Encapsulation in OOP with example

Encapsulation is one of the four important OOP concepts. Encapsulation is a mechanism for combining variables(data) and methods (code) into a single unit. It is better illustrated by the picture below.

Diagram

Description automatically generated

Diagram

Description automatically generated

Encapsulation is used to hide the values or status of a structured data object in a class, to prevent unauthorized access to it. It is the process of hiding information and protecting the data and behavior of the object. Publicly accessible methods are usually provided for access to class (so-called getters and setters) values, and these methods are used to obtain and modify values within other client class objects. Encapsulation can be achieved by declaring all the variables in the class as private and writing public methods in the class to set and get the values of the variable. It is more defined with setter and getter methods. Encapsulations perform very important role in OOP concepts. It controls the way of data accessibility, Modifies the code based on needs, helps us to achieve a loose couple, Gets the simplicity of our application and it also allows to modify a portion of the code without interrupting other functions or code in the program. Data hiding, increased flexibility, reusability Testing code is easy are benefits of encapsulation. In data hiding, user has no idea about the internal implementation of the class. The user does not even know how to store values in class variables. User only knows that we are sending a value processing system and with that value the variables start. Encapsulate class is easy to test, so it is also better to test units. Encapsulation also improves the re-usability and is easy to change with new requirements.