SOLID PRINCIPLES

S – single responsibility principle

A class/method should do one thing(job) and do it well. Its based more on the context of it eg. You could have a userManagement controller that would have all the methods in that class (Add, edit, delete) and you could also have a controller for each user function(AddUserController etc.) they say methods should have 2-3 lines but that’s not really true.. it more depends on the context of the code.

O – open closed principle

Open for extension and closed for modification.

Simply means a class should be easily extendable without modifying the class itself.

L – liskov substitution principle

This allows you to replace objects of a parent class with objects of a sub class without breaking the app. This requires the subclass to behave the same way as the parent class.

Eg of this was the duck example.. the parent class is a duck and the method is quack.

When a diff type of duck inherits from the duck class it still quacks with a quaaack.

But if a rubber duck inherits from a duck .. it cannot quack so it returns a not implemented exception which then breaks the app.

I – interface segregation

This basically states that no class should be forced to depend on methods that it does not use.

You should instead create a new interface and let you class implement multiple interfaces.

D – dependency inversion

This is a way to decouple modules. We use dependency injection to use other classes methods in our app.

Dependency injection is used by injecting dependencies of a class through the class constructor as an input parameter