

Array & Vector

array<int> a; ?



No, you cannot declare an array in C++ using `array<int> a;`. This syntax is incorrect for array declaration. In C++, arrays are typically declared using one of the following methods:

1. C-style Arrays

You can declare an array using the traditional C-style syntax:

cpp

Copy code

```
int a[10]; // Declares an array of 10 integers
```

2. `std::array`

If you want to use the C++ standard library's `std::array`, you need to include the appropriate header and use the correct syntax:

cpp

Copy code

```
#include <array>

std::array<int, 10> a; // Declares an array of 10 integers using std::array
```

3. `std::vector`

Alternatively, you can use `std::vector` if you need a dynamic array:

cpp


Copy code

```
#include <vector>

std::vector<int> a; // Declares a dynamic array of integers using std::vector
```

Here's an example of how each can be used:


cpp

 Copy code

```
int main() {
    int a[5] = {1, 2, 3, 4, 5}; // Initialize the array
    for(int i = 0; i < 5; ++i) {
        std::cout << a[i] << " ";
    }
    return 0;
}
```

``std::array`` Example

cpp


 Copy code

```
#include <iostream>
#include <array>

int main() {
    std::array<int, 5> a = {1, 2, 3, 4, 5}; // Initialize the array
    for(const auto& elem : a) {
        std::cout << elem << " ";
    }
    return 0;
}
```

``std::vector`` Example

cpp

 Copy code

```
#include <iostream>
#include <vector>

int main() {
    std::vector<int> a = {1, 2, 3, 4, 5}; // Initialize the vector
    for(const auto& elem : a) {
        std::cout << elem << " ";
    }
    return 0;
}
```

Choose the appropriate type based on your needs—fixed-size arrays with ``std::array`` or dynamic arrays with ``std::vector``.