Object Oriented Programming

—

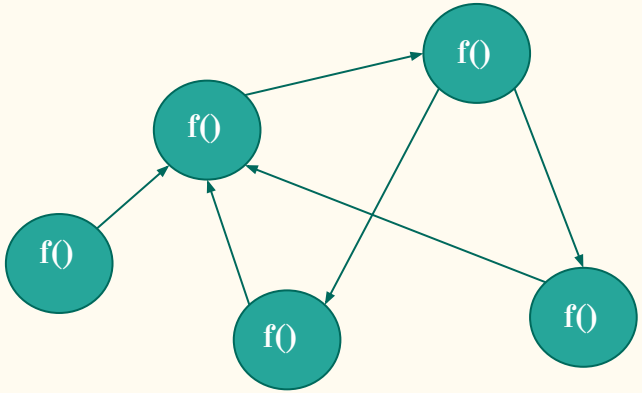
Introduction

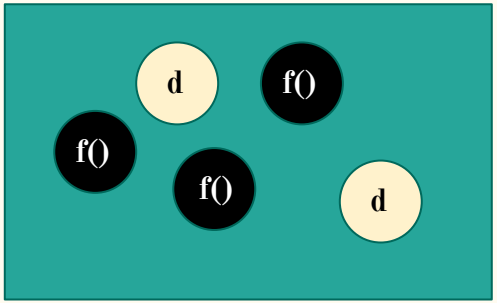
Introduction

Procedural model

Ccontains procedures that perform on data.

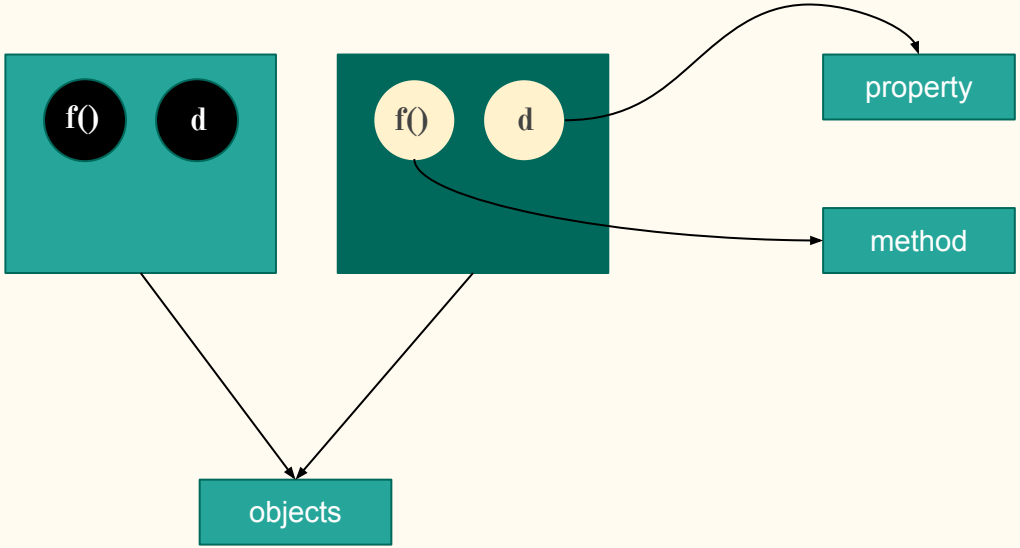
Here functions are interdependent of each other.





OOP model

OOP stands for Object Oriented Programming. It is all about creating objects that content both data and objects.



Advantages

Why OOP model is better than procedural model ?

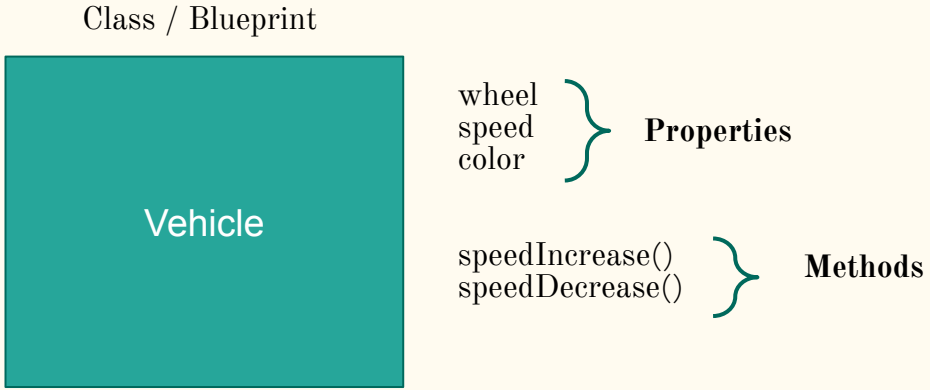
- We can not give data security in Procedural model.
- Procedural model difficult to maintain for large projects
- Procedural model is difficult to scale
- Creating microservices using procedural model is hard

Beside these

- OOP model is flexible
- Reuse of code
- Effective

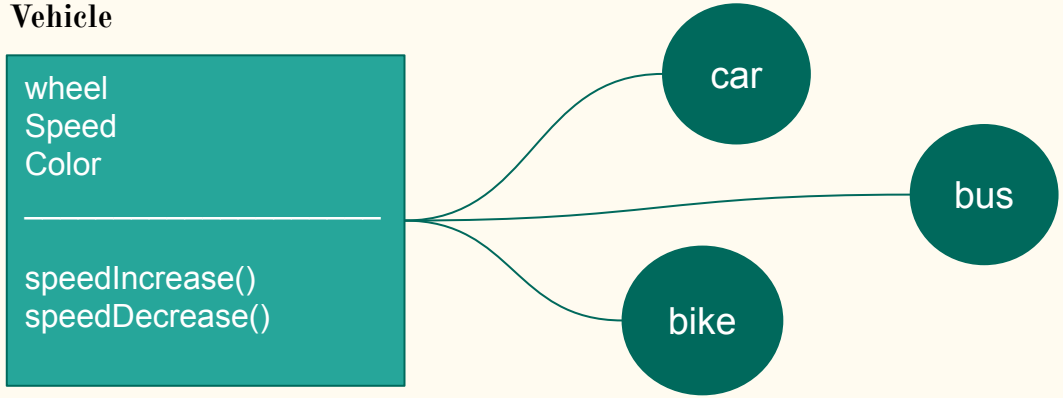
Class

Class: It is a **blueprint** for creating objects, providing initial values for state (member variables or attributes), and implementations of behavior (member functions or methods).



Object

Object is an **instance of a class**. An object in OOPS is nothing but a self-contained component which consists of methods and properties to make a particular type of data useful. For example color name, table, bag, barking. When you send a message to an object, you are asking the object to invoke or execute one of its methods as defined in the class.



Here, **car**, **bus**, **bike** are instances or objects of vehicle class

Define a class

A class is defined by using the **class** keyword, followed by the name of the class and a pair of curly braces ({}). All its properties and methods go inside the braces:

```php
<?php
class Vehicle {
  // Properties
  public $speed = 0;
  public $wheel = 4;

  // Methods
  function speedInc($val) {
    $this->speed += $val;
  }
  function speedDec($val) {
    return $this->val;
  }
}
?>
```

Create objects

We can create multiple objects from a class. Each object has all the properties and methods defined in the class, but they will have different property values.

Objects of a class is created using the **new** keyword.

```php
<?php
  $car = new Vehicle();
  $bus = new Vehicle();
  $bike = new Vehicle();
?>
```

```php
<?php
  $car = new Vehicle();
  echo car.wheel    // 4
?>
```

Ref: Read More From Here

this keyword

In PHP, **$this** keyword references the current **object** of the class. The **$this** keyword allows you to access the properties and methods of the current object within the class using the object operator (->):

```
$this->property
$this->methods()
```

The **$this** keyword is only available within a class. It doesn't exist outside of the class. If you attempt to use the **$this** outside of a class, you'll get an error.

When you access an object property using the **$this** keyword, you use the **$** with the this keyword only. And you don't use the **$** with the property name. For example:

```
$this->speed
```

Ref : Read More From Here