Remaining topics

Victor Eijkhout, Susan Lindsey

Fall 2025

last formatted: November 20, 2025

Array class

1. Array class

Arrays:

```
1 #include <array>
2 std::array<int,5> fiveints;
```

- Size known at compile time.
- Vector methods that do not affect storage
- Zero overhead.

2. Array initialization

With template parameters:

```
1 array<int,3> i3 = {1,2,3};
2 // or
3 array<int,3> i3 { {1,2,3} };
4 // or
5 array<int,3> i3{1,2,3};
```

With C++17 template argument deduction:

```
1 array i3 = {1,2,3};
2 // Note: DOES NOT COMPILE:
3 array not4{1.5,2,3,4};
```

3. Random walk implementations

Random walk implementations

Vector:

```
1 // rand/compare.cpp
2 vector<float> random_walk_1( int
      steps ) {
3   vector<float> result(3);
4   for (int s=0; s<steps; ++s) {
5     vector<float> update(3);
6     random_vector<vector<float>>(
       update );
7     for ( auto i : {0,1,2} )
8       result[i] += update[i];
9   }
10  return result;
11 }
```

Array:

```
1 // rand/compare.cpp
2 array<float,3> random_walk_2( int
      steps ) {
3   array<float,3> result = {0,0,0};
4   for (int s=0; s<steps; ++s) {
5     array<float,3> update{};
6     random_vector<array<float,3>>
        ( update );
7     for ( auto i : {0,1,2} )
8       result[i] += update[i];
9   }
10  return result;
11 }
```

Enum

4. Name collisions for enums

Global names:

```
1 // enum/enumnot.cpp
2 enum trafficlight { red,yellow,green };
3 cout << red << "," << yellow << "," << green << '\n';
4 enum flag { red,white,blue }; // Collision!
```

5. Enum class

```
1 // enum/enumclass.cpp
2 enum class flowers { grass, poppy, bluebonnet };
3 vector<flowers> field(10,flowers::grass);
4 field[1] = flowers::poppy;
5 field[5] = flowers::bluebonnet;
```

More

6. Type alias

```
1 using idxint = std:int64_t
```