

# PHP : Constructors & Destructors

**Presented by :**  
**Ajitha Kumary P**  
**Lecturer in Computer Application**  
**GDC Srisailam Project**

**Constructors** are special member functions for initial settings of newly created object instances from a class, which is the key part of the object-oriented concept in PHP5.

Constructors are the very basic building blocks that define the future object and its nature. You can say that the Constructors are the blueprints for object creation providing values for member functions and member variables

Once the object is initialized, the constructor is automatically called.

**Destructors** are for destroying objects and automatically called at the end of execution.

Both are special member functions of any class with different concepts but the same name except destructors are preceded by a ~ **Tilda operator**.

**Syntax:**

`__construct():`

```
function __construct()  
{  
// initialize the object and its properties by assigning  
//values  
}
```

```
–  
_destruct():  
function __destruct()  
{  
// destroying the object or clean up resources here  
}
```

**Note:** The constructor is defined in the public section of the Class. Even the values to properties of the class are set by Constructors.

## Constructor types:

- **Default Constructor:** It has no parameters, but the values to the default constructor can be passed dynamically.
- **Parameterized Constructor:** It takes the parameters, and also you can pass different values to the data members.
- **Copy Constructor:** It accepts the address of the other objects as a parameter.

**Inheritance:** As Inheritance is an object-oriented concept, the Constructors are inherited from parent class to child class derived from it. Whenever the child class has constructor and destructor of their own, these are called in order of priority or preference.

**Pre-defined Default Constructor:** By using function `__construct()`, you can define a constructor.

**Note:** In the case of Pre-defined Constructor(`__construct`) and user-defined constructor in the same class, the Pre-defined Constructor becomes Constructor while user-defined constructor becomes the normal method.

## Program

```
<?PHP
class Tree
{
    function Tree()
    {
        echo "Its a User-defined Constructor of the class Tree";
    }

    function __construct()
    {
        echo "Its a Pre-defined Constructor of the class Tree";
    }
}

$obj= new Tree();
?>
```

## Output

**Its a Pre-defined Constructor of the class Tree**

## Advantages of using Constructors:

- Constructors provides the ability to pass parameters which are helpful in automatic initialization of the member variables during creation time .
- The Constructors can have as many parameters as required and they can be defined with the default arguments.
- They encourage re-usability avoiding re-initializing whenever instance of the class is created .
- You can start session in constructor method so that you don't have to start in all the functions every time.
- They can call class member methods and functions.
- They can call other Constructors even from Parent class.



## Destructor:

Destructor is also a special member function which is exactly the reverse of constructor method and is called when an instance of the class is deleted from the memory.

Destructors (`__destruct ( void): void`) are methods which are called when there is no reference to any object of the class or goes out of scope or about to release explicitly.

- They don't have any types or return value. It is just called before deallocating memory for an object or during the finish of execution of PHP scripts or as soon as the execution control leaves the block.
- Global objects are destroyed when the full script or code terminates.
- Cleaning up of resources before memory release or closing of files takes place in the destructor method, whenever they are no longer needed in the code. The automatic destruction of class objects is handled by PHP Garbage Collector.

**Note:** The destructor method is called when the PHP code is executed completely by its last line by using PHP **exit()** or **die()** functions.

## Program

```
<?php
class SomeClass
{

    function __construct()
    {
        echo "In constructor, ";
        $this->name = "Class object! ";
    }

    function __destruct()
    {
        echo "destroying " . $this->name . "\n";
    }
}
$obj = new Someclass();

?>
```

## Output

In constructor, destroying Class object!

## Advantages of destructors:

- Destructors give chance to objects to free up memory allocation , so that enough space is available for new objects or free up resources for other tasks.
- It effectively makes programs run more efficiently and are very useful as they carry out clean up tasks.

**Conclusion:** In the real programming world, Constructors and Destructor methods are very useful as they make very crucial tasks easier during coding. These encourage re-usability of code without unnecessary repetition.

Both of them are implicitly called by compiler even they are not defined in the class.

## Comparison between `__constructors` and `__destructors`:

CONSTRUCTORS	DESTRUCTORS
Accepts one or more arguments.	No arguments are passed. Its void.
function name is <code>_construct()</code> .	function name is <code>_destruct()</code>
It has same name as the class.	It has same name as the class with prefix <code>~</code> tilda.
Constructor is involved automatically when the object is created.	Destructor is involved automatically when the object is destroyed.
Used to initialize the instance of a class.	Used to de-initialize objects already existing to free up memory for new accommodation.
Used to initialize data members of class	Used to make the object perform some task before it is destroyed.

<b>CONSTRUCTORS</b>	<b>DESTRUCTORS</b>
Constructors can be overloaded.	Destructors cannot be overloaded.
It is called each time a class is instantiated or object is created.	It is called automatically at the time of object deletion.
Allocates memory	It deallocates memory.
Multiple constructors can exist in a class	Only one Destructor can exist in a class
The concept of copy constructor is allowed where an object is initialized from the address of another object	No such concept is allowed