Assignment With Solutions
Module - Introduction Of OOPS

1. Write a simple PHP class which displays the following string.
'MyClass class has initialized !'

Solution:

```
<?php
class MyClass {
  public function __construct()
  {
   echo 'MyClass class has initialized !'."\n";
  }
 }
$userclass = new MyClass;
?>
```

2. Write a simple PHP class which displays an introductory message like "Hello All, I am Scott", where "Scott" is an argument value of the method within the class.

```
<?php
class user_message {
 public $message = 'Hello All, I am ';
 public function introduce($name)
 {
  return $this->message.$name;
 }
 }
$mymessage = New user_message();
echo $mymessage->introduce('Scott')."\n";
?>
```

3. Write a PHP class that calculates the factorial of an integer.

```
<?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();
?>
```

4. Write a PHP class that sorts an ordered integer array with the help of sort() function.

Sample array : array(11, -2, 4, 35, 0, 8, -9)
Output : Array ([0] => -9 [1] => -2 [2] => 0 [3] => 4 [4] => 8 [5] => 11 [6] => 35)

```
<?php
class array_sort
{
   protected $_asort;

   public function __construct(array $asort)
    {
      $this->_asort = $asort;
    }
   public function alhsort()
    {
      $sorted = $this->_asort;
      sort($sorted);
      return $sorted;
     }
}
$sortarray = new array_sort(array(11, -2, 4, 35, 0, 8, -9));
print_r($sortarray->alhsort())."\n";
?>
```

5. Calculate the difference between two dates using PHP OOP approach.

Sample Dates : 1981-11-03, 2013-09-04

Expected Result : Difference : 31 years, 10 months, 1 days

```
<?php

$sdate = new DateTime("1981-11-03");
$edate = new DateTime("2013-09-04");
$interval = $sdate->diff($edate);
echo "Difference : " . $interval->y . " years, " . $interval->m." months, ".$interval->d." days
";
?>
```

6. Write a PHP Calculator class which will accept two values as arguments, then add them, subtract them, multiply them together, or divide them on request.

For example :

```
$mycalc = new MyCalculator( 12, 6);
echo $mycalc- > add(); // Displays 18
echo $mycalc- > multiply(); // Displays 72
```

```
<?php
class MyCalculator {
private $_fval, $_sval;
public function __construct( $fval, $sval ) {
$this->_fval = $fval;
$this->_sval = $sval;
}
public function add() {
return $this->_fval + $this->_sval;
}
public function subtract() {
return $this->_fval - $this->_sval;
}
public function multiply() {
return $this->_fval * $this->_sval;
}
public function divide() {
return $this->_fval / $this->_sval;
}
}
$mycalc = new MyCalculator(12, 6);
echo $mycalc-> add()."\n"; // Displays 18
echo $mycalc-> multiply()."\n"; // Displays 72
echo $mycalc-> subtract()."\n"; // Displays 6
echo $mycalc-> divide()."\n"; // Displays 2
?>
```

7. Write a PHP script to convert a string to Date and DateTime.
Sample Date : '12-08-2004'
Expected Output : 2004-12-08
Note : PHP considers '/' to mean m/d/Y format and '-' to mean d-m-Y format.

```
<?php
$dt = DateTime::createFromFormat('m-d-Y', '12-08-2004')->format('Y-m-d');
echo $dt;
?>
```