OOPs NOTES

What is OOPs?

Object-oriented programming (OOP) is a computer programming model that organizes software design around data, or objects, rather than functions and logic. An object can be defined as a data field that has unique attributes and behaviour.

What is a class?

A class is a collection of objects. Classes don't consume any space in the memory.

It is a user defined data type that act as a template for creating objects of the identical type.

A large number of objects can be created using the same class. Therefore, Class is considered as the blueprint for the object.

What is an object?

An object is a real-world entity which have properties and functionalities. Object is also called an instance of class. Objects take some space in memory.

For eg .

- Fruit is class and its object s are mango, apple, banana
- Furniture is class and its objects are table, chair, desk

What is the difference between a class and an object?

Class:

1. It is a collection of objects.
2. It doesn't take up space in memory.
3. Classes are declared just once

Object:

1. It is an instance of a class.
2. It takes space in memory.
3. Objects can be declared as and when required

What is the difference between a class and a structure?

Class is a collection of objects.

Structure is a collection of variables of different data types under a single unit

Class is used to combine data and methods together.

Structure is used to grouping data.

Class's objects are created on the heap memory.

Structure's objects are created on the stack memory.

A class can inherit another class.

A structure can't inherit another structure.

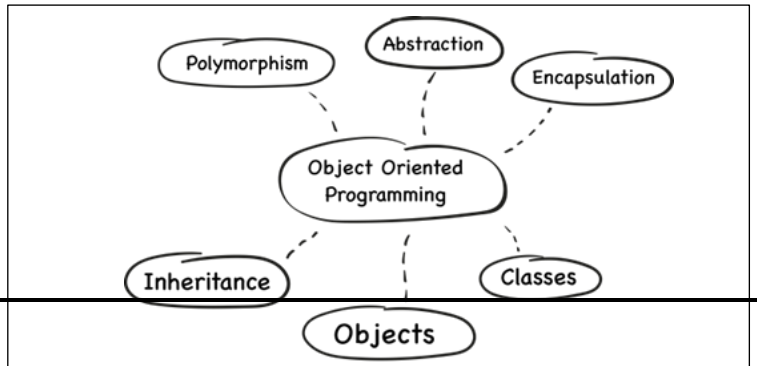
A class has all members private by default

A structure has all members public by default

Classes are ideal for larger or complex objects

Structures are ideal for small and isolated model objects

Following are the basic features of OOPs –



There are typically Three Principles in object-oriented programming (OOP):

Encapsulation: This principle focuses on the bundling of related data and functions into an object, allowing access only through the object's methods. It helps in hiding the internal implementation details and protects the data from unauthorized access.

Inheritance: This principle provides the ability to create new classes based on the existing classes. It allows the derived class (subclass) to inherit the properties and behaviours of the base class (superclass). Inheritance promotes code reusability and supports the concept of "is-a" relationship.

Polymorphism: Polymorphism means the ability of an object to take on many forms. In OOP, it refers to the ability to use a single interface or method to represent different types of objects. It allows objects of different classes to be treated as objects of a common superclass through method overriding and method overloading.

Abstraction is not considered as a separate principle in OOP, but rather a concept that is utilized in all the pillars mentioned above.

Abstraction: Abstraction is the process of hiding the internal details and complexities of an object and providing a public interface to interact with the object. It helps in achieving data security, modularity, and code maintainability.

Data Hiding: Process of hiding data from outside world.

Encapsulation:

The main advantage of encapsulation is that data is hidden and protected from randomly access by outside non-member methods of a class. Encapsulation is the process of binding data and methods in a single unit. In encapsulation, data(variables) are declared as private and methods are declared as public

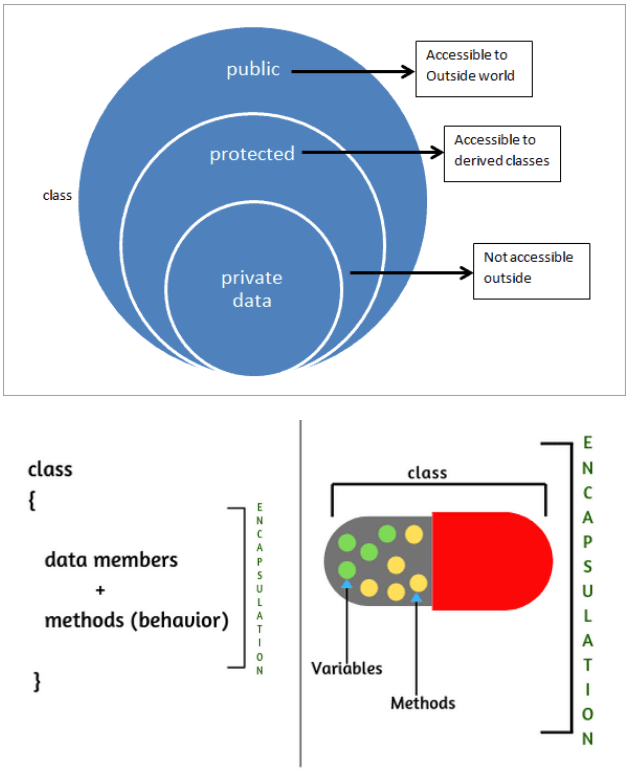


Fig: Encapsulation

Tightly Encapsulated Class:

A class is said to be tightly encapsulated if and only if each and every variable declared as private whether class contain corresponding getter setting and setter method or not and

whether these method declare public or not these things are not required to check

Note: if the parent class is not TEC then no child class is TEC

Inheritance:

Inheritance is the procedure in which one class inherits the attributes and methods of another class.

In other words, It is a mechanism of acquiring properties or behaviours of existing class to a new class

The Base Class, also known as the Parent Class is a class, from which other classes are derived.

The Derived Class, also known as Child Class, is a class that is created from an existing class

There are four types of inheritance in OOP:

- Single Level Inheritance
- Hierarchical Inheritance
- Multi-Level Inheritance
- Multiple Inheritance
- Hybrid inheritance

Abstraction:

Allows to hide unnecessary data from the user. This reduces program complexity efforts.

it displays only the necessary information to the user and hides all the internal background details.

If we talk about data abstraction in programming language, the code implementation is hidden from the user and only the necessary functionality is shown or provided to the user.

In other words, it deals with the outside view of an object (Interface)

Eg.

All are performing operations on the ATM machine like cash withdrawal etc.

but we can't know internal details about ATM

phone call we don’t know the internal processing

We can achieve data abstraction by using

1. Abstract class
2. Interface