Manish Jaiswal

OOP Interview

Encapsulation: The Secret to Secure and Maintainable Code?

① What is Encapsulation?

Encapsulation bundles **data & methods into a class** while restricting direct access.

Why does it matter?:

- Prevents unauthorized access.
- Reduces unintended modifications
- Improves maintainability

💡 **Analogy:** A vending machine hides internal mechanics and exposes only necessary controls.

② How Encapsulation Works?

🔐 Access Modifiers:

Modifier	Scope
`private`	Only inside the class
`protected`	Inside class & subclasses
`internal`	Inside the same assembly
`public`	Accessible from anywhere

③ Example

```
1  public class BankAccount
2  {
3      private decimal balance; // Hidden data
4
5      public void Deposit(decimal amount)
6      {
7          if (amount > 0) balance += amount;
8      }
9
10     public decimal GetBalance() => balance; // Controlled access
11 }
```

④ Why Direct Access is Dangerous?

```
// 🚨 Bad Practice (No Encapsulation)
public class BankAccount { public decimal balance; }
var acc = new BankAccount();
acc.balance = -1000; // ❌ Allowed but wrong!

// ✅ Encapsulated Approach

public class BankAccount
{
    private decimal balance;
    public decimal Balance => balance; // Read-only access
    public void Deposit(decimal amt) { if (amt > 0) balance += amt; }
}
```

Now, negative balance is impossible!

⑤ Common Mistakes

Exposing fields directly:

- Use private fields with getters/setters.

Making setters public:

- Restrict modification access.

Overusing getters & setters:

- Expose only what's needed.

🚀 **Best Practice:**

- Use readonly properties for fixed data.
- Keep business logic inside the class to maintain control.

6 Encapsulation vs. Abstraction

Feature	Encapsulation	Abstraction
Purpose	Hides data & controls access	Hides implementation complexity
How?	`private`, `protected`, `public`	Abstract classes, interfaces
Example	Bank balance is private	ATM interface hides transaction logic

⑦ Real-World Use Cases

💳 **Banking Systems:**

Prevent unauthorized access to account balances.

🏢 **Enterprise Apps:**

Protect business logic within domain models.

🎮 **Game Dev:**

Encapsulate player stats to avoid unintended modifications.

8 Common Questions ❓

Top Interview Questions

- What is encapsulation, and why is it important?
- How is it different from abstraction?
- Can you give a real-world example?

Let me know if this helped!