

# PHP - Anonymous Classes

The release of version 7.0 is an important milestone in the evolution of PHP language, when a lot of new features were introduced. The feature of Anonymous class was also made available in PHP version 7.0.

As the term "anonymous" suggests, it is a class without a (programmer declared) name. The usual practice is to define a class with a certain identifier, so that it can be used repeatedly. The anonymous class, on the other hand is for one-time use only.

```
$obj = new class() {  
    /* class body */  
};
```

Apart from this class not having a name, it is similar to a normal named class, in the sense it can contain properties and methods. Its functionality is no different from that of an object of a named class.

An anonymous class might be used over a named class especially when the class does not need to be documented, and when the class is used only once during execution. Anonymous classes are useful when simple, one-off objects need to be created.

## Example

In the following code, an anonymous class is instantiated and stored in `$obj` object. The class includes definitions of `addition()` and `division()` methods, which are called with the `$obj` object.

&lt;/&gt;

Open Compiler

```
<?php  
$obj = new class(10) {  
    private int $x;  
    function __construct($x) {  
        $this->x = $x;  
    }  
  
    public function addition($x) {  
        return $this->x+$x;  
    }  
    public function division($x) {
```

```
        return $this->x/$x;
    }
};

echo "Addition: " . $obj->addition(20) . PHP_EOL;
echo "Division: " . $obj->division(20) . PHP_EOL;
?>
```

It will produce the following **output** –

```
Addition: 30
Division: 0.5
```

## Anonymous Class as a Child Class

An anonymous class can do everything that a normal class can. It can extend another class, implement an interface or even use a trait.

### Example

In the example below, the anonymous class is a child class, extending a parent already available.

&lt;/&gt;

Open Compiler

```
<?php
class myclass {
    public function hello() {
        echo "Hello World!" . PHP_EOL;
    }
}

$obj = new class("Neena") extends myclass {
    private string $nm;
    function __construct($x) {
        $this->nm = $x;
    }
    public function greeting() {
        parent::hello();
        echo "Welcome " . $this->nm . PHP_EOL;
    }
};
```

```
$obj->greeting();  
?>
```

It will produce the following **output** –

```
Hello World!  
Welcome Neena
```

## Example

Although the anonymous class doesn't have any user defined name, PHP does assign it an internal name, which can be obtained with the built-in `get_class()` function as follows –

[Open Compiler](#)

```
<?php  
$obj = new class() {  
    function greeting() {  
        echo "Hello World" . PHP_EOL;  
    }  
};  
$obj->greeting();  
echo "Name of class: " . get_class($obj);  
?>
```

It will produce the following **output** –

```
Hello World  
Name of class: class@anonymousC:\xampp\htdocs\hello.php:2$0
```

PHP parser assigns the internal name randomly.