Contents

[Engineering Practices 3](#_Toc73096660)

[DAY 1 – GIT, Agile Methodology, Architecture Patterns 3](#_Toc73096661)

[.Net Core Framework 3](#_Toc73096662)

[DAY 2 – Framework Fundamentals and Project Types 3](#_Toc73096663)

[DAY 3 – Getting hands-on with various Project Types 4](#_Toc73096664)

[C# Language 4](#_Toc73096665)

[DAY 4 - Fundamentals 4](#_Toc73096666)

[DAY 5 – Access Specifies and OOPS 5](#_Toc73096667)

[DAY 6 – New Features (7.0 – 9.0) 6](#_Toc73096668)

[DAY 7 – Lambdas, Delegates and Events 7](#_Toc73096669)

[Day 8 – Generics and Collections 7](#_Toc73096670)

[DAY 9 – Asynchronous programming and Multithreading 8](#_Toc73096671)

[DAY 10 – Reflection, Serialization, Files and Streams 8](#_Toc73096672)

[Design Patterns 9](#_Toc73096673)

[DAY 11 9](#_Toc73096674)

[--- Code Assessment --- 9](#_Toc73096675)

[Log4Net Course Content 9](#_Toc73096676)

[DAY 12 9](#_Toc73096677)

[Asp.Net Core 10](#_Toc73096678)

[DAY 13 - Introduction 10](#_Toc73096679)

[DAY 14 - APIs 10](#_Toc73096680)

[DAY 15 – Entity Framework 11](#_Toc73096681)

[--- Code Assessment --- 11](#_Toc73096682)

[Asp.net Core MVC 11](#_Toc73096683)

[DAY 16 - Introduction 11](#_Toc73096684)

[DAY 17 – Tag Helpers and Partial Views 12](#_Toc73096685)

[DAY 18 – Layouts and Filters 12](#_Toc73096686)

[DAY 19 – Authentication, Authorization and Caching 12](#_Toc73096687)

[--- Code Assessment --- 13](#_Toc73096688)

[Unit Testing 13](#_Toc73096689)

[DAY 20 13](#_Toc73096690)

[Azure 13](#_Toc73096691)

[DAY 21 13](#_Toc73096692)

[CI CD using Jenkins 14](#_Toc73096693)

[DAY 22 14](#_Toc73096694)

[SQL Server 14](#_Toc73096695)

[DAY 23 14](#_Toc73096696)

[Mongo DB 15](#_Toc73096697)

[DAY 24 15](#_Toc73096698)

[--- Code Assessment --- 15](#_Toc73096699)

# Engineering Practices

## DAY 1 – GIT, Agile Methodology, Architecture Patterns

What is Source Control?

Introduction to Git, Github

Installation, configuration

Initializing the Git Repository

Adding Files, Committing files through Git Bash

Cloning a repository

Git Branching

Demos & Labs

Agile & Unit Testing

Introduction to Agile

Where is automation required in the phases of this cycle?

What is Unit Testing

Understanding TDD

Introduction to CI / CD

Principles of CI / CD

Benefits of CI/CD

When is CI / CD not feasible?

Tools in the Market

Tools that we would use

Monolithic Architecture

Microservice Architecture

Standard Enterprise Project Layered Architecture

# .Net Core Framework

## DAY 2 – Framework Fundamentals and Project Types

.Net Core Introduction

Understanding .Net Core Architecture

VS Code as the IDE

IL, CLR and portability over Operating systems & hardware

Assemblies - Private and shared Assemblies

An example of IL code

Solution Explorer & project structure

Type System

Introduction to .Net CLI

CLI Commands

Creating projects using CLI

Executing projects using CLI

## DAY 3 – Getting hands-on with various Project Types

Understanding different Project Types

Creating your first project in C#

Executing the project

Understanding Build & Bin

Understanding modes of Execution – Debug & Release

When to use Debug & Release Modes

Introduction to Nuget

Introduction to Nuget Command Line

Installing Packages via command line

Understanding VS Code extensions for .Net

# C# Language

## DAY 4 - Fundamentals

Introduction to C#

Datatypes in C#

By reference, by Value parameters

Type Casting Vs Type conversion

Out parameters

Variable number of parameters

Creating your first Application in C#

Creating Functions

Understanding string interpolation

Understanding Anonymous methods

.Net as an Object Oriented Platform

Importance of Classes

Parts of a Class

What are not parts of class

Understanding how memory is allocated

Stack & Heap

How to Dispose objects in memory

Stages of GC

Adding References to other projects

Creating a Asp.Net Core Web Api Project

Adding functions in the form of API

Viewing Results on browser

Demos & Labs

## DAY 5 – Access Specifies and OOPS

Access Specifiers in C#

Public

Private

Protected

Internal

Static

Constructors & Destructors

Object Oriented Programming

Parts of OOPs

Pillars of OOPs

Understanding Abstraction, Encapsulation, Inheritance & Polymorphism

UML structures

Inheritance between classes

Inheritance between constructors

Calling one constructor from another

Flavours of Overloading & Overriding

Importance of static Vs Singleton – Sample application

Concrete Vs Abstract Classes

Creating Abstract Classes

Understanding Interfaces as contracts

Abstract Classes Vs Interfaces – When to use which one

Sealed Classes

Runtime Polymorphism

Virtual functions

Abstract functions

What is dependency Injection

Using Dependency injection in C#

Tools in the market for DI

Demos with multi-level inheritance with virtual & abstract functions

Demos, Experiments & Labs

## DAY 6 – New Features (7.0 – 9.0)

New features in C#

Autoproperties & Immutability

String Interpolation

Nameof()

Nullability & null-conditional operators

Index initializers

Tuples

Discards

Local Functions

Pattern Matching

Exception Handling

Types of Exceptions

Throw…catch

When to use throw & catch – Best Practices

Demos & Labs

## DAY 7 – Lambdas, Delegates and Events

Lambda Functions

What are Lambdas?

Syntax & Usage

Creating Lambda Expressions

Built-in Lambdas

Creating Lambda Functions & Properties

Understanding Delegates

Creating Delegates

What are events

Creating Events

How Events use Delegates

Creating Delegates as lambdas

Action<>, Func<>, Predicate<>

Creating Loops as Lambda expressions

Demos & Labs

## Day 8 – Generics and Collections

Generics

What are Generics

Problems with return data types

Object Type – Issues, Addressed by Generics

Introduction to Collections

Arrays Vs Collections

Generic Collections

Creating Generic Functions

Creating Generic Classes

Understanding IEnumerable, IEnumerator, IList

Using List<>, Dictionary<>, NameValueCollection<>

Demos & Labs

## DAY 9 – Asynchronous programming and Multithreading

Asynchronous Programming

What is Asynchronous Programming

Threading concepts

Spawning Threads, Wait handles, Fork

Understanding Tasks and Task Manager

Introduction to async & Await

Monitoring async awaits using Thread windows

Introduction to Task Parallel Library

Creating Tasks – Task<>, Task<T>

Using TaskFactory

Using Lambdas for Creating & Running Tasks

Verifying Parallization across cores

Handling exceptions

Task Cancellation

Using Concurrent Collections

Demos & Labs

## DAY 10 – Reflection, Serialization, Files and Streams

Reflection

What is Reflection?

Usecases where reflection can be used

Limitations of Reflection

Runtime Type Discovery

Dynamic Object Creation

Dynamic invocation of methods and properties

Serialization

JSON Serialization

Files and Streams

Demos & Labs

Introduction to Var & dynamic keywords

What is DLR

Practical usage of dynamic keyword

Intercommunication between languages

# Design Patterns

## DAY 11

What are design Patterns

SOLID Principles

Usage and decision making

GOF Patterns overview

Creational Vs Structural Vs Behavioral Patterns

Abstract Factory Pattern – UML, Implementation, Usecase

Adapter pattern – UML, Implementation, Usecase

Demos & Labs

### --- Code Assessment ---

# Log4Net Course Content

## DAY 12

What is Logging?

Practical need for logging as a monitoring & support tool

Manually logging vs using logging tools

Introduction to Log4Net

Similar players in the market

Log Levels

Configurations for use

Create a Log with Log4net

Parameterized Logs

Logging to files Vs Logging to DB

Some Best Practices

Creating a Logging System

Demos & Labs

# Asp.Net Core

## DAY 13 - Introduction

Introduction to Asp.Net Core

Creating a Web Api Project

What is SOA?

API Driven Architecture

What is REST?

Understanding HTTP & Http Verbs, HTTP Headers, Status Codes

Introduction to Postman

Introduction to Server Side Routing

Creating your First API

Creating Apis using GET, POST, PUT DELETE

Intercepting Requests Via Postman

Understanding Content Types & Serialization

Demos & Labs

## DAY 14 - APIs

Creating Routes for Apis

What is Model Binding?

FromBody, FromRoute, FromQuery, FromForm Binders

Adding Model Validations

What is a middleware?

How to plug in a middleware?

Built-in Middlewares

Creating a simple middleware

What is deployment and hosting

Types of Deployment

Deploying App to a server

Demos & Labs

## DAY 15 – Entity Framework

Introduction to Entity Framework Core

What problems does EF Core solve?

Architecture & Working

When to use EF Core

When NOT to use EF Core

What is a DB Context

Adding EF Core to the Web Api Project

EF Core Approaches

Creating APIs using EF Core

Creating Database Tables

Demos & Labs

### --- Code Assessment ---

# Asp.net Core MVC

## DAY 16 - Introduction

Difference between Web Api & MVC

Introduction to MVC Architecture Pattern

Creating an MVC Project

Convention Over Configuration

Understanding Actions and non-Actions in controllers

Creating Models & Validations

Introduction to Razor View Engine

Creating Views using Razor

Injecting Controllers into Views

Demos & Labs

## DAY 17 – Tag Helpers and Partial Views

Sharing Data using ViewData, ViewBag

What are Tag Helpers

Creating Views using Tag Helpers

What are Partial Views

Declaring Partial Views

Referencing Partial Views

Introduction to Areas in MVC Core

Creating Areas

Demos & Labs

## DAY 18 – Layouts and Filters

What are Layouts

How are Layouts helpful

Creating Layout Sections

Creating Ignore Sections

Adding Scripts to MVC

Introduction to Filters

Action Filters

Result Filters

Exception Filters

Resource Filters

Demos & Labs

## DAY 19 – Authentication, Authorization and Caching

Authentication & Authorization in Asp.Net Core

Configuring Authentication

Types of authentication

Adding authentication to application

Adding Authorization Filters

Caching in MVC Core

Types of Caching

Creating an In-Memory cache

Data Validation

Application State and Error Handling – (Error Handling will be part of model Binders & covered Web Api Validations. Application state applies to .Net frameworks upto 4.8.)

Demos & Labs

### --- Code Assessment ---

# Unit Testing

## DAY 20

Unit testing Review

Introduction to TDD

Introduction to NUnit

Creating an NUnit test Project

Adding Assertions

Running Tests

Introduction to Mocks

Why are mocks used

Adding Moq to Test Project

Creating Mocks

# Azure

## DAY 21

Introduction to Cloud Computing and understanding Cloud concepts

Introduction to Azure Microsoft Azure

Overview of Azure Core Services

Creating an account

Compute

Networking

Data & Storage Services

Accounts & identity

Compliance, security & cost

Deploying WebApp to Cloud AppService

Demos & Labs

# CI CD using Jenkins

## DAY 22

What is Jenkins?

Benefits of using Jenkins

Jenkins Pipelines

Jenkins Integrations

Demos & Labs

# SQL Server

## DAY 23

RDBMS Concepts (MSSql Server)

Introduction to RDBMS

Overview of Database Models

Overview of ER Diagram and Normalization

Introduction to SQL

SQL Data types

Introduction to DQL, DDL, DML, DCL

CRUD operations with database

INSERT/UPDATE/DELETE/SELECT

Using functions and ordering the result

Group By, Order By, Having

Use of LIKE, WHERE & IN clause

Dropping / truncating a table

SQL Joins - Inner, Left, Right and Self joins

Database Views

Demos & Labs

# Mongo DB

## DAY 24

MongoDB Basics

NoSQL Database

Introduction to NoSQL

Types of NoSQL Database (Concepts only)

Column, Document, Key-Value, Graph, Multi-Model

Introduction to MongoDB (version – 3.6)

Introduction to MongoDB

Installation and set up

Data Modelling and Architecture

Advantages of MongoDB - Why & Where to use.

Querying in MongoDB

Creating & Dropping Database & Collection

Insert, Query, Update & Delete Document

find(), limit(), skip(), sort(), ensureIndex() methods

### --- Code Assessment ---