### Code Snippet A

php

class Book {

private $price;

protected $title;

function \_construct(string $param1 = "PHP Basics", int $param2 = 380) {

$this->title = $param1;

$this->price = $param2;

}

public function getPrice() {

echo "Price: $this->price

";

}

public function getTitle() {

echo "Title: $this->title

";

}

}

class mybook {

public function getmytitle($b) {

echo "Title: $b->title

";

}

}

$b1 = new mybook();

$b = new Book();

$b1->getmytitle($b);

?>

### Code Snippet B

php

class Book {

private $price;

protected $title;

function \_construct(string $param1 = "PHP Basics", int $param2 = 380) {

$this->title = $param1;

$this->price = $param2;

}

public function getPrice() {

echo "Price: $this->price

";

}

public function getTitle() {

return $this->title; // Returning the title

}

}

class mybook {

public function getmytitle($b) {

echo "Title: " . $b->getTitle() . "

"; // Accessing the protected

}

}

$b1 = new mybook();

$b = new Book();

$b1->getmytitle($b);

?>

Differences in Approach

1. Accessing the Title Property:

- Snippet A: Directly accesses the protected property

$title

using

$b->title

. This is incorrect because

$title

is protected and cannot be accessed directly from outside the class.

- Snippet B: Uses the

getTitle()

method to access the protected property. This is the correct approach as it adheres to encapsulation principles.

2. Return Statement:

- Snippet A: The

getTitle()

method echoes the title directly.

- Snippet B: The

getTitle()

method returns the title, allowing for more flexible use of the title value (e.g., concatenation or further processing).

### Access Modifiers

- Both snippets use the same access modifiers (

private

for

$price

and

protected

for

$title

), but the way they access these properties differs significantly.

- In Snippet A, the direct access to the protected property violates object-oriented principles, while Snippet B correctly uses a public method to access the protected property.

Potential Errors

1. Constructor Name:

- Both snippets have a typo in the constructor name. It should be

\_\_construct

(with two underscores) instead of

\_construct

. This means that the constructor will not be called when an object of the

Book

class is instantiated.

2. Incorrect Method Call in Snippet A:

- In Snippet A, the line

$b1->getmytitle($b);

attempts to access the protected property directly, which will lead to an error. This is corrected in Snippet B by using

$b->getTitle()

.

3. Syntax Errors:

- In Snippet B, there is a stray character

4

after the constructor definition, which should be removed.

- The method call

$b1->aetmvtitle(sb);

in Snippet B is also incorrect and should be corrected to

$b1->getmytitle($b);

.