SOLID principles

Single-responsibility principle- a class/method should do just one thing

Open-closed principle- entities should be open for extension, closed for modification

The ability to add new usability but not change existing functionality

Liskov substitution principle- objects should be replaceable by instances of their subtypes without altering the correctness of the code

Interface segregation principle- heavy interfaces should be split up into lighter interfaces

Dependency inversion principle- instead of high level code depending on low level code, or vice versa, both should depend on interfaces

GAC- global assembly cashe

Strongname- filenames include version numbers

General Principles

Separation of concerns- minimize “tight coupling”. Want to decrease the dependency on other classes

Comment code and for public code, use XML comments

DRY- don’t repeat yourself

KISS- keep it simple, stupid