

Tổng thể nội dung đề ôn tập

Đề bao phủ 4 mảng kiến thức cốt lõi:

1. **PHP cơ bản – Mảng – Hàm**
 1. Mảng 1 chiều, đa chiều
 2. Duyệt mảng, unset, isset
 3. Truyền tham trị, tham chiếu
 4. \$_GET, \$_POST, \$_REQUEST
2. **Lập trình hướng đối tượng (OOP) trong PHP**
 1. Constructor
 2. access modifier (private, protected, public)
 3. static, const
 4. Ké thừa, gọi phương thức
3. **PHP + CSDL (PDO)**
 1. Kết nối CSDL
 2. Prepared Statement
 3. fetch / fetchAll
 4. Chống SQL Injection
4. **Tư duy đọc code – suy luận kết quả**
 1. Trắc nghiệm
 2. Đoán output
 3. Hiểu bản chất, không học vẹt.

ÔN TẬP PHP – WEB PROGRAMMING

1. Mảng & GET

Cho mảng sau:

```
$students = [  
    ['id'=>1, 'name'=>'An', 'score'=>7.5],  
    ['id'=>2, 'name'=>'Bình', 'score'=>9.0],  
    ['id'=>3, 'name'=>'Chi', 'score'=>6.8]  
];
```

5. Hiển thị danh sách sinh viên dạng bảng HTML
 6. Thêm liên kết sắp xếp theo score tăng / giảm thông qua \$_GET
-

2. Trắc nghiệm & lý thuyết

2.1 Đoạn code sau in ra kết quả gì?

```
$a = [ "2"=>2, "5"=>5, 7];  
$a[] = 10;  
echo array_sum(array_keys($a));
```

- A. 14
 - B. 24
 - C. 17
 - D. 12
-

2.2 Kết quả của đoạn code?

```
$a = [1,2,3,4,5];  
$s = 0;  
foreach($a as $k=>$v) {  
    if($v % 2 == 0)  
        $s += $k;  
}  
echo $s;
```

2.3 Phát biểu nào đúng về `protected`?

- A. Truy cập ở mọi nơi
 - B. Chỉ truy cập trong class
 - C. Truy cập trong class và class con
 - D. Chỉ truy cập ngoài class
-

3. OOP

Cho đoạn code:

```
class Test {  
    public static $count = 0;  
    public $x;  
  
    function __construct($x){  
        $this->x = $x;  
        self::$count++;  
    }  
  
    function show(){
```

```
        echo $this->x . " - " . self::$count;
    }
}

$a = new Test(5);
$b = new Test(10);
$a->show();
```

1. Kết quả in ra?
 2. Giải thích vì sao \$count không phải là 1?
-

4. PDO & SQL

4.1 Viết đoạn code kết nối PDO tới CSDL school

4.2 Viết prepared statement thêm 1 sinh viên (id, name) vào bảng students

5. Đọc code & REQUEST

URL:

?x=1&y=2&x=5&z[]=3

Code:

```
$x = $_REQUEST['x'];
$y = $_GET['y'];
$z = $_REQUEST['z'][0];
echo "$x-$y-$z";
```

Kết quả là gì? Vì sao?

✓ PHÂN TÍCH

1 – Gợi ý đáp án

```
if(isset($_GET['sort'])){  
    usort($students, function($a,$b){  
        return ($_GET['sort']=='asc')  
            ? $a['score'] <=> $b['score']  
            : $b['score'] <=> $a['score'];  
    });  
}
```

📌 Giải thích

- usort() cho phép sắp xếp mảng nhiều chiều
 - Logic xử lý phải đặt **trước khi hiển thị HTML**
-

2.1 Phân tích

- Keys: "2" , "5" , 0 , 6
 - Tổng = 2 + 5 + 0 + 6 = 13 ✗ → sai
👉 Lưu ý: "2" và "5" được ép kiểu thành số
-

2.2 Giải thích

- Giá trị chẵn: 2 (index 1), 4 (index 3)
 - Tổng index: 1 + 3 = 4
-

2.3 🔒 protected:

- Dùng trong **class cha**
 - Truy cập được từ **class con**
 - Không truy cập từ bên ngoài
-

3.1 Kết quả: 5-2

3.2 Giải thích

- \$count là static → dùng chung cho toàn class
 - Tạo 2 object → \$count = 2
 - \$a->show() in:
 - \$this->x = 5
 - self::\$count = 2
-

4.1 Kết nối PDO

```
$pdo = new PDO(  
    "mysql:host=localhost;dbname=school",  
    "root", ""  
) ;
```

4.2 Prepared Statement

```
$sql = "INSERT INTO students VALUES (?, ?)";  
$stmt = $pdo->prepare($sql);  
$stmt->execute([1, 'Nam']);
```

📌 Vì sao dùng prepare?

- Chống SQL Injection
 - Chạy nhiều lần hiệu quả
-

5. Kết quả: 5-2-3

Giải thích

- \$_REQUEST['x'] → lấy giá trị cuối
- \$_GET['y'] = 2
- z[] là mảng → phần tử đầu là 3