DCIT 201: Programming 1

Assignment 7

**NAME: AKAFARI TOUFIQ SIA**

**STUDENT ID: 11336088**

1. **Create a class called *Book*. Give it the following protected attributes:** 
   1. **title**
   2. **yearPublished**
   3. **author**

Provide getters and setters for these attributes. Eg. *getTitle*()

Create a constructor for your Book class, which accepts arguments for *title*, *yearPublished* and *author*. Make the constructor arguments optional, by supplying defaults. Finally, add a method called *summary*() that returns the *title*, *yearPublished* and *author* attributes formatted in a string. Write a test script that creates an instance of Book, and echoes the result of calling *summary*().

**Solution**

<?php

class Book {

// Protected attributes

protected $title;

protected $yearPublished;

protected $author;

// Constructor with optional arguments and defaults

public function \_\_construct($title = 'Unknown Title', $yearPublished = 'Unknown Year', $author = 'Unknown Author') {

$this->title = $title;

$this->yearPublished = $yearPublished;

$this->author = $author;

}

// Getter and Setter for title

public function getTitle() {

return $this->title;

}

public function setTitle($title) {

$this->title = $title;

}

// Getter and Setter for yearPublished

public function getYearPublished() {

return $this->yearPublished;

}

public function setYearPublished($yearPublished) {

$this->yearPublished = $yearPublished;

}

// Getter and Setter for author

public function getAuthor() {

return $this->author;

}

public function setAuthor($author) {

$this->author = $author;

}

// Method to get a summary of the book

public function summary() {

return "Title: {$this->title}, Year Published: {$this->yearPublished}, Author: {$this->author}";

}

}

// Test script

$book = new Book("The Great Gatsby", 1925, "F. Scott Fitzgerald");

echo $book->summary();

?>

1. **Write a PHP class *Rectangle* that has properties for *length* and *width*. Implement methods to calculate the rectangle's *area* and *perimeter*.**

**Solution**

<?php

class Rectangle {

// Properties

private $length;

private $width;

// Constructor

public function \_\_construct($length, $width) {

$this->length = $length;

$this->width = $width;

}

// Getter and Setter for length

public function getLength() {

return $this->length;

}

public function setLength($length) {

$this->length = $length;

}

// Getter and Setter for width

public function getWidth() {

return $this->width;

}

public function setWidth($width) {

$this->width = $width;

}

// Method to calculate the area of the rectangle

public function calculateArea() {

return $this->length \* $this->width;

}

// Method to calculate the perimeter of the rectangle

public function calculatePerimeter() {

return 2 \* ($this->length + $this->width);

}

}

// Example usage

$rectangle = new Rectangle(5, 8);

echo "Length: " . $rectangle->getLength() . " units" . PHP\_EOL;

echo "Width: " . $rectangle->getWidth() . " units" . PHP\_EOL;

echo "Area: " . $rectangle->calculateArea() . " square units" . PHP\_EOL;

echo "Perimeter: " . $rectangle->calculatePerimeter() . " units" . PHP\_EOL;

?>

1. **Write a PHP class called *Shape* with an abstract method *calculateArea*(). Create two subclasses, *Triangle* and *Rectangle*, that implement the *calculateArea*() method.**

**Solution**

<?php

// Abstract Shape class

abstract class Shape {

// Abstract method for calculating the area

abstract public function calculateArea();

}

// Triangle class, a subclass of Shape

class Triangle extends Shape {

// Properties

private $base;

private $height;

// Constructor

public function \_\_construct($base, $height) {

$this->base = $base;

$this->height = $height;

}

// Method to calculate the area of the triangle

public function calculateArea() {

return 0.5 \* $this->base \* $this->height;

}

}

// Rectangle class, a subclass of Shape

class Rectangle extends Shape {

// Properties

private $length;

private $width;

// Constructor

public function \_\_construct($length, $width) {

$this->length = $length;

$this->width = $width;

}

// Method to calculate the area of the rectangle

public function calculateArea() {

return $this->length \* $this->width;

}

}

// Example usage

$triangle = new Triangle(4, 6);

$rectangle = new Rectangle(5, 8);

echo "Triangle Area: " . $triangle->calculateArea() . " square units" . PHP\_EOL;

echo "Rectangle Area: " . $rectangle->calculateArea() . " square units" . PHP\_EOL;

?>

1. **Write a PHP class called *Vehicle* with properties like *brand*, *model*, and *year*. Implement a method to display the vehicle details.**

**Solution**

<?php

class Vehicle {

// Properties

public $brand;

public $model;

public $year;

// Constructor

public function \_\_construct($brand, $model, $year) {

$this->brand = $brand;

$this->model = $model;

$this->year = $year;

}

// Method to display vehicle details

public function displayDetails() {

echo "Brand: {$this->brand}\n";

echo "Model: {$this->model}\n";

echo "Year: {$this->year}\n";

}

}

// Example usage

$car = new Vehicle("Toyota", "Camry", 2022);

$car->displayDetails();

?>

1. **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**

**Solution**

<?php

class MyCalculator {

// Properties

private $num1;

private $num2;

// Constructor

public function \_\_construct($num1, $num2) {

$this->num1 = $num1;

$this->num2 = $num2;

}

// Method to add two numbers

public function add() {

return $this->num1 + $this->num2;

}

// Method to subtract two numbers

public function subtract() {

return $this->num1 - $this->num2;

}

// Method to multiply two numbers

public function multiply() {

return $this->num1 \* $this->num2;

}

// Method to divide two numbers

public function divide() {

if ($this->num2 != 0) {

return $this->num1 / $this->num2;

} else {

return "Cannot divide by zero.";

}

}

}

// Example usage

$mycalc = new MyCalculator(12, 6);

echo $mycalc->add(); // Displays 18

echo $mycalc->subtract(); // Displays 6

echo $mycalc->multiply(); // Displays 72

echo $mycalc->divide(); // Displays 2

?>