Web Technology II (BIT301)

Instructor: Prakash Neupane

Objects

```
<?php
class SayHello{
   function hello(){
      echo "Hello World";
   }
}
$obj=new SayHello;
$obj->hello();
?>
```

Objects

- An object is an instance (or occurrence) of a class.
- In this case, it’s an actual user data structure with attached code.
- Objects and classes are a bit like values and data types.
- There’s only one integer data type, but there are many possible integers.
- Similarly, your program defines only one user class but can create many different (or identical) users from it.

Objects

- The data associated with an object are called its properties.
- The functions associated with an object are called its methods.
- When you define a class, you define the names of its properties and give the code for its methods.

Objects

- Syntax

  To declare an object of a class we need to use new statement

```
<?php
class Class_name
{
   ..
}
$obj=new Class_name;
?>
```

Objects

- A class is a template for objects, and an object is an instance of class.
- A class is defined by using the class keyword, followed by the name of the class and a pair of curly braces ({}).
- All its properties and methods go inside the braces:

```
 <?php
class Fruit {
  // code goes here...
}
?>
```

Objects

```
<?php
class Fruit {
  // Properties
  public $name;
  public $color;

 // Methods
  function set_name($name) {
    $this->name = $name;
  }
  function get_name() {
    return $this->name;
  }
}
?>
```

Objects

```
<?php
class Fruit {
  // Properties
  public $name;
  public $color;
 // Methods
  function set_name($name) {
    $this->name = $name;
  }
  function get_name() {
    return $this->name;
  }
}
```

```
$apple = new Fruit();
$banana = new Fruit();
$apple->set_name('Apple');
$banana->set_name('Banana');

echo $apple->get_name();
echo $banana->get_name();
?>
```

PHP - The $this Keyword

The $this keyword refers to the current object, and is only available inside methods.

```
 <?php
class Student {
  public $name;
}
$apple = new Student();
?>
```

- So, where can we change the value of the $name property?
- There are two ways:
  - Inside the class (by adding a set_name() method and
  - use $this):

PHP - The $this Keyword

```
 <?php
class Student {
  public $name;
  function set_name($name) {
    $this->name = $name;
  }
}
$s1 = new Student();
$s1->set_name("Ram");

echo $s1->name;
?>
```

```
 <?php
class Student {
  public $name;
}
$s1 = new Student();
$s1->name = "RAM";

echo $s1->name;
?>
```

we can use the **instanceof** keyword to check if an object belongs to a specific class:

```
<?php
class CTC{



}
$mango_tree = new CTC();
var_dump($mango_tree instanceof CTC);
?>
```

Anonymous Class

- In PHP, you can create anonymous classes using the class keyword without specifying a class name.
- Anonymous classes are useful when you need to create a small, one-time-use class without giving it a name.
- They are often used for creating objects with specific behavior or overriding methods on the fly.

Anonymous Class

```
$object = new class {
    // Properties and methods go here
};
```

Anonymous Class

```php
<?php
$object = new class {
    public $property = 'Hello, World!';

    public function greet() {
        echo $this->property;
    }
};

$object->greet(); // Outputs: Hello, World!

?>
```

Inheritance

```
<?php
class Animal {
    public $name;
    public function __construct($name) {
        $this->name = $name;
    }
    public function speak() {
        echo "{$this->name} speaks\n";
    }
}
class Dog extends Animal {
    public function speak() {
        echo "{$this->name} barks\n";
    }
}
class Cat extends Animal {
    public function speak() {
        echo "{$this->name} meows\n";
    }
}
// Creating objects
$dog = new Dog("Dog");
$cat = new Cat("Cat");
$obj = new Animal("This is parent class");

// Calling methods
$dog->speak();
$cat->speak();
$obj->speak();
?>
```

Strings and Arrays

References

- Kevin Tatroe , Peter MacIntyre, Programming PHP: Creating Dynamic Web Pages, O′Reilly, 2021