How to create classes?

• class

• object

• method

• property

```
class Car {
        // The code
}
```

• We declare the class with the class keyword.

• We write the name of the class and capitalize the first letter.

• If the class name contains more than one word, we capitalize each word. This is known as upper camel case. For example, JapaneseCars, AmericanIdol, EuropeTour, etc.

• We circle the class body within curly braces. We put our code within the curly braces

How to add properties to a class?

We call properties to the variables inside a class. Properties can accept values like strings, integers, and booleans (true/false values), like any other variable. Let's add some properties to the Car class.

```
class Car {
        public $comp;
        public $color = "beige";
        public $hasSunRoof = true;
}
```

• We put the public keyword in front of a class property.

• The naming convention is to start the property name with a lower case letter.

• If the name contains more than one word, all of the words, except for the first word, start with an upper case letter. For example, $color or $hasSunRoof.

• A property can have a default value. For example, $color = 'beige'.

• We can also create a property without a default value. See the property $comp in the above example

How to create objects from a class?

We can create several objects from the same class, with each object having its own set

of properties.

In order to work with a class, we need to create an object from it. In order to create an object, we use the new keyword. For example:

$bmw = new Car ();

• We created the object $bmw from the class Car with the new key word.

• The process of creating an object is also known as instantiation.

We can create more than one object from the same class.

$bmw = new Car ();

$mercedes = new Car ();

How to get an object's properties?

echo $bmw -> color;

echo $mercedes -> color;

How to set an object property?

In order to set an object property, we use a similar approach.

For example, in order to set the color to 'blue' in the bmw object:

```
$bmw -> color = 'blue';
```

and in order to set the value of the $comp property for both objects:

```
$bmw -> comp = "BMW";

$mercedes -> comp = "Mercedes Benz";
```

Once we set the value of a property, we can get its value.

In order to get the color of the $bmw, we use the following line of code:

```
echo $bmw -> color;
```

Result:

Blue

We can also get the company name and the color of the second car object.

```
echo $mercedes -> color;

echo $mercedes -> comp;
```

Result:

beige

Mercedes Benz

How to add methods to a class?

The classes most often contain functions. A function inside a class is called a method. Here we add the method hello() to the class with the prefix public.

```
class Car {

public $comp;

public $color = "beige";

public $hasSunRoof = true;

public function hello()

{
```

```
return "beep";

}

}
```

• We put the public keyword in front of a method.

• The naming convention is to start the function name with a lower case letter.

• If the name contains more than one word, all of the words, except for the first word, start with an upper case letter. For example, helloUser() or flyPanAm().

```
$car1 = new Car ();

$car2 = new Car ();

echo $car1 -> hello();

echo $car2 -> hello();
```

Result:

```
beep

beep
```

Here is the full code

```
<?php

// Declare the class

class Car {

// Properties

public $comp;

public $color = "beige";

public $hasSunRoof = true;

// Method that says hello

public function hello()
```

```
{
return "beep";
}
}
// Create an instance
$bmw = new Car ();
$mercedes = new Car ();
// Get the values
echo $bmw -> color; // beige
echo "<br />";
echo $mercedes -> color; // beige
echo "<hr />";
// Set the values
$bmw -> color = "blue";
$bmw -> comp = "BMW";
$mercedes -> comp = "Mercedes Benz";
// Get the values again
echo $bmw -> color; // blue
echo "<br />";
echo $mercedes -> color; // beige
echo "<br />";
echo $bmw -> comp; // BMW
echo "<br />";
echo $mercedes -> comp; // Mercedes Benz
echo "<hr />";
// Use the methods to get a beep
```

```
echo $bmw -> hello(); // beep
echo "<br />";
echo $mercedes -> hello(); // beep
```

The $this keyword

The $this keyword indicates that we use the class's own methods and properties, and allows us to have access to them within the class's scope

```
$this -> propertyName;
$this -> methodName();
```

• Only the this keyword starts with the $ sign, while the names of the properties and methods do not start with it.

This is the full code:

```
class Car {
// The properties
public $comp;
public $color = "beige";
public $hasSunRoof = true;
// The method that says hello
public function hello()
{
return "Beep I am a <i>" . $this -> comp .
"</i>, and I am <i>" . $this -> color ;
}
```

```
}
// We can now create an object from the class
$bmw = new Car();
$mercedes = new Car();
// Set the values of the class properties
$bmw -> color = "blue";
$bmw -> comp = "BMW";
$mercedes -> comp = "Mercedes Benz";
// Call the hello method for the the $bmw object
echo $bmw -> hello();
```