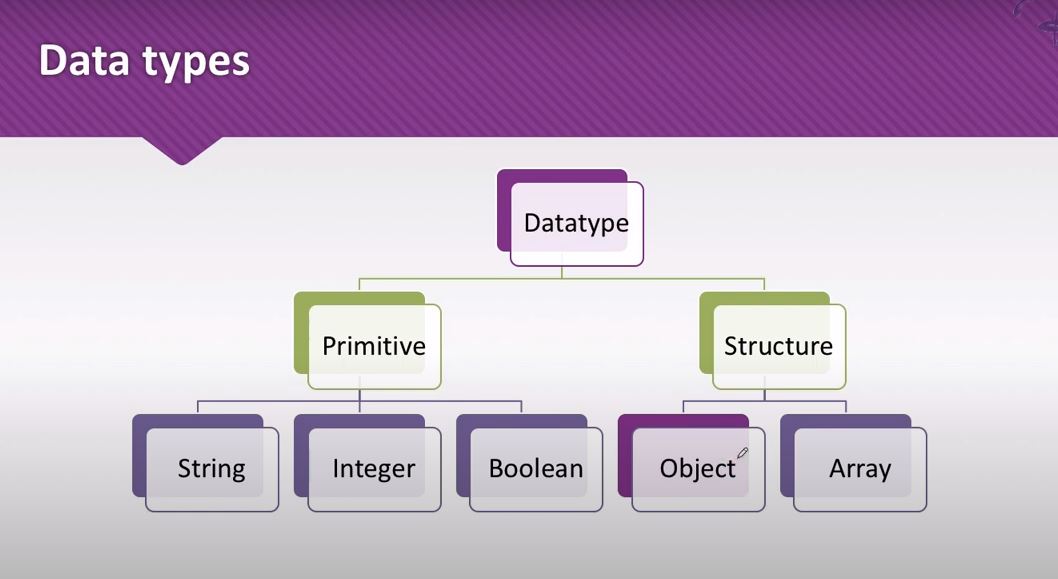
**OOP PHP**

**How this object came from?**

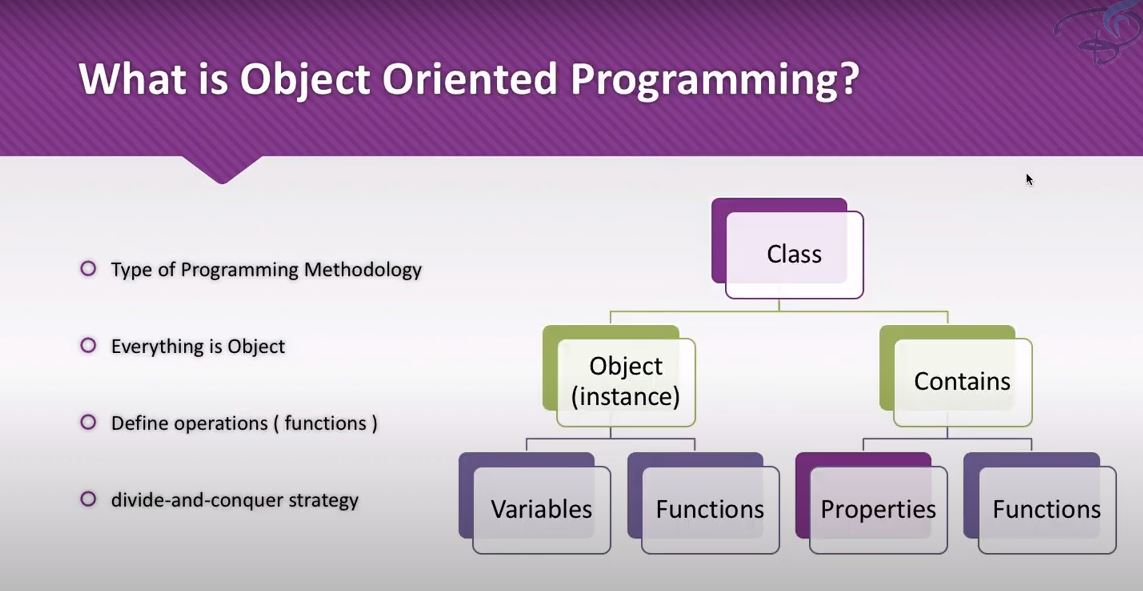


**What is object?**



Like an airplane. It has some behaviour have some attributes and have some data which define the airplane like size of it, speed, company etc. Behaviour is to fly, take passenger to one place to another place it’s like functions.

**What is object-oriented programming?**

****

As we have seen anything can be object. Likewise in programming world anything can be created as an object and this object-oriented programming is actually is a methodology.

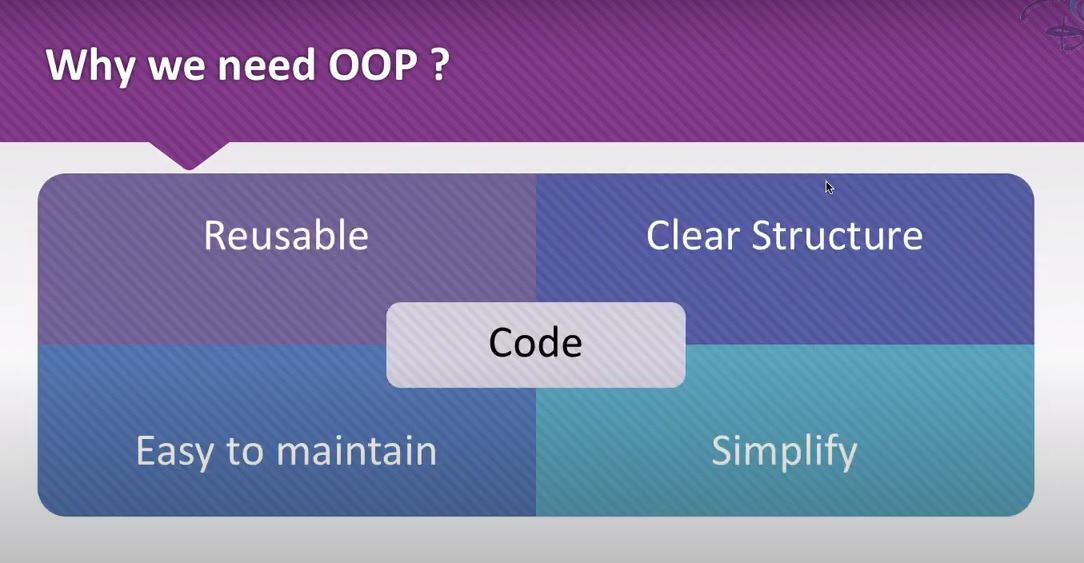
And what is methodology?

* Methodology is like style of creating your code or managing you code.

How this is work actually?

Object-oriented programming to divide your big problem or big project into small jumps of the classes. A class is just collection of the attributes and function that it can have or blue prient . If you have taken an instance or you can simply say a copy of the class then it will called and object

**Why we need oop?**

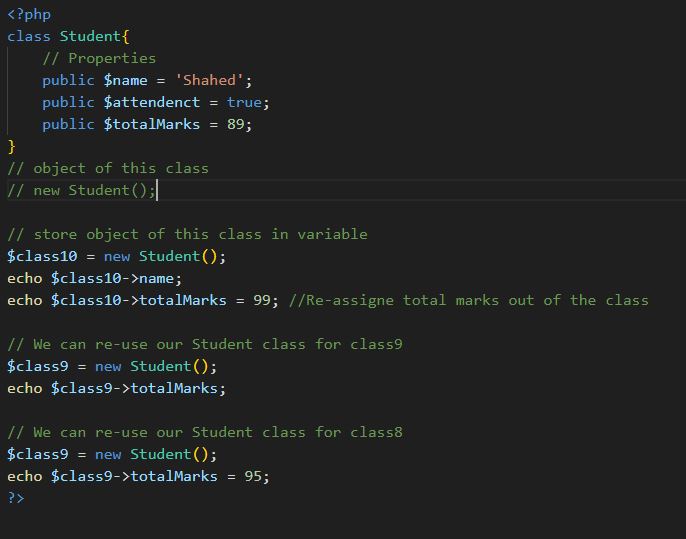


If you are using a object-oriented programming system then your code can be reuseable, you codes will have a clear structure which means you can manage those codes very very efficiently the code becomes simple to read to write and simple to manage and easy to maintion

**Why do we need to create the class?**

Think about your project as a school. School has various classes so think about why schools need classes because for a particular class syllabus is different actually if you say this for every class the work they are going to do is different obviously if the first class is doing something then that second class is not going to do that. so if you are in a different class then every class has a different syllabus we are going to create a class and what is the benefit if you are dealing with a certain functionality that belongs to a certain class then you just have to go to that class you don’t have to search in your whole school or your project so we need the class to divide our big project into small terms of the smaller problems so everyone knows that a smaller problem is easy to solve. So to solve a big problem they will chunk out our project into smaller classes okay? so come back to the school every class has some students which means every student has unique characteristics so in PHP or OOP every class has some functions like every school class has some work to do and some syllabus to study okay so here are object-oriented classes can have many functions in our programming class we can have various attributes of that class

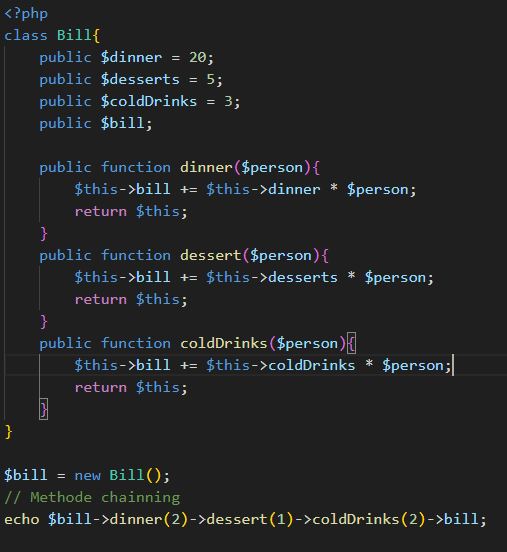
**How we can create the object for a class and how we can define the properties of a class?**



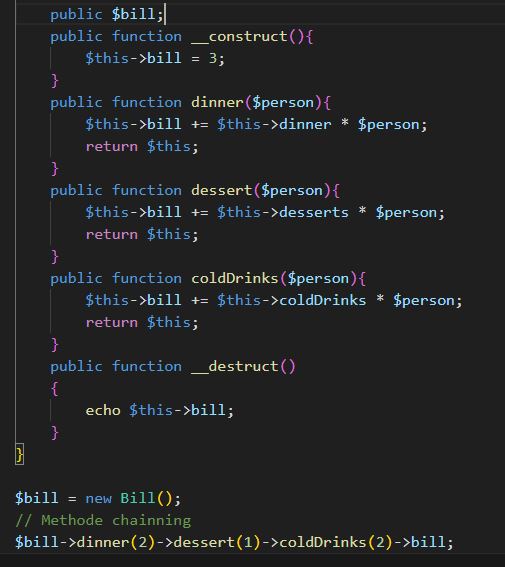
**Functions and Methodes:**



**Methode Chaining:-** Method chaining is a technique in PHP that allows the concatenation of multiple methods to increase the readability of code and avoid putting all the code in a single function

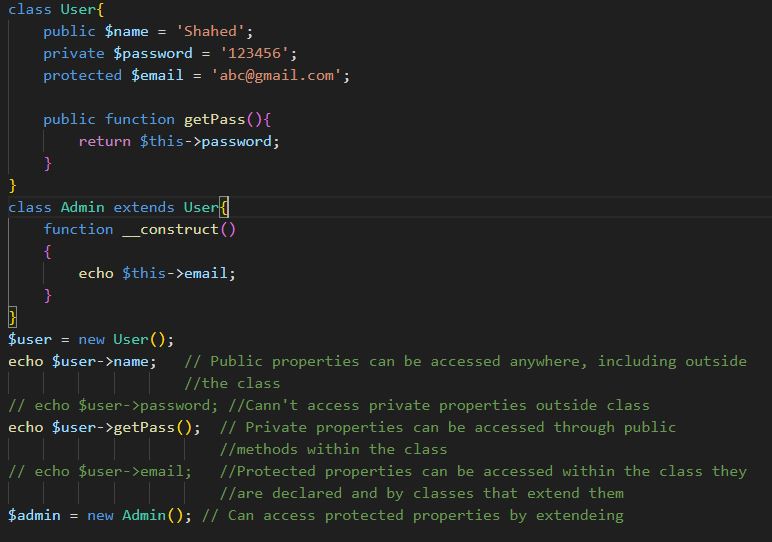


**\_\_construct and \_\_destruct methodes:- Those** method is automatically called when an object is created or destroyed. This function always starts with two underscores. \_\_construct() called when object is created and \_\_destruct() called when object is destroyed.

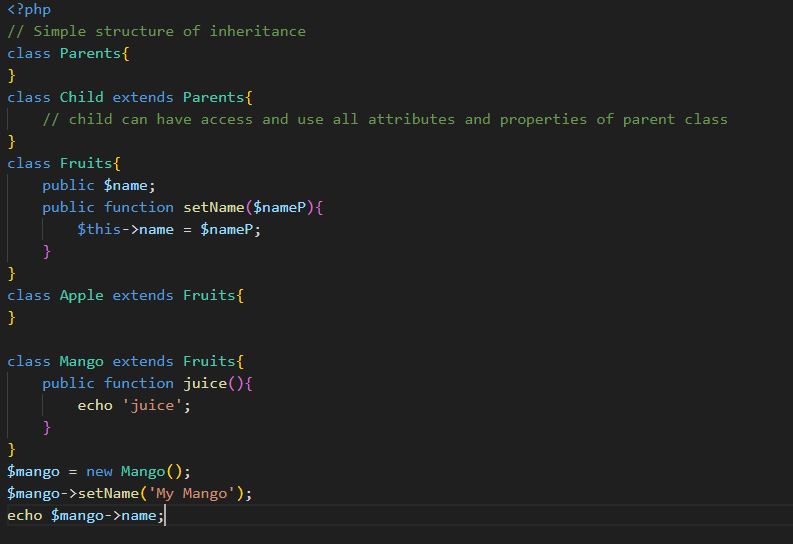


**Access Modifiers: -**

1. Public properties can be accessed anywhere, even outside the class.
2. Private properties can only be accessed within the class they are declared.
3. However, private properties can be accessed through public methods within the class.
4. Protected properties can be accessed within the class they are declared and by classes that extend them.



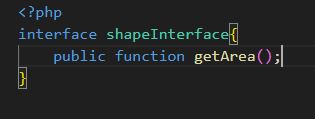
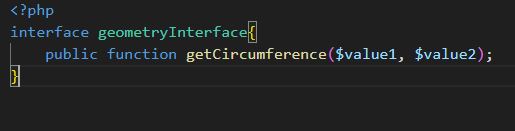
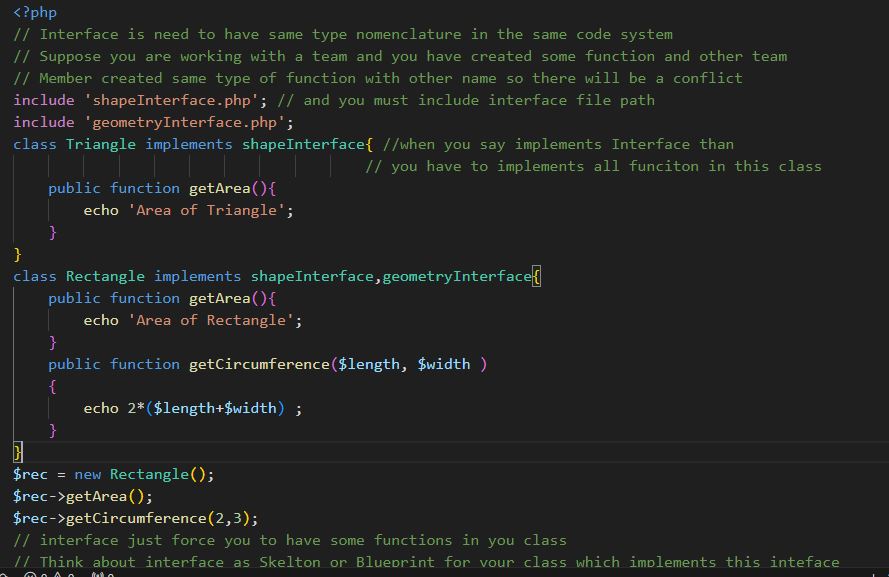
**Inheritance:-**   
In PHP, inheritance is a key principle of object-oriented programming. Here’s a simple example to illustrate this concept:  
  
In this code, Child is a subclass of Parents, and Apple and Mango are subclasses of Fruits. The subclasses inherit all the public and protected properties and methods from their parent class. This allows for code reusability and efficiency.



**Interface:**

Interfaces are like a blueprint or skeleton for your classes. They enforce a certain structure, ensuring that all classes implementing the interface have the same type of methods. This is incredibly useful when working in a team, as it prevents conflicts due to different naming conventions. 🤝

Remember, interfaces just force you to have some functions in your class. They're like a blueprint guiding you to build your classes in a certain way. And in the world of programming, a little guidance can go a long way! 🚀



**Abstruct methode:-**

Imagine you’re building a toy car 🚗. You have a blueprint that tells you what parts the car needs (like wheels, body, and engine), but it doesn’t tell you the specifics (like the color of the car, the size of the wheels, or the type of engine). This blueprint is like an abstract class in PHP.

Now, you decide to build a specific type of car, say a race car 🏎️. You know it needs all the parts from the blueprint, but you also add specifics like red color, big wheels, and a powerful engine. This is like a class that extends the abstract class. It follows the blueprint but adds its details.

In PHP, this blueprint is an abstract class with methods (the parts). Some methods are fully defined (concrete methods), and some are just named without a definition (abstract methods). When another class extends this abstract class, it must provide the details for any abstract methods.

So, when you’re coding in a team, abstract classes help ensure everyone knows what parts are needed, even if they add their own details. It’s all about teamwork and consistency! 🚀

