**LARAVEL**

**\*INCLUDE**

**\*REQUIRE**

**\*INCLUDE\_ONCE**

**\*REQUIRE\_ONCE**

**OOP in PHP**

­­­­base – \*Classes (we make concepts using this base)

\*Objects (concepts like polymorphism, inheritance, encapsulation, data abstraction, etc)

**Syntax of class:**

<?php

Class PhpClass{

//variables

//methods

}

?>

**Eg of class :**

<?php

Class Student {

Var $roll\_no;

function getRno(){

$data;

//body

}

}

?>

Z

**Object creation:**

$std = new Student(); //object initialized

**Constructor Function:**

Function \_\_construct () {

//body

// constructor never returns any value

}

**Inheritance:**

class Child extends Parent {

//child class now have properties of parent

}

**Public Members:**

//default var scope = public

var $var1; //public (default)

public var $var2; //available to outside classes

private var $var3; //exclusively available for the same class

protected var $var4; //available to only same class and child class

**Interface:**

interface student {

public function getRollNo();

}

Class stdInfo implements student {

public function getRollNo(){

//body

// compulsory to define such function

}

}

**Abstract class:**

Abstract class’s objects cannot be created

Only way to use is extend.

Contains abstract and normal functions.

Abstract functions need to be defined by the child class.

Normal methods can directly be used by child class.

**Eg:**

abstract class Student {

abstract function getRollNo();

function getName(){

//body

}

}

class StdInfo extends Student {

function getRollNo(){

//compulsory to define abstract functions

//body

}

}

**Static keyword:**

Sustains values, Created Once, Accessible to all objects of same class, modification alters the value for all other objects.

class Foo {

public static $foo = “You are Great“;

public function getStaticvValue(){

echo self::$foo;

}

}

echo “We want you to know that ” . Foo::$foo;

**Constants in php:**  created by “const” keyword.

**‘this’ keyword:** $this -> var\_name;