[Engineering Management]

```
Estimation

        .Scope Concept
        .Estimates, Targets, and Commitments
        .Overestimate vs Underestimate
        .Decomposition and Recomposition
        .Analogy-based estimations
        .Story based estimations

Process Planning

        .Cone of Uncertainty
        .Source of Estimation Errors
        .Dis economies of Scale
        .Count, Compute, Judge techniques
        .Delphi method
        .Challenges with Estimating Size
        .Challenges with Estimating Effort
        .Challenges with Estimating Schedule
        .Story based scope definition: scoping project, release planning
        .Documenting and presenting estimation results
        .PERT analysis
```

[Software Requirements Engineering]

```
        .Requirement definition
        .Levels of Requirements: Business, User, and Functional requirements
        .Most common requirements risks
        .Characteristics of Excellent Requirements
        .Benefits from a High-Quality Requirements Process
        .System requirements
        .Non-functional requirements
        .Product champion
        .User classes and their characteristics
        .Software Quality Attributes
        .Assumptions, constraints and their roles
        .Requirements artifacts:Product Vision and Scope document
```

[Design]

```
    Algorithms

            .Algorithms complexity
            .Array sorting methods
            .Tree structure
            .Binary search
            .Hash table
            .Stack
            .Queue
            .Linked List
            .Graphs basic
            .Mutithreading algorithms
            .Regular Expressions implementation using Nondeterministic finite-state
automata
            .String searching tries
            .Searching strings algorithms
            .Divide and conquer method
            .Balanced trees

    DB Design

            .Relational terminology: Entities , Attributes , Records
            .Relationships
            .Understanding normalization concept
            .Data Integrity
            .Implementing Data Integrity
            .Competent
            .Understanding RDBMS architecture
            .Transactions: ACID
            .Transactions: Recovery
            .Transactions: Locking
            .Transactions: Isolation
            .Transactions: Concurrency
            .Optimization database (performance, volume)
            .Optimizing the Database Design by Denormalizing

    OOD

            Abstraction
            Encapsulation
            Inheritance vs. Aggregation
            Modularity
            Polymorphism
            Types vs. Classes
            Abstraction Qualities
            Separation of concerns principle
            Single responsibility principle
            Competent
            GoF Design Patterns
            Architectural Patterns: Layered Architecture
            Architectural Patterns: MVC
            Architectural Patterns: IoC
            SOLID principles
            Anti-patterns

    Security

            Information security
            Access Control Lists
            .NET Framework Cryptography Model
            DPAPI
            Authentication types
            Competent
            Public-key Infrastructure
            SSL and TLS
            Strong-Named Assemblies
```

[Construction - Core]

```
    Concurrency

            Understand threading
            Using Tasks, async/await
            Parallel class
            Volatile class
            Interlocked class
            Parallel Language Integrated Query (PLINQ)
            Synchronizing resources
            Delegates
            Events
            Lambda expressions
            Deadlock problem
            Race Condition
            Thread class basics
            Thread pooling
            Background Worker
            Threading in CLR
            Continuation with Tasks
            Synchronization Context
            Using Multiple Asynchronous Methods
            Exceptions with Asynchronous Methods
            Cancellation
            Collections (advanced async)

    Internationalization

            Localization
            Global Resource
            Local Resource Expression (Explicit, Implicit)
            Internationalization
            Globalization
            User interface issues
            Natural language issues
            Data formatting issues
            String-related issues
            System. Globalization Namespace
            Creating a satellite assembly
            Creating a file based resource manager
            Using Calendars for Specific Cultures
            Formatting Numeric Data for a Specific Culture
            Comparing and Sorting Data for a Specific Culture
            System.Text Namespace

    NET C#

            Declare namespaces, classes, interfaces, static and instance class
members
            Types casting
            Value and reference types. Class vs Struct usage
            Properties and automatic properties
            Structured Exception Handling
```

```
        Collections and Generics
        Dictionaries. Comparison of Dictionaries
        Building enumerable types
        Building cloneable objects
        Building comparable types
        Nullable types
        Delegates, events and lambdas
        Indexers and operator overloading
        Anonymous types
        Extension methods. Practices.
        Custom Type Conversions (implicit/explicit keywords)
        Strings and StringBuilder. String concatenation practices.
        Serialization
        System.IO namespace
        LINQ to Objects
        General Coding conventions for C#
        C# 6.0 new features
        Building and Configuring Class Libraries
        Type Reflection
        Late Binding
        Custom attributes
        Dispose and Finalizable patterns
        Garbage collection
        .Net Diagnostics
        Implementing logging
        Exception handling guidelines
        Regular Expressions
        LINQ to Xml

Networking

        Understanding networks: layers and protocols
        Basic understanding of TCP/IP model and protocols
        Basic knowledge of physical layer protocols and media
        Application layer protocols basics (HTTP, FTP, Telnet)
        Understanding HTTP and WWW
        Basic troubleshooting tools (ICMP, ping, traceroute)
        Client/Server model
        Sockets, IP and port addressing
        Using proxy server
        File transfer services: FTP, TFTP
        Name resolution services: DNS, whois
        Remote access services: Telnet, SSH, rdesktop, VNC

Product Depoloying

        Create, configure, and publish a web package
        Creating Web Setup Project
        Publishing Web Services
        Manage packages by using NuGet
        Configure a web application for deployment
        Choose a deployment strategy for a Windows Azure web application
        Share assemblies between multiple applications and servers

Refactoring

        Refactoring Concept (what/when/why)
        Smells Catalog and possible re-factorings
        Move Method
        Move Field
        Encapsulate Field
        Encapsulate Collection
        Extract Method
        Inline Method
        Inline Temp
        Replace Temp with Query
        Split Temporary Variable
        Decompose Conditional Expression
        Consolidate Conditional Expression
        Consolidate Duplicate Conditional Fragments
        Remove Control Flag
        Replace Conditional with Polymorphism
        Making Method Calls Simpler
        Dealing with Generalization
        Replace Constructors with Creation Methods
        Move Creation Knowledge to Factory
        Encapsulate Classes with Factory
        Encapsulate Composite with Builder
        Replace Conditional Logic with Strategy
        Replace State-Altering Conditionals with State
        Replace Implicit Tree with Composite
        Replace Conditional Dispatcher with Command
        Form Template Method
        Extract Composite
        Replace Hard-Coded Notifications with Observer
        Red green refactoring
        Managing Technical Debt
```

[Construction - Web]

```
    CSS
        Simple Style rules.
        Selectors cascading and inheritance
        Elements positioning, floating and layering
        Tables properties
        Flexible Box Layout
        Color (rgba, hsl, hsla, transparent, currentColor)
        Fonts
        Text
        Backgrounds and Borders
        Selectors
        Complex rules and defining styles structure
        Box models
        Typography & Fonts
        Basic User Interface
        Transitions
        Animations
        Transforms
        mixins
        SCSS or less
        Bootstrap: main features and concepts
        Bootsrap: using of grid
        Bootstrap: customization

HTML

        Basic elements
        Design Patterns
        Page Layouts with tables
        Page Layouts with divs
        Frames
        Embedded multimedia
        forms & form elements
        Section elements
        Grouping content elements
        Text-level semantic element
        Interactive elements
        Video, Audio
        Standards and Browser compatibility
        HTML5: Forms
        Web-components
        templates
        custom elemets
        shadow DOM

JavaScript

        Identifiers and Reserved Words
        Javascript types
        Reference vs value
        Property Attributes of an object
        Type Conversion
        Increment and Decrement Operators
        Variable Scoping
        Context of calling (*.call(), *.apply(), *.bind())
        Event handling (bubbling and capturing)
        Exception Handling
        Loops
        Object - Date
        Object - Array
        Closures
        ES6: let and scope
        ES6: Arrow functions
        Function hoisting
        Strict mode of javascript
        URI Handling Function Properties
        Dependancy injection (common, AMD, namespase)
        WebWorkers
        Inheritance
        Currying and chaining
        Regular Expressions
        template strings
        modules

Web Services

        Web Services fundamentals (concept, SOAP, WSDL, etc)
        Understanding ASP.NET Web API
        SOAP-Based Web Services
        ASP.NET WebAPI fundamentals (Controllers, Routes, Models)
        WSDL
        WCF basics
        WCF Contracts (Service, Data, Message)
        Creating and configuring WCF Services
        Exposing WCF Services (Endpoints and Bindings)
        Consuming Services
        Handling WCF Exceptions
        WCF Sessions and Instancing
        Web API-based services
        Web API Security
        WCF Instrumentation (Debugging, Tracing, Monitoring)
        WCF Data Services
        Implement caching
        Default configuration model in WCF

Web server applications (.NET)

        Understanding Web Communications
        Configure projects, solutions, and reference assemblies
        Working with Web Configuration Files
        Implementing User Profiles, Authentication, and Authorization
        Working with User Profiles
        Using ASP.NET Membership
        The FormsAuthentication, Membership, Roles Classes
        Configure Authentication and Authorization
        Understanding the MVC pattern
        Building loosely coupling components
        The MVC Application Structure
        Implementing Controllers/Views/Models
        Model Binding
        MVC Life cycle
        Data Annotations and Validation
        ASP.NET Routing
        Bundling and Minification
        HTTP Request Processing in IIS
        HTTP Handlers and HTTP Modules
        Handling Events and Managing State
        Using Master Pages, Themes and Caching
        What is wrong Asp.Net Web Forms?
        URL and Ajax Helper Methods
        Attribute routing
        Filters
        Model Validation
        Using Asynchronous Methods
        Owin and Katana
        SignalR
        ASP.NET Web API
        Writing an API Controller
        Adding Web API to an ASP.NET Project
        Diagnostics
        Using the Error Logging Modules and Handlers
        Configurations in MVC Project
        Adding a Class Library Project
        Deployment to Azure
```

[Construction - DB]

```
    Data Access Layer

            ADO.NET Connected Classes
            ADO.NET Disconnected Classes
            Handling Exceptions
            Manage transactions
            Executing asynchronous queries
            Manage update conflicts between online data and offline data (
Synchronization Services)

            Entity Framework

                - Model First VS Database First VS Code First Approaches
:white_check_mark:
                - Entity Data Modeling Fundamentals  :white_check_mark:
                - Querying an Entity Data Model
                - Loading Entities and Navigation Properties
                - Modeling a Many-to-Many, Self-Referencing Relationship
                - Stored Procedures in the Entity Framework
                - Code First Migrations and Deployment with the Entity Framework
                - Async and Stored Procedures with the Entity Framework
                - Handling Concurrency with the Entity Framework 6

            NET Micro ORMs
            Dapper

    SQL

            Creating database objects
            Altering and Destroying RDBMS Objects
            DDL, DML, DCL understanding
            SQL data types
            Data manipulation (insert, update, delete)
            Retrieving data, Aggregations
            Joins understanding
            Combining the results of multiple queries
            Sessions, transactions, locks
            Implementing stored procedures, user-defined functions, triggers
            Cursors
```

[Verification]

```
    Automated-Testing.NET

            Unit testing fundamentals and elementary tests
            Organizing and building unit tests (Nunit
            Managing test suites
            Working with test data
            Running unit tests
            Reporting unit results
            Mocks/Stubs (Nmock, POCMock, etc.)
            Code Coverage Tools (NCover)
            Load Testing ASP.NET Web Applications
            UI automation testing and tools (Coded UI, Selenium, Ranorex)

    Code-Quality.NET

            Code Quality .NET
            Guidelines for Names
            Automated coding standards enforcement (StyleCop)
            Code Reviews and Toolset
            Use Work Items (TODO, BUG etc.)
            Preemptive Error Detection (FxCop)
            Desirable characteristics of a design (minimal complexity, ease of
maintenance, minimal connectedness etc)
            Creating high quality classes
            Creating high quality methods
            Guidelines for initializing variables
            Exceptions and error handling techniques
            Best practices of working with data types
            Code commenting practices
```

[Configuration Management]

```
    Managing versions

            Fundamental concepts: revisions, working copy, repository, branch,
baseline, trunk
            Versioning Models
            Distributed Version Control basics
            Distributed systems advantages and weak sides
            VCS Management life-cycle on major tools (clone, commit, update, revert,
merge, resolve, etc)
            Branching/Merging strategies
            Blaming (annotate)
            Revision graph/log actions (Git)
            Integrating with Issue Tracking Systems
            Source control Best Practices

    Product builds and Continious Integration

            Automated build concept
            Building with build tool (ant, make, rake, ...), cleaning up built
product
            Scripting multiphase (build, testing, deployment, …) build process
            Integrating building of product installer
            Generating release notes and/or other release documentation
            Development of scheduled builds (night builds)
            Monitoring build process
            Build status reporting (notification)
            Integrating deployment stage, moving product to release area
            Integrating with source control, versioning, tagging, building release
            Continuous integration concept, best practices and framework
```