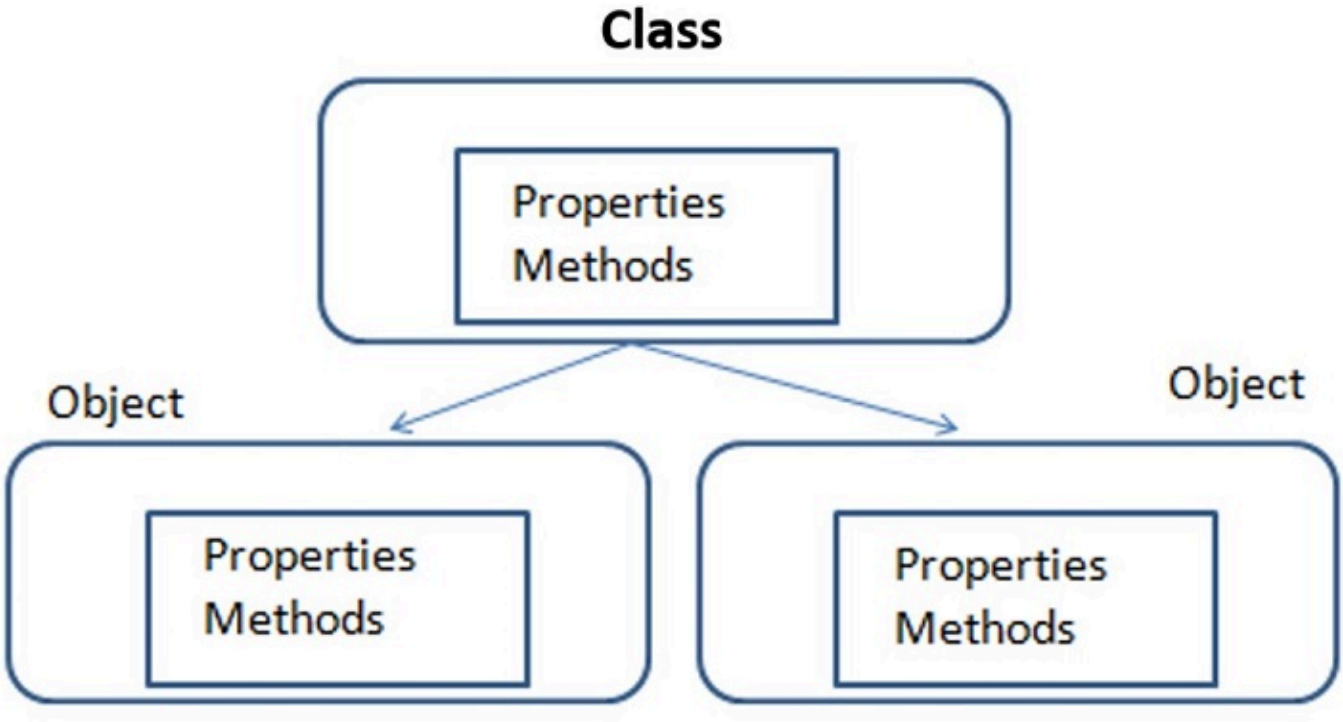
PHP – Classes and Objects

The concept of classes and objects is central to PHP's object-oriented programming methodology. A **class** is the template description of its objects. It includes the properties and functions that process the properties. An **object** is the instance of its class. It is characterized by the properties and functions defined in the class.



Defining a Class in PHP

To define a class, PHP has a keyword "**class**". Similarly, PHP provides the keyword "**new**" to declare an object of any given class.

The general form for defining a new class in PHP is as follows –

```
<?php
   class phpClass {
      var $var1;
      var $var2 = "constant string";

      function myfunc ($arg1, $arg2) {
         [..]
      }
      [..]
```

```
   }
?>
```

The keyword **class** is followed by the name of the class that you want to define. Class name follows the same naming conventions as used for a PHP variable. It is followed by a pair of braces enclosing any number of variable declarations (properties) and function definitions.

Variable declarations start with another reserved keyword **var**, which is followed by a conventional **$variable** name; they may also have an initial assignment to a constant value.

Function definitions look much like standalone PHP functions but are local to the class and will be used to set and access object data. Functions inside a class are also called methods.

Example

Here is an example which defines a class of Book type –

```
class Book {

   /* Member variables */
   var $price;
   var $title;

   /* Member functions */
   function setPrice($par){
      $this->price = $par;
   }

   function getPrice(){
      echo $this->price ."<br/>";
   }

   function setTitle($par){
      $this->title = $par;
   }

   function getTitle(){
      echo $this->title ." <br/>";
   }
}
```

The pseudo-variable **$this** is available when a method is called from within an object context. **$this** refers to the calling object.

The Book class has two **member variables** (or properties) - **$title** and **$price**. The member variables (also sometimes called instance variables) usually have different values for each object; like each book has a title and price different from the other.

The Book class has functions (functions defined inside the class are called **methods**) setTitle() and setPrice(). These functions are called with reference to an object and a parameter, used to set the value of title and price member variables respectively.

The Book class also has **getTitle()** and **getPrice()** methods. When called, they return the title and price of the object whose reference is passed.

Once a class is defined, you can declare one or more objects, using new operator.

```
$b1 = new Book;
$b2 = new Book;
```

The **new** operator allocates the memory required for the member variables and methods of each object. Here we have created two objects and these objects are independent of each other and they will have their existence separately.

Each object has access to its member variables and methods with the "**->**" operator. For example, the **$title** property of **b1** object is "**$b1->title**" and to call setTitle() method, use the "**$b1->setTitle()**" statement.

To set the title and price of **b1** object,

```
$b1->setTitle("PHP Programming");
$b1->setPrice(450);
```

Similarly, the following statements fetch the title and price of **b1** book –

```
echo $b1->getPrice();
echo $b1->getTitle();
```

Example

Given below is the complete PHP script that defines Book class, declares two objects and calls the member functions.

</> Open Compiler

```php
<?php
   class Book {

      /* Member variables */
      var $price;
      var $title;

      /* Member functions */
      function setPrice($par){
         $this->price = $par;
      }

      function getPrice(){
         echo $this->price ."\n";
      }

      function setTitle($par){
         $this->title = $par;
      }

      function getTitle(){
         echo $this->title ."\n";
      }
   }

   $b1 = new Book;
   $b2 =new Book;

   $b1->setTitle("PHP Programming");
   $b1->setPrice(450);
   $b2->setTitle("PHP Fundamentals");
   $b2->setPrice(275);
   $b1->getTitle();
   $b1->getPrice();
   $b2->getTitle();
   $b2->getPrice();
?>
```

It will produce the following **output** –

```
PHP Programming
450
```

PHP Fundamentals
275