Introduction

Find out what this section is all about below!

C++17 adds a few wrapper types that make it possible to write more expressive code. In this chapter, you'll see `std::optional`, which models a `nullable` type. With this utility, your objects can easily express that they don't have any value. Such behaviour is more straightforward to achieve than using some unique values (like `-1`, `null`).

In this chapter, you'll learn:

- Why we need nullable types
- How does `std::optional` work and what does it do
- Operations on `std::optional`
- The performance cost of using the type
- Example use cases

Let's get started with the std::optional utility introduced in C++17.