The concept of objects is fundamental to object-oriented programming (OOP), and it revolves around three key aspects: state, behavior, and identity.

State:

Definition: For any given moment, the state of an object is formed from its characteristics. Example: The state of 'car' can include properties like color, speed, fuel level and present gear.

Behavior:

Definition: This is a terminology that refers to the actions or methods that are performed by an object. They describe how the given object relates with others or its environment.

Example: Behaviors for cars can include accelerate, brake, changeGear, turnLeft.

Identity:

Definition: An object's identity is what separates it from other objects. It helps differentiate between multiple instances of the same class in the system.

Example: Every car on the road has certain unique identity such as a license plate number or Vehicle Identification Number (VIN).

Key Points:

• Encapsulation: Objects encapsulate both state and behavior. The object encapsulates the internal state and uses methods to control access to it, promoting data security

and integrity.

- Abstraction: Objects make real-world things seem less real by modeling their most important traits and interactions. This lets developers focus on what is important and ignore the details that are not necessary.
- Inheritance: Through inheritance, objects can inherit attributes and behaviors from other objects, fostering code reuse and creating a hierarchical structure.
- Polymorphism: Objects of different classes can be treated as objects of a common base class through polymorphism, enabling flexibility and extensibility in the code.

Related posts:

- 1. Abstraction and encapsulation
- 2. Object Oriented Programming & Methodolog Viva Voce
- 3. How to install compiler for code blocks
- 4. Object Oriented Programming
- 5. Differences between Procedural and Object Oriented Programming
- 6. Features of Object Oriented Paradigm
- 7. Inheritance in Object Oriented Programming
- 8. Object Oriented Programming
- 9. Introduction to Object Oriented Thinking & Object Oriented Programming
- 10. Difference Between Object-Oriented Programming (OOP) and Procedural Programming
- 11. features of Object oriented paradigm
- 12. Merits and demerits of Object Oriented methodology
- 13. Access modifiers
- 14. Static members of a Class
- 15. Instances in OOP
- 16. Message Passing in OOP

Concept of Objects: State, Behavior & Identity of an object

17. Construction and destruction of Objects