PRACTICAL-9

OOP with PHP -3

Interface, Exception Handling and Practice of OOP fundamental

1. Create a class stack having a property called value(array), as well as having a push ,pop and display method. Display method displays the element of the array. [Note : for deleting elements replace the respective value with '0']. Create an object of stack and perform push, pop, and display operation on the object. Create proper HTML form which input the value to be pushed. Display stack element in table format. Take necessary variables required to perform operation. Throw an exception when trying to push the element from an empty array. Use Appropriate technique to show multiple operations via session or object serialization etc.

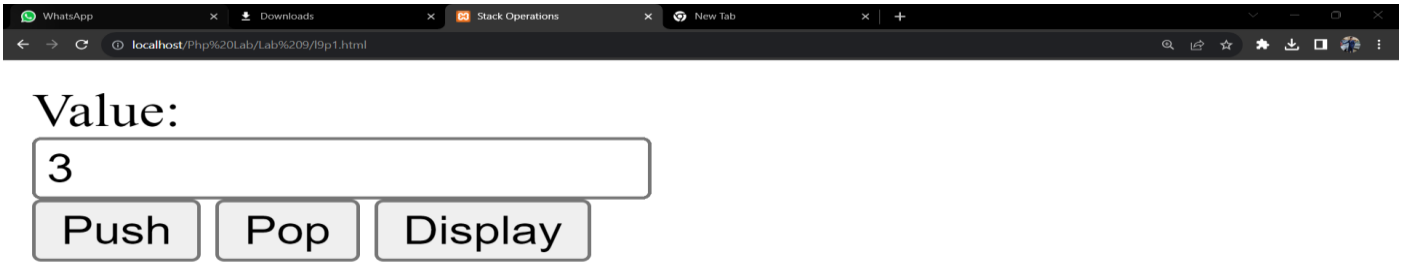
Q1.php

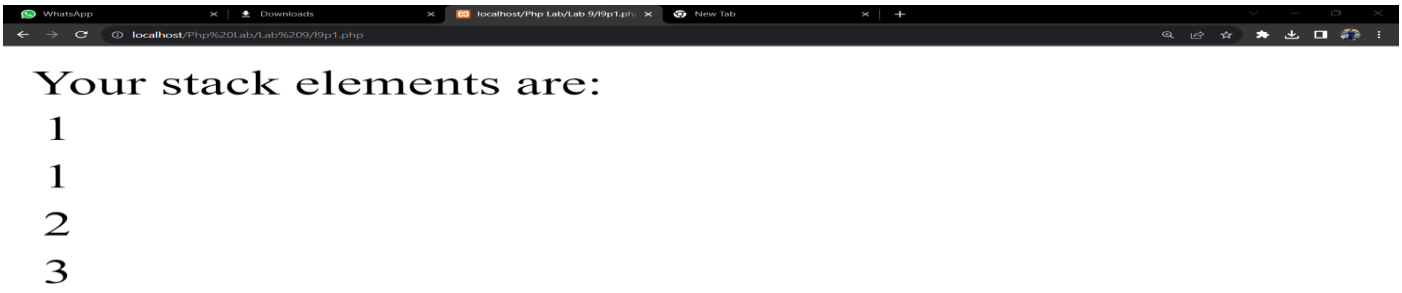
```
<!DOCTYPE html>
<html>
<head>
  <title>Stack Operations</title>
</head>
<body>
  <form action="l9p1.php" method="post">
    <label for="value">Value:</label><br>
    <input type="text" id="value" name="value"><br>
    <input type="submit" name="push" value="Push">
    <input type="submit" name="pop" value="Pop">
    <input type="submit" name="display" value="Display">
  </form>
</body>
</html>
```

```
<?php
class Stack {
  private $value = array();
  public function push($item) {
    array_push($this->value, $item);
    echo "Inserted successfully";
  }
  public function pop() {
    if(empty($this->value)) {
      throw new Exception("Stack is empty");
    } else {
      array_pop($this->value);
      echo "Element is popped out";
    }
  }
  public function display() {
    if(empty($this->value)) {
      echo "Your stack is empty";
    } else {
```

```
            echo "Your stack elements are: ";
            echo "<table>";
            foreach($this->value as $item) {
                echo "<tr><td>" . $item . "</td></tr>";
            }
            echo "</table>";
        }
    }
}
session_start();
if(!isset($_SESSION['stack'])) {
    $_SESSION['stack'] = new Stack();
}
$stack = $_SESSION['stack'];
if(isset($_POST['push'])) {
    $stack->push($_POST['value']);
} elseif(isset($_POST['pop'])) {
    try {
        $stack->pop();
    } catch(Exception $e) {
        echo 'Error: ' .$e->getMessage();
    }
} elseif(isset($_POST['display'])) {
    $stack->display();
}
?>
```

Output:





2. Create an interface called operation having a method plus(), minus(), div(), mul(). Create a class Arithmetic having property x and y Implement the method of interface to perform arithmetic operations on two values

and display answers. Create a class called String having property str1 and str2. Implements the method of operation interface Plus() : concat two strings and display the answer. Minus() : find out the position of str2 in str1. Mul() : find out the number of occurrences of str2 in str1. Div() : find out the last word form str1. Do appropriate handling of exceptions.

Q2.php

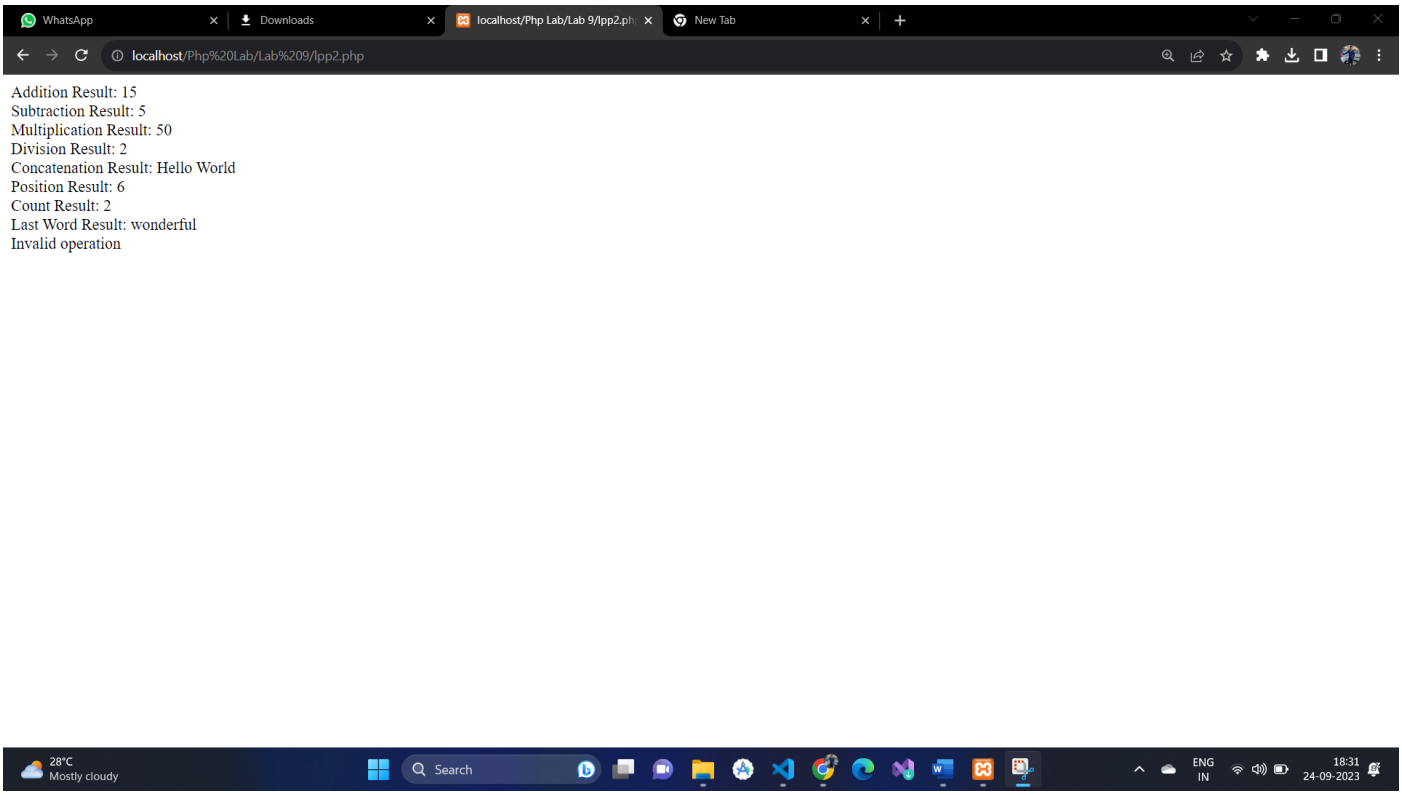
```
<?php
<?php

interface operation

{

   function plus();

   function minus();

   function mul();

   function div();

}

class Arithmatic implements operation

{

   public $x, $y;


   function __construct($x, $y, $op)

   {

      $this->x = $x;

      $this->y = $y;


      switch ($op) {

         case '+':

            echo "Addition Result: " . $this->plus() . "</br>";

            break;

         case '-':

            echo "Subtraction Result: " . $this->minus() . "</br>";

            break;

         case '*':

            echo "Multiplication Result: " . $this->mul() . "</br>";

            break;

         case '/':

            echo "Division Result: " . $this->div() . "</br>";
```

```
            break;
        default:
            echo "Invalid operation</br>";
            break;
      }
   }
   function plus()
   {
      return $this->x + $this->y;
   }
   function minus(){
      return $this->x - $this->y;
   }
   function mul(){
      return $this->x * $this->y;
   }
   function div(){
      if ($this->y == 0) {
         return "Division by zero error";
      }
      return $this->x / $this->y;
   }
}
class MyString implements operation
{
   public $str1, $str2;
   function __construct($str1, $str2, $op)
   {
      $this->str1 = $str1;
      $this->str2 = $str2;
      switch ($op) {
         case '+':
            echo "Concatenation Result: " . $this->plus() . "</br>";
            break;
```

```
            case '-':
                echo "Position Result: " . $this->minus() . "</br>";
                break;
            case '*':
                echo "Count Result: " . $this->mul() . "</br>";
                break;
            case '/':
                echo "Last Word Result: " . $this->div() . "</br>";
                break;
            default:
                echo "Invalid operation</br>";
                break;
        }
    }
    function plus()
    {
        return $this->str1 . $this->str2;
    }
    function minus(){
        return strpos($this->str1, $this->str2);
    }
    function mul()
    {
        return substr_count($this->str1, $this->str2);
    }
    function div(){
        $words = explode(' ', $this->str1);
        if (!empty($words)) {
            return end($words);
        } else {
            throw new Exception("No words found in the string.");
        }
    }
}
```

```
$arithAdd = new Arithmatic(10, 5, '+');

$arithSub = new Arithmatic(10, 5, '-');

$arithMul = new Arithmatic(10, 5, '*');

$arithDiv = new Arithmatic(10, 5, '/');

$strConcat = new MyString("Hello", " World", '+');

$strPosition = new MyString("Hello World", "World", '-');

$strCount = new MyString("Hello World, World is wonderful", "World", '*');

$strLastWord = new MyString("Hello World, World is wonderful", "", '/');

$invalidOperation = new Arithmatic(10, 5, '%');
```

output:



3. Create an HTML form which takes the input of Employee name, age. Form should have a drop down list to select the Employee type, as Developer /worker. Create php script for following:

- Class Employee
- Property : name,age
- Method : constructor() , display()

- Class Developer inherits Employee
- Property: skill[] array, salary, degree
- Method: constructor, disp_salary() , disp_skill() .

- Class Worker inherits Employee
- Property: woring_hr , per_hr_sprice

- Methods: constructor , calsalary() dispsalary() At the end based on drop down value it create the object of employee/manager / worker and display the details.

Q3.php

```
<?php
class Employee
{
   protected $name;
   protected $age;

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

   public function display()
   {
      echo "Name: $this->name<br>";
      echo "Age: $this->age<br>";
   }
}

class Developer extends Employee
{
   private $skills = [];
   private $salary;
   private $degree;

   public function __construct ($name, $age, $skills, $salary, $degree)
   {
      parent:: __construct($name, $age);
      $this->skills = $skills;
      $this->salary = $salary;
      $this->degree = $degree;
   }

   public function disp_salary()
```

```
    {
        echo "Salary: $this->salary<br>";
    }

    public function disp_skill()
    {
        echo "Skills: " . implode(", ", $this->skills) . "<br>";
        echo "Degree: $this->degree<br>";
    }
}

class Worker extends Employee
{
    private $working_hours;
    private $per_hour_price;

    public function __construct ($name, $age, $working_hours, $per_hour_price)
    {
        parent:: __construct($name, $age);
        $this->working_hours = $working_hours;
        $this->per_hour_price = $per_hour_price;
    }

    public function calsalary()
    {
        return $this->working_hours * $this->per_hour_price;
    }

    public function dispsalary()
    {
        echo "Salary: ". $this->calsalary() . "<br>";
    }
}

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = $_POST["name"];
    $age = $_POST["age"];
    $employee_type = $_POST["employee_type"];

    if ($employee_type == "Developer") {
        // Example Developer data
        $skills = ["PHP", "JavaScript"];
        $salary = 60000;
        $degree = "Bachelor's in Computer Science";

        $developer = new Developer ($name, $age, $skills, $salary, $degree);

        echo "<h2>Developer Details</h2>";
        $developer->display ();
        $developer->disp_skill();
        $developer->disp_salary();
```

```
    } elseif ($employee_type == "Worker") {
        // Example Worker data
        $working_hours = 40;
        $per_hour_price = 15;

        $worker = new Worker($name, $age, $working_hours, $per_hour_price);

        echo "<h2>Worker Details</h2>";
        $worker->display();
        $worker->dispsalary();
    } else {
        echo "Invalid employee type selected.";
    }
}
?>

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

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Employee Details</title>
    <!-- Include Bootstrap CSS -->
    <link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">
</head>

<body>
    <div class="container mt-5">
        <h2 class="text-center">Employee Details</h2>
        <form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
            <div class="form-group">
                <label for="name">Employee Name:</label>
                <input type="text" class="form-control" id="name" name="name" required>
            </div>

            <div class="form-group">
                <label for="age">Age:</label>
                <input type="number" class="form-control" id="age" name="age" required>
            </div>

            <div class="form-group">
                <label for="employee_type">Employee Type:</label>
                <select class="form-control" id="employee_type" name="employee_type">
                    <option value="Developer">Developer</option>
                    <option value="Worker">Worker</option>
                </select>
            </div>

            <button type="submit" class="btn btn-primary">Submit</button>
        </form>
```

```
    </div>

    <!-- Include Bootstrap JS and jQuery (for optional Bootstrap features) -->
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.3/dist/umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>

</html>
```

Output: