Date: 29th March 2023
Roll No. and Name: 20BCE204 Dhyan Patel
Course Code and Name: 2CSDE69 LAMP Technology
Practical No. 5
AIM:
(A) Derive a class square from class Rectangle. Create one more class circle. Create an interface with only one method called area(). Implement this interface in all the classes. Include appropriate data members and constructors in all classes. Write a program to accept details of a square, circle and rectangle and display the area.

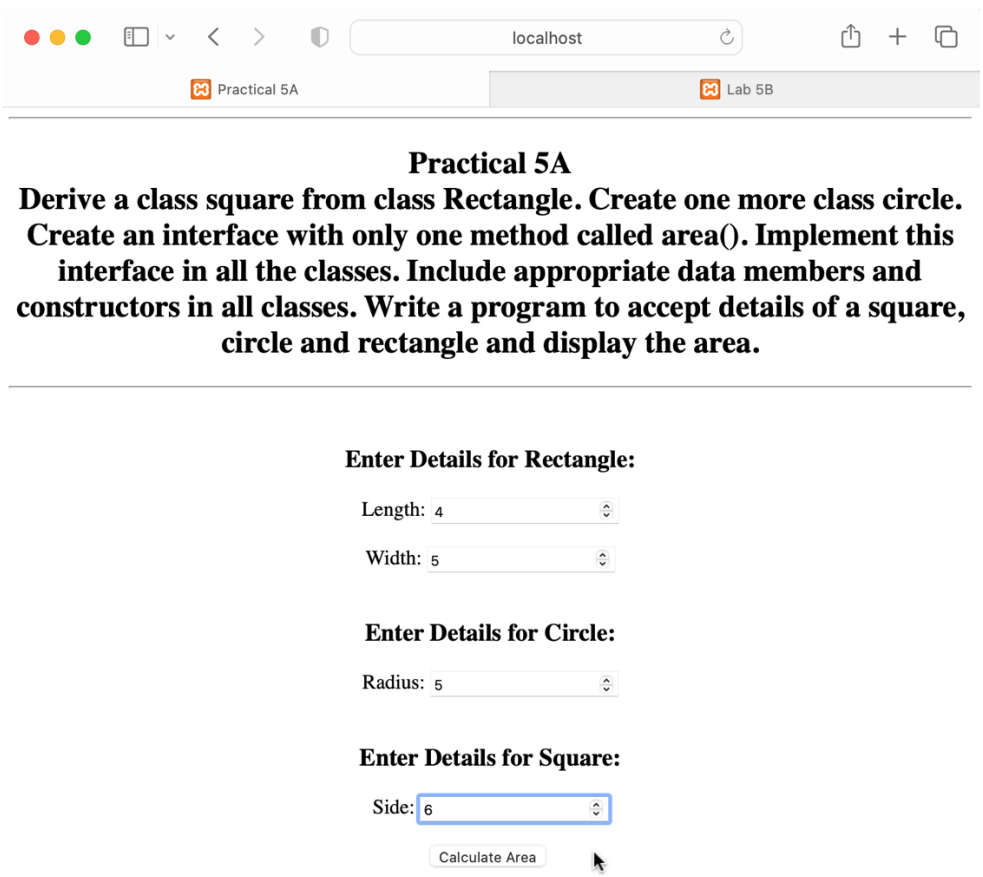
Methodology followed:

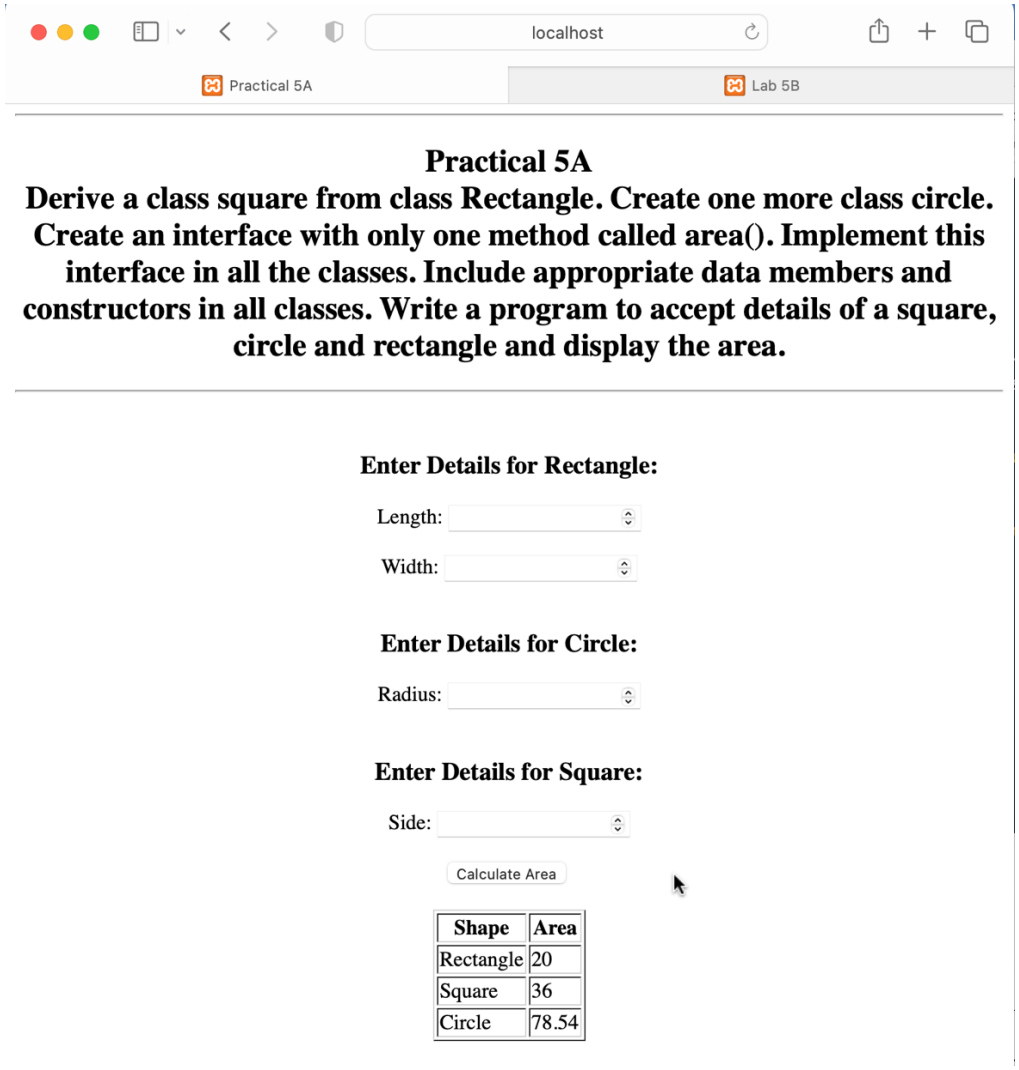
```
<!DOCTYPE html>
<html>
    <head>
        <title>Practical 5A</title>
        <link rel="stylesheet" type="text/css" href="./Style2.css">
    </head>
    <body>
        <form method="post">
            <center>
                <hr>
                <h2><b>Practical 5A<br>Derive a class square from class Rectangle.
                Create one more class circle. Create an interface with only one method called
area().
                Implement this interface in all the classes. Include appropriate data members and
constructors in all classes.
                Write a program to accept details of a square, circle and rectangle and display the
area.</b></h2>
                <hr>
                <br>
                <h3>Enter Details for Rectangle:</h3>
                Length: <input type="number" name="rLength"><br><br>
                Width: <input type="number" name="rWidth"><br><br>
                <h3>Enter Details for Circle:</h3>
                Radius: <input type="number" name="cRadius"><br><br>
                <h3>Enter Details for Square:</h3>
                Side: <input type="number" name="sSide"><br><br>
                <input type="submit" name="submit" value="Calculate Area">
                <br><br>

                <?php

                    interface Shape {
                        public function area();
                    }


                    class Rectangle implements Shape {
                        private $length;
                        private $width;
```

```
    public function __construct($length, $width) {
        $this->length = $length;
        $this->width = $width;
    }

    public function area() {
        return (int)$this->length * (int)$this->width;
    }
}


class Square extends Rectangle {

    public function __construct($side) {
        parent::__construct($side, $side);
    }
}


class Circle implements Shape {
    private $radius;

    public function __construct($radius) {
        $this->radius = $radius;
    }

    public function area() {
        return pi() * (int)$this->radius * (int)$this->radius;
    }
}



if(isset($_POST['submit'])) {

    $rLength = $_POST['rLength'];
    $rWidth = $_POST['rWidth'];
    $cRadius = $_POST['cRadius'];
    $sSide = $_POST['sSide'];

    $rectangle = new Rectangle($rLength, $rWidth);
    $circle = new Circle($cRadius);
    $square = new Square($sSide);


    echo "<table border=1>";
    echo "<tr><th>Shape</th>";
    echo "<th>Area</th></tr>";



    if ($rLength!="" and $rWidth!= ""){
```

```
                    echo "<tr><td>Rectangle</td><td>".$rectangle->area()."</td>";
                    }
                    if ($sSide != ""){
                    echo "<tr><td>Square</td><td>".$square->area()."</td>";
                    }
                    if ($cRadius != ""){
                    echo "<tr><td>Circle</td><td>". round($circle->area(),2)."</td>";

                    }

                    echo "</table>";

                }
            ?>
        </center>
    </form>
  </body>
</html>
```

Output:





(B)

Define a class Rectangle with its length and breadth. Provide appropriate constructor(s), which gives facility of constructing rectangle object passing value of length and breadth externally to constructor. Provide appropriate accessor & mutator methods to Rectangle class.

Provide methods to calculate area & to display all information of Rectangle. Create five Rectangle objects and call appropriate methods.

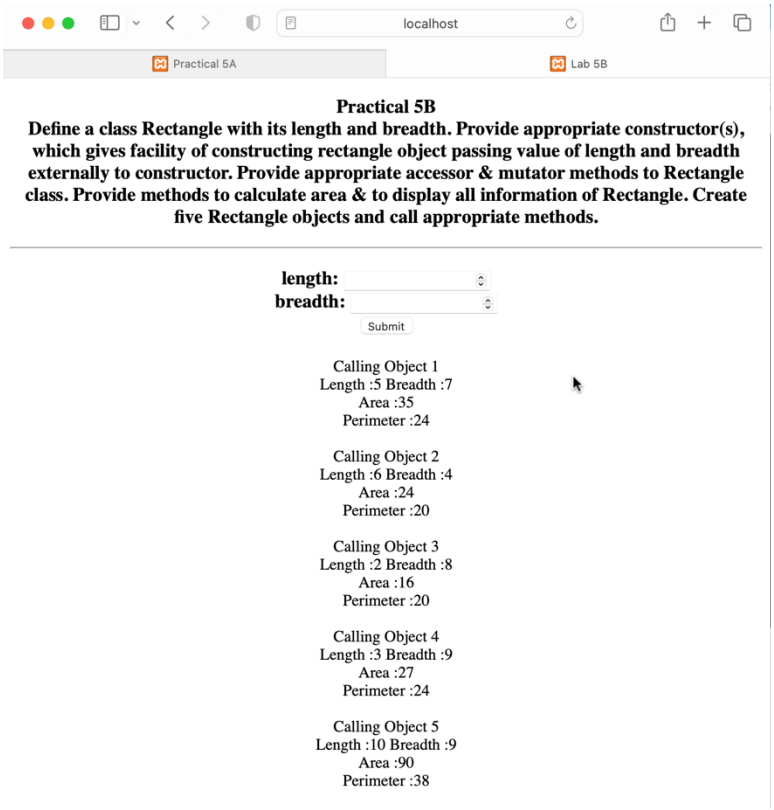
Methodology followed:

```
<!DOCTYPE html>
<html lang="en">

    <head>
        <title>Lab 5B</title>
    </head>
    <body>
        <center>

            <h3><b>Practical 5B<br>Define a class Rectangle with its length and breadth.
                    Provide appropriate constructor(s), which gives facility of constructing
rectangle object passing value of length and breadth externally to constructor.
                    Provide appropriate accessor & mutator methods to Rectangle class.
```

```
                Provide methods to calculate area & to display all information of Rectangle.
                Create five Rectangle objects and call appropriate methods.
        </b></h3>
        <hr>
        <div class="Form">
            <form method="post"  action="<?php echo $_SERVER['PHP_SELF']?>">
            <h3>
                length: <input type="Number" name="length"></input><br>
                breadth: <input type="Number" name="breadth"></input><br>
                <div class="Button">
                    <button type="submit">Submit</button>
            <h3>
                </div>
            </form>
        </div>
            <?php
            class Rectangle{
                public $length;
                public $breadth;
                function __construct($length, $breadth)
                {
                    $this->length = $length;
                    $this->breadth = $breadth;
                }
                public function area()
                {
                    return $this->length * $this->breadth;
                }
                public function perimeter()
                {
                    return 2*($this->length + $this->breadth);
                }
                public function setLength($length){
                    $this->length = $length;
                }

                public function setBreadth($breadth){
                    $this->breadth = $breadth;
                }

                public function getLength(){
                    return $this->length;
                }

                public function getBreadth(){
                    return $this->breadth;
                }

                public function __toString(){
            return ("Length :".$this->getLength()." "."Breadth :".$this->getBreadth()."<br>".
                    "Area :".$this->area()."<br>"."Perimeter :".$this->perimeter());
                }
            }
```

```
                $rect1= new Rectangle(5,7);
                $rect2= new Rectangle(6,4);
                $rect3= new Rectangle(2,8);
                $rect4= new Rectangle(3,9);
                $rect5= new Rectangle(10,9);

                echo "Calling Object 1<br>".$rect1;
                echo"<br><br>";
                echo "Calling Object 2<br>".$rect2;
                echo"<br><br>";
                echo "Calling Object 3<br>".$rect3;
                echo"<br><br>";
                echo "Calling Object 4<br>".$rect4;
                echo"<br><br>";
                echo "Calling Object 5<br>".$rect5;
                echo"<br><br>";




                if ($_SERVER["REQUEST_METHOD"] == "POST") {
                    $l = htmlspecialchars($_REQUEST['length']);
                    $b = htmlspecialchars($_REQUEST['breadth']);
                    if (empty($l)) {
                        echo "Enter Length";
                    }
                    if (empty($b)) {
                        echo "Enter Breadth";
                    }
                    else{
                        $rectangle = new Rectangle($l, $b);
                        echo $rectangle;
                    }
                }
            ?>
        </center>
    </body>

</html>
```

Output:



Practice Examples:

1. Different Access Modifiers:

```php
<?php

    class Person {
        public $name;
        protected $age;
        private $address;

        public function __construct($name, $age,
$address){
            $this->name = $name;
            $this->age = $age;
            $this->address = $address;
        }

        public function getName() {
            return $this->name;
        }

        protected function getAge() {
            return $this->age;
        }

        public function getAddress() {
            return $this->address;
        }
```

Output:

Dhyan
Private variable accesed from public method
Ahmedabad

Protected variable
24

When trying to access Protected variable

Dhyan
Fatal error: Uncaught Error: Cannot access protected property Person::$age in C:\xampp\htdocs\index.php:34 Stack trace: #0 {main} thrown in **C:\xampp\htdocs\index.php** on line **34**

When trying to access Private variable

Dhyan
Fatal error: Uncaught Error: Cannot access private property Person::$address in C:\xampp\htdocs\index.php:35 Stack trace: #0 {main} thrown in **C:\xampp\htdocs\index.php** on line **35**

```php
    }

    class Employee extends Person {
        public function getAgeFromParent() {
            return $this->getAge();
        }
    }

    $person = new Person("Dhyan", 20, "Ahmedabad");
    echo $person->name;
    //echo $person->age;
    echo $person->address;
    echo"<br>";
    //echo $person->getName();
    // echo $person->getAge();
    //echo"Private variable accesed from public
method<br>";
    //echo $person->getAddress();
    echo"<br>";
    echo"<br><br> Protected variable<br>";
    $employee = new Employee("Alisha", 24, "Boston");
    echo $employee->getAgeFromParent();

?>
```

2. Introspection Functions:

```php
<?php
    class ParentClass {
        public $a = 204;
        public function Method1() {
            echo "Hello, world!<br>";
        }
        private function Method2() {
            // do something private
        }
    }

    $obj = new ParentClass();

    // get class name
    echo "Class name: " . get_class($obj) . "<br>";

    // check if object is an instance of a class
    if ($obj instanceof ParentClass) {
        echo "Object is an instance of
".get_class($obj)."<br>";
    }

    // get class methods
    $methods = get_class_methods($obj);
```

Output:

Class name: ParentClass
Object is an instance of ParentClass
Class methods: Method1
Method2 exists in class
Class properties: a

```php
    echo "Class methods: " . implode(", ",
$methods) . "<br>";

    // check if a method exists in a class
    if (method_exists($obj, 'Method2')) {
        echo "Method2 exists in class<br>";
    }

    // get class properties
    $properties = get_class_vars('ParentClass');
    echo "Class properties: " . implode(", ",
array_keys($properties)) . "\n";


?>
```

Conclusion:

We learned about OOP concepts in PHP like inheritance, access modifiers, interfaces, methods and Introspection Functions.