TITLE : Oops Concepts

AUTHOR : Balaji V

DESIGNATION : Trainee | Impact Training

CREATED ON : 19/11/2024

MODIFIED ON : 21/11/2024

REVIEWED BY :

REVIEWED ON :

OOPS – Object Oriented Programming

OOPS allows developers to model real-world entities and their behaviors using classes and objects.it promotes code reusability, maintainability.

Class:

It is a User defined data type. In Other words, class is a code template for creating objects.

Object:

Objects are variables that contains data and functions that can be used to manipulate data

Abstraction:

Abstraction is the concept of hiding the complex implementation details and showing only the essential features of an object.

Example:

Money Transaction

When a Money Transfer happens only the necessary information is shown to the User. The methods involved in making a transaction are hidden from the User.

Encapsulation:

It refers to bundling of data members (Variables) and methods (Function) that operates on the data into a single unit or class. It also involves restricting access to some of an object's components using access modifiers (Public, Private, Protected).

Example:

Library system

In a library system, the details of a book (such as Title, Author, ISBN, etc) are encapsulated within a Book class. Users interact with the book objects through methods like borrowBook() or returnBook(), but they don't need direct access to the internal details of the book itself.

Inheritance:

Inheritance is a concept in which one class (sub class) acquires the properties and behaviors (Data members, Methods) of another class (super class). It promotes code reusability.

Example:

Electronic Devices

Consider a base class named Electronic, and then Phone, Laptop and Tablet classes inherit from it. All devices have common properties like powerOn() or powerOff(), but each subclass can implement specific features, like makeCall() for Phone or useTouchscreen() for Tablet.

Polymorphism:

Polymorphism means 'Many forms. It is the ability of different objects to respond to the same method or function call in different ways.

Example:

Shape Drawing

Consider a base class name as Shape with a method draw() . Subclasses like Circle and Rectangle can provide their own specific implementations of draw() . When you call draw() on any Shape object, the correct version of the method is executed based on the object's actual type.

References:

- Javatpoint
- Datacamp
- W3school