📘 Complete C# Roadmap (Basics → Advanced)

🔹 1. C# & .NET Fundamentals

- Introduction to C# and .NET
- Architecture of .NET Framework
  - CLR (Common Language Runtime)
  - CLS (Common Language Specification)
  - CTS (Common Type System)
  - MSIL, Assemblies, Metadata, Manifest, PE Files
  - Application Domain
  - Types of Applications
  - Evolution of Framework Versions
- Base Class Library (BCL)
- Components of .NET

🔹 2. Getting Started with C#

- Overview of C#
- Structure of a C# Program
- Developing Console Applications
- Using Visual Studio IDE (Projects & Solutions)
- Entry Point Method (Main)
- Command Line Parameters
- Compiling & Building Projects
- Namespaces

🔹 3. Data Types & Variables

- Variables and Data Types
- Value Types vs Reference Types
- Datatypes in C#

- Strings & StringBuilder
- Operators
- Statements

🔹 4. Control Flow

- Control Statements (if, else, switch)
- Loops (for, while, do-while, foreach)
- Break, Continue, Goto
- Control Structures PDF

🔹 5. Arrays & Collections Basics

- Arrays (1D, 2D, Jagged)
- Strings vs Arrays
- ArrayList, List, Dictionary, HashTable
- Collections in C#

🔹 6. Methods & Functions

- Procedures and Functions
- Method Overloading
- Extension Methods
- Anonymous Methods
- Lambda Expressions
- Named Parameters
- var and dynamic

🔹 7. Object-Oriented Programming (OOP)

- Understanding OOP (Identity, State, Behavior)
- Classes & Objects

- Partial Classes & Methods
- this Reference
- Constructors & Destructors
- Properties & Indexers
- Access Modifiers (public, private, protected, internal)
- Inheritance (Single, Multi-level)
- Calling Base Class Constructor
- Sealed Classes (Non-Inheritable)
- Abstract Classes
- Interfaces & Interface Inheritance
- Difference: Abstraction vs Inheritance
- Polymorphism (Compile-time & Runtime)
- Operator Overloading
- Inner Classes
- Encapsulation

🔹 8. Advanced C# Features

- Anonymous Types
- Static Members (Shared)
- Attributes and Usage
- Enums & Structs
- Memory Management & Garbage Collection
- Assemblies (Private, Shared)
- Strong Names & GAC (Global Assembly Cache)
- Deploying Assemblies

🔹 9. Exception Handling

- try-catch-finally

- Raising Exceptions with throw
- Pre-defined Exception Classes
- Custom Exception Classes
- Using Statement (for resources cleanup)

◆ 10. File Handling & Streams

- File I/O
- Creating Files & Folders
- Reading & Writing Files
- Streams Introduction
- Byte Streams
- StreamReader / StreamWriter
- Serialization & Deserialization

◆ 11. Delegates, Events & LINQ

- Delegates
- Anonymous Methods & Lambda
- Events in C#
- LINQ (Language Integrated Query)
    - IEnumerable & IEnumerator
    - Working with Lists & Dictionaries

◆ 12. Advanced Collections & Data Structures

- List vs Array vs ArrayList
- Stacks
- Queues
- HashTables
- Generic Classes & Methods

🔹 13. Database Connectivity

- Introduction to SQL & NoSQL
- Connecting with a Database
- Creating Tables
- Insert, Update, Delete Queries
- Parameterized Queries
- Handling Exceptions in DB Connections

🔹 14. Modern C# & Best Practices

- C# 9 Features (records, init-only setters, pattern matching improvements)
- Regular Expressions
- Math Class, Random Class, DateTime
- Auto Implemented Properties & Collections
- Dependency Injection Basics

🔹 15. Testing & Version Control

- NUnit Basics
- Writing Unit Tests in C#
- Using Git with Visual Studio

📅 Suggested Learning Schedule (12 Weeks Plan)

👉 Assuming you study **5–6 hours per week**, here's a breakdown:

Weeks 1–2: Fundamentals

- Introduction, CLR, CLS, CTS, .NET Overview
- Visual Studio Projects, Main Method

- Data Types, Variables, Strings, Operators
- Control Flow Statements

Weeks 3–4: Arrays & Methods

- Arrays & Strings
- Methods, Parameters, Overloading
- Anonymous Methods, Lambdas, Extension Methods

Weeks 5–6: OOP Core

- Classes & Objects
- Constructors, Destructors
- Properties, Indexers
- Inheritance, Abstract, Interfaces
- Encapsulation & Polymorphism

Weeks 7–8: Advanced OOP & Assemblies

- Partial Classes, Sealed Classes
- Operator Overloading, Inner Classes
- Attributes, Enums, Structs
- Garbage Collection
- Assemblies & GAC

Weeks 9–10: Exceptions & File Handling

- Exception Handling (try, catch, throw, custom)
- File I/O, Streams (Reader/Writer, Byte Streams)
- Serialization / Deserialization

Weeks 11: Delegates, LINQ & Collections

- Delegates, Events
- LINQ, IEnumerable, IEnumerator
- Collections (List, ArrayList, Dictionary, HashTable)
- Generics, Stacks, Queues

Week 12: Database + Testing + Git

- Database Connectivity (SQL, Queries, Exceptions)
- NUnit Unit Testing
- Git Basics in Visual Studio
- Final Revision + Mini Project (e.g., Student Management System)