Summary

Let's summarise the chapter.

A few core elements to know about `std::optional`:

- `std::optional` is a wrapper type to express “null-able” types
- `std::optional` won’t use any dynamic allocation
- `std::optional` contains a value or it’s empty
- use `operator *`, `operator->`, `value()` or `value_or()` to access the underlying value.
- `std::optional` is implicitly converted to bool so that you can easily check if it contains a value or not

---

Get ready for a short quiz on this chapter.