**C# 3.5 (2007) - LINQ and Functional Features**

C# 3.5 was a major step forward with Language Integrated Query (LINQ) and functional programming concepts.

* **LINQ (Language Integrated Query)** → Query operations on collections
* **Lambda Expressions** → Shorter syntax for inline functions
* **Extension Methods** → Add methods to existing types without modifying them
* **Anonymous Types** → Types without explicitly defining classes
* **Implicitly Typed Variables (var)** → Type inference for local variables
* **Automatic Properties** → Auto-implemented properties (get; set;)
* **Object and Collection Initializers** → Simplified object instantiation
* **Partial Methods** → Methods that can be optionally implemented in partial classes

**C# 4.0 (2010) - Dynamic Programming**

C# 4.0 focused on improving interop with dynamic language.

* **Dynamic Type (dynamic)** → Runtime type resolution
* **Named and Optional Parameters** → Simplifies method overloading
* **Covariance and Contravariance** → Enables flexible generic type assignments

**C# 6.0 (2015) - Syntax Enhancements & Readability**

C# 6.0 brought syntactical improvements for cleaner and more readable code.

* **Expression-bodied Methods and Properties** → Single-line method/property definitions
* **Null-Conditional Operator (?.)** → Safe navigation (avoids null reference exceptions)
* **Auto-Property Initializers** → Default values for properties
* **String Interpolation ($"{}")** → Improved string formatting
* **nameof Operator** → Safer refactoring by avoiding magic strings
* **Exception Filters** → More precise exception handling
* **Await in Catch and Finally Blocks** → Async exception handling

**C# 7.0 (2017) - Performance & Productivity**

C# 7.0 introduced several performance improvements and new ways to structure code.

* **Tuples and Deconstruction** → Return multiple values from methods easily
* **Pattern Matching** → is expression and switch enhancements
* **Local Functions** → Functions within functions for better encapsulation
* **Out Variables** → Declare out parameters inline
* **Ref Returns and Locals** → Return references instead of values for performance
* **Throw Expressions** → Throw exceptions inline

**C# 7.1 - 7.3 Minor Enhancements**

* **Default Expressions** → default keyword supports type inference (default(int))
* **Async Main Method** → static async Task Main()

**C# 8.0 (2019) - Null Safety and Modern Constructs**

C# 8.0 focused on null safety, asynchronous programming, and new functional features.

* **Nullable Reference Types** → Compiler warnings for potential null reference issues
* **Async Streams (await foreach)** → Async iteration over collections
* **Switch Expressions** → More concise switch statements
* **Using Declarations** → Auto-dispose resources without nesting in blocks
* **Interface Default Methods** → Implement default logic inside interfaces