose you want to input a sentence from the user, but you don't know the exact characters needed in an array.

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

int main() {
   char sentence [10];
   cout << "Please write your sentence:" << endl;
   cin >> sentence;
   cout << sentence;
}</pre>
```

Explanation

- If you initialize an array with fewer characters than the size of an input sentence, then you may get an error.
- If you initialize an array with more characters than the actual size of an input sentence, then the unused memory is wasted.

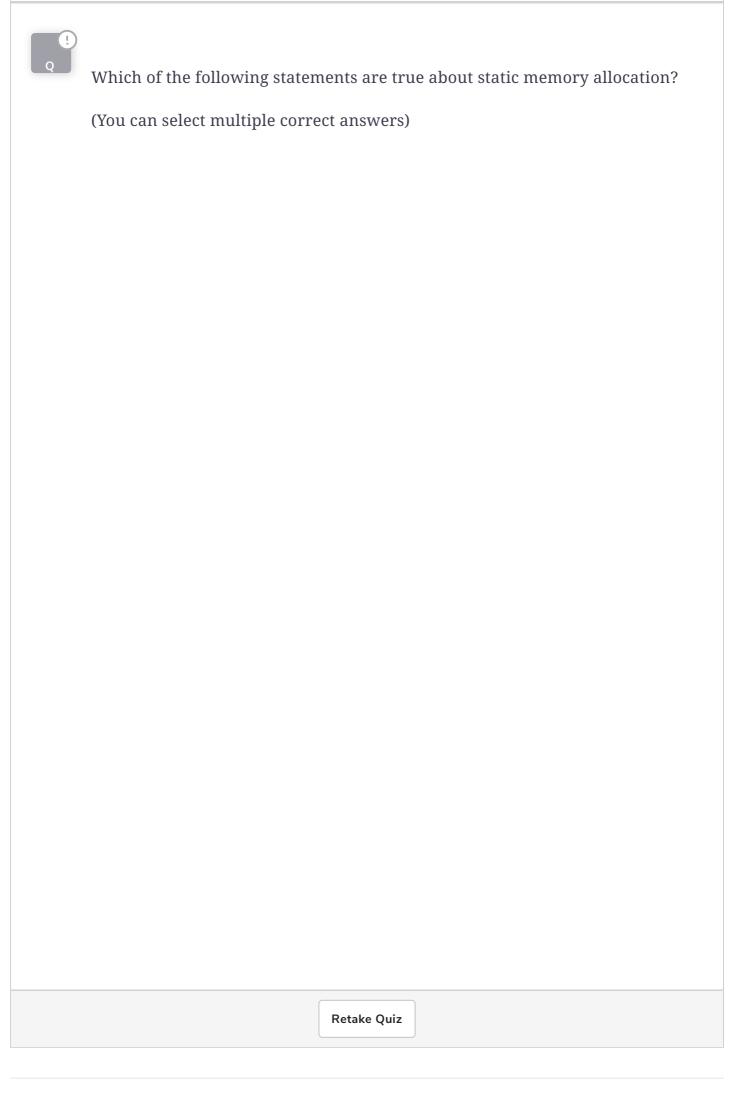
Here, dynamic allocation comes in.

Dynamic allocation

In **dynamic allocation**, we can get as much memory as we want during the program execution.

In dynamic allocation:

- We can get more memory during the running program from the free store.
- Memory is allocated and deallocated by the programmer during the run-time.



Let's get into the details of the allocation of dynamic memory in C++.