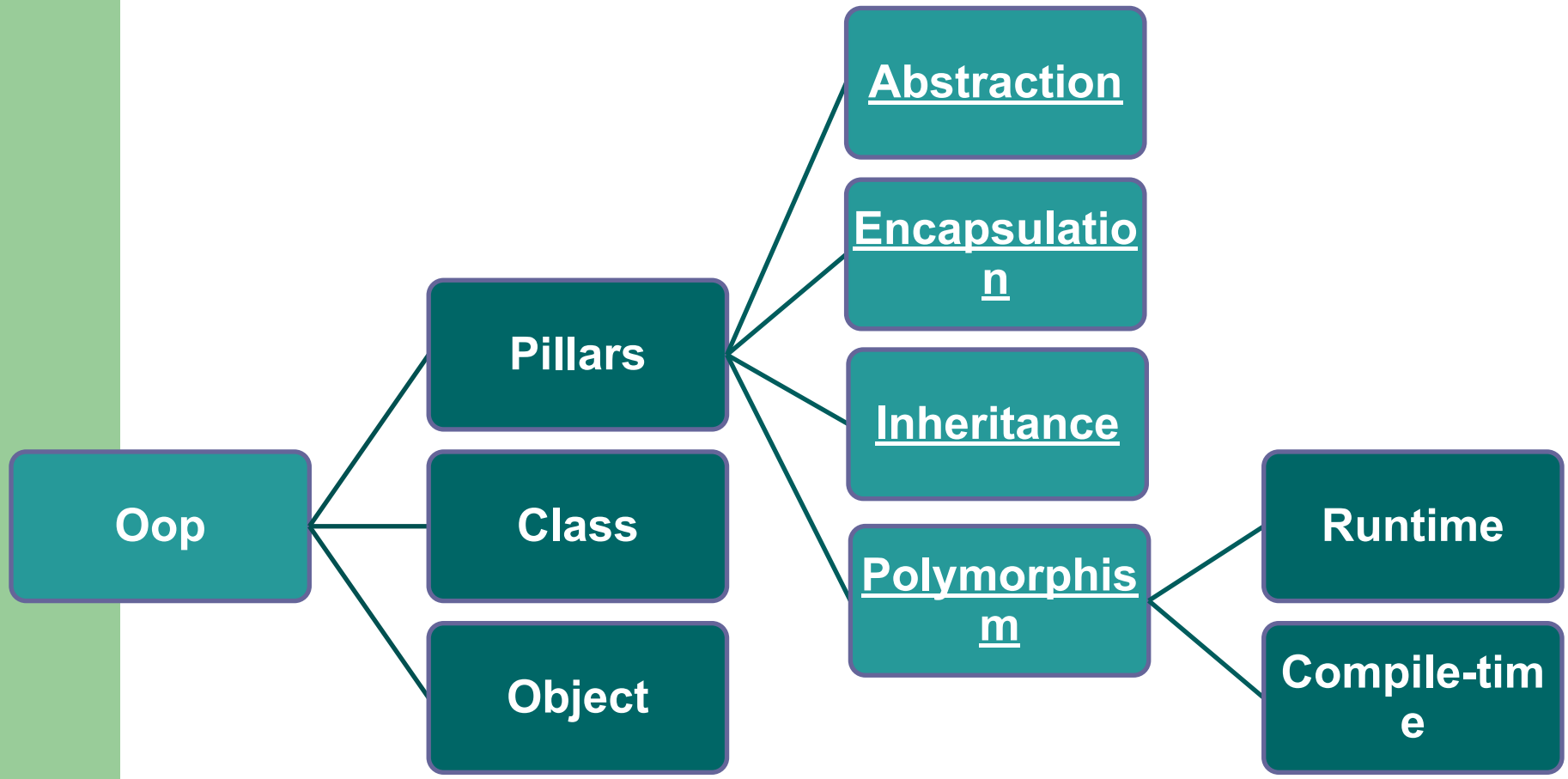
Information and Communication Technologies (ICT)
Object Oriented Programming

Dec 2022

Object-Oriented Programming

- Object-oriented programming is a programming paradigm based on the concept 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.
- There are many object-oriented programming languages including JavaScript, C++, Java, and Python..

OOPS concepts are as follows:



Object

An object is a basic unit of Object-Oriented Programming that represents real-life entities. Objects are persons, places, or things that are relevant to the system we are analyzing. Typical objects may be customers, items, orders, and so on.

An object has an identity, state, and behavior.

For example “Dog” is a real-life Object, which has some characteristics like color, Bark, Sleep, and Eats.

Class

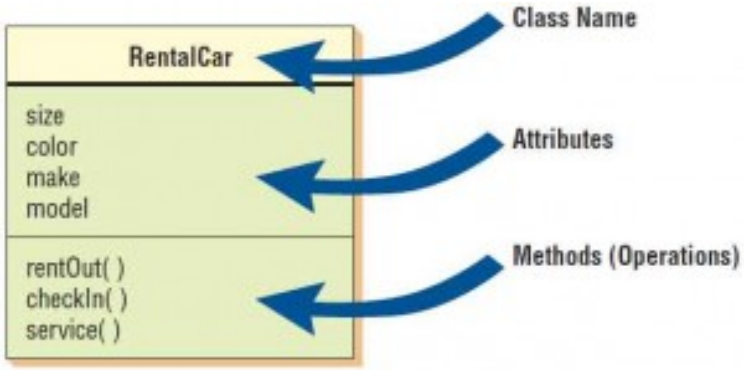
A class is a user-defined blueprint or prototype from which objects are created. It represents the set of properties or methods that are common to all objects of one type. Objects are typically part of a group of similar items called classes. The desire to place items into classes is not new.

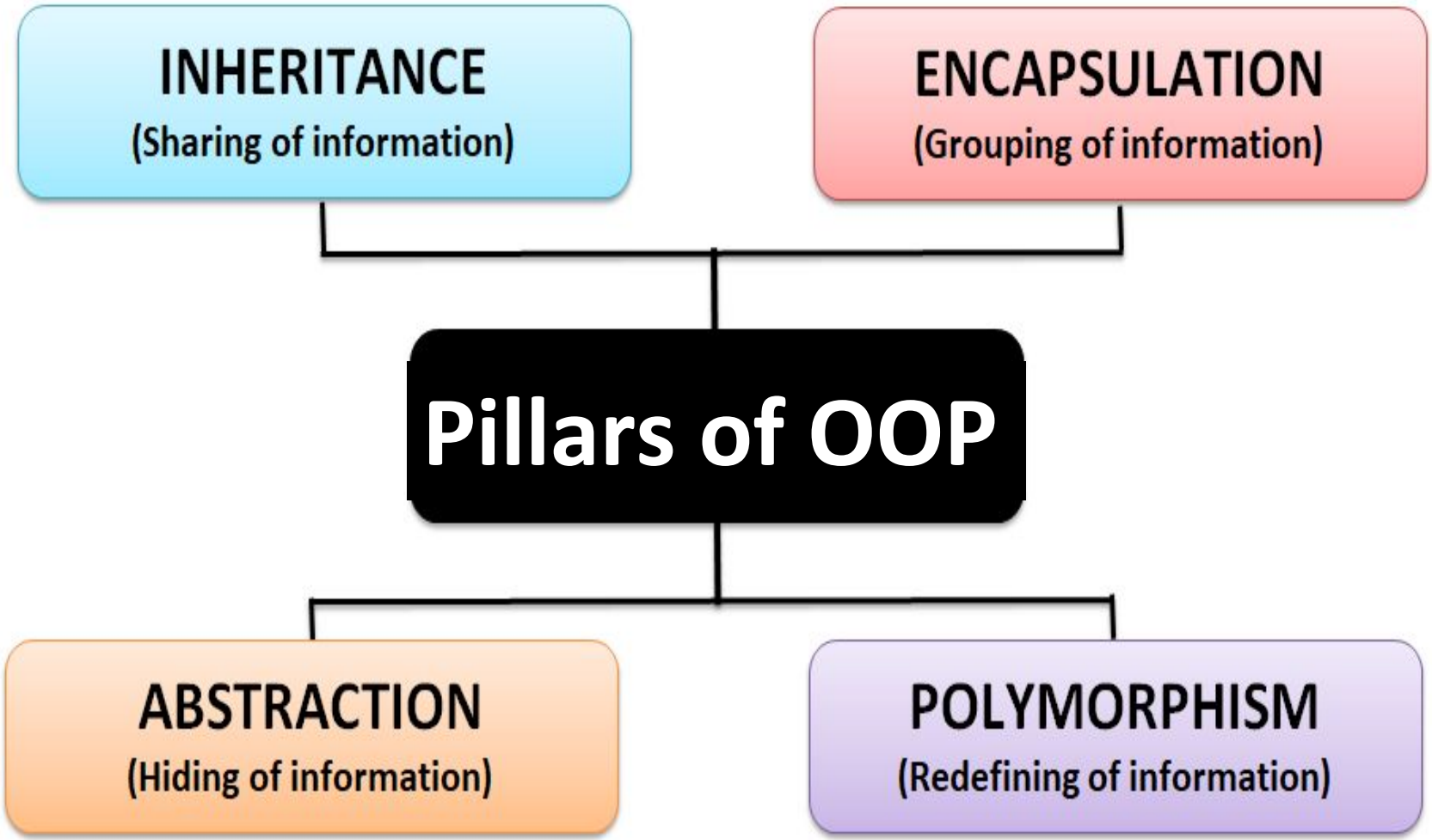
For Example: Consider the Class of Cars.

There may be many cars with different names and brands. All of them will have 4 wheels, Speed Limit, Mileage range, etc.

So here, Car is the class, and wheels, speed limits, mileage are their properties.

A method is a collection of statements that perform some specific task and return the result to the caller.




INHERITANCE
(Sharing of information)
ENCAPSULATION
(Grouping of information)
Pillars of OOP
ABSTRACTION
(Hiding of information)
POLYMORPHISM
(Redefining of information)

Abstraction

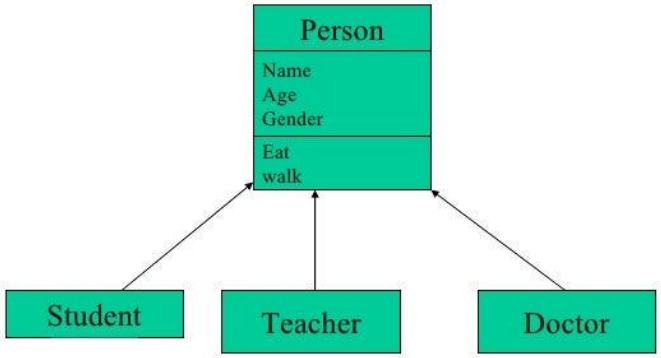
Abstraction is the method of hiding the implementation details and showing only the functionality.

Consider a real-life example:

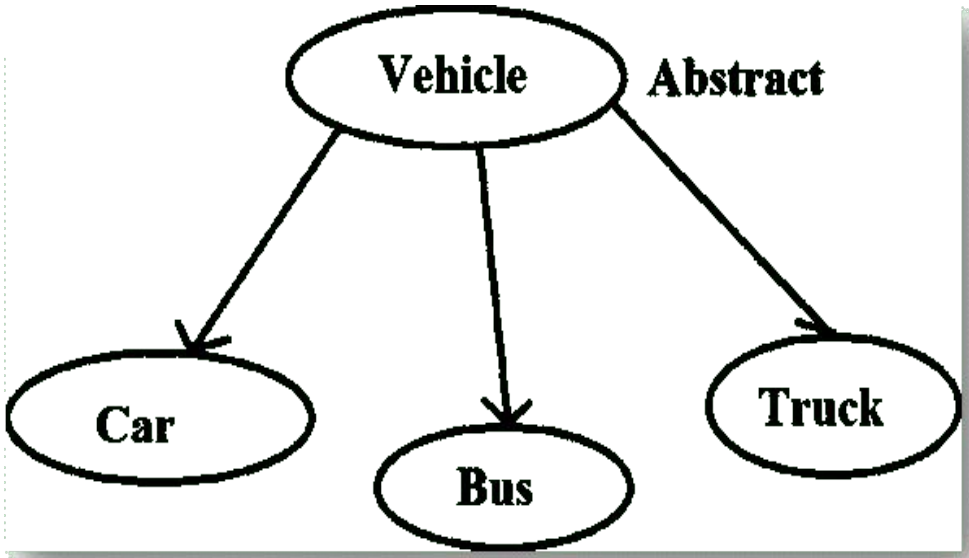
You know how car look like and how to drive but you don't know how to build a car.

In Java, abstraction is achieved by interfaces and abstract classes.

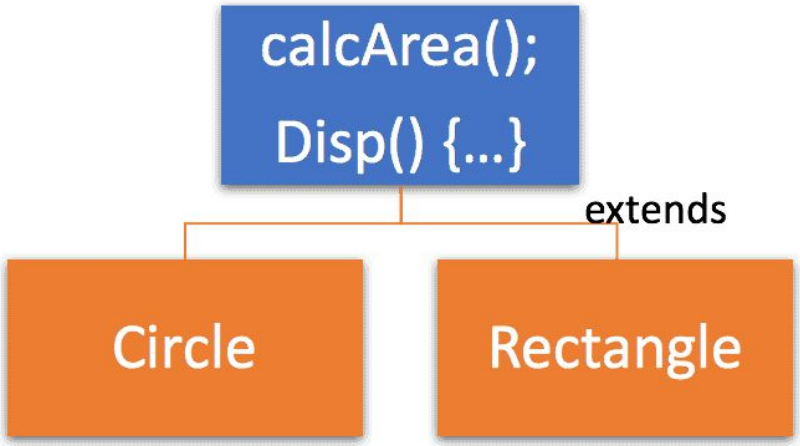
Example – Abstract Classes



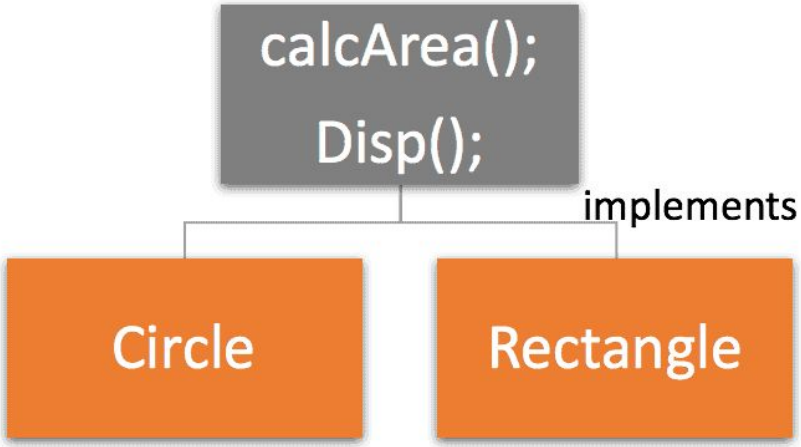
- Here, Person is an abstract class



Abstract

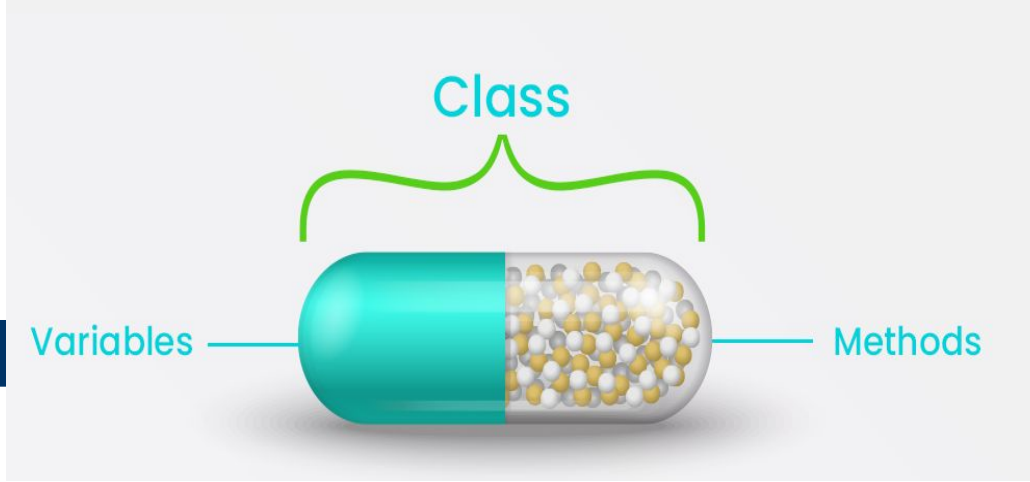


Interface



Parameters	Abstract Class	Interface
1. Keyword Used	abstract	interface
2. Type of Variable	Static and Non-static	Static
3. Access Modifiers	All access modifiers	Only public access modifier
4. Speed	Fast	Slow
5. When to use	To avoid Independence	For Future Enhancement

Encapsulation



- Encapsulation means containing all important information inside an object, and only exposing selected information to the outside world.
- Data in a class is hidden from other classes, which is similar to what **data-hiding** does.
- For example, you have an attribute that is not visible from the outside of an object. You bundle it with methods that provide read or write access. Encapsulation allows you to hide specific information and control access to the internal state of the object.

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 also from outside the class.

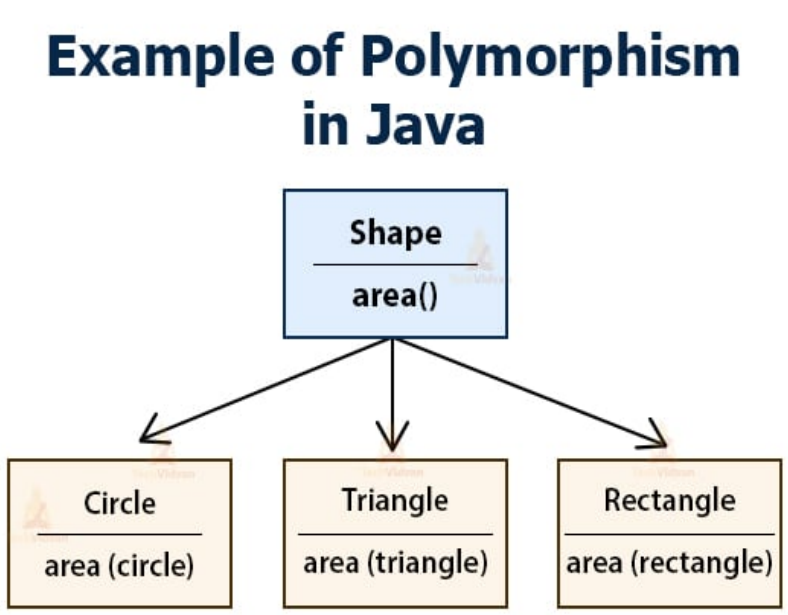
Inheritance

Classes can have children; that is, one class can be created out of another class. The capability of a class to derive properties and characteristics from another class is called Inheritance.

- **Superclass:** The class whose features are inherited is known as superclass (also known as base or parent class).
- **Subclass:** The class that inherits the other class is known as subclass (also known as derived or extended or child class). The subclass can add its own fields and methods in addition to the superclass fields and methods.
- **Reusability:** We can derive our new class from the existing class. By doing this, we are reusing the fields and methods of the existing class.

Polymorphism

- The word polymorphism means having many forms. In simple words, we can define polymorphism as the ability of a message to be displayed in more than one form.
- 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 Overloading and method overriding



For example, A person at the same time can have different characteristics. Like a man at the same time is a father, a husband, an employee.
So the same person posses different behavior in different situations. This is called polymorphism.

Benefits of OOP