

SOLID Principles (C++)

Object-Oriented Design Principles

- S – Single Responsibility Principle
- O – Open/Closed Principle
- L – Liskov Substitution Principle
- I – Interface Segregation Principle
- D – Dependency Inversion Principle

Single Responsibility Principle (SRP)

- A class should have only one responsibility.

Example:

- User class handling data + email
- Separate User and EmailService

Code Example

```
class User {  
public:  
    void saveUser();  
};
```

```
class EmailService {  
public:  
    void sendEmail();  
};
```

Open/Closed Principle (OCP)

- Open for extension, closed for modification.
- Software entities should be open for extension, closed for modification.

Example:

- if/else for shapes
- Add new Shape class without changing existing code

Code Example

```
class Shape {  
public:  
    virtual void draw() = 0;  
};  
  
class Circle : public Shape {  
public:  
    void draw() override;  
};
```

Liskov Substitution Principle (LSP)

- Derived classes must be substitutable for base classes.
- Subclasses should replace base classes without breaking behavior.

Example:

- Square inheriting Rectangle
- Separate implementations

Code Example:

```
class Bird {  
public:  
    virtual void fly();  
};
```

```
class Sparrow : public Bird {  
public:  
    void fly() override;  
};
```

Interface Segregation Principle (ISP)

- Use small, specific interfaces.
- Clients should not depend on unused methods.

Code Example:

```
class Printer {  
public:  
    virtual void print() = 0;  
};
```

```
class Scanner {  
public:  
    virtual void scan() = 0;  
};
```

Dependency Inversion Principle (DIP)

- Depend on abstractions, not concrete classes.

Example:

- Direct DB dependency
- Use interface for DB

Code Example:

```
class Database {  
public:  
    virtual void connect() = 0;  
};
```

```
class App {  
    Database* db;  
public:  
    App(Database* d) : db(d) {}  
};
```

Summary

- Industry best practices
- Maintainable code
- Scalable design
- Easy testing
- Reduced coupling