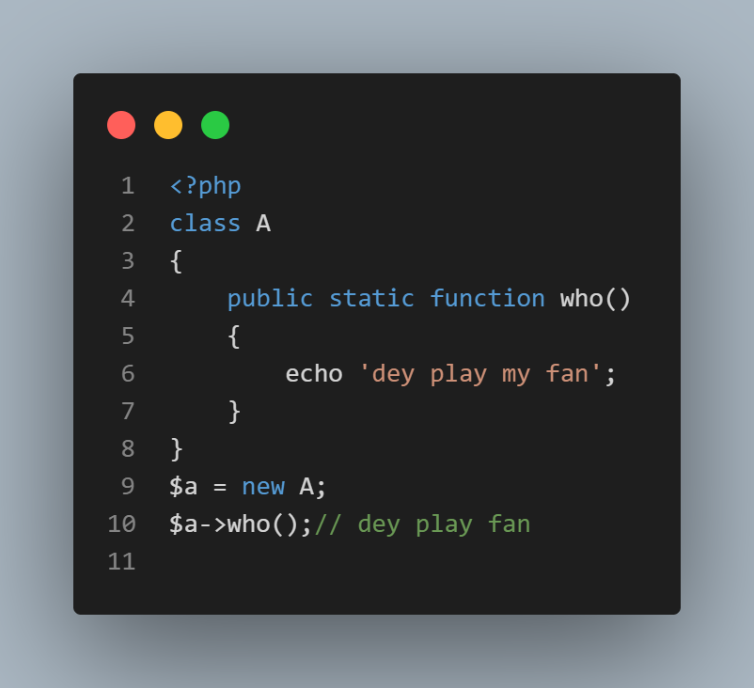
Late Static Bindings

In PHP, when a class is declared, the simplest way to instantiate that class and call a function within it is by creating an object of that class using the new keyword, followed by the class name, and then calling the function using the object's instance. For example:



Before going into Late static bindings, lets dive into 2 concepts for a better comprehension…..

* Early Binding
* Late Binding……

Early Binding

In early binding, the code uses self keyword which ensures that the method and property calls are resolved at compile time based on the class where they are defined rather than at runtime based on the class of the object.



Output

The self keyword drives the method from where it is defined and not the instantiated class. Therefore, it compiled it before it run the code

Late Binding

For late binding, we use the static keyword to achieve that. Example.



Output

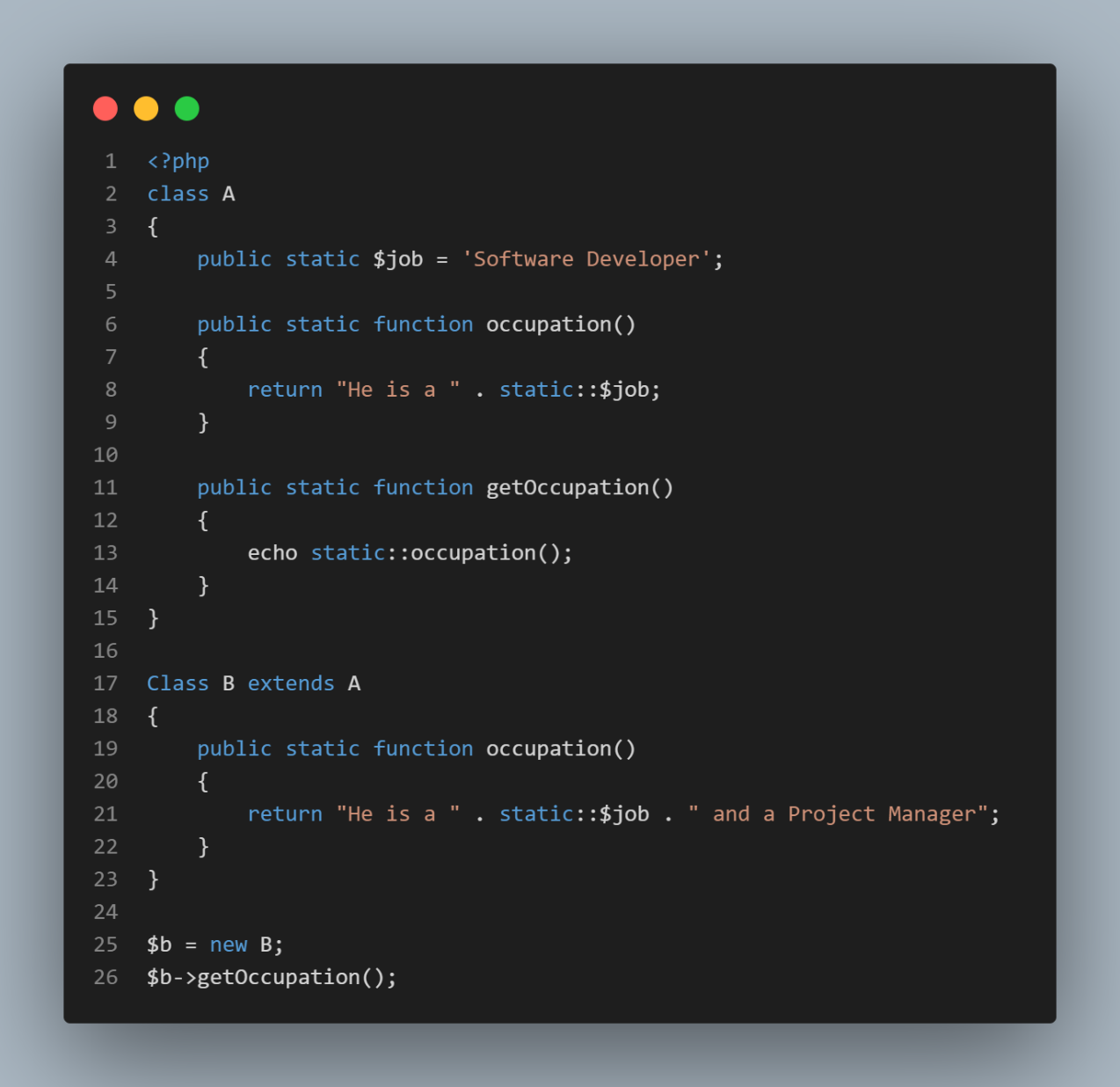
He is a software Developer and a Project Manager

In this case, the **getOccupation()** method in class **A** calls the **occupation()** method using the **static::occupation()** syntax, which ensures that the method call is resolved at runtime based on the actual class of the object, rather than the class where the method is defined.

We have the object of class **B** and it took the method from class **B** which is based on the class executing the code

Late Static Binding

In late static binding, all properties and method uses the static keyword



Output

He is a Software Developer and a Project Manager