

```
<?php
```

```
// ABSTRACTION + ENCAPSULATION
```

```
abstract class Vehicle {
```

```
    protected $brand;
```

```
    protected $speed;
```

```
    // Constructor with Encapsulation
```

```
    public function __construct($brand, $speed) {
```

```
        $this->setBrand($brand);
```

```
        $this->setSpeed($speed);
```

```
    }
```

```
    // Encapsulation: Getters and Setters
```

```
    public function getBrand() {
```

```
        return $this->brand;
```

```
    }
```

```
    public function setBrand($brand) {
```

```
        if (!empty($brand)) {
```

```
            $this->brand = $brand;
```

```
        }
```

```
    }
```

```
    public function getSpeed() {
```

```
        return $this->speed;
```

```
}
```

```
public function setSpeed($speed) {
```

```
    if ($speed >= 0) {
```

```
        $this->speed = $speed;
```

```
    }
```

```
}
```

```
// Abstract Method (Abstraction)
```

```
abstract public function drive();
```

```
}
```

```
// INHERITANCE + POLYMORPHISM
```

```
class Car extends Vehicle {
```

```
    public function drive() {
```

```
        return $this->brand . " car is driving at " . $this->speed . " km/h.";
```

```
    }
```

```
}
```

```
class Motorcycle extends Vehicle {
```

```
    public function drive() {
```

```
        return $this->brand . " motorcycle is zooming at " . $this->speed . " km/h.";
```

```
    }
```

```
}
```

```
// POLYMORPHISM in action
```

```
function testDrive(Vehicle $vehicle) {  
    echo $vehicle->drive() . "\n";  
}
```

```
// Create objects
```

```
$car = new Car("Toyota", 120);
```

```
$bike = new Motorcycle("Yamaha", 90);
```

```
// Test polymorphic behavior
```

```
testDrive($car); // Output: Toyota car is driving at 120 km/h.
```

```
testDrive($bike); // Output: Yamaha motorcycle is zooming at 90 km/h.
```

```
?>
```