Experiment No: 08

Write a simple PHP program to

a. Inherit members of super class in subclass.

b. Create constructor to initialize object of class by using object oriented concepts.

Resources required:

Hardware	**Software**
Computer System	Any database tools such as XAMPP

Practical Significance:

Inheritance:

- Inheritance is a mechanism of extending an existing class by inheriting a class.
- We create a new sub class with all functionality of that existing class, and we can add new members to the new sub class.
- When we inherit one class from another we say that inherited class is a subclass and the class who has inherits is called parent class.

Constructor and Destructor:

- A constructor and a destructor are special functions which are automatically called when an object is created and destroyed.
- Constructors are special member functions for initialize variables on the newly created object instances from a class.
- When creating a new object, it’s useful to set up certain aspects of the object at the same time. For example, you might want to set some properties to initial values, fetch some information from a database to populate the object, or register the object in some way.
- Similarly, when it’s time for an object to disappear, it can be useful to tidy up aspects of the object, such as closing any related open files and database connections, or unsetting other related objects.
- An object’s constructor method is called just after the object is created, and its destructor method is called just before the object is freed from memory.

Theoretical Background:

Inheritance:

To declare that one class inherits the code from another class, we use the **extends** keyword.

Syntax :

```
class Parent
{
  // The parent’s class code
}
 class Child extends Parent
{
  // The  child can use the parent's class code
}
```

The child class can make use of all the non-private (public and protected) methods and properties that it inherits from the parent class. This allows us to write the code only once in the parent, and then use it in both the parent and the child classes.

Constructor and Destructor:

- Normally, when you create a new object based on a class, all that happens is that the object is brought into existence. (Usually you then assign the object to a variable or pass it to a function.) By creating a constructor method in your class, however, you can cause other actions to be triggered when the object is created.

  To create a constructor, simply add a method with the special name _ _construct() to your class. (That ’ s two underscores, followed by the word “ construct, ” followed by parentheses.) PHP looks for this special method name when the object is created; if it finds it, it calls the method.

```
class MyClass
{
function __construct()
{
echo “Welcome to PHP constructor. <br / > ”;
}
}
$obj = new MyClass; // Displays “Welcome to PHP constructor.”
```

Program Code:

a)

Output :

Practical related questions:

1. Define the relationship between a class and an object?

A class acts as a blueprint that defines the properties, states, and behaviors that are common to a number of objects. An object is an instance of the class. For example, you have a class called Vehicle and Car is the object of that class. You can create any number of objects for the class named Vehicle, such as Van, Truck, and Auto.

The new operator is used to create an object of a class. When an object of a class is instantiated, the system allocates memory for every data member that is present in the class.

2. What is the purpose of $this & extends?

$this-:

- The $this keyword allows us to approach the class properties and methods from within the class.
- The $this keyword indicates that we use the class's own methods and properties and allows us to have access to them within the class's scope.
- $this is a pseudo-variable which is a reference to the current object.
- $this variable is used to call non-static method, if you are trying to call static method then it will throw the error that means $this variable is not available inside the static method.

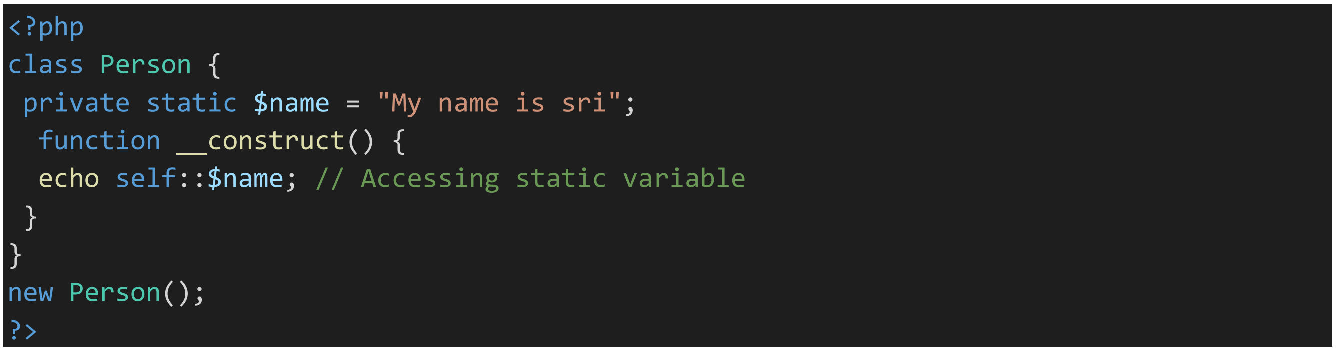
Extends-:

- In order to declare that one class inherits the code from another class, we use the **extends** keyword.

3. State the use of 'self' with suitable example.

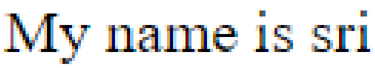
self
self keyword is not preceded by any symbol rather we can use as it is.
To access class variables and methods using the self keyword, we use the scope resolution operator ::
It is used to refer the static members of the class.
PHP self refers to the class members, but not for any particular object. This is because the static members(variables or functions) are class members shared by all the objects of the class.
Example : self::<class_member>

Code-:



```
<?php
class Person {
 private static $name = "My name is sri";
  function __construct() {
  echo self::$name; // Accessing static variable
 }
}
new Person();
?>
```

Output-:

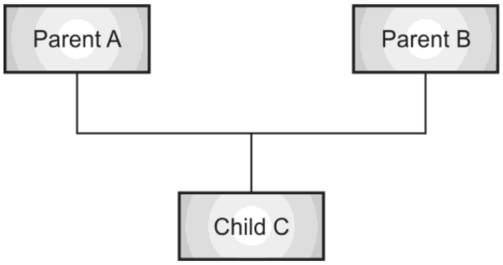


```
My name is sri
```

Exercise:

1. How to implement multiple inheritance in PHP?

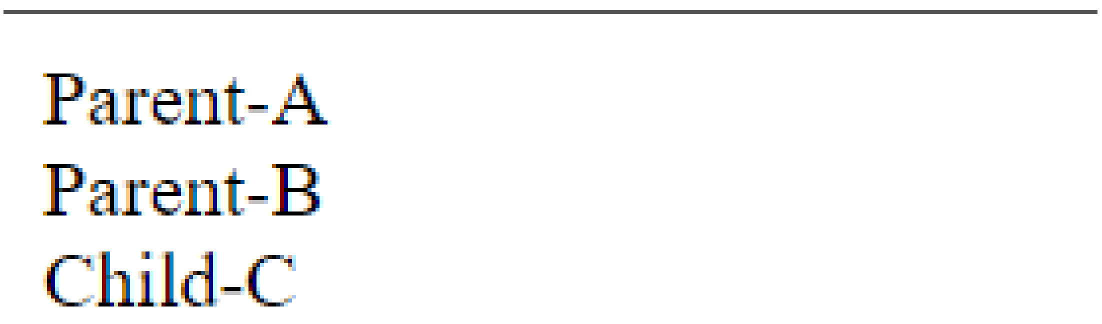
The mechanism of inheriting the features of more than one base class into a single class is known as multiple inheritance. PHP doesn’t support multiple inheritance but by using Interfaces in PHP or using Traits in PHP instead of classes, we can implement it.



Code -:

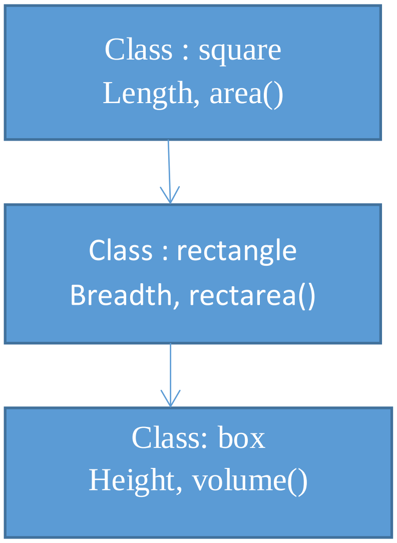
```
class A {
  public function disp1()
   {
     echo "Parent-A <br>";
  }
}

// Trait B i.e. Parent B
trait B
{
  public function disp2()
  {
     echo " Parent-B <br>";
  }
}

class C extends A
 {
   use B;
   public function disp3()
  {
      echo "\nChild-C";
  }
}
$obj = new C();
$obj->disp1();
$obj->disp2();
$obj->disp3();
?>
```

Output-:



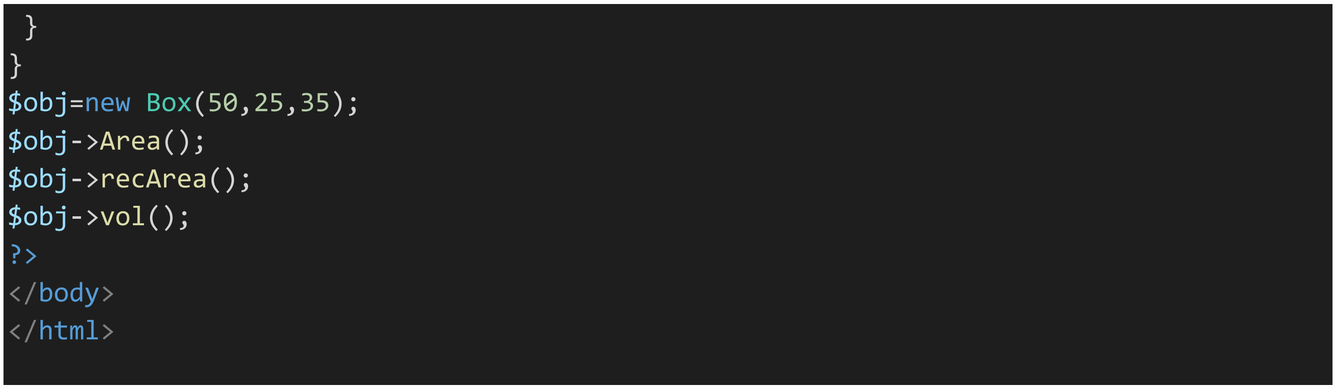
```
Parent-A
Parent-B
Child-C
```

2. Implement following inheritance:



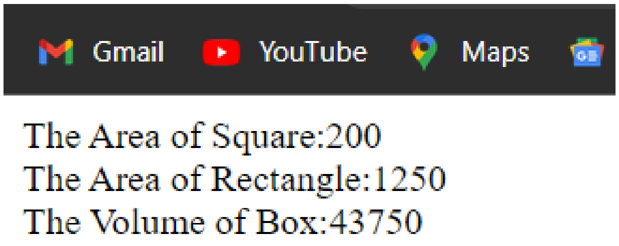
Code-:

```
<!DOCTYPE html>
<html>
<body>
<?php


class Square{
public $length;
public function __construct($length) {
 $this->length = $length;
 }
 public function Area(){
 echo" The Area of Square:";
 echo($this->length*4);
 echo "<br>";
 }
}


class Rectangle extends Square{
public $width;
public function __construct($width,$length) {
 $this->width = $width;
 $this->length = $length;
 }
 public function recArea(){
 echo"The Area of Rectangle:";
 echo($this->length*$this->width);
 echo "<br>";
 }
}


class Box extends Rectangle{
public $height;
public function __construct($length,$width,$height) {
$this->length = $length;
 $this->width = $width;
 $this->height = $height;
 }
 public function vol(){
 echo"The Volume of Box:";
 echo($this->length*$this->width*$this->height);
```



```
 }
}
$obj=new Box(50,25,35);
$obj->Area();
$obj->recArea();
$obj->vol();
?>
</body>
</html>
```

Output-:



```
The Area of Square:200
The Area of Rectangle:1250
The Volume of Box:43750
```