OOP Recap for Interviews

1. OOP Principles

- Encapsulation: Binding data and methods together, and restricting direct access to some of the object's components.
- Abstraction: Hiding the complex implementation details and showing only the necessary features of the object.
- Inheritance: Mechanism where a new class inherits properties and behavior (methods) from an existing class.
- Polymorphism: The ability of different classes to respond to the same function call in different ways (method overriding or overloading).

2. Example Code (PHP)

```
class Animal {
  public function makeSound() {
    echo "Some sound";
  }
}

class Dog extends Animal {
  public function makeSound() {
    echo "Bark";
  }
}

$pet = new Dog();
$pet->makeSound(); // Outputs: Bark
```

3. SOLID Principles

- S: Single Responsibility - A class should have one reason to change.
- O: Open/Closed - Classes should be open for extension, closed for modification.
- L: Liskov Substitution - Subclasses should be substitutable for their base classes.
- I: Interface Segregation - Use many specific interfaces, not one general-purpose interface.

- D: Dependency Inversion - Depend on abstractions, not concretions.

4. Common Interview Questions

- What is the difference between abstract class and interface?
- How does polymorphism work in OOP?
- Explain encapsulation with a real-life example.
- When should inheritance be used over composition?
- What are access modifiers in OOP?
- How do SOLID principles improve code quality?

5. Final Tips

- Revise your previous projects that used OOP.
- Practice explaining OOP concepts clearly and with examples.
- Use diagrams or whiteboard if allowed during the interview.
- Don't just explain theory - show practical understanding.