

ANSWER 7 –

Inheritance is a fundamental concept in object-oriented programming that allows objects or classes to inherit properties and methods from other objects or classes. It establishes a hierarchical relationship between classes, where a subclass (or child class) can inherit and extend the characteristics of a superclass (or parent class).

In simple words, inheritance can be understood as follows:

Code Reusability: Inheritance allows you to reuse code by creating new classes based on existing classes. The new class, known as a subclass, inherits the properties and methods of the existing class, known as a superclass. This promotes code reuse, as you don't need to rewrite the common properties and behaviors in every class.

Class Hierarchy: Inheritance establishes a hierarchy of classes, where subclasses are derived from a superclass. This creates a structure that represents the relationships and classifications between different types of objects. Subclasses inherit the characteristics of their superclass and can add or modify properties and methods specific to their own requirements.

Overriding and Extension: Inheritance allows subclasses to override or extend the behavior of their superclass. Subclasses can provide their own implementation of methods inherited from the superclass, allowing customization and specialization. This enables you to tailor the behavior of classes to suit specific needs while maintaining the common functionality inherited from the superclass.

Polymorphism: Inheritance plays a significant role in enabling polymorphism, which allows objects of different types to be treated as objects of a common superclass. This promotes code flexibility and interchangeability, as objects of different subclasses can be used interchangeably through their shared superclass.

The purpose of inheritance is to promote code reusability, modularity, and extensibility. It allows you to create a hierarchy of classes that organizes and categorizes objects based on their shared characteristics. Inheritance facilitates the creation of specialized classes that inherit and build upon the properties and behavior of more general classes, resulting in efficient and maintainable code. It promotes the principles of abstraction, encapsulation, and polymorphism in object-oriented programming.