



## Object Oriented Design - Note

Made With Love by AH

last edited July 22, 2023

# Object Oriented Design - Note

Made With Love by AH

last edited July 22, 2023

## Contents

<b>1</b>	<b>Maintainable Code With SOLID Concept</b>	<b>2</b>
1.1	Single Responsibilities Principle . . . . .	2
1.2	Open Close Principle . . . . .	2
1.3	Liskov Substitution Principle . . . . .	2
1.4	The Interface Segregation Principle . . . . .	2
1.5	Dependencies Inversion Principle . . . . .	2

# 1 Maintainable Code With SOLID Concept

## 1.1 Single Responsibilities Principle

pecah fungsi ke beberapa method, jangan buat seluruh fungsi dalam 1 method

## 1.2 Open Close Principle

gunakan interface untuk membuat object yang memiliki method yang sama.

## 1.3 Liskov Subtitution Principle

menurut barbara liskov pada presentasinya di konferensi *Data Abstraction and Hierarchy* setiap object dari subtype harus *suitable* dan dapat bekerja dengan superclassnya hal ini disebutnya sebagai *Liskov Subtitution Principle*. salah satu cara untuk menghindari *Liskov Subtitution Principle* atau LSP adalah dengan menggunakan prinsip *Interface* dibandingkan menggunakan *Inheritance*

## 1.4 The Interface Segregation Principle

dalam ISP atau interface segregation Principle, jangan gunakan method atau variable yang tidak diperlukan dalam class, dalam C# dapat digunakan *Inheritance of Interface*.

```
public interface ITwoDeeShape {
    public double Width{get; set;}
    public double Height{get; set;}
}

public class SquareISP : ITwoDeeShape{
    // implement all interface
}

public interface IThreeDeeShape : ITwoDeeShape{
    // add new properties to our new interface
    public double Depth{get; set;}
}
```

## 1.5 Dependencies Inversion Principle

- High-level classes or modules should not depend on low-level classes or modules and both should depend on abstractions.
- Abstractions should not depend on implementation details but instead details should depend upon abstractions.

## **2   Getting Creative With Creational Pattern**