- Solution

This is the solution for the exercise in the previous lesson.

WE'LL COVER THE FOLLOWING ^

- Solution
 - Explanation
- Further information:

Solution

```
#include <array>
                                                                                                 G
#include <iostream>
#include <set>
#include <unordered_set>
#include <vector>
int main(){
  std::cout << std::endl;</pre>
  std::array<int, 5> myArray = {-10, 5, 1, 4, 5};
  for (auto i: myArray) std::cout << i << " ";</pre>
  std::cout << "\n\n";</pre>
  std::vector<int> myVector = {-10, 5, 1, 4, 5};
  for (auto i: myVector) std::cout << i << " ";</pre>
  std::cout << "\n\n";</pre>
  std::set<int> mySet = {-10, 5, 1, 4, 5};
  for (auto i: mySet) std::cout << i << " ";</pre>
  std::cout << "\n\n";</pre>
  std::unordered_multiset<int> myUnorderedMultiSet = {-10, 5, 1, 4, 5};
  for (auto i: myUnorderedMultiSet) std::cout << i << " ";</pre>
  std::cout << "\n";</pre>
  std::cout << std::endl;</pre>
```







Explanation

- In line 11, an std::array of size 5 and type int is created with the given data.
- In line 15, an std::vector of type int is created using the given data. The integers are not inserted in numerical order.
- In line 19, an std::set of type int is created using the given data.
 Integers are inserted in numerical order and duplicate elements, such as
 5, are not inserted in the set.
- In line 23, an std::unordered_multiset of type int is created using the
 given data. The keys are not sorted. Duplicate keys are allowed in a
 std::unordered_multiset.

Further information:

initializer_list

In the next chapter, we'll look into the details of const, constexpr, and volatile.