C# Design Patterns

The following notes are my own phrasing of various C# design patterns based on the understanding I have gained through independent study. They are intended for the purpose of revision and proof of my understanding.

SOLID Design Principles

**Single Responsibility Principle** – Any class should only be responsible for a single part of the program’s functionality. New functionality should be added in new classes. This aids readability and in the event of a problem, the source of the issue will be easy to locate based on the nature of the problem (e.g. if text is not being saved to a .txt file correctly, you know that the issue will be in the class that handles persistent data).

**Open-Closed Principle**

**Liskov Substitution Principle**

**Interface Segregation Principle**

**Dependency Inversion Principle**