Encapsulation

In this lesson, you'll get familiar with one component of data hiding, encapsulation.

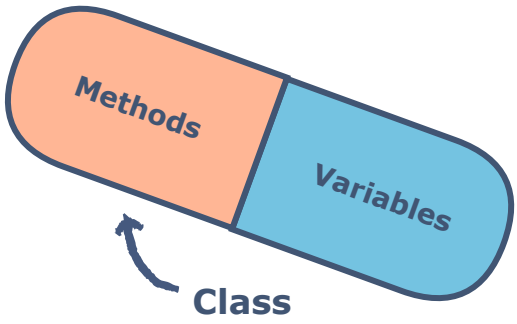
WE'LL COVER THE FOLLOWING

Definition

Encapsulation is a pillar of the object-oriented programming paradigm and is used to achieve data hiding.

> **Encapsulation** in OOP refers to binding the **data** and the **methods manipulating that data** together in a single **unit** (class).

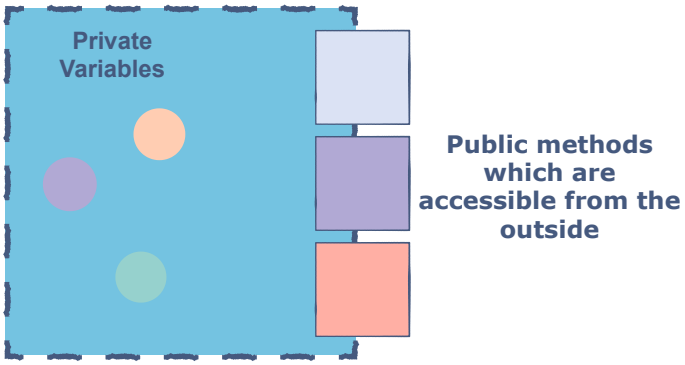
Depending upon this **unit**, objects are created. Encapsulation is normally done to hide the state and representation of an object from the outside. A class can be thought of as a **capsule** having *methods* and *data members* inside it.



As a rule of thumb, a good convention is to declare all the *fields or instance variables* of a class `private`. This will restrict the direct access of these class members from the code outside that class.

At this point, a question can be raised that if the methods and variables are encapsulated in a class then, *"how can they be used outside of that class"*?

The answer to this is simple; one has to implement `public` methods to let the outside world communicate with this class. These methods can be *getters, setters,* and any other custom methods implemented by the programmer.



Advantages of Encapsulation

- Classes are easier to change and maintain.
- We can specify which data member we want to keep hidden or accessible.
- We decide which variables can be read or written by the outside world (increases flexibility).

In the next lesson, we'll learn more about **encapsulation** with the help of some examples.