Glossary

General

Interface: abstract type (i.e. no data/logic) that defines behaviour through method signatures
Spaghetti Code - messy, hard to read and difficult to maintain code
Extendable Code - adding new features easily
Strategy pattern: Select algorithm at runtime
Unit testing - Tests individual methods/sources of logic (i.e. only testing one piece of logic, not a journey)
Behavior-driven development - tests written in plain language- commonly using Gherkin syntax - Given/When/Then
Legacy Codebase - outdated or code that is no longer updated - usually has old tech stack
Mocking - simulating the behaviour of objects

Code Craftsmanship

YAGNI - You aren't gonna need it
KISS principle - Keep it simple, stupid
DRY - Don't repeat yourself
Boy Scout - Leave things in a better state than you found it

SOLID

Single-responsibility principle: Every class should have only **one** responsibility
Open–closed: software entities (classes, modules, functions, etc.) should be open for extension, but closed for modification
Liskov substitution: Functions that use pointers or references to base classes must be able to use objects of derived classes without knowing it.
Interface segregation: Many client-specific interfaces are better than one general-purpose interface.
Dependency inversion: Depend upon abstractions (interfaces), not concrete classes.

Resources

Margaret Hamilton interview - interesting interview with Margaret Hamilton regarding her work on the Apollo code and being the first female software engineer and managing an all-male team. She also talks about managing being a mother and her career.

Clean Code Summary - Really good read/tips on clean code

KISS Code Example from slides - A detailed example of the KISS principle in C#

SOLID Resources

Overview - SOLID: The First 5 Principles of Object Oriented Design
Strategy Pattern - Strategy Pattern C# Example

SOLID examples (both violations & fixes) in Python - SOLID Python Examples
SOLID examples (both violations & fixes) in JavaScript - SOLID Javascript Examples

Single-Responsibility Principle C# Example - Single Responsibility Examples
Open-Closed Principle C# Example - The Open/Closed Principle Examples
Liskov Substitution Principle C# Example - Liskov Substitution Principle Examples
Interface Segregation Principle C# Example - Interface Segregation Examples
Dependency Inversion Principle C# Example - Dependency Inversion Examples

TDD and Code Craftsmanship Resources

TDD & Code Craftsmanship - Clean Code & TDD
The Benefits - The Benefits of TDD, Unit Testing, and Code Quality

Different testing layers - Difference between Component and Unit Testing
Different layers of testing - TestPyramid

Mocking - What is Mocking?
Mocking C# - Mocking in C#
Mocking Python - Mocking in Python
Mocking JavaScript - Mocking using Jest in JavaScript

BDD

SpecFlow - C# BDD testing framework - SpecFlow Example Projects
BDD language syntax - Gherkin Syntax