**DCIT 201 – PROGRAMMING ONE**

**1128 3519 – STUDENT ID**

1. <?php

class Book {

protected $title;

protected $yearPublished;

protected $author;

public function \_\_construct($title = "", $yearPublished = "", $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: {$this->title}\nYear Published: {$this->yearPublished}\nAuthor: {$this->author}";

}

}

$

?>

1. <?php

class Rectangle {

private $length;

private $width;

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);

}

}

$rectangle = new Rectangle(5, 8);

echo "Rectangle Area: " . $rectangle->calculateArea() . "\n";

echo "Rectangle Perimeter: " . $rectangle->calculatePerimeter();

?>

1. <?php

// Abstract class Shape

abstract class Shape {

// Abstract method to calculate the area

abstract public function calculateArea();

}

// Subclass Triangle

class Triangle extends Shape {

private $base;

private $height;

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

$this->base = $base;

$this->height = $height;

}

// Implementation of calculateArea() for Triangle

public function calculateArea() {

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

}

}

// Subclass Rectangle

class Rectangle extends Shape {

private $length;

private $width;

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

$this->length = $length;

$this->width = $width;

}

// Implementation of calculateArea() for Rectangle

public function calculateArea() {

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

}

}

// Example usage

$triangle = new Triangle(4, 6);

$rectangle = new Rectangle(5, 8);

// Calculate and output the areas

echo "Triangle Area: " . $triangle->calculateArea() . "\n";

echo "Rectangle Area: " . $rectangle->calculateArea();

?>

1. <?php

class Vehicle {

private $brand;

private $model;

private $year;

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

$this->brand = $brand;

$this->model = $model;

$this->year = $year;

}

// Getter and setter for brand

public function getBrand() {

return $this->brand;

}

public function setBrand($brand) {

$this->brand = $brand;

}

// Getter and setter for model

public function getModel() {

return $this->model;

}

public function setModel($model) {

$this->model = $model;

}

// Getter and setter for year

public function getYear() {

return $this->year;

}

public function setYear($year) {

$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);

// Display vehicle details

$car->displayDetails();

?>

1. <?php

class MyCalculator {

private $value1;

private $value2;

public function \_\_construct($value1, $value2) {

$this->value1 = $value1;

$this->value2 = $value2;

}

// Method to add the two values

public function add() {

return $this->value1 + $this->value2;

}

// Method to subtract the second value from the first

public function subtract() {

return $this->value1 - $this->value2;

}

// Method to multiply the two values

public function multiply() {

return $this->value1 \* $this->value2;

}

// Method to divide the first value by the second (assuming the second value is not zero)

public function divide() {

if ($this->value2 != 0) {

return $this->value1 / $this->value2;

} else {

return "Cannot divide by zero.";

}

}

}

// Example usage

$mycalc = new MyCalculator(12, 6);

// Display results

echo "Addition: " . $mycalc->add() . "\n";

echo "Subtraction: " . $mycalc->subtract() . "\n";

echo "Multiplication: " . $mycalc->multiply() . "\n";

echo "Division: " . $mycalc->divide() . "\n";

?>