Language Map for JavaScript

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. int myNum = 22; string myText = “Welcome”; char myLetter = ‘K’;
Data Types *List all of the data types (and ranges) supported by this language.*	int (4 bytes) – stores whole numbers from -2,147,483,648 to 2,147,483,647 long (8 bytes) – stores whole numbers -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 float (4 bytes) – stores fractional numbers, sufficient for storing 6 to 7 decimal digits double (8 bytes) – stores fractional numbers, sufficient for storing 15 decimal digits bool (1 bit) – stores true or false char (2 bytes) – stores a single character/letter, surrounded by single quotes string (2 bytes per character) – stores a sequence of characters, surrounded by double quotes

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:
	if (15 > 11)
	{
	  Console.WriteLine("15 is greater than 11");
	}
	
	if-else:
	int phrase = 6;
	if (phrase < 11)
	{
	  Console.WriteLine("Hello!");
	}
	else
	{
	  Console.WriteLine("Goodbye.");
	}
	
	if-else-if:
	int phrase = 21;
	if (phrase < 18)
	{
	  Console.WriteLine("Hello!");
	}
	else if (phrase < 26)

	{
	  Console.WriteLine("How are you?");
	}
	else
	{
	  Console.WriteLine("Goodbye.");
	}
	
	nested-if:
	int i = 30;
	if (i == 30)
	{
	  if (i < 24)
	  Console.WriteLine("The number is smaller than 24");
	  else
	  Console.WriteLine("The number is greater than 24");
	}
	
	switch:
	int season = 3;
	switch (season)
	{
	  case 1:
	  Console.WriteLine("Spring");
	  break;
	  case 2:
	  Console.WriteLine("Summer");
	  break;
	  case 3:
	  Console.WriteLine("Fall");
	  break;
	  case 4:
	  Console.WriteLine("Winter");
	  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.***
	while loop:
	int i = 0;
	while (i < 3)
	{
	  Console.WriteLine(i);

	i++;
	}
	
	do while loop:
	int i = 0;
	do
	{
	Console.WriteLine(i);
	i++;
	}
	while (i < 3);
	
	for loop:
	for (int i = 0; i < 3; i++)
	{
	Console.WriteLine(i);
	}
	
	foreach loop:
	string[] colors = {"Yellow", "Purple", "Green", "Orange"};
	foreach (string i in colors)
	{
	Console.WriteLine(i);
	}

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 declaring arrays in multiple ways, provide an example of way.*	string[ ] names = {“Rachel”, “Kyle”, “Jonathan”, “Ruby”}; string [ ] names = new string [4]; string [ ] names = new string [4] {“Rachel”, “Kyle”, “Jonathan”, “Ruby”}; string [ ] names = new string[ ] {“Rachel”, “Kyle”, “Jonathan”, “Ruby”}; int [ ] numbers = {5, 10, 15, 20};
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).*	C# Arrays: Big-Oh is O(1) for accessing elements at a specific index. O(n) for search methods. If adding an element to the end of the array, Big O is O(1). If adding to the beginning of the array, all elements must be shifted and Big O is O(n). C# Stack: If the count is less than the capacity of the stack, Push() is O(1). If capacity needs to be increased to accommodate new element, Push() is O(n). Pop is O(1). C# Queue: Big-Oh is O(n) for accessing a specific element. Enqueue and dequeue is O(1). C# Hashtable: Search methods are O(1).

	C# Dictionary: Search methods are O(1) if the dictionary is implemented as a hashtable. If the dictionary is an unsorted linked list, search methods are O(n). If it is implemented as a sorted array, search methods are O(log n). Addition, removal and traversal are O(n) for a sorted array-based dictionary. C# LinkedList: Accessing elements are O(n). Adding an element to the beginning of the list is O(1). Adding an element to the end of the list is O(n).

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.
	class Wizard
	{
	public string power;
	
	public Wizard(string p)
	{
	power = p;
	}
	
	}
	
	class WizardDriver
	{
	static void Main(string[] args)
	{
	Wizard Hermione = new Wizard(“levitation”);
	}
	
	}

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?*	C# has a runtime environment called Common Language Runtime (CLR). It is part of the .NET framework. Other languages that use CLR are Visual C++, Visual Basic, J#, JavaScript
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.*	AutoMapper: A mapping library that simplifies the process of mapping objects between different types. It eliminates the need for writing repetitive mapping code and reduces the risk of errors. NUnit: A testing framework that enables developers to write and run unit tests. It provides assertions, test fixtures, and runners, and supports various test categories, such as integration, performance, and security. Dapper: An object mapper for .NET. It provides a set of tools for working with data in a C# application.
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*	Microsoft uses C# for web and game development. Service Titan uses C# for web services and android app development. City National Bank uses C# for building cloud-based applications.

*for***. *E.g. Company X uses C# for its line of business applications.***	