Language Map for C#

Variable Declaration
Is this language strongly typed or dynamically typed? Provide at least three examples (with different data types or keywords) of how variables are declared in this language.
	C# is strongly typed, like Java.
	
	bool isStronglyTyped = true;
	int counter = 0;
	string message = “Hello World”;

Data Types *List all of the data types (and ranges) supported by this language.*	**byte**: 0 to 255 (1 byte) **sbyte**: -128 to 127 (1 byte) **short**: -32,768 to 32,767 (2 bytes) **ushort**: 0 to 65,535 (2 bytes) **int**: -2,147,483,648 to 2,147,483,647 (4 bytes) **uint**: 0 to 4,294,967,295 (4 bytes) **long**: -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 (8 bytes) **ulong**: 0 to 18,446,744,073,709,551,615 (8 bytes) **float**: ±1.5 x 10^−45 to ±3.4 x 10^38 (4 bytes) **double**: ±5.0 x 10^−324 to ±1.7 x 10^308 (8 bytes) **decimal**: ±1.0 x 10^-28 to ±7.9 x 10^28 (16 bytes) **char**: single character (2 bytes). Single quotes. **string**: sequence of characters (immutable). Double quotes. **bool**: true or false (1 byte)

Selection Structures
Provide examples of all selection structures supported by this language (if, if else, etc.) ***Don’t just list them, show code samples of how each would look in a real program.***
	if statement:
	if (condition){
	  // Code to execute if condition is true
	}
	
	if-else statement (can also do else-if):
	if (condition){
	  // Code;
	}else{
	  // Code;
	}

	switch statement:
	switch (value){
	  case value1:
	  // Code;
	  break;
	  case value2:
	  // Code;
	  break;
	  default:
	  // Code;
	  break;
	}

Repetition Structures
Provide examples of all repetition structures supported by this language (loops, etc.) ***Don't just list them, show code samples of how each would look in a real program.***
	for loop:
	for (int i = 0; i < 10; i++) {
	  // Code;
	}
	
	foreach loop (used to loop items in a data set, like an array):
	foreach (int number in numbers) {
	  // Code;
	}
	
	while loop:
	while (count < 5) {
	  // Code;
	}
	
	do-while loop:
	do { // Code;
	} while (i < 10);

Arrays
If this language supports arrays, provide ***at least two examples*** *of creating an array with a primitive or String data types (e.g. float, int, String, etc.) If the language supports*
	Note – Can declare array values with [] or {}
	
	// Declare a single-dimensional array of 5 integers.
	int[] array1 = new int[5];

declaring arrays in multiple ways, provide an example of way.	// Declare and set array element values.
	int[] array2 = [1, 2, 3, 4, 5, 6];
	// Declare a two-dimensional array.
	int[,] multiDimensionalArray1 = new int[2, 3];
	
	// Declare and set array element values.
	int[,] multiDimensionalArray2 = { { 1, 2, 3 }, { 4, 5, 6 } };
	
	// Declare a jagged array.
	int[][] jaggedArray = new int[6][];
	
	// Set the values of the first array in the jagged array structure.
	jaggedArray[0] = [1, 2, 3, 4];

Data Structures *If this language provides a standard set of data structures, provide a list of the data structures and their Big-Oh complexity (identify what the complexity represents).*	**1. Array** • **Access:** O(1) - Accessing an element by index. • **Search:** O(n) - Searching for an element. • **Insertion:** O(n) - Inserting an element (due to shifting). • **Deletion:** O(n) - Deleting an element (due to shifting). **2. List<T> (Dynamic Array)** • **Access:** O(1) • **Search:** O(n) • **Insertion:** O(n) • **Deletion:** O(n) **3. LinkedList<T>** • **Access:** O(n) - No direct access by index. • **Search:** O(n) • **Insertion:** O(1) (if you have a reference to the node). • **Deletion:** O(1) (if you have a reference to the node). **4. Dictionary<TKey, TValue> (Hash Table)** • **Access:** O(1) (average case) • **Search:** O(1) (average case) • **Insertion:** O(1) (average case) • **Deletion:** O(1) (average case) **5. HashSet<T>** • **Access:** O(1) (average case) • **Search:** O(1) (average case)

	• **Insertion:** O(1) (average case) • **Deletion:** O(1) (average case) **6. Queue<T>** • **Enqueue (Insertion):** O(1) • **Dequeue (Deletion):** O(1) • **Access:** O(n). **7. Stack<T>** • **Push (Insertion):** O(1) • **Pop (Deletion):** O(1) **8. SortedList<TKey, TValue>** • **Access:** O(log n) • **Search:** O(log n) • **Insertion:** O(n) (due to maintaining order). • **Deletion:** O(n) **9. SortedDictionary<TKey, TValue>** • **Access:** O(log n) • **Search:** O(log n) • **Insertion:** O(log n) • **Deletion:** O(log n) **10. Stack (in System.Collections)** • **Push (Insertion):** O(1) • **Pop (Deletion):** O(1) • **Access:** O(n) (to access elements).

Objects
If this language support object-orientation, provide an example of how you would write a simple object with a default constructor and then how you would instantiate it.
	public class Animal
	{
	
	// Fields **(These are not required if using automatic properties {get; set;})**
	private string species;
	private int age;
	
	// Properties
	public string Species { get; set; }
	public int Age { get; set; }
	
	// Default constructor

	public Animal()
	{
	Species = "string";
	Age = 0;
	}
	}
	
	class Application
	{
	static void Main(string[] args)
	{
	// Instantiate the Animal object using the default constructor
	Animal lion = new Animal();
	
	// Set property
	lion.Age = 12;
	
	}
	}

Runtime Environment *What runtime environment does this language compile to? For example, Java compiles to the Java Virtual Machine.* *Do other languages also compile to this runtime? If so, what these other languages?*	The Common Language Runtime (CLR) C++ Visual Basic VB.NET F#
Libraries/Frameworks *What are the popular libraries or frameworks used by programmers for this language? List at least three (3) and describe what they are used for.*	ASP.NET Core – Web Applications .NET Core - Cross-platform Applications Entity Framework – An ORM (Object-Relational Mapping) to simplify working with databases
Domains *What industries or domains use this programming language? Provide at least three specific examples of companies that use this language and what they use it for.*	Microsoft – Created and develops C# and uses it in its products and services Accenture – Uses C# for enterprise applications Stack Overflow – Uses C# for various website features on the frontend and backend Unity Technologies – Game development Intuit – Financial software products