What is Data Hiding?

In this section, we will learn two concepts which help us create an efficient class in C++

WE'LL COVER THE FOLLOWING

Data hiding is an essential process in the OOP cycle.

> In layman's terms, data hiding refers to the concept of **hiding the inner workings of a class** and simply providing an **interface** through which the outside world can interact with the class without knowing what's going on inside.

Our goal is to make the interaction between different classes as simple as possible. This can only be achieved if the interfaces of the classes are simple. One class does not need to know anything about the underlying algorithms of another class. However, the two can still communicate.

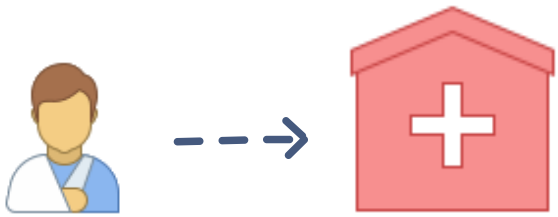
Real Life Example

Let's apply this to a real-world scenario. Take the doctor-patient model. In case of an illness, the patient consults the doctor, after which he or she is prescribed the appropriate medicine.

The patient only knows the process of going to the doctor. The logic and reasoning behind the doctor's prescription of a certain medicine are unknown to the patient. A patient will not understand the medical details the doctor uses to reach his/her decision on the treatment.

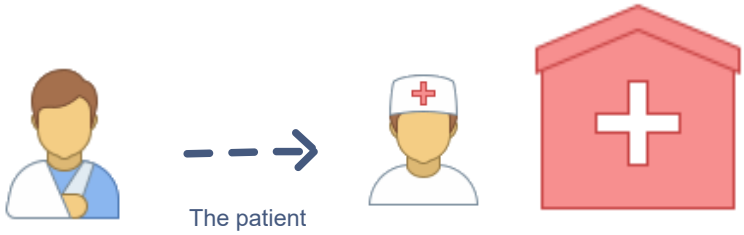
This is a classic example of the patient class interacting with the doctor class

without knowing the inner workings of the doctor class.



A patient visits the hospital for a cure.

1 of 5

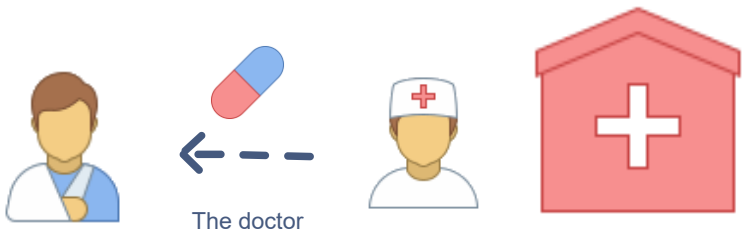


The patient consults the doctor.

2 of 5

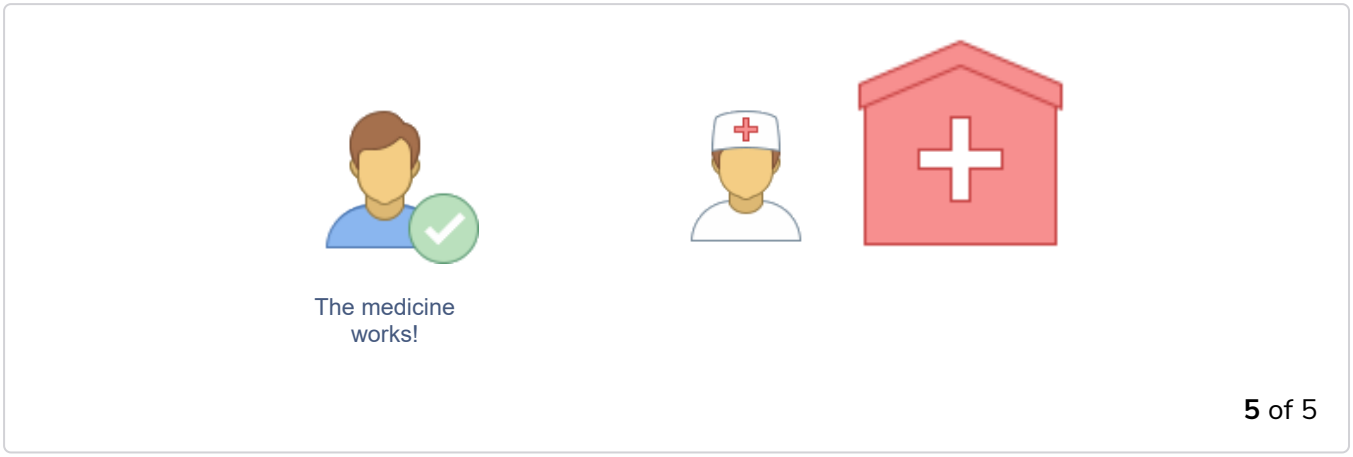
The doctor diagnoses the illness

3 of 5



The doctor prescribes a medicine

4 of 5



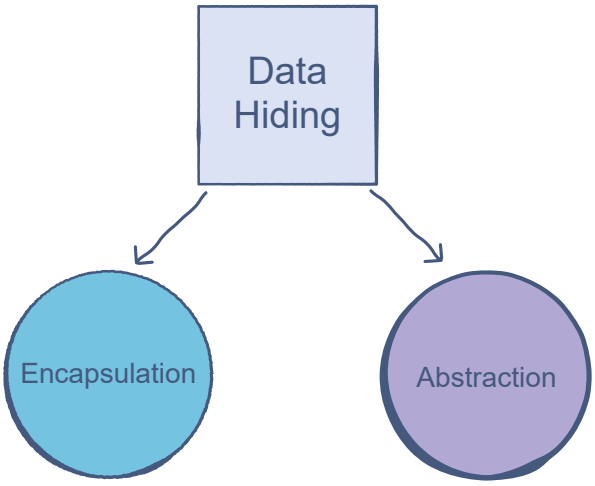
The transaction shown above seems fairly simple. Data hiding is useful because it can apply the same simplicity to transactions between objects of different classes.

Components of Data Hiding

Data hiding can be divided into two primary components:

1. Encapsulation
2. Abstraction

When used together, they allow us to make efficient classes for further use in our application.



Now, we'll study these two topics in details.