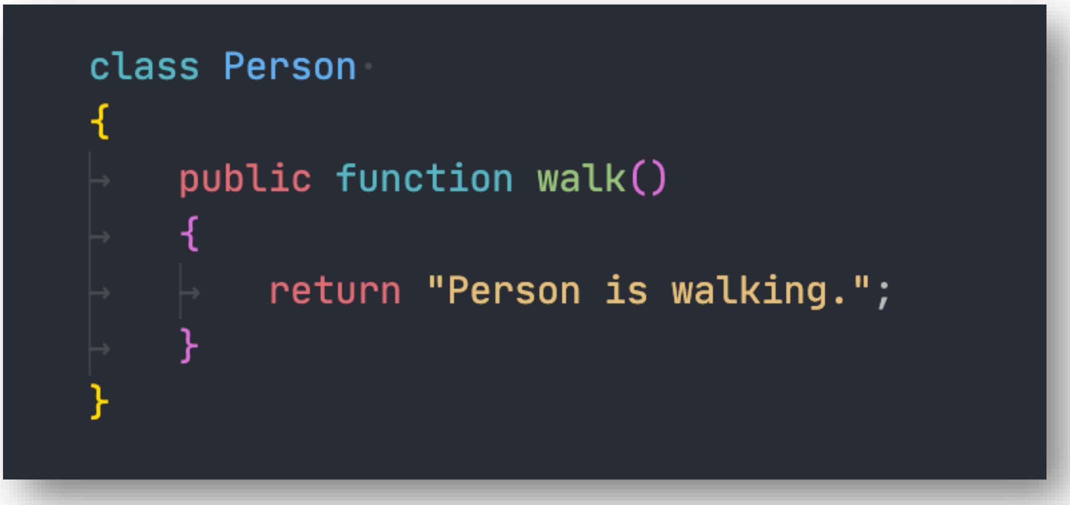
Seakmeng Chheang

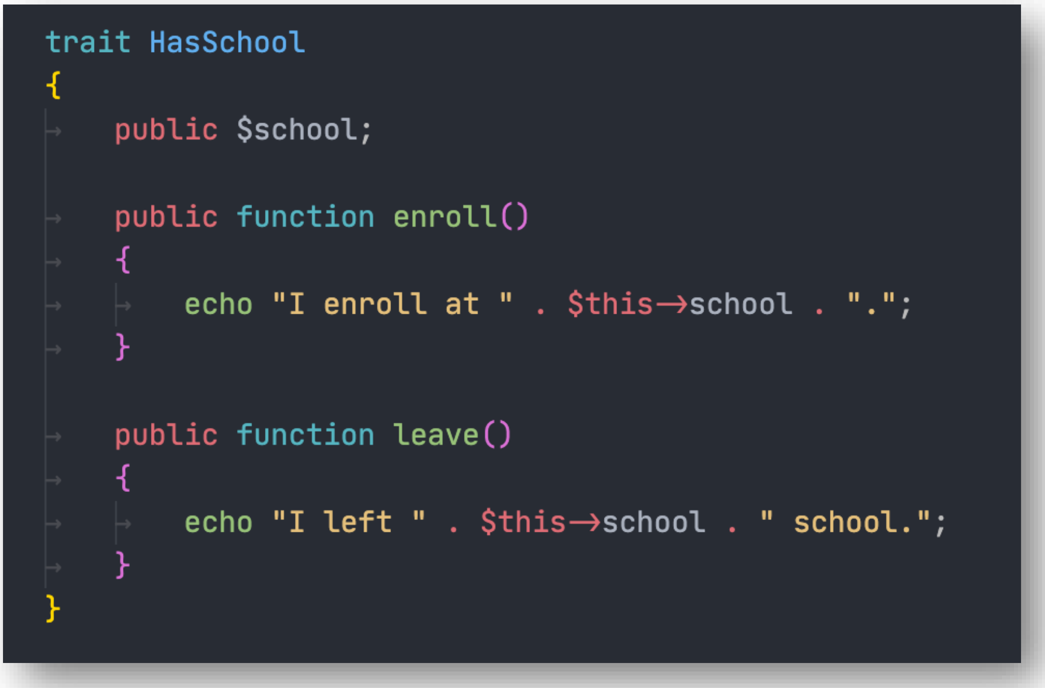
Exercise 1:
Explain the advantage and disadvantage of multiple inheritance. Write a PHP program to demonstrate that.

- **Advantages:**
    1. Allows a class to inherit the functionality of more than one base class thus allowing for modeling of complex relationships.
    2. Reuses code a lot without having to copy-and-paste similar code to implement interfaces.
    3. Creates more readability of the code



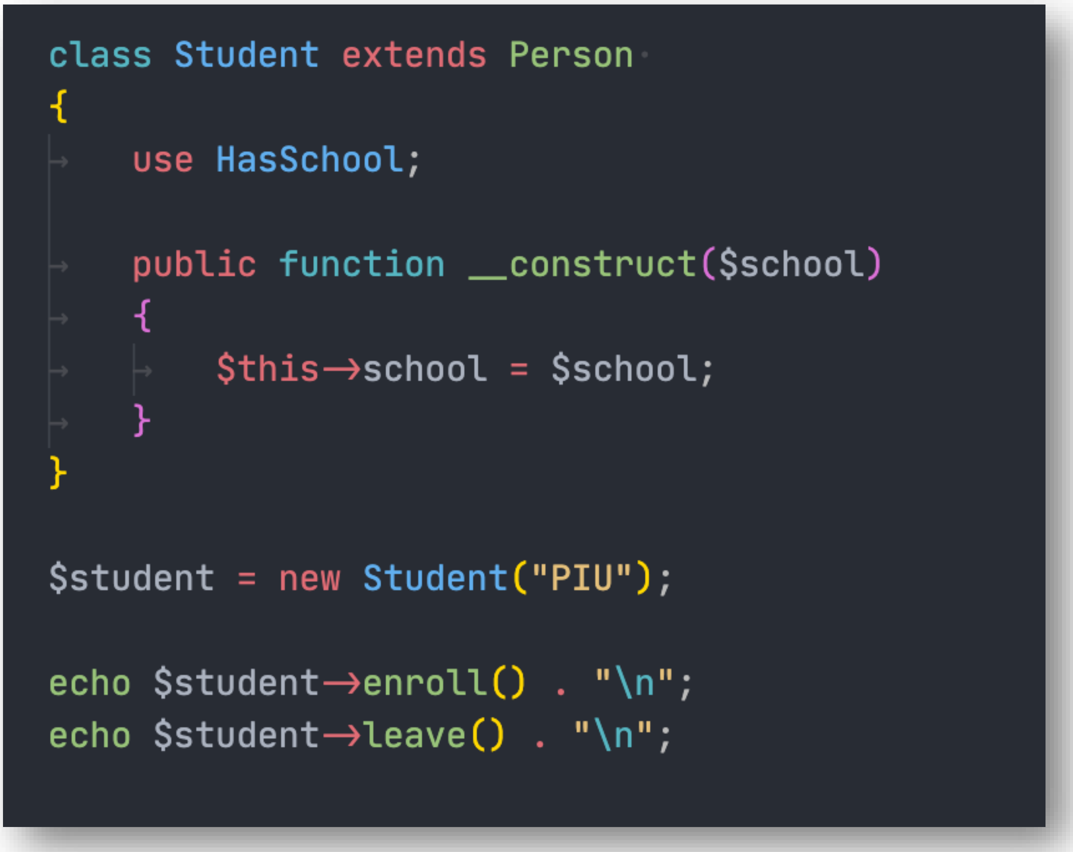
```php
class Person
{
    public function walk()
    {
        return "Person is walking.";
    }
}
```

Figure 1: Person Class



```php
trait HasSchool
{
    public $school;

    public function enroll()
    {
        echo "I enroll at " . $this->school . ".";
    }

    public function leave()
    {
        echo "I left " . $this->school . " school.";
    }
}
```

Figure 2: HasSchool Trait



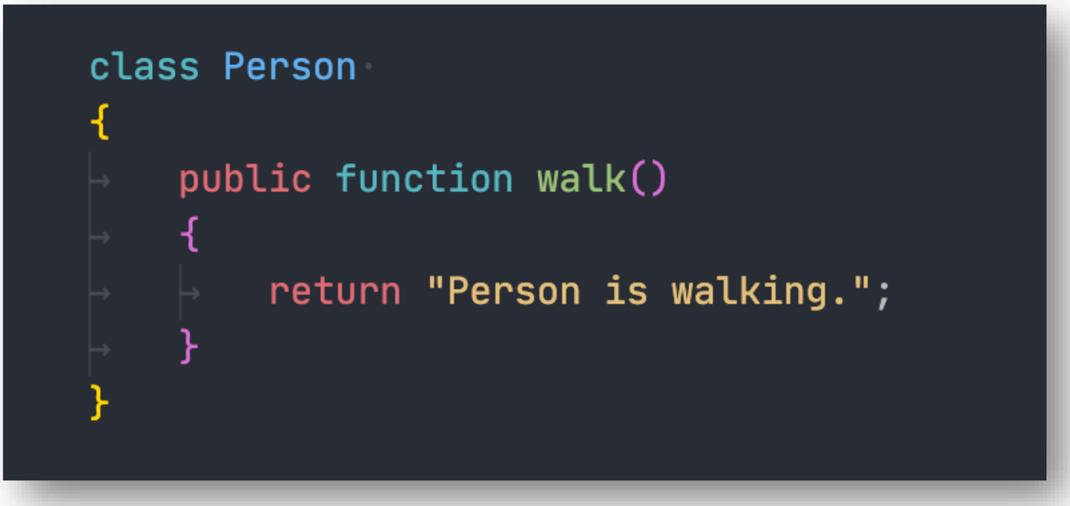
```php
class Student extends Person
{
    use HasSchool;

    public function __construct($school)
    {
        $this->school = $school;
    }
}

$student = new Student("PIU");

echo $student->enroll() . "\n";
echo $student->leave() . "\n";
```

Figure 3: Student Class which inherits from Person and HasSchool

⇨ We see that we can have the flexibility of the Student class without paying off the readability of the code. Take a glance at the code, we immediately know that Student class has the nature of a person and has the school to enroll also. Furthermore, HasSchool trait can also be reused with for example Instructor class or even Academic Staffs.

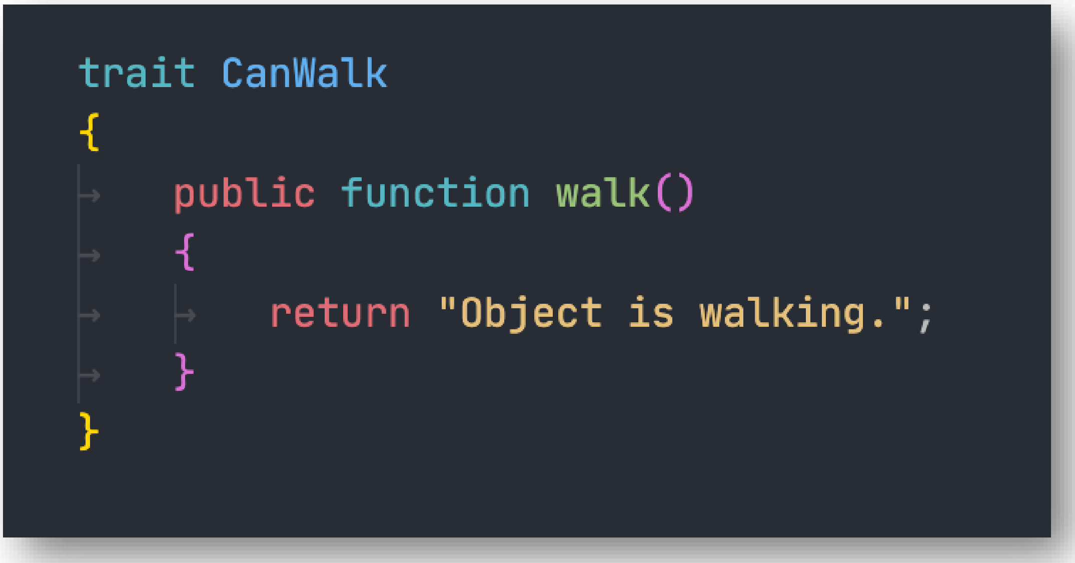
❖ **Disadvantages:**

1. Can lead to a lot of confusion when two base classes implement a method with the same name.
2. If one of the super classes or base classes happens to change, the subclass may have to change as well.
3. Causes Diamond problem. For example, class Student inherits from class Person and Trait CanWalk. However, class Person and Trait CanWalk has one same method. So, the question arises which method from which class will be called from class C?



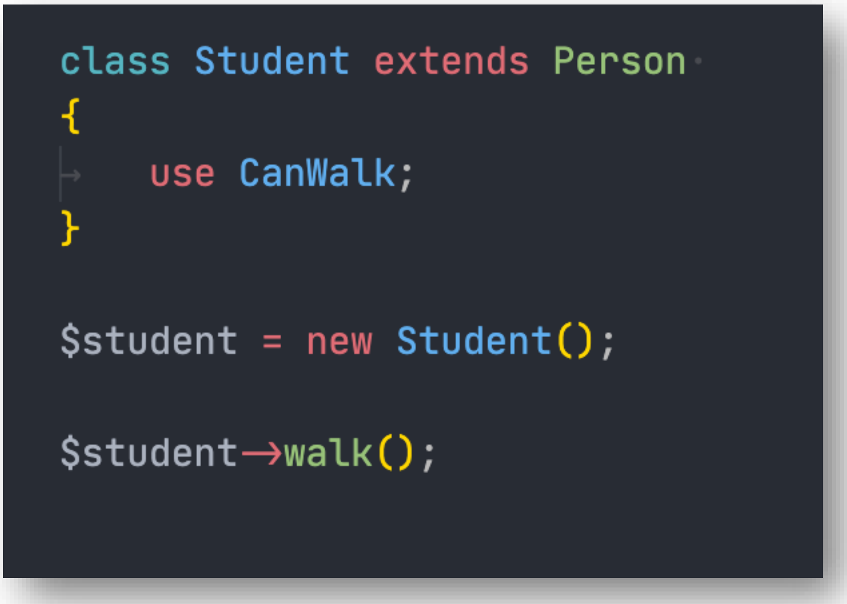
```
class Person
{
    public function walk()
    {
        return "Person is walking.";
    }
}
```

Figure 4: Person Class



```
trait CanWalk
{
    public function walk()
    {
        return "Object is walking.";
    }
}
```

Figure 5: CanWalk Trait



```php
class Student extends Person
{
    use CanWalk;
}

$student = new Student();

$student->walk();
```

Figure 6: Student Class inherits from Person and CanWalk

Guess what? Your co-workers will have to execute this program just to know which walk method that student calls. Imagine, they forgot it and have to do this every time, Ahhhh!!!!