UNIVERSIDAD DE DAGUPAN

SCHOOL OF INFORMATION TECHNOLOGY EDUCATION

ITP03 | OBJECT-ORIENTED PROGRAMMING

WEEK2 | LABORATORY ACTIVITY 2

Name : _Jol Cydrik F. Dela Cruz _____

Year, Course & blk : ___21 – ITE - 05_____

Subject : __OOP_____

Date : __12/09/2024_____

SCORE:

WEEK2 | Laboratory Activity 2: Comprehensive Application of Access Modifiers, Constructors, Destructors, and Method Overloading

LABORATORY ACTIVITY 2

```
<?php
class Book {
   public $title;
   protected $author;
   private $price;

   public function __construct($title, $author, $price) {
      $this->title = $title;
      $this->author = $author;
      $this->price = $price;
   }

   public function getDetails() {
      return "Title: $this->title, Author: $this->author, Price: \$$this->price";
   }

   public function setPrice($price) {
      $this->price = $price;
   }

   public function __call($method, $args) {
      if ($method == 'updateStock') {
         echo "Stock updated for '{$this->title}' with arguments: " . implode(", ", $args);
      } else {
         echo "Method '$method' does not exist.\n";
      }
   }
}

class Library {
   private $books = [];
   public $name;
```

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

    public function addBook(Book $book) {
        $this->books[] = $book;
    }

    public function removeBook($title) {
        foreach ($this->books as $key => $book) {
            if ($book->title == $title) {
                unset($this->books[$key]);
                echo "Book '$title' removed from the library";
                return;
            }
        }
        echo "Book '$title' not found in the library.\n";
    }

    public function listBooks() {
        if (empty($this->books)) {
            echo "No books in the library";
            return;
        }
        echo "Books in the library after removal:\n";
        foreach ($this->books as $book) {
            echo $book->getDetails()."\n";
        }
    }

    public function __destruct() {
        echo "The library '{$this->name}' is now closed.\n";
    }
}

$book1 = new Book("The Great Gatsby", "F. Scott Fitzgerald", 12.99);
$book2 = new Book("1984", "George Orwell", 8.99);

$library = new Library("City Library");
$library->addBook($book1);
$library->addBook($book2);

$book1->updateStock(50);

echo "\n";

$library->listBooks();

$book1->setPrice(12.99);
$library->removeBook("1984");
```

```
echo "\n";

$library->listBooks();

?>
```

EXPLANATION

•Constructor

The constructor takes three parameters: $title, $author, and $price. These values are assigned to the object's properties when a Book object is created.

```
public function __construct($title, $author, $price) {
   $this->title = $title;
   $this->author = $author;
   $this->price = $price;
}
```

•Methods

getDetails(): This method returns a formatted string with the book's title, author, and price.
setPrice(): This method allows modifying the price of the book.

•Magic Methods

This is a magic method that is triggered when an undefined or inaccessible method is called.

```
public function __call($method, $args) {
   if ($method == 'updateStock') {
      echo "Stock updated for '{$this->title}' with arguments: " . implode(", ", $args);
   } else {
      echo "Method '$method' does not exist.\n";
   }
}
```

• Composition

The Library class has a $books property, which is an array that stores Book objects. The addBook() method demonstrates how a Book object is added to the library.

```
public function addBook(Book $book) {
   $this->books[] = $book;
}
```

• Destructor

The destructor is a method that is invoked when an object is about to be destroyed, or when the script ends.

```
public function __destruct() {
   echo "The library '{$this->name}' is now closed.\n";
}
```