OOPS ASSIGNMENT

Q1. What Is Object-Oriented Programming?

ANS:- Object-oriented programming (OOP) is a computer programming model that organizes software design around data, or objects, rather than functions and logic. An object can be defined as a data field that has unique attributes and behavior.

Q2. What Are Properties of Object-Oriented Systems?

ANS:- 4 major principles make a language Object Oriented. These are Encapsulation, Data Abstraction, Polymorphism, and Inheritance. These are also called as four pillars of Object-Oriented Programming.

Q3. What Is the Difference Between Class and Interface?

ANS:- A class describes the attribute and behaviors of an object. An interface contains behaviors that the class implements.

Q4. What Is Overloading?

ANS:- Method overloading is a salient feature in Object-Oriented Programming (OOP). It lets you declare the same method multiple times with different argument lists.

Q5. What Is T_PAAMAYIM_NEKUDOTAYIM (Scope Resolution Operator

::)

and give an example?

ANS:-

```php
<?php
class demo{
      public static $bar=10;
      public static function func(){

        echo static::$bar."\n";
    }
}
class Child extends demo{
      public static $bar=20;
  }
demo::func();
Child::func();
  ?>
```

Q6. What are the differences between abstract classes and interfaces?

ANS:- Abstract class can have final, non-final, static and non-static variables.
The interface has only static and final variables.

Q7. Define Constructor and Destructor?

ANS:- Constructor helps to initialize the object of a class. Whereas destructor is used to destroy the instances.

Q8. How to Load Classes in PHP?

ANS:- introduced the magic function __autoload() which is automatically called when your code references a class

Q9. How to Call Parent Constructor?

ANS:-

```php
<?php
 class parents{
    function __construct(){
         echo "I am parent class"."</br>";
     }
 }
 class child extends parents{
    function __construct(){
        parents::__construct();
         echo "I am child class";
     }
 }
 $a = new child();
?>
```

Q10. Are Parent Constructors Called Implicitly When Create An Object Of Class?

ANS:- Parent constructors are not called implicitly if the child class defines a constructor. ... If the child does not define a constructor then it may be inherited from the parent class just like a normal class method (if it was not declared as private).

Q11. What Happen, If Constructor Is Defined As Private Or Protected?

ANS:- If a constructor is declared as private, then its objects are only accessible from within the declared class. You cannot access its objects from outside the constructor class.

Q12. Define New-style Constructor & Old-style Constructor. Check which One Will Be Called?

ANS:- new Style constructor

Q13. Create Abstract class and method?

ANS:-

```php
<?php
abstract class A{
    abstract function area();
}

class circle extends A {
    function area(){
        echo "area of circle";
```

```
    }
}
class squar extends A {
    function area(){
        echo "area of squar";
    }
}
class triangle extends A {
    function area(){
        echo "area of triangle";
    }
}

$a = new circle;
$a->area();
?>
```

Q14. Define 3 types of visibility of data member & member function.

AMS:- Public, private, and protected.

Q15. Create a method that will never be inherited?

ANS:- Inherited in OOP = When a class derives from another class.

Q16. What is the difference between Abstract class and Interface?

ANS:- Abstract class can have final, non-final, static and non-static variables.

The interface has only static and final variables.

Q17. Create parent class for car and child class for car_model and user functionality in car_modelclass.

ANS:-

```php
<?php
  class car{
      public $model;
       function setmodel($model){
          $this->model=$model;
      }
  }
  class car_model extends car{
      function getcar(){
      echo $this->model;
      }
  }

  $car1=new car_model();
  $car1->setmodel('bmw');
 $car1->getcar();
?>
```

Q18. Override the parent's properties and methods in the child class?

ANS:- Method overriding allows a child class to define a method that overrides (or replaces) the method already provided by its parent class.

Q19. Prevent the child class from overriding the parent’s methods?

ANS:- In order to prevent the method in the child class from overriding the parent's methods, we can prefix the method in the parent with the final keyword.

Q20. Declare classes and methods as abstract?

ANS:-

```php
<?php
abstract class A{
    abstract function area();
}

class circle extends A {
    function area(){
        echo "area of circle";
    }
}
class squar extends A {
    function area(){
        echo "area of squar";
    }
}
class triangle extends A {
    function area(){
        echo "area of triangle";
    }
}

$a = new circle;
```

```php
$a->area();
?>
```

Q21. Can we have nonabstract methods inside an abstract class? Explain With Example

ANS:- Yes we can have an abstract class without Abstract Methods as both are independent concepts

Q22. How to create child classes from an abstract class?

ANS:-

```php
<?php
abstract class A{
    abstract function area();
}

class circle extends A {
    function area(){
        echo "area of circle";
    }
}
class squar extends A {
    function area(){
        echo "area of squar";
    }
}
class triangle extends A {
    function area(){
        echo "area of triangle";
    }
}
```

```php
$a = new circle;
$a->area();
?>
```

Q23. What are PHP Magic Methods/Functions? List them

ANS:- special methods which override PHP's default's action when certain actions are performed on an object.

Magic Method
Description
__toString()
is invoked when an object of a class is treated as a string.
__invoke()
is invoked when an object is called as a function
__set_state()
is called for a class exported by var_export()
__clone()
is called once the cloning is complete

Q24. Write a program for Static keywords in PHP?

ANS:-

```php
<?php

class myclass{
    public static $str = "hello world";

    public static function func(){
        echo myclass::$str;
```

```
    }
}
echo myclass::$str."</br>";
echo myclass::func();
?>
```

Q25. Create multiple Traits and use them in a single class?

ANS:-

```
<?php

trait message1{
    public function msg1(){
        echo"OOPs is easy";
    }
}

class welcome{
    use message1;
}

$obj = new welcome;
$obj-> msg1();
?>
```

Q26. Write PHP Script of Object Iteration?

ANS:-

```
<?php
class A{
    public $name ="sam";
    public $name1 = "raj";
```

```
    function show(){
        foreach($this as $key=>$value){
            print_r($value);
        }
    }
}
$objA = new A;
$objA->show();
?>
```

Q27. Create PHP script for objects passed in PHP – by reference or by value?

ANS:-

```
<?php
class A {
    public $foo = 1;
}

$a = new A;
$b = $a;      // $a and $b are copies of
the same identifier
              // ($a) = ($b) = <id>
$b->foo = 2;
echo $a->foo."\n";


 $c = new A;
 $d = &$c;     // $c and $d are references
//                 // ($c,$d) = <id>

 $d->foo = 2;
```

```php
 echo $c->foo."\n";
?>
```

Q28. What is data modeling?

ANS:- Data modeling is the process of creating a simplified diagram of a software system and the data elements it contains, using text and symbols to represent

Q29. Use of The $this keyword

ANS:- eliminate the confusion between class attributes and parameters with the same name

Q30. Declare and implement an interface and implement more than one interface in the same class.

ANS:-

```php
<?php
interface stray{
    function greed();
}
interface pet{
    function behaviar();
}

class cat implements stray,pet{
    function greed(){
        echo "greed";
    }
    function behaviar(){
        echo "behavor";
```

```
    }
}

$c = new cat;
$c->greed();
$c->behaviar();
?>
```

Q31. How to implement the polymorphism principle?

ANS:- polymorphism principle, we can choose between abstract classes and interfaces.

Q32. Explain the scope resolution operator with an example.

ANS:-

```
<?php

class democlass {
    const PI = 3.14;
}

echo democlass::PI;

?>
```

Q33. Access child class property to parent.

ANS:-