**5 Design Principles SOLID**

## S is Single Responsibility Principle- Class should have only one reason to change, Like Member function offers class and Booking class

## ****O is for Open/Closed Principle** : like different kind of payment class should be created like for Wallet , Cr card if any ne wpayment method should come so don't need to modify the class just another class**

## 

public abstract class Shape

{

public abstract double Area();

}

Inheriting from Shape the Rectangle and Circle classes now looks like this:

public class Rectangle : Shape

{

public double Width { get; set; }

public double Height { get; set; }

public override double Area()

{

return Width\*Height;

}

}

public class Circle : Shape

{

public double Radius { get; set; }

public override double Area()

{

return Radius\*Radius\*Math.PI;

}

}

## ****L is for Liskov Substitution Principle -** Derived class are completely substitutable for their base classes**

## 

## ****I is for Interface Segregation Principle** example like payment method with cash Payment and card payment or differtn interfaces for differtn work**

**D is for Dependency Inversion-** The principle states:

1. High-level modules should not depend on low-level modules. Both should depend on abstractions.
2. **High-level modules should not depend on low-level modules. Both should depend on abstractions.**
3. Abstractions should not depend on details. Details should depend on abstractions.
4. http://www.oodesign.com/dependency-inversion-principle.html