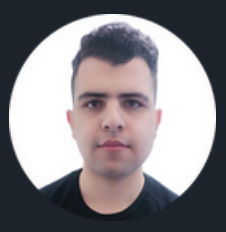
Single Responsibility Principle

A class/method should only have one reason to change.

```
public class Logger
{
    public void Log(string message)
    {
        // Logging logic
    }
}

public class Authenticator
{
    public bool Authenticate(string username, string password)
    {
        // Authentication logic
        return true;
    }
}
```



Elliot One

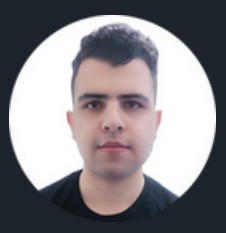
Open/Closed Principle

Software entities (classes, modules, methods) should be open for extension but closed for modification.

```
public abstract class Shape
{
    public abstract double CalculateArea();
}

public class Rectangle : Shape
{
    public double Width { get; set; }
    public double Height { get; set; }

    public override double CalculateArea()
    {
        return Width * Height;
    }
}
```



Elliot One

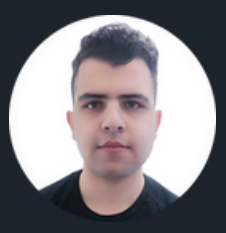
Liskov Substitution Principle

Subtypes must be substitutable for their base types without altering the correctness of the program.

```
public interface IFlyable
{
    void Fly();
}

public class Bird : IFlyable
{
    public void Fly()
    {
        // Flying logic
    }
}

public class Penguin : IFlyable
{
    public void Fly()
    {
        // Do nothing, as penguins can't fly
    }
}
```



Elliot One

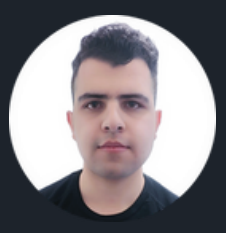
Interface Segregation Principle

A class should not be forced to implement interfaces it does not use.

```
public interface IWorker
{
    void Work();
}

public interface IEater
{
    void Eat();
}

public class Robot : IWorker
{
    public void Work()
    {
        // Work logic
    }
}
```



Elliot One

Dependency Inversion Principle

High-level modules should not depend on low-level modules. Both should depend on abstractions.

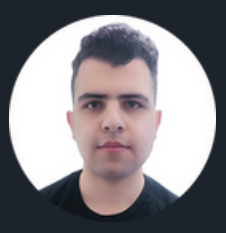
```
public interface ISwitchable
{
    void TurnOn();
}

public class LightBulb : ISwitchable
{
    public void TurnOn()
    {
        // Turn on logic
    }
}

public class Switch
{
    private readonly ISwitchable device;

    public Switch(ISwitchable device)
    {
        this.device = device;
    }

    public void Press()
    {
        device.TurnOn();
    }
}
```



Elliot One

Elliot One

Enjoyed Reading This?

Reshare and Spread Knowledge.