

```
1 <?php
2
3 class user
4 {
5     public $name;
6
7     public function __construct($name=NULL)
8     {
9         if(empty($name))
10            $this->name = 'John Doe';
11        else $this->name = $name;
12    }
13 }
14
```

[www.engineersworld.in](http://www.engineersworld.in)



## Object Oriented PHP

1. What Is Object Oriented	3
2. What Is Class	3-4
3. What Is Object	4
4. What Is Method	5
5. Access Modifier	
• Public	6
• Private	6
• Protected	6
6. Static member	7
7. Self keyword	7
8. this variable	8
9. constant	8-9
10. constructor	9-10
11. destructor	10-11
12. Inheritance	12-13
13. Include Keyword	13

## Object Oriented :

object Oriented का मतलब यह है की एक एसी तकनीक जिसमे ऑब्जेक्ट पर फोकस किया जाए object Oriented तकनीक कहलाती है object Oriented तकनीक में हर चीज एक ऑब्जेक्ट होती है php में भी अब Object Oriented technic का use होता है object oriented Technic में हर समस्या को एक object मान कर हल किया जाता है जिससे उस समस्या को हल करना आसान होता है

### object Oriented Programming की विशेषता :

1. object Oriented programming का use करने से प्रोग्राम पढ़ने और समझने में आसान होता है
2. object Oriented का use करने से प्रोग्राम में कोड reusability की fecelity मिलती है

## class :

object Oriented प्रोग्रामिंग में class सबसे important होती है class collection of method and statement होते हैं class का use कर के किसी भी प्रोग्राम का कोड लिखा जाता है class का use करने से प्रोग्राम सिक्वोर रहता है

### class का syntax निम्नलिखित है

```
class_keyword class_name
{
    method;
    statement;
}
```

### Example :

```
class Example
{
    public $a=101; //variable declaration
    function disp() //method
```

```
{
    echo $a;    //statement
}
}
```

## Object :

object Oriented प्रोग्रामिंग जेसा की नाम से हो स्पष्ट है object Oriented अर्थात एक एसी प्रोग्रामिंग technic जिसमे object पर focus किया जाए

object php में किसी न किसी class का reffrence होता है जिसका use कर के class के मेंबर को एक्सेस किया जाता है object में class से सम्बंधित method तथा statement को content कर के रखता है class के मेम्बर को उस class के object का use कर के ही एक्सेस किया जाता है

### **syntax :**

```
reffrence_variable =new constructor_name();
```

### **Example :**

```
<?php
class Example
{
    public function fl()
    {
        $a=10;
        $b=5;
        echo "Sum =" .($a+$b);
    }
}
$obj=new Example();
$obj->fl();
?>
```

## Method :

जिस तरह c language में function होती है use तरह oops php में method कहा जाता है दुसरे शब्दों में कहे तो object oriented प्रोग्रामिंग में function का method कहा जाता है method का use किसी ना किसी special task को पूरा करने के लिए

किया जाता है php में method को create करने के लिए function keyword का use किया जाता है

**syntax :**

```
function function_name(argument_list)
{
    statement;
    return statement;
}
```

**Example :**

```
<?php
class Example
{
    public function f1()
    {
        $a=10;
        $b=5;
        echo "Sum =" .($a+$b);
    }
}
$obj=new Example();
$obj->f1();
?>
```

**Access Modifier :**

Access Modifier का use class के member का scop निर्धारित करने के लिए किए जाता है मेम्बर के scop से तात्पर्य यह है की class के member कहा तक एक्सेस किये जा सकते है इसका निर्धारण करने के लिए Access Modifier का use किया जाता है

php में Access Modifier 3 प्रकार के होते है

1. public
2. private
3. protected

1. public :

public मेम्बर वह member होते हैं जिनको पुरे प्रोग्रम में कही से भी एक्सेस किया जा सकता है public member public member कहलाते हैं public member define करने के लीये statement के पहले public keyword का use किया जाता है

### 2. protected :

इस function का use protected member बनाने के लिए किया जाता है protected member को उस class में एक्सेस किया जाता है जिसमे वह define किये गए है उस class के अलावा वह class उस member को एक्सेस के सकती है जिस class से उस class को inherit किया गया हो

### 3. private :

private member वह member होते हैं जिनको केवल उसी class में एक्सेस किया जाता है जिस class में वह डिक्लेअर किये गए है उसके आलावा कोई भी class उस member को एक्सेस नहीं कर सकती है

## Static statement :

oops php में static member बनाने के लिए static keyword का use किया जाता है static member वह member होते हैं जिनको एक्सेस करने के लिए object की अवश्यकता नहीं होती है static मेम्बर को एक्सेस करने के लिए class के नाम का use किया जाता है और class के नाम के साथ scop rigolution operator ( :: ) का use किया जाता है

### Example :

```
<?php
class Example
{
    public static $a="Hello";
}
echo Example::$a;
?>
```

## **self Keyword :**

जब भी हमें किसी class के static member को उसी class में एक्सेस करना होता है तो उसे एक्सेस करने के लीये self keyword का use किया जाता है

### **Example :**

```
<?php
class Example
{
    public static $a="Hello";
    public function f1()
    {
        echo self::$a;
    }
}
Example::f1();
?>
```

## **\$this Keyword :**

जब हमें किसी class के सामान्य member को उसी class में एक्सेस करना होता है तो php में \$this का use किया जाता है

### **Example :**

```
<?php
class Example
{
    public $a="Hello";
    public function f1()
    {
        echo $this->a;
    }
}
$obj = new Example();
$obj->f1();
?>
```

## Constant :

Php या अन्य प्रोग्रामिंग language में constant एक एसी entity होती है जो पुरे प्रोग्राम में चेंज नहीं होते है constant को एक बार initialization करने के बाद उसके मान को बदला नहीं जा सकता है

php में constant 2 प्रकार के होते है

1. class content
2. non – class constant

### 1. class constant :

यह ऐसे constant होते है जिनको class के अन्दर define किया जाता है class constant कहलाते है constant को define करने के लिए const keyword का use किया जाता है

#### Example :

```
<?php
class Example
{
    const name="Deependra"; //class Constant
}
echo Example::name;
?>
```

## Non – Class Constant :

non – class constant वह constant होते है जिनको class के बाहर define किया जाता है non class constant कहलाते है non class constant को define करने के लिए एक method का use किया जाता है जिसे define() कहा जाता है यह method 2 पेरामीटर लेती है

1. variable name
2. value

#### syntax :

```
define('variable name', 'value');
```

#### Example :



```
<?php
define("hello","Hello World");//non-class Constant
echo hello;
?>
```

### Constrictor And Deconstrictor

php में Constrictor का use किया जाता है php में Constrictor एक विशेष प्रकार का function होता है जिसे execute करने के लिए call करने की आवश्यकता नहीं होती है यह खुद ही execute हो जाते हैं php में Constrictor बनाने के लिए function \_\_construct का use करते हैं

#### **syntax :**

```
public function __construct()
{
    statement;
}
```

#### **Example :**

```
<?php
class Exampel
{
    public function __construct()
    {
        echo "This Is Constract";
    }
}
```

?>

**Destructor :**

Destructor php में constructor की तरह एक special function है किन्तु Destructor का कार्य constructor से विपरीत होता है php में destructor बनाने के लिए function \_\_destruct लिखते हैं

**syntax :**

```
public function __destruct()  
{  
    statement;  
}
```

**Example :**

```
<?php  
class Exampel  
{  
    public function __destruct()  
    {  
        echo "This Is Construct";  
    }  
}  
  
?>
```

## Inheritance

### What Is Inheritance :

object oriented programming का सबसे बड़ा उद्देश्य code Reuseblity है अर्थात एक बार कोड को लिख कर use आवश्यकता अनुसार बार बार आसानी से किया जा सके Inheritance oops के इसी गुण को पूरा करता है

Inheritance में 2 या 2 से अधिक class होती है जिसमे पहली class base class होती है तथा दूसरी class derived class होती है Inheritance का use कर के base class की सभी property derived class को Inheritance

कर दिया जाता है base class की सभी property को derived class से Inheritance करने का तात्पर्य यह है की base क्लास की property derived class को दे दी जाती है Inheritance कहलाता है

base class को derived class से Inheritance करने के लिए extends keyword का use किया जाता है

### Syntax :

```
class A
{
    statement;
}
class B extends A
{
    Statement;
}
```

### Example :

```
<?php
class A
{
    public $a="Hello";
}
class B extends A
{
    public function f1()
    {
        echo $this->a;
    }
}
$obj=new B();
$obj->f1();
?>
```

### Include Statement :

Include statement का use php में एक एक webpage में किसी दुसरे webpage को include करने के लिए करते हैं

### Example :

```
<?php
include "a1.php";
class A extends Example
{
    public function f1()
    {
        echo $this->a;
    }
}
$obj=new A();
$obj->f1();
?>
```

### a1.php File :

```
<?php
class Example
```

```
{  
    public $a="Hello";  
}  
?>
```