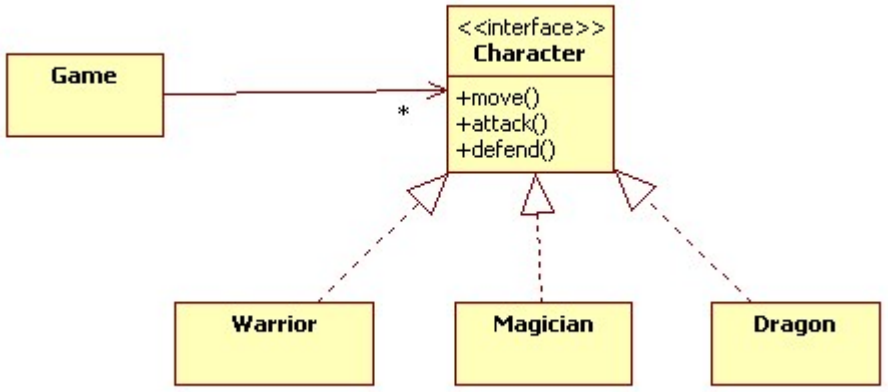
Deeper Understanding of SRP

- A class should have only one reason to change.
- Example: Logger and OrderProcessor responsibilities remain separate.

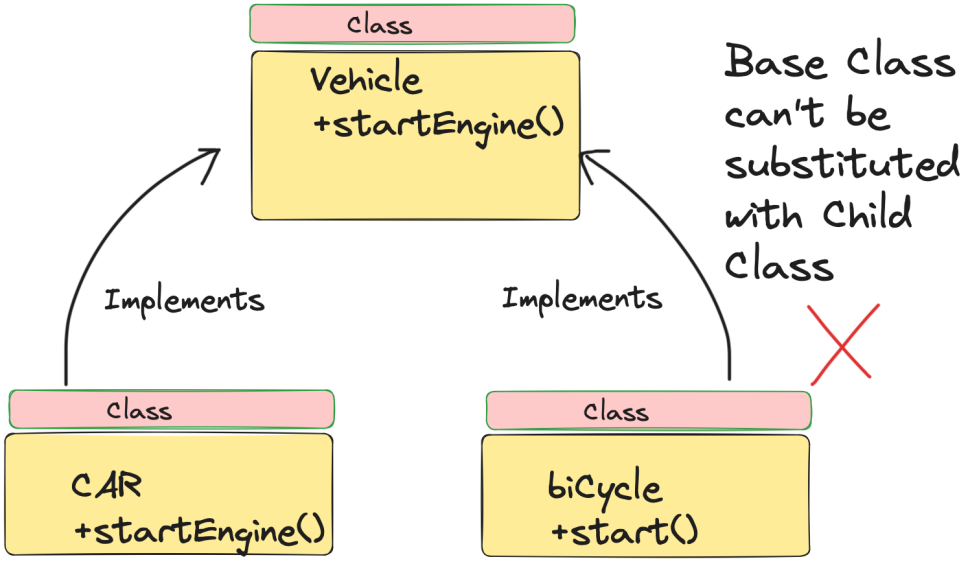
Applying OCP

- Design for extension without modifying existing code.
- Example: Add new shapes to a drawing app without altering existing code.



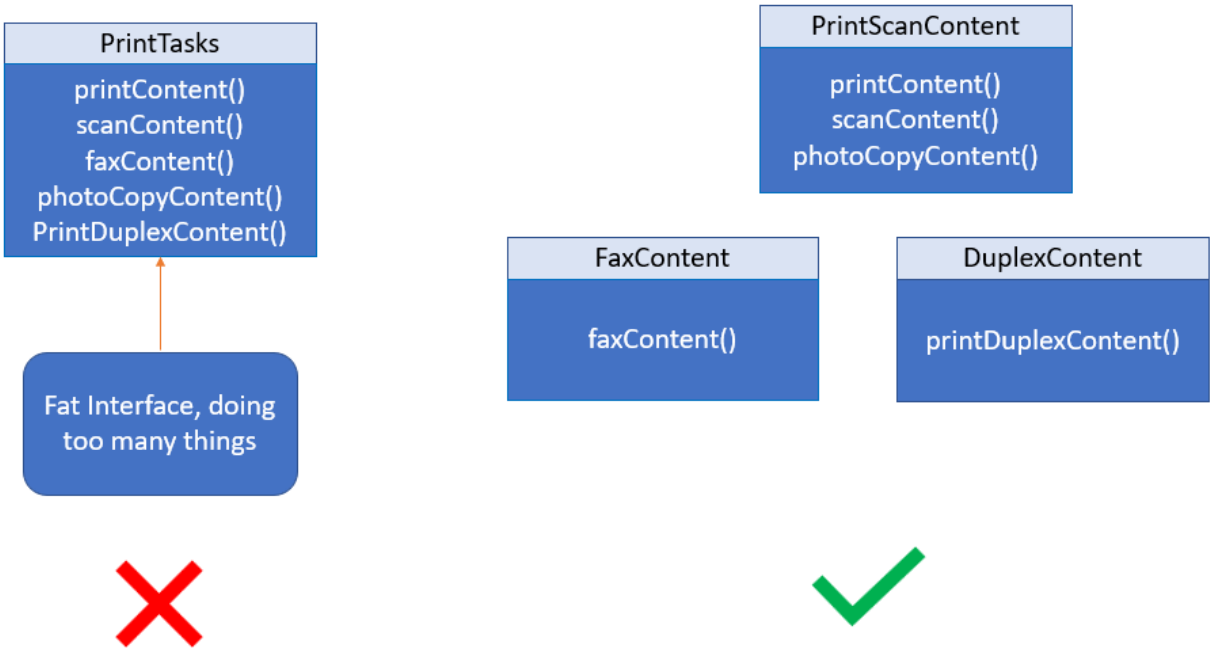
LSP in Practice

- Derived classes substitute their parents without altering behavior.
- Example: Vehicles like Cars and Bicycles share the same interface.



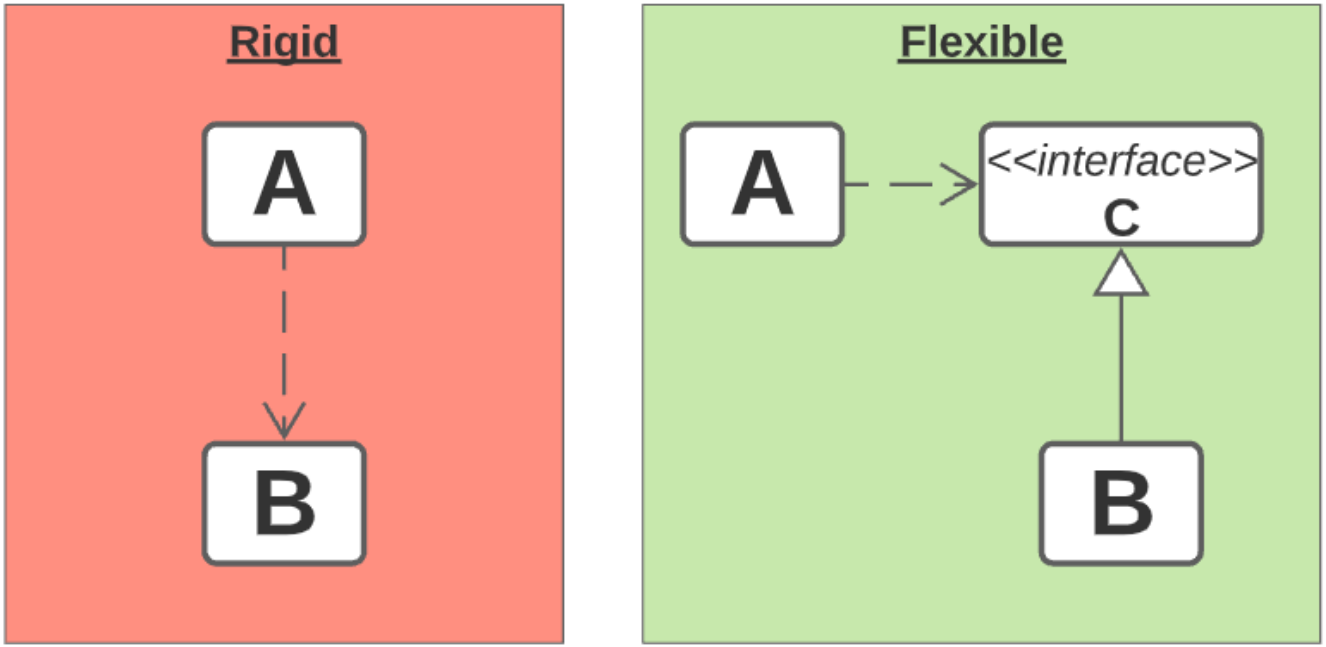
ISP: Simplifying Interfaces

- Break large interfaces into smaller, more specific ones.
- Example: Separate Printable and Scannable functionalities for devices.



DIP: Flexible Dependencies

- Depend on abstractions rather than concretions.
- Example: PaymentService relies on a PaymentProcessor interface.



What are Anti-Patterns?

- Recurring solutions that often cause more problems than they solve.
- Example: God Object, Spaghetti Code.

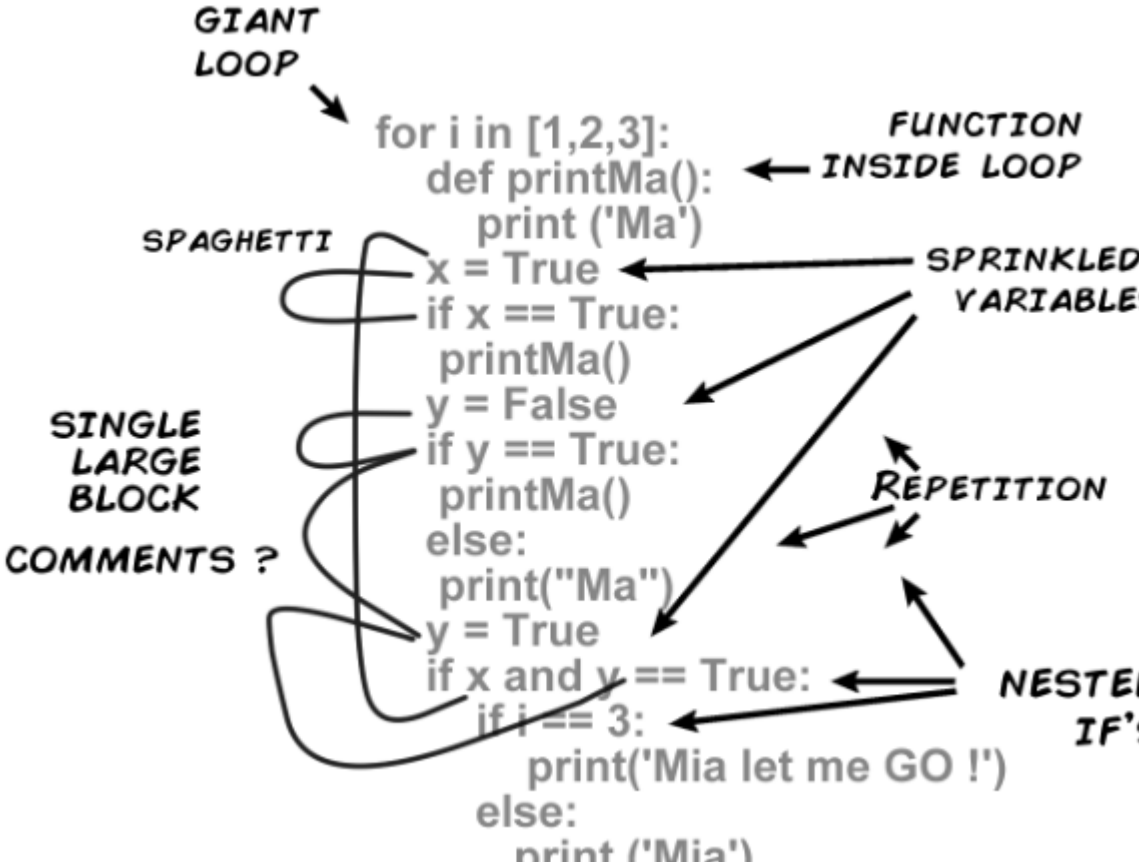
  https://bariscimen.medium.com/10-most-common-anti-patterns-every-software-engineer-must-avoid-182091438c2b

The God Object Problem

- A class that knows too much or does too much.
- Solution: Refactor into smaller, focused classes.

Spaghetti Code

- Unstructured and difficult-to-read code.
- Solution: Modularize and use clear control flows.

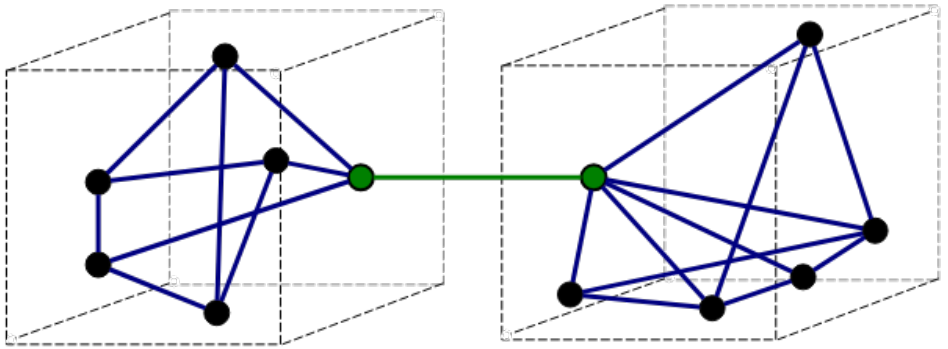


Benefits of Modularization

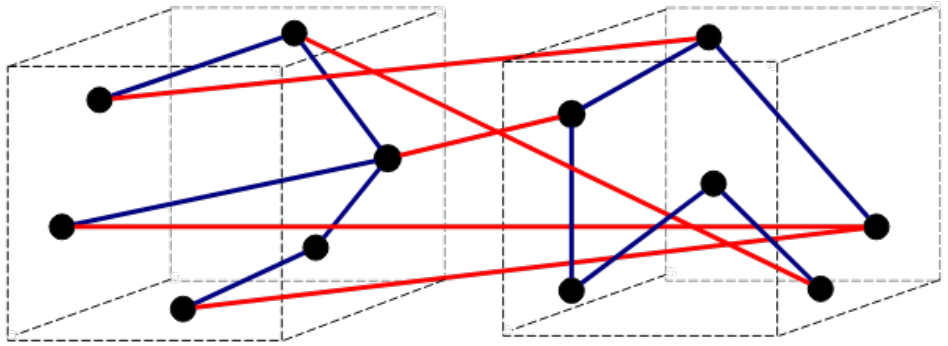
- Improves readability, testing, and maintainability.
- Example: Splitting billing, inventory, and user management into modules.

Cohesion in Design

- Cohesion measures how related the functionalities of a module are.
- High cohesion ensures modules do one thing well.



a) Good (loose coupling, high cohesion)



b) Bad (high coupling, low cohesion)

What is Refactoring?

- Improving the structure of code without changing its behavior.
- Stat: 70% of developers refactor to reduce bugs (Fowler, 2018).

Examples of Refactoring

- Rename variables for clarity.
- Extract methods to remove duplication.

- Example, we have one calculate area function. We define a class for each type of shape (rectangle, circle) and embed logic therein.
- Easier to maintain, don’t have to remember logic of ‘how to for…’

Reducing Redundancy in Code

- Use functions and templates for repetitive tasks.
- Example: Parameterize queries in database code.

Reducing Redundancy in Code

```
user_id = 123

query = f"SELECT * FROM users WHERE id = {user_id};"
```

- Not great. Let’s improve.

```
username = "johndoe"

email = "johndoe@example.com"

query = "SELECT * FROM users WHERE username = ? AND
email = ?;"
```

DRY Across Teams

- Standardize tools and practices to avoid redundancy.
- Example: Shared libraries for common utilities.