**Object Oriented Programming with C#**

# **Basic environment setup**

[VSCode](https://docs.google.com/document/d/1CBRX_ppdLyCrggPYaI0jw-qGfJQRi1xx5Ju00AJdKYs/edit?usp=sharing)

* + [🐈Git](https://docs.google.com/document/d/13fDvzLMC4Yw8vmM9C-oLWxUFm5lOfFmuk_4_SokYTiw/edit?usp=sharing)

# **Formulatrix Development Standards**

[Dev Standard](https://docs.google.com/document/d/1wZn3nZ37hFKE7dzf6yPGXm49gYSMjno-GoEu50GWSUw/edit?usp=sharing)

* + [Coding Standards](https://docs.google.com/document/d/15SJlqUUjzL6Va9akhahni58OLuRDjOMgwrFFewy3SOU/edit?usp=sharing)

# **Reference for Learning C#**

* + C# 10 in a Nutshell
  + [C# Corner](https://www.c-sharpcorner.com/)
  + [Dot Net Tutorials By Real-Time Experts | .NET Framework](https://dotnettutorials.net/)

# **Essential Chapters Introduction**

* C# project structure
  + Solution file
  + Project files
  + References
    - Built in library
    - External library
    - NuGet
  + Project output: Assembly (dll, exe)

**2 Days**

## **Introducing C# and the .NET Framework**

### [🏛️Object Oriented Programming](https://docs.google.com/document/d/1fz-_jMjH3FpdFoxrzqBa0Ehfulj2pwt38QNh18byMBI/edit?usp=sharing)

### [🆎Typesafety in C#](https://docs.google.com/document/d/1emmUsmBK0fwRamdFmd05Z8AX-_TQ4R_E2w_1CtxLxRo/edit?usp=sharing)

## **C# Language Basics**

### Syntax

### Type Basics

### Numeric Types

### Boolean Type and Operators

### Strings and Characters

### Arrays

### Variables and Parameters

### Expressions and Operators

### Null Operators

### Statements

### Namespaces

* + - **C# project structure**

**2 Days**

## **Creating Types in C#**

### Classes

### Inheritance

### The object Type

### Structs

### Access Modifiers

### Interfaces

### Enums

### Nested Types

### [⚙ Generics](https://docs.google.com/document/d/1purl6pqV70rcvQmtlsM1HUxj2CucqUqC8MXj9sJoLBs/edit?usp=sharing)

**3 Days**

## **Advanced C#**

### [✉️ Delegates](https://docs.google.com/document/d/1yIj-yzKNfBd3EJghscvfWEpeHqX7XfFx291MIqgnLoY/edit?usp=sharing)

### [🔔 Event Handler](https://docs.google.com/document/d/1AuWLxOfwsZHsl-Z0A5Q-yLh4ZBFzflsMD7dmvGwLErU/edit?usp=sharing)

### [⚠️try Statements and Exceptions](https://docs.google.com/document/d/1nOQwVQjrxhWhjBdALFUPMQU6-b2vc_o2LkwHSevtPoI/edit?usp=sharing)

### [🔁 Enumerator and Iterators](https://docs.google.com/document/d/191xLBlOJDim7ITGvhZkf0ieHaWucHnTXcQdXB3auiOE/edit?usp=sharing)

### [🈳 Nullable Value Types](https://docs.google.com/document/d/1MsOjgrWK2btftphOyZfkONqCBdM5Jq--PTvLLJbVwx0/edit?usp=sharing)

### [🔂Operator Overloading](https://docs.google.com/document/d/1xI8omYprWsuZTXS8khsC5ByP0cV73n0UksMMjQARZng/edit?usp=sharing)

## **Framework Overview**

### **The CLR and Core Framework** [.NET Core and .NET Standard: What Is the Difference?](https://www.infoq.com/news/2017/10/dotnet-core-standard-difference/) [.NET Framework, .NET Core and .NET Standard compatibility matrix](https://docs.microsoft.com/en-us/dotnet/standard/net-standard) [~~What is MONO?~~](https://www.mono-project.com/) [🌐 .NET Overview](https://docs.google.com/document/d/1_11D8_Nb4dRDjotUyMxlAjwjvASpB0N-bCO1HVGtQLY/edit?usp=sharing)

**1 Day**

## **Framework Fundamentals**

### [⌨️ String and Text Handling](https://docs.google.com/document/d/1BgscaqPpEeSZr5zxndSrBOWX4DWDbzywcGOENLquS0s/edit?usp=sharing)

### [🗓️ Date and Times](https://docs.google.com/document/d/1jbHlHiOe8JxvT5_hcn-l5TXWwfzSl89YUZ5Zawu9Dgk/edit?usp=sharing)

### [ℹ️ Formatting and Parsing](https://docs.google.com/document/d/1vAc8e-VV_3gIDkMgkuyydEeCvUZoLKkpH98Uv300uQ8/edit?usp=sharing)

### [📔 Other Conversion Mechanism](https://docs.google.com/document/d/13IGoLsHQx6XgL3mODdfbGzRPtEWtnnaxAZkBeR0A37o/edit?usp=sharing)

### [🔢 Working with Numbers](https://docs.google.com/document/d/1Tpf1AokBejTcFmfP8Kgr7uyVPf2Kk98OA5HaMUghVmk/edit?usp=sharing)

### [📑Enums](https://docs.google.com/document/d/1p0jWOmC9aTqfNi1Io-smaQb8gzOMn-INaEkaaSuFXEs/edit?usp=sharing)

### [🔷 Equality Comparison](https://docs.google.com/document/d/1gNYczKT-xZIVKSzSh4iZlsYWn7tJSlGPpw9QFUv0Zgc/edit?usp=sharing)

### [🔍 Utility Classes](https://docs.google.com/document/d/1S7eKdqMiCSKIWycL2LsQUtZHXvqLbO7uUuQ8yk1qR64/edit?usp=sharing)

**2 Days**

## **Collections**

### [📼 Collection](https://docs.google.com/document/d/1mnSKctPceEi6RQuJHl4yNUW7KtMoCaQXRbquAZoP30w/edit?usp=sharing)

#### [🤟Enumeration](https://docs.google.com/document/d/1MmcmlpPICvW7r0pqFJvKIgxskc1n_pRmg3TckYillVU/edit?usp=sharing)

#### [🧮 Array Class](https://docs.google.com/document/d/1s0OVuxRrPr6Tgcrdoe48WJ3PaOH0cx4nkxRc94G3GIc/edit?usp=sharing)

#### [🀄Lists, Queues, Stacks, and Sets](https://docs.google.com/document/d/1niUE4uAUWk32Mtzycz90tCozUYR9_OEClLZxHZfJDpM/edit?usp=sharing)

#### [🦺 Customizable Collections and Proxies](https://docs.google.com/document/d/1qLsZY9PUZJPywmmrAXrNit8fvEAB_2JxOEYJr7qxecM/edit?usp=sharing)

#### [🧑🏻‍ Immutable Collections](https://docs.google.com/document/d/1j9FR2BVF0mhHFoLbaiCWMEAFgKdQiIcc1b2kfS6A7NM/edit?usp=sharing)

#### [📑Plugging in Equality and Order](https://docs.google.com/document/d/1TnNPJRwxaXLO18_tMXTxEcEFGdACoUsTsfOEBHM4uF0/edit?usp=sharing)

#### [📗ArrayList](https://docs.google.com/document/d/1lrkwkXmsV9FzaG1h9S5xjXLCwTw50g4hFIVurXhs6kA/edit?usp=sharing)

#### [#️⃣HashTable](https://docs.google.com/document/d/1Z5A0BUFwUQMS_co7TlHrjSKVj46J53v8b9utjy6gRZc/edit?usp=sharing)

* + - Supplemental Readings for Choosing the Right Collection
      * [Choosing the Right Collection](https://www.codeproject.com/Articles/1095822/Choosing-the-Right-Collection)

## **2 Days**

## **Disposal and Garbage Collection**

### [🗑️ Garbage Collector](https://docs.google.com/document/d/1HjtCFSLqeKNoQHQwmMtrzbGJR_5HzYQLghtgO3JC66E/edit?usp=sharing)

#### Highlights

* + - * Managed vs unmanaged
        + Struct is not handled by GC!
      * Disposable pattern
      * Don’t mess GC.Collect()
      * GC freeze the world! – performance
    - Supplemental Readings
      * [Garbage Collector Fundamentals](https://docs.microsoft.com/en-us/dotnet/standard/garbage-collection/fundamentals)
      * [Benchmarking Demo](http://www.nimaara.com/2016/01/01/high-performance-logging-log4net/)

**1 Days**

## **Diagnostics and Code Contracts**

### [#️⃣Conditional Compilation](https://docs.google.com/document/d/1X7pPpWKapkm33sm1N7plOyh0At_N5zXn0VX43_o00ZE/edit?usp=sharing)

### [🐛Debug and Trace Classes](https://docs.google.com/document/d/1xoKNPPxN1GdbTP7FoCnVSIb52j2c5-kZTVR0H-_0vns/edit?usp=sharing)

### [🌪️ Assertions](https://docs.google.com/document/d/1VZcGgGd7UvpIzsbEfMmJy0h4JbAN6syU0iVjPOs5kUk/edit?usp=sharing)

## **2 Days**

## **Concurrency and Asynchrony**

### [🧠Concurrency and Asynchrony](https://docs.google.com/document/d/10dgKpdx0n32neDKknb5CZT7wYwUYraJwmQKElRyC01w/edit?usp=sharing)

#### [🧵Threading](https://docs.google.com/document/d/1azJMWY7goOeZ1GSuWv0g2V63wCP8dwnFLz1VWm2WVyc/edit?usp=sharing)

#### [🧵 Multithreading](https://docs.google.com/document/d/1YEXCbnFvnxyQgM9iOtFrLTA2D5NbbCy-8mHhp1A42cY/edit?usp=sharing)

#### [💼Tasks](https://docs.google.com/document/d/1iF1OczL6hN9OFRLYByQ6mqHdj9ro6rXT8FbYhO1fS44/edit?usp=sharing)

#### [🔄Asynchronous Programming](https://docs.google.com/document/d/1oqsg_cjZmNvD2N9pbYZO_1ce0huyE_wibWeKeRYEojg/edit?usp=sharing)

* + - Supplemental Readings
      * [Monitor.TryEnter, lock, using, and Deadlock](http://www.interact-sw.co.uk/iangblog/2004/03/23/locking)
      * [Safe Thread Synchronization](https://learn.microsoft.com/en-us/archive/msdn-magazine/2003/january/net-column-safe-thread-synchronization)

## **1 Day**

## **Streams and I/O**

* + - [🎏 Streams and I/O](https://docs.google.com/document/d/14prlSoD5Z9MlIcFn9ocGBozieg6eoIrzr1VWYwYdM5g/edit?usp=sharing)

### [📁File Handling](https://docs.google.com/document/d/1tqcmBL96xTSbtVPri2fKlPsFXEF_7NQzI0_Lu5O3R1I/edit?usp=sharing)

## **1 Day**

## **Serialization and Assemblies**

* + - [🛷Serialization](https://docs.google.com/document/d/1SsQ4CXod4wD-7NCnFPv5YjyoyjIRWW1oJpygE-ZjlMY/edit?usp=sharing)
    - [🥂Assemblies](https://docs.google.com/document/d/1aaDlhD-giOf53KFbkgDWbYu5q7XpYTZOTlCAWl85XDE/edit?usp=sharing)

**PROJECT** (individual)

General guideline:

* Time constraints: **3 days**

# **3 Days**

# **Advanced Reading**

* + - Understanding [SOLID](https://www.c-sharpcorner.com/UploadFile/damubetha/solid-principles-in-C-Sharp/) and [KISS](http://en.wikipedia.org/wiki/Kiss_principle) principles
    - [Tell don't ask](https://martinfowler.com/bliki/TellDontAsk.html)
    - [Unit testing](https://docs.microsoft.com/en-us/dotnet/core/testing/unit-testing-with-nunit)[and more…](https://www.codementor.io/copperstarconsulting/intro-to-unit-testing-c-code-with-nunit-and-moq-part-1-y2b9iv8iq)  
      **\*assignment**: lengkapi project sebelumnya dengan unit testing
  + Logging with [Log4Net](https://stackify.com/log4net-guide-dotnet-logging/)
    - [Easy Logger](https://github.com/NimaAra/Easy.Logger)
    - [log4net](https://docs.google.com/document/d/1Nm_8jurI6wl_zT3PDkzcEaXxsEFEUd1SV_YEMqpwqEA/edit?usp=sharing)
    - Contextual logging
      * [Context and correlation – structured logging concepts in .NET (5) (nblumhardt.com)](https://nblumhardt.com/2016/08/context-and-correlation-structured-logging-concepts-in-net-5/)
      * [Logging in C# .NET Modern-day Practices: The Complete Guide | Michael's Coding Spot (michaelscodingspot.com)](https://michaelscodingspot.com/logging-in-dotnet/)
      * [C# Logging Best Practices: Smarter errors and logs to fix apps faster (stackify.com)](https://stackify.com/csharp-logging-best-practices/)

**2 Days**

* + Entity Framework
* [🆔 Entity Framework](https://docs.google.com/document/d/1Oi_U5U_6b6Dxp-uszqh2pKLxVZ4zoihdMKeHoFgsDxs/edit?usp=sharing)//ongoing
* [Entity Framework Tutorial](https://www.entityframeworktutorial.net/)
* [Get started with Entity Framework 6 - EF6 | Microsoft Learn](https://learn.microsoft.com/en-us/ef/ef6/get-started?redirectedfrom=MSDN)
* [Entity Framework using C# (c-sharpcorner.com)](https://www.c-sharpcorner.com/article/entity-framework-introduction-using-c-sharp-part-one/)
* Database options: [SqlLite](https://sqlite.org/download.html) or [Postgre](https://www.postgresql.org/download/)

# **Reference for Design Pattern**

**3 Days**

* + Head First Design Patterns
  + [🎨Design Pattern](https://docs.google.com/document/d/1B79XMGp4a6nDDhRnx-fSKIkPrn11i7U_vumdkqNf0WU/edit?usp=sharing)
    - [🖌️Creational Design Pattern](https://docs.google.com/document/d/1LSTLK4IR8glcU5tEZiKxf4tmxRXHMp0TCE8Z9kF8bkg/edit?usp=sharing)
  + C#3.0 Design Patterns - Judith Bishop
  + RefactoringGuru or Deep Dive into Design Patterns

**Project**

**2 Days**

# **Debugging Techniques**

* + - Reference :
      * [Debugging Technique](https://docs.google.com/document/d/1x-5ysM_cFtWi0BDJJYyHq4CeZZ7-IaWR4jM4Vo-2Jfk/edit?usp=sharing) //ongoing
      * [9 rules for debugging anything (Book Review) | tygertec](https://www.tygertec.com/9-rules-debugging/)
      * Book : Debugging—The Nine Indispensable Rules for Finding Even the Most Elusive Software and Hardware Problems
* Understand the System
* Make it Fail
* Quit Thinking and Look
* Divide and Conquer
* Change One Thing at a Time
* Keep an Audit Trail
* Check the Plug
* Get a Fresh View
* If You Didn't Fix It, It Ain't Fixed

**Project**

## **How to Write a Clean Code**

[🧹Clean Code](https://docs.google.com/document/d/1_YHuHV-lPtn4oz9oP5gT7nv8eMghJpvzPn8Jaj1PUPI/edit?usp=sharing)

* **Project** ( team)