Abstraction in C++

1. Delivering only essential information to the outer world while masking the background details.

2. It is a design and programming method that separates the interface from the implementation.

3. Real life e.g., various functionalities of AirPods but don’t know the actual implementation/ working

    1. To drive a car, one only needs to know the driving process and not the mechanics of the car engine

Abstraction in Header files

1. Function’s implementation is hidden in header files.

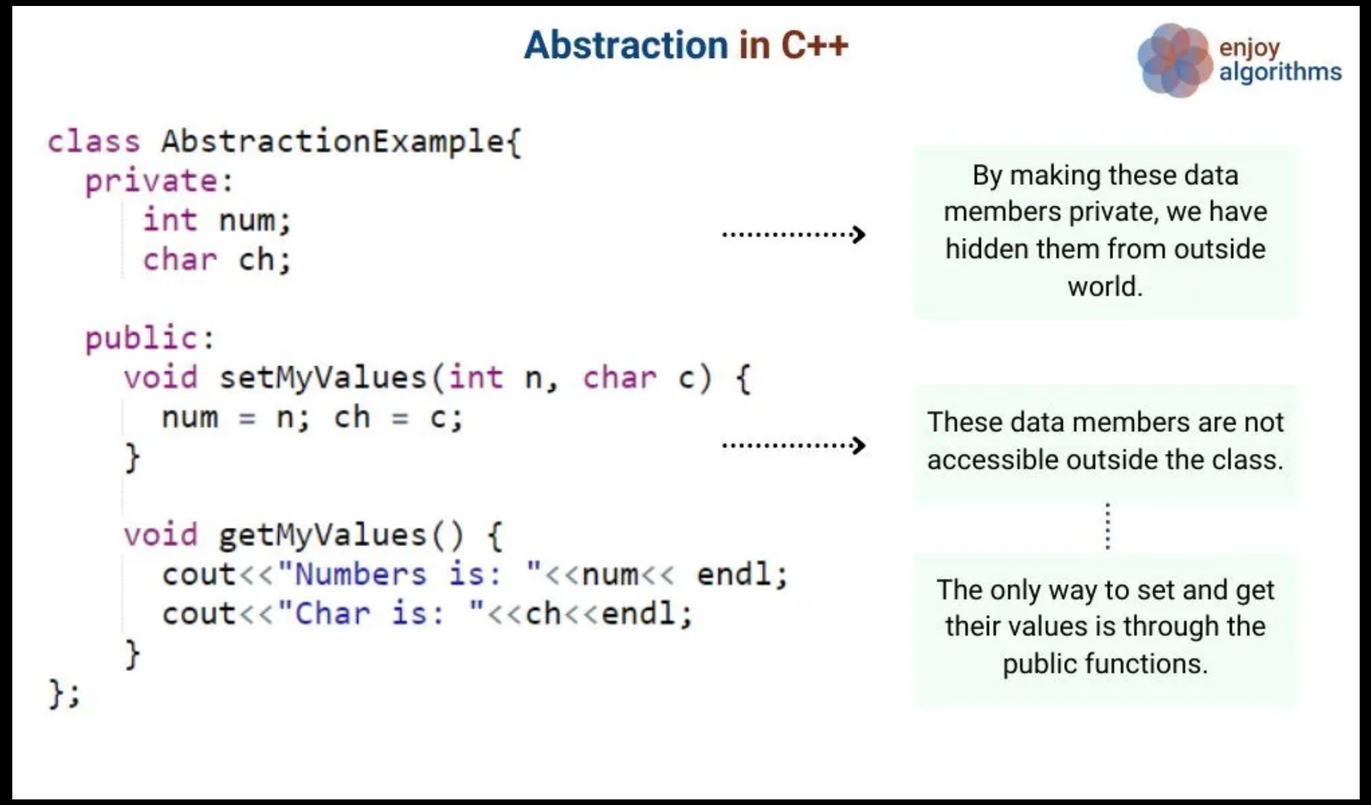
2. We could use the same program without knowing its inside working.

3. E.g., Sort(), for example, is used to sort an array, a list, or a collection of items, and we know that if we give a container to sort, it will sort it, but we don't know which sorting algorithm it uses to sort that container.

Abstraction using Classes

1. Grouping data members and member functions into classes using access specifiers.

2. A class can choose which data members are visible to the outside world and which are hidden.



What is Abstract Class?

1. Class that contains at least one pure virtual function, and these classes cannot be instantiated.

2. It has come from the idea of Abstraction.

Design Strategy

1. Abstraction divides code into two categories: interface and implementation. So, when creating your component, keep the interface separate from the implementation so that if the underlying implementation changes, the interface stays the same.

2. In this instance, any program that uses these interfaces would remain unaffected and would require recompilation with the most recent implementation.