

Encapsulation:

Definition: encapsulation is one of the basic principles of object-oriented programming (OOP) that describes the bundling of data in a class and its related functions.

Purpose: It protects an object's internal state by allowing only access to its method implementations. Therefore, it is only possible for a method that belongs to a class to change the state of an object.

Access Control: Typically, encapsulation has something to do with visibility control over elements (attributes and methods) by means of access modifiers like public, private, and protected among others within a given class.

DATA ABSTRACTION

Definition: Data abstraction is a process that simplifies intricate systems by defining classes according to their important properties and behaviors, while disregarding irrelevant details.

Essential Features: It is about focusing on the necessary components of a class without considering irrelevant information. This is made possible through abstract classes and interfaces in OOP.

Hierarchy: Data abstraction allows creation of class hierarchies where each level of abstraction represents a more generalized and abstract concept.

Example: In a real-world scenario, if we have a "Vehicle" class, we abstract the essential features shared by all vehicles (e.g., speed, capacity), without getting into specifics like whether it's a car, truck, or bicycle.

Relationship between Encapsulation and Data Abstraction:

Connection: Object-oriented programming (OOP) tightly relates the concepts of encapsulation and data abstraction.

Implementation: To achieve data abstraction, encapsulation is involved. This is through encompassing data and methods into a class limiting access to them hence enabling creation of abstract data types.

Modularity: It is therefore necessary to consider modularity when developing software applications. These should be done by creating classes that represent generic entities while keeping their features hidden.

Related Posts:

1. Object Oriented Programming & Methodolog Viva Voce
2. How to install compiler for code blocks
3. Object Oriented Programming
4. Differences between Procedural and Object Oriented Programming
5. Features of Object Oriented Paradigm
6. Inheritance in Object Oriented Programming
7. Object Oriented Programming
8. Introduction to Object Oriented Thinking & Object Oriented Programming
9. Difference Between Object-Oriented Programming (OOP) and Procedural Programming
10. features of Object oriented paradigm
11. Merits and demerits of Object Oriented methodology
12. Concept of Objects: State, Behavior & Identity of an object
13. Access modifiers

- 14. Static members of a Class
- 15. Instances in OOP
- 16. Message Passing in OOP
- 17. Construction and destruction of Objects