

Guide2Code - C# Programming Roadmap

◆ Phase 1: Beginner Level

📌 Topics to Learn:

1. Introduction to C# (Installation, .NET Framework, Visual Studio)
2. Basic Syntax & Structure (Main method, Data Types, Variables)
3. Operators (Arithmetic, Comparison, Logical, Bitwise)
4. Control Statements (if-else, switch-case)
5. Loops (for, while, do-while, foreach, break, continue)
6. Methods (Functions, Parameters, Return Types, Recursion)
7. Arrays & Lists (1D, 2D Arrays, Lists, Dictionary)
8. Exception Handling (try-catch, finally, throw)
9. File Handling (Reading, Writing, Appending Files)
10. Basic Debugging (Breakpoints, Watch Window, Console Logging)

🚀 Beginner Project Ideas:

- 🧮 **Simple Calculator** – Perform basic arithmetic operations
- 📊 **Student Grade Calculator** – Input marks and display grades
- 🎲 **Number Guessing Game** – User guesses a number, program gives hints
- 🌡️ **Temperature Converter** – Convert Celsius ↔ Fahrenheit
- 📅 **Age Calculator** – Calculate age based on birth year
- 🏦 **Bank Account System** – Basic deposit and withdrawal operations







◆ Phase 2: Intermediate Level

📌 Topics to Learn:

1. Object-Oriented Programming (Classes, Objects, Inheritance, Polymorphism)
2. LINQ (Language-Integrated Query) – Querying Collections & Databases
3. Generics (Generic Classes, Methods, Collections)
4. Delegates & Events (Event Handling, Anonymous Methods)

5. Multithreading & Asynchronous Programming (Task, async-await)
6. Working with APIs (REST API, JSON Handling, HttpClient)
7. Entity Framework (Database ORM, CRUD Operations)
8. Windows Forms & WPF (Building Desktop Applications)
9. Unit Testing (NUnit, xUnit, Test-Driven Development)
10. Advanced Exception Handling (Custom Exceptions, Logging)

Intermediate Project Ideas:



-  **Library Management System** – Store and manage books
-  **Contact Management System** – Add, edit, delete contacts
-  **To-Do List App** – Task management using C# GUI
-  **Bank Management System** – Manage user accounts and transactions
-  **Tic-Tac-Toe Game** – Playable in the console or GUI
-  **Web Scraper** – Extract information from websites using C#





Phase 3: Advanced Level

Topics to Learn:

1. ASP.NET Core (MVC, Web API, Dependency Injection)
2. Microservices Architecture (Docker, Kubernetes, API Gateway)
3. Security in C# (JWT Authentication, OAuth, Data Encryption)
4. Blazor Framework (Web App Development)
5. Game Development with Unity & C#
6. Machine Learning with ML.NET
7. Cloud Computing (Azure, AWS Deployment)
8. C# Performance Optimization & Profiling
9. IoT with C# (Raspberry Pi, Sensor Integration)
10. System-Level Programming (Working with Memory, Low-Level Operations)

Advanced Project Ideas:

-  **Chat Application** – Real-time messaging with SignalR
-  **E-commerce Website** – Full-fledged shopping platform using ASP.NET

-  **Stock Market Predictor** – Machine Learning with ML.NET
-  **Secure Login System** – Authentication using JWT and OAuth
-  **Face Recognition System** – AI-based image processing in C#
-  **IoT-Based Smart Home System** – Home automation using C#