OOP Pillars Cheat Sheet - Gauri Patil

1. Encapsulation - 'Protect Secrets'

Encapsulation means bundling data and methods in a class and hiding internal details.

Independent concept.

Use private variables (e.g., __balance)

Analogy: Sita hides her secret sword techniques.

2. Abstraction - 'Show Button, Hide Code'

Abstraction means showing only the necessary parts of the system.

Independent concept.

Use abstract classes or minimal interfaces.

Analogy: You see 'Attack' button but not how Sita trains.

3. Inheritance - 'Pass the Torch'

Inheritance allows one class to reuse code from another class.

Independent concept.

Use class B(A): to inherit.

Analogy: Sita inherits power from ancestors.

4. Polymorphism - 'Change Style'

Polymorphism allows the same method to behave differently based on context.

Not independent (needs inheritance).

Use method overriding.

Analogy: Sita adapts her fighting style for each demon.

Dependencies Summary

OOP Pillars Cheat Sheet - Gauri Patil

- Encapsulation: No dependencies
- Abstraction: No dependencies
- Inheritance: No dependencies
- Polymorphism: Depends on Inheritance

"Think like a designer, code like a warrior!" - Gauri Patil's OOP mantra