Got it — you want a **clear, consolidated list of C# language features from version 9 through 14**, with **short runnable code examples** for each.
I'll cover **major features only** (not every minor compiler tweak) so you can quickly see what's new and how to use it.

C# 9 (2020, .NET 5)

1. Record Types

```
public record Person(string FirstName, string LastName);

var p1 = new Person("John", "Doe");
var p2 = p1 with { LastName = "Smith" }; // Non-destructive mutation
Console.WriteLine(p1); // Person { FirstName = John, LastName = Doe }
```

2. Init-only Setters

```
public class Car
{
    public string Model { get; init; }
}

var car = new Car { Model = "Tesla" };
// car.Model = "BMW"; // ❌ Compile error
```

3. Top-level Statements

```
using System;

Console.WriteLine("Hello, World!");
```

4. Pattern Matching Enhancements

```
object obj = 42;
if (obj is int { } number and > 40)
    Console.WriteLine($"Number > 40: {number}");
```

C# 10 (2021, .NET 6)

1. Global Using Directives

```
// global using System; // in a separate file (e.g., GlobalUsings.cs)
Console.WriteLine("No need to add 'using System;' in every file.");
```

2. File-scoped Namespace

```
namespace MyApp;

class Program { }
```

3. Record Structs

```
public readonly record struct Point(int X, int Y);
```

4. Constant Interpolated Strings

```
const string Name = "World";
const string Greeting = $"Hello, {Name}!";
Console.WriteLine(Greeting);
```

C# 11 (2022, .NET 7)

1. Required Members

```
public class User
{
    public required string Name { get; init; }
}

var u = new User { Name = "Alice" }; // ✅ Must set Name
```

2. List Patterns

```
int[] numbers = { 1, 2, 3 };
if (numbers is [1, 2, _])
    Console.WriteLine("Starts with 1, 2");
```

3. Raw String Literals

```
string json = """
{
  "name": "Alice",
  "age": 30
}
""";
Console.WriteLine(json);
```

C# 12 (2023, .NET 8)

1. Primary Constructors for Classes

```
public class Person(string name)
{
    public string Name { get; } = name;
}

var p = new Person("Bob");
Console.WriteLine(p.Name);
```

2. Collection Expressions

```
int[] nums = [1, 2, 3];
List<string> names = ["Alice", "Bob"];
```

3. Default Lambda Parameters

```
var greet = (string name = "World") => Console.WriteLine($"Hello,
{name}!");
greet();
greet("Alice");
```

C# 13 (2024, .NET 9)

1. Params Collections

```
void PrintAll(params ReadOnlySpan<int> numbers)
{
```

```
    foreach (var n in numbers) Console.Write($"{n} ");
}
PrintAll(1, 2, 3, 4);
```

2. Partial Properties

```
partial class MyClass
{
    public partial int MyValue { get; set; }
}
```

C# 14 (2025, .NET 10)

(Based on preview features — subject to change)

1. First-class `Span<T>` and `ReadOnlySpan<T>`

```
void Process(ReadOnlySpan<int> data)
{
    foreach (var d in data) Console.Write($"{d} ");
}

int[] arr = { 1, 2, 3 };
Process(arr); // Implicit conversion to ReadOnlySpan<int>
```

2. Extension Members in Interfaces

```
public interface ILogger
{
    void Log(string message) => Console.WriteLine($"[LOG]
{message}");
}
```

```
class ConsoleLogger : ILogger { }

var logger = new ConsoleLogger();
logger.Log("Hello from C# 14!");
```

✅ **Tip:** If you want, I can prepare a **single C# console app** that demonstrates **all features from C# 9 to 14 in one runnable file** so you can test them directly.

Do you want me to prepare that combined runnable example?