

1. Git	3
Introduction to Git	3
Learn Git Branching	3
Intro to Git Recap	3
Recommended apps to work easier with Git:	3
2. Programing Principles and Use Cases	3
UML UseCase Diagram	3
Association vs Aggregation vs Composition	3
Aggregation vs Composition	3
KISS - Keep It Simple, Stupid :)	3
YAGNI - You aren't gonna need it	3
DRY - Don't repeat yourself	3
3. Agile Principle	3
The 12 Agile Principles	3
Agile Fundamentals	3
Agile Course	3
4. Clean Code in .NET	3
Naming Convention	3
Learn the Art of Writing Clean Code	3
5. Object Oriented Programming (OOP)	4
OOP Basics	4
OOP Intermediate	4
Homework :)	4
6. C#	4
C# 10 Ultimate Guide - Beginner to Advanced	4
7. SQL	4
SQL Basics	4
What is Normalization in SQL ?	4
SQL Tutorial	4
SQL In C#	4
Homework:	4
8. Entity Framework (EF)	4
Introduction in EF	4
EF Tutorial	4
9. SOLID Principles	5

SOLID Principles Basics	5
SOLID Principles Basics	5
10. Design Patterns	5
Design Patterns in C# Tutorial	5
Design Patterns Docs	5
11. REST Architecture and WEB API	5
Introduction of REST & Web API	5
What are RESTful APIs	5
.NET Web API & Entity Framework Core	5
Common web application architectures	5
Onion Architecture in ASP.NET Core	5
12. Serialization	5
Introducing JSON	5
Serialization in C#	5
13. Parallel Programming	5
Task Class	5
Asynchronous programming with async and await	5
C# Async / Await	5
14. Javascript	6
JavaScript Basics for Beginners	6
15. HTML and CSS	6
HTML Basics	6
CSS Basics	6
16. Angular	6
Angular - The Complete Guide	6

1. Git

[Introduction to Git](#)

[Learn Git Branching](#)

[Intro to Git Recap](#)

Recommended apps to work easier with Git:

- <https://desktop.github.com/>
- <https://www.sourcetreeapp.com/>

2. Programing Principles and Use Cases

[UML UseCase Diagram](#)

[Association vs Aggregation vs Composition](#)

[Aggregation vs Composition](#)

[KISS - Keep It Simple, Stupid :\)](#)

[YAGNI - You aren't gonna need it](#)

[DRY - Don't repeat yourself](#)

3. Agile Principle

[The 12 Agile Principles](#)

[Agile Fundamentals](#)

[Agile Course](#)

4. Clean Code in .NET

[Naming Convention](#)

[Learn the Art of Writing Clean Code](#)

5. Object Oriented Programming (OOP)

[OOP Basics](#)

[OOP Intermediate](#)

[Homework :\)](#)

6. C#

[C# Masterclass](#) - Without WPF and Unity

7. SQL

[SQL Basics](#)

[What is Normalization in SQL ?](#)

[SQL Tutorial](#)

[SQL In C#](#)

Homework:

- Create a database diagram using <https://app.diagrams.net/>
- Create the diagrams for NF1, NF2 and NF3
- You must have at least 4 tables

Notes: You must choose a theme for example Education. As tables I choose to have Courses, Professors and Students. You must create a database for NF1, then from NF1 change it to NF2 and from NF2 to NF3.

8. Entity Framework (EF)

[Introduction in EF](#)

[EF Tutorial](#)

9. SOLID Principles

[*SOLID Principles Basics*](#)

[*SOLID Principles Basics*](#)

10. Design Patterns

[*Design Patterns in C# Tutorial*](#)

[*Design Patterns Docs*](#)

11. REST Architecture and WEB API

[*Introduction of REST & Web API*](#)

[*What are RESTful APIs*](#)

[*.NET Web API & Entity Framework Core*](#)

[*Common web application architectures*](#)

[*Onion Architecture in ASP.NET Core*](#)

12. Serialization

[*Introducing JSON*](#)

[*Serialization in C#*](#)

13. Parallel Programming

[*Task Class*](#)

[*Asynchronous programming with async and await*](#)

[*C# Async / Await*](#)

14. Javascript

[*JavaScript Basics for Beginners*](#)

15. HTML and CSS

[*HTML Basics*](#)

[*CSS Basics*](#)

16. Angular

[*Angular - The Complete Guide*](#)