Delete dynamic array C++

how to delete a dynamically created array in c++?

In C++, dynamically created arrays are allocated using the `new` keyword, and **you must use** `delete[]` to properly deallocate them. Here's how you do it:

1. Create a dynamically allocated array:

```cpp
int* arr = new int[10];  // dynamically allocates an array of 10 integers
```

2. Delete the dynamically allocated array:

```cpp
delete[] arr;
```

⚠️ Important:

- Always use `delete[]` for arrays (not just `delete`).
- Using `delete` (without `[]`) on an array leads to **undefined behavior**.

Full Example:

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

int main() {
    int* arr = new int[5];  // Allocate array

    // Use the array
    for (int i = 0; i < 5; ++i) {
        arr[i] = i * 2;
        cout << arr[i] << " ";
    }
    cout << endl;

    delete[] arr;  // Deallocate array
    return 0;
}
```

Let me know if you're using smart pointers (like `` `std::unique_ptr` ``) or vectors, which are safer alternatives.

do I need to assign nullptr to array to avoid address not getting deleted?

Great question! While **you don't *need* to assign `nullptr` after `delete[]`**, doing so can help avoid bugs, especially **dangling pointer access**. Let me explain both the behavior and best practices.

🔧 What Happens on `delete[]`:

When you do this:

```cpp
delete[] arr;
```

- The memory is deallocated.
- But `arr` still holds the **same (now invalid) address** — it becomes a **dangling pointer**.

✅ Good Practice:

After `delete[]`, it's a good idea to set the pointer to `nullptr` to avoid accidentally using the dangling pointer later.

```cpp
delete[] arr;
arr = nullptr;  // Safe practice
```

Now:

- If you try to use `arr`, it’s `nullptr`, and most programs will crash safely or check for null.
- It prevents hard-to-find bugs caused by accessing freed memory.

🧠 Summary:

Action	Required?	Recommended?	Why?
`delete[] arr`	✅ Yes	✅ Yes	Frees memory

Action	Required?	Recommended?	Why?
`` `arr = nullptr` ``	❌ No	✅ Yes	Prevents dangling pointer access

Let me know if you want to see what happens when you *don’t* assign `` `nullptr` ``.