- Exercise

Let's solve the exercise on tag dispatching in this lesson.

WE'LL COVER THE FOLLOWING

- Problem Statement

Problem Statement

Extend the example template for Tag Dispatching given below. The following program shows a typical example of tag dispatching on iterators. Call the `advance_` algorithm in the `main` function with different containers such as `std:.vector`, `std::list`, and `::forward_list` to put the iterator 5 positions further.

```
// TemplatesTagDispatching.cpp

#include <iterator>
#include <forward_list>
#include <list>
#include <vector>
#include <iostream>

template <typename InputIterator, typename Distance>
void advance_impl(InputIterator& i, Distance n, std::input_iterator_tag) {
        std::cout << "InputIterator used" << std::endl;
    while (n--) ++i;
}

template <typename BidirectionalIterator, typename Distance>
void advance_impl(BidirectionalIterator& i, Distance n, std::bidirectional_iterator_tag) {
        std::cout << "BidirectionalIterator used" << std::endl;
    if (n >= 0)
        while (n--) ++i;
    else
        while (n++) --i;
}

template <typename RandomAccessIterator, typename Distance>
void advance_impl(RandomAccessIterator& i, Distance n, std::random_access_iterator_tag) {
        std::cout << "RandomAccessIterator used" << std::endl;
    i += n;
}
```

```
}

template <typename InputIterator, typename Distance>
void advance_(InputIterator& i, Distance n) {
    typename std::iterator_traits<InputIterator>::iterator_category category;
    advance_impl(i, n, category);
}

int main(){
    // call the above functions to move them 5 position further
}
```

In the next lesson, we’ll look at the solution to this exercise.