Summary

Let's summarise the chapter.

Some of the highlights are as follows:

- `std::any` is not a template class
- `std::any` uses Small Buffer Optimisation, so it will not dynamically allocate memory for simple types like ints, doubles... but for larger types, it will use extra `new`.
- `std::any` might be considered 'heavy', but offers a lot of flexibility and type-safety.
- you can access the currently stored value by using `any_cast` that offers a few "modes": for example it might throw an exception or return `nullptr`.
- use it when you don't know the possible types - in other cases consider `std::variant`.

We will conclude with compiler support in the next lesson.