### **ASSIGNMENT OF PHP: Object oriented programming**

**1:- Explain OOPS concept with an example.**

**ANSWER 1:**PHP is a server-side scripting language, mainly used for web development but also used as a general-purpose programming language. Object-Oriented Programming (PHP OOP), is a type of programming language principle added to php5, that helps in building complex, reusable web applications.he Object Oriented concepts in PHP are:

**a.Class** − This is a programmer-defined data type, which includes local functions as well as local data. You can think of a class as a template for making many instances of the same kind (or class) of object.

**b.Object** − An individual instance of the data structure defined by a class. You define a class once and then make many objects that belong to it. Objects are also known as instance.

**c.Inheritance** − When a class is defined by inheriting existing function of a parent class then it is called inheritance. Here child class will inherit all or few member functions and variables of a parent class.

**d.Polymorphism** − This is an object oriented concept where same function can be used for different purposes. For example function name will remain same but it make take different number of arguments and can do different task.

**e.Overloading** − a type of polymorphism in which some or all of operators have different implementations depending on the types of their arguments. Similarly functions can also be overloaded with different implementation.

**f.Data Abstraction** − Any representation of data in which the implementation details are hidden (abstracted).

**g.Encapsulation** − refers to a concept where we encapsulate all the data and member functions together to form an object.

**h.Constructor** − refers to a special type of function which will be called automatically whenever there is an object formation from a class.

**i.Destructor** − refers to a special type of function which will be called automatically whenever an object is deleted or goes out of scope

**2:- Explain Classes & Objects with an example.**

**ANSWER 2:**

**a.Class** − This is a programmer-defined data type, which includes local functions as well as local data. You can think of a class as a template for making many instances of the same kind (or class) of object.

**b.Object** − An individual instance of the data structure defined by a class. You define a class once and then make many objects that belong to it. Objects are also known as instance.

**CODE :**

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');

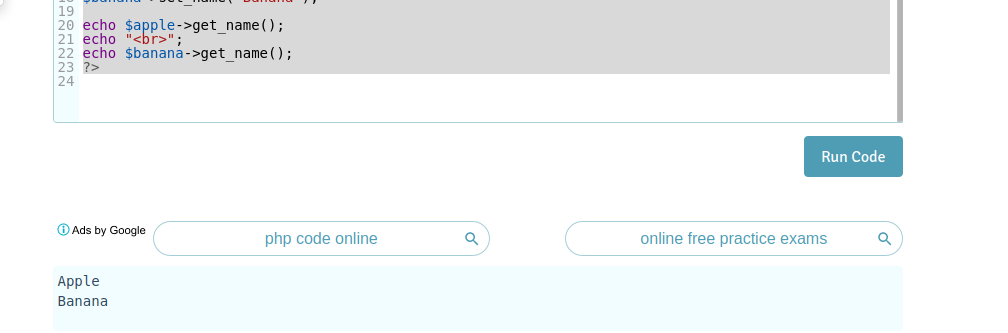
$banana->set\_name('Banana');

echo $apple->get\_name();

echo "<br>";

echo $banana->get\_name();

?>



**3:- What is a Namespace?**

**ANSWER 3:**A namespace is used to avoid conflicting definitions and introduce more flexibility and organization in the code base. Just like directories, namespace can contain a hierarchy know as subnamespaces. PHP uses the backslash as its namespace separator.

**Code :**<?php

namespace MyNamespaceName;

function hello()

{

echo 'Hello I am Running from a namespace!';

}

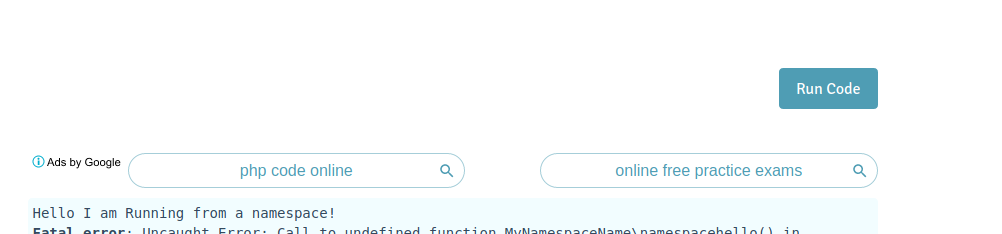
// Resolves to MyNamespaceNamehello

hello();

// Explicitly resolves to MyNamespaceNamehello

namespacehello();

?>



**4:- Explain constructor and destructor with an example.**

**ANSWER 4:**

**CODE :**<?php

class Person {

// first name of person

private $fname;

// last name of person

private $lname;

// Constructor

public function \_\_construct($fname, $lname) {

echo "Initialising the object...<br/>";

$this->fname = $fname;

$this->lname = $lname;

}

// Destructor

public function \_\_destruct(){

// clean up resources or do something else

echo "Destroying Object...";

}

// public method to show name

public function showName() {

echo "My name is: " . $this->fname . " " . $this->lname . "<br/>";

}

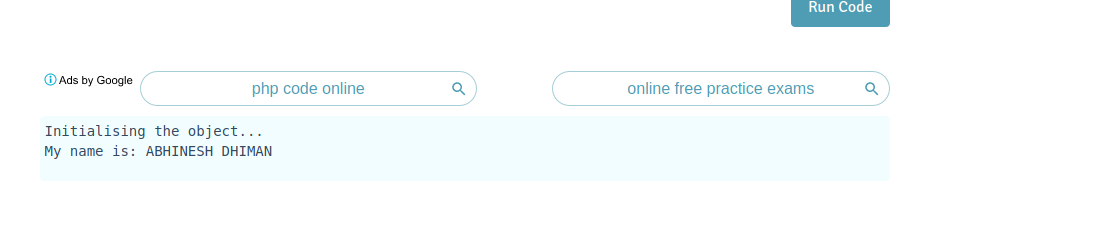
}

// creating class object

$A = new Person("ABHINESH", "DHIMAN");

$A->showName();

?>



**5:- Write a program that initialize class property using constructor.**

**ANSWER 5:**

<?php

class Fruit {

public $name;

public $color;

function \_\_construct($name, $color) {

$this->name = $name;

$this->color = $color;

}

function get\_name() {

return $this->name;

}

function get\_color() {

return $this->color;

}

}

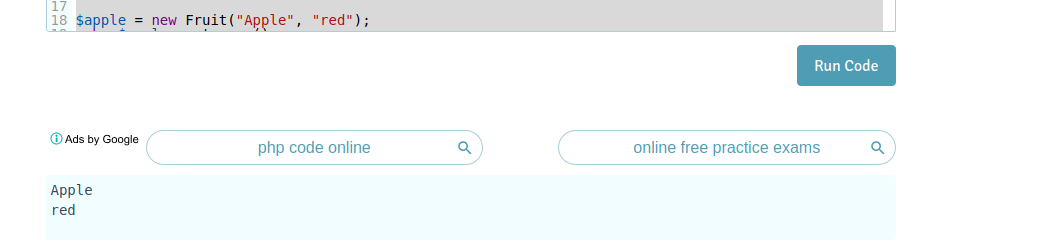
$apple = new Fruit("Apple", "red");

echo $apple->get\_name();

echo "<br>";

echo $apple->get\_color();

?>



**6:- Write a PHP program to return Factorial value of number in a Object oriented way. Factorial logic should be in separate function.**

**ANSWER 6:**

<?php

class factorial\_of\_a\_number

{

protected $\_n;

public function \_\_construct($n)

{

if (!is\_int($n))

{

throw new InvalidArgumentException('Not a number or missing argument');

}

$this->\_n = $n;

}

public function result()

{

$factorial = 1;

for ($i = 1; $i <= $this->\_n; $i++)

{

$factorial \*= $i;

}

return $factorial;

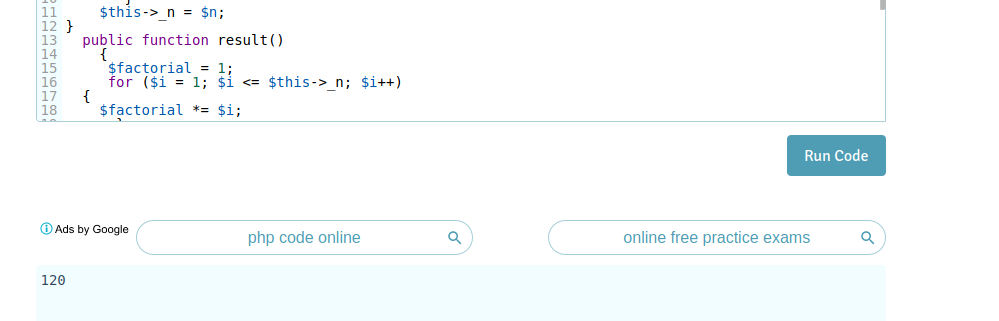
}

}

$newfactorial = New factorial\_of\_a\_number(5);

echo $newfactorial->result();

?>



**7:- Write a class that extends Abstract class**

**ANSWER 7:**

**CODE:** <?php

// Abstract class

abstract class Base {

function \_\_construct() {

echo "this is abstract class constructor ";

}

// This is abstract function

abstract function printdata();

}

class Derived extends base {

function \_\_construct() {

echo " Derived class constructor";

}

function printdata() {

echo " Derived class printdata function";

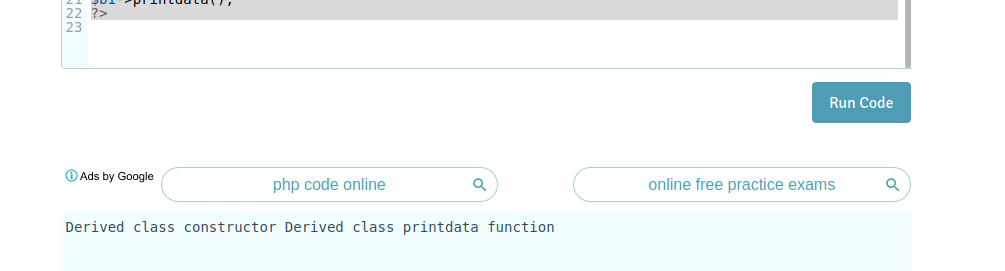
}

}

$b1 = new Derived;

$b1->printdata();

?>



**8:- Implement Inheritance in PHP program to display the Message when Parent class method is called and when the Child method is called.**

**ANSWER 8:**

<?php

class Fruit {

public $name;

public $color;

public function \_\_construct($name, $color) {

$this->name = $name;

$this->color = $color;

}

public function intro() {

echo "The fruit is {$this->name} and the color is {$this->color}.";

}

}

// Strawberry is inherited from Fruit

class Strawberry extends Fruit {

public function message() {

echo "Am I a fruit or a berry? ";

}

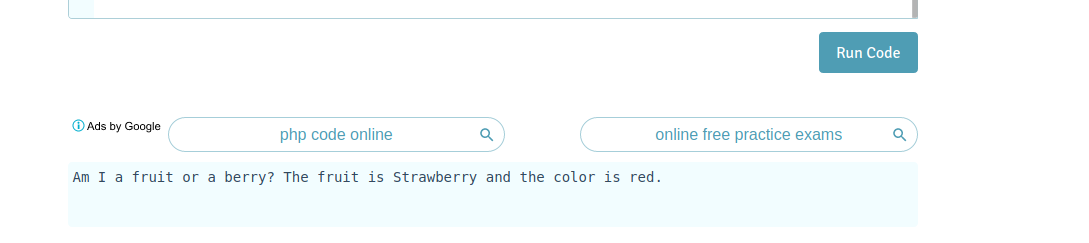
}

$strawberry = new Strawberry("Strawberry", "red");

$strawberry->message();

$strawberry->intro();

?>



**9:- What is method chaining explain with an example.**

**ANSWER 9:**

<?php

class person

{

private $name="";

private $age="";

public function setName($name="")

{

$this->name=$name;

return $this;

}

public function setAge($age="")

{

$this->age=$age;

return $this;

}

public function getInfo()

{

echo "Hello, My name is ".$this->name." and my age is ".$this->age." years.";

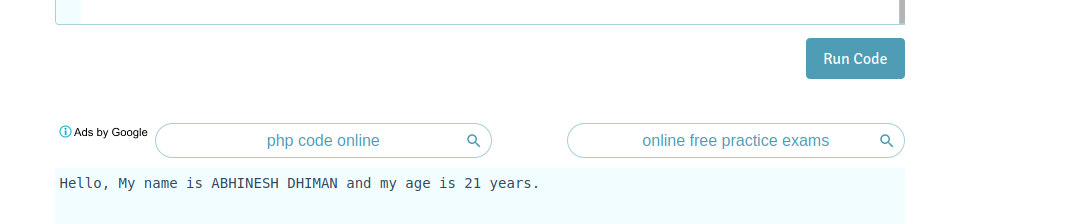
}

}

$person = new person();

$person->setName("ABHINESH DHIMAN")->setAge("21")->getInfo();

?>



**10:- What is $this keyword.**

**ANSWER 10:** To access or change a class method or property within the class itself, it's necessary to prefix the corresponding method or property name with "$this" which refers to this class**.**

**11:- Write program to explain Multi Level Inheritance.**

**ANSWER 11:**<?php

class Abhinesh {

public function sayhello() {

echo "abhinesh class ";

}

}

trait Priyanshu {

public function sayfor() {

echo "priyanshu trait ";

}

}

class S extends Abhinesh {

use Priyanshu;

public function g() {

echo "s class ";

}

}

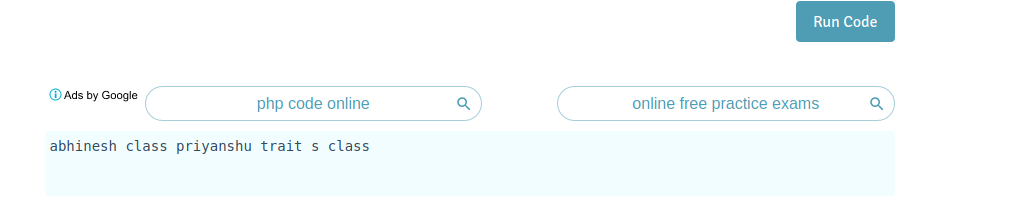
$test = new S();

$test->sayhello();

$test->sayfor();

$test->g();

?>



**12.- Develop prototype for flight-booking service. Here, we don't own any flights. we are just a service provider mediator**

**Define two methods :**

**a. checkFlightsAvailability() - to check available flights from two vendors like "Air India", "IndiGO".**

**b. bookFlights() - to book a seat from respective vendor.**

**ANSWER 12:**

**PROTOTYPE CODE OF ABOVE SCENARIO:**

<?php

class Flight

{

protected $v1 =”airindiaflights”;

protected $v2=”indigoflights”;

public function flightsavlable( no.of\_flights )

{

$c1 =0 ;

$c2=0;

for($i=1;$i<=$total\_no\_of\_flights ;$i++)

{

if($vendor\_name ==$v1)

{ $c1++;}

elseif($vendor\_name== $v2)

{ $c2++;}

}

echo “ flight available for “.$v1.”are”.$c1;

echo “ flight available for “.$v2.”are”.$c2;

}

public function bookFlights( $choice, $available\_seats,$seats\_to\_be \_booked);

{

$vendor\_name =$choice;

if($available\_seats ==0)

{

echo” all seats are full”;

}

else

{

if ($seats\_to\_be\_booked > $available\_seats)

{

echo “ only “.$seats\_to\_be\_booked -$available\_seats. “seats are available”;

}

else

{

$available\_seats-=$seats\_to\_be\_booked;

echo” seats bookesd succesfully”;

}

}

?>

**13:- Define three traits and use it in class and override it's function.**

**ANSWER 13:**

**Code:**

<?php

trait f

{

public function demo()

{

echo("1st printed");

}

}

trait s

{

public function demo2()

{

echo("2nd printed");

}

}

trait t

{

public function demo3()

{

echo("3rd printed");

}

}

class c

{

use f,s,t;

} {

$o=new c;

$o->demo();

$o->demo2();

$o->demo3();

}

class c2

{

use f,s,t;

public function demo()

{

echo"function overloaded";

}

}

$obj1= new c2();

$obj1->demo();

$obj1->demo2();

$obj1->demo3();

?>

