

Design Patterns - Hallmarks of Good Architecture

SOLID Principles of Object-Oriented Programming

Practice Session

Single Responsibility Principle (SRP)

Design Patterns - Hallmarks of Good Architecture

```
class User {  
    String name;  
    String email;  
  
    User(this.name, this.email);  
  
    void saveUserToDatabase() {  
        // save user to the database  
    }  
  
    void showWelcomeMessage() {  
        print('Welcome, $name!');  
    }  
}
```

Single Responsibility Principle (SRP)

Hints:

1. Identify different responsibilities in the given code.
2. Create separate classes for each responsibility.
3. Ensure that each class has only one reason to change.

Design Patterns - Hallmarks of Good Architecture

```
class User {
    String name;
    String email;

    User(this.name, this.email);
}

class UserRepository {
    void saveUserToDatabase(User user) {
        // save user to the database
    }
}

class UserView {
    void showWelcomeMessage(User user) {
        print('Welcome, ${user.name}!');
    }
}
```

Single Responsibility Principle (SRP)

1. The original code violates the Single Responsibility Principle because the `User` class had more than one responsibility: holding user data, saving user data to the database, and displaying a welcome message to the user. This makes the class harder to maintain and modify in the future.

Design Patterns - Hallmarks of Good Architecture

```
class User {
    String name;
    String email;

    User(this.name, this.email);
}

class UserRepository {
    void saveUserToDatabase(User user) {
        // save user to the database
    }
}

class UserView {
    void showWelcomeMessage(User user) {
        print('Welcome, ${user.name}!');
    }
}
```

Single Responsibility Principle (SRP)

2. In the refactored solution, we have separated the responsibilities into three different classes: User, UserRepository, and UserView.
3. User class is responsible for holding user data, UserRepository class is responsible for saving user data to the database, and UserView class is responsible for displaying a welcome message to the user.