Object Oriented Programming

Object-Oriented Programming (OOP) is a programming paradigm in computer science that relies on the concept of **classes** and **objects**.

It is used to structure a software program into simple, reusable pieces of code blueprints (usually called classes), which are used to create individual instances of objects.

The main aim of OOP is to bind together the data and the functions that operate on them so that no other part of the code can access this data except that function.

Class

Classes contain fields for attributes and methods for behaviors.

Object

Objects are **instances of a class** created with specific data.

Four Principles of OOP

The four pillars of object-oriented programming are:

Inheritance:

child classes inherit data and behaviors from the parent class

Encapsulation:

containing information in an object, exposing only selected information

Abstraction:

only exposing high-level public methods for accessing an object

Polymorphism:

many methods can do the same task

Inheritance

Inheritance allows classes to inherit features of other classes. Put another way, parent classes extend attributes and behaviors to child classes. **Inheritance supports reusability**.

Encapsulation :

Encapsulation means containing all important information **inside an object**, and only exposing selected information to the outside world.

Encapsulation requires defining some fields as private and some as public.

- Private/ Internal interface: methods and properties accessible from other methods of the same class.
- Public / External Interface: methods and properties accessible from outside the class.

Abstraction

Abstraction is an extension of encapsulation that uses classes and objects, which contain data and code, to hide the internal details of a program from its users.

This is done by creating a **layer of abstraction** between the user and the more complex source code, which helps protect sensitive information stored within the source code.

Reduces complexity and improves code readability

Facilitates code reuse and organization

Data hiding improves data security by hiding sensitive details from

Enhances productivity by abstracting away low-level details

Polymorphism

Polymorphism means designing objects to **share behaviors**. Using inheritance, objects can override shared parent behaviors with specific child behaviors. Polymorphism allows the same method to execute different behaviors in two ways: method overriding and method overloading.