# Object-Oriented Programming Principles and Their Applications

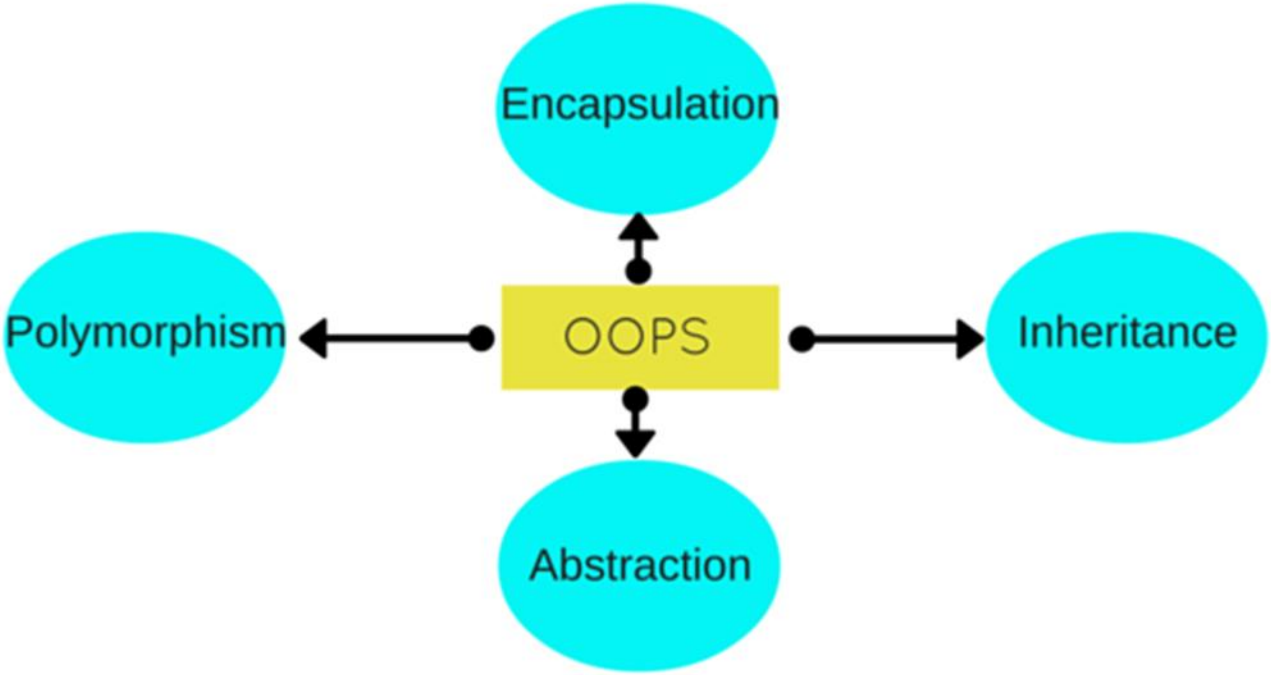
1. **Abstraction:**
   Abstraction refers to the act of representing essential features without including the background details. Hiding unnecessary data from the users and making the application user friendly then it is called abstraction.
   Example: Data abstraction is a technique of creating new data types that are well suited to an application. It allows creation of user defined data types. Having the properties of built in data types and set of permitted operations. Such as, In a program If I created to manage a library, I defined a Book class. The class contained many attributes, like ISBN, title, author, etc., but when displaying the catalog, I only showed the title and author.

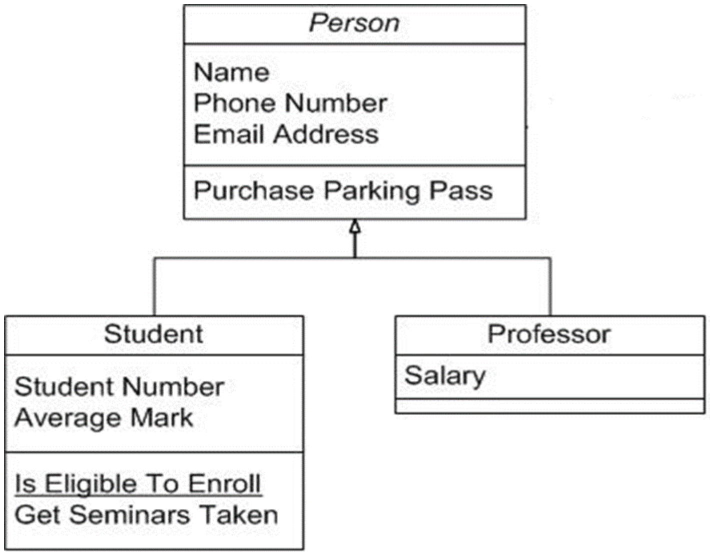
2. **Encapsulation:**
   Encapsulation is the condition of being enclosed (as in a capsule) with programming perspective, Enclosing “attributes” and “methods” within a class would be called Encapsulation.
   Example: In the same library program, the Book class had methods like checkout() and returnBook(), which modified the book's status attribute. The status attribute was private and could only be changed using these methods.

3. **Polymorphism:**
   It is mainly an ability of an object to take many forms. Polymorphism allows objects to be treated as instances of their parent class. This facilitates the execution of the same method in different classes through a common interface.
   Example: In a drawing the program when we use ShapeDraw(),this object can be also used as LineDraw(), CircleDraw(), RectangleDraw().

4. **Inheritance:**
   Inheritance is when an object acquires the property of another object. Inherited class is called parent class or super class or base class. Class that inherits a parent class is called as child class or sub class or derived class.
   Example: If I had a Media class in my library program that had general attributes like title and availability. The Book and DVD classes inherited from this Media class.

# Concept Map


Encapsulation
Polymorphism
OOPS
Inheritance
Abstraction


*Person*
Name
Phone Number
Email Address
Purchase Parking Pass
Student
Student Number
Average Mark
Is Eligible To Enroll
Get Seminars Taken
Professor
Salary

# Annotations

- Abstraction ↔ Encapsulation: Encapsulation can be viewed as a form of abstraction where the internal details of an object are hidden.
- Encapsulation ↔ Inheritance: Encapsulation is often maintained in inheritance, where private fields are not accessible, but protected ones are.
- Inheritance ↔ Polymorphism: Polymorphism often relies on inheritance, as objects of derived classes can be treated as objects of the parent class.
- Polymorphism → Method: Polymorphism is achieved through methods that can be implemented differently in various classes.
- Method → Class: Methods are contained within classes.
- Class → Object: Objects are instances of classes.