- Exercise

Let's answer a few questions in this lesson.

WE'LL COVER THE FOLLOWING

- Try it yourself

Try it yourself

- `std::forward_list` has a reduced interface, so we can't use it with a lot of STL algorithms.
- What are the characteristics of the methods of `std::forward_list`?

Recall and implement all of these below.

```
#include <iostream>

int main() {

  // Try here yourself

  return 0;
}
```

Pointers that might help us to solve the above exercise:

- Accessing of an arbitrary element is slow because we might have to iterate forward through the whole list.
- To `add` or `remove` an element is fast, if the iterator points to the right place.
- If we `add` or `remove` an element, the iterator remains valid.

- We can only move forward with the iterator.

This concludes our discussion on sequential containers. In the next chapter, we’ll discuss the basics of associative containers.