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 a strongly typed language. o Note: This means you have to declare the type of each variable explicitly when you create is because the variable is determined at compile time and cannot be changed. • How are variables declared in C#: o Syntax for creating a variable: ▪ type variableName = value; o Integer example: ▪ int primeNumber = 7; o Double example: ▪ double price = 10.99; o String example: ▪ String petName = “Drake”; o Character example: ▪ char initial = ‘M’; o Boolean example: ▪ bool isActive = true;
Data Types *List all of the data types (and ranges) supported by this language.*	• Integral Types (Whole numbers) o byte ▪ 0 - 255 o sbyte ▪ -128 - 127 o short ▪ -32,768 – 32,767 o ushort ▪ 0 – 65,535 o int ▪ Whole numbers between -2,147,483,648 - 2,147,483,648 o long ▪ -9,223,372,036,854,775,808 - 9,223,372,036,854,775,807 o ulong ▪ 0 – 18,446,744,073,709,551,615 • Floating Point Types (Decimals) o float ▪ Store up to 7 decimal digits o double ▪ Store up to 15 decimal digits

	o decimal ▪ Store up 28-29 digits • Boolean Type o Bool ▪ Holds true or false value • Data and Time Type o DateTime ▪ Represents any time or date • Character and Reference Types o char ▪ stores single character (letter), surrounded with single quotes o string ▪ store sequence of characters, surrounded with double quotes o object ▪ Holds any data type o dynamic ▪ bypasses compile-time type checking • Other o strut ▪ Contains other value types or reference types o enum ▪ consist set of named constants called the enumerator list

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**
	o example:
	
	int score = 83;
	
	if (score >= 60)
	{
	    MessageBox.Show(“You passed the math test!);
	}
	
	• **if-else statement**
	o example:
	
	int score = 58;
	
	if (score >=60)
	{
	        MessageBox.Show(“You passed the math test!”);

```
} else {
        MessageBox.Show(“You did not pass the math test.”);
}
```

- **if-else if-else statement**
  - example:

```
if (rdoLarge.Checked)
        MessageBox.Show(“You chose large for your shirt size.”);
Else if (rdoMedium.Checked)
        MessageBox.Show(“You chose medium for your shirt size.”);
Else if (rdoSmall.Checked)
        MessageBox.Show(“You chose small for your shirt size.”);
```

- **switch statement**
  - example:

```
String shirtSize = “large”;

Switch (shirtSize)
{
        case “small”:
                Console.WriteLine(“A small shirt was selected”);
                break;
        case “medium”:
                Console.WriteLine(“A medium shirt was selected”);
                break;
        case “large”:
                Console.WriteLine(“A large shirt was selected”);
                break;
        default:
                Console.WriteLine(“A shirt size has not been selected”);
                break;
}
```

- **ternary operator (?)**
  - example:

```
int age = 18;

string result = (age >= 18) ? “Adult” : “Minor”;
```

```
Console.WriteLine(result);
```

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
  - Example:

```
for (int I = 0; i < 5; i++);
{
        Console.WriteLine(“Iteration: ” + i);
}
```

- While loop
  - Example:

```
int count = 1;

while (count < 3)
{
        Console.WriteLine(“Count: ” + count);
        count++;
}
```

- Do While loop
  - Example:

```
int num = 0;

do
{
        Console.WriteLine(“Count: ” + num);
} while (num <10);
```

- For Each loop
  - Example:

```
string[] candy = {“Twix”, Snickers”, “Skittles”, “Starburst”);
foreach (String i in candy)
{
        Console.WriteLine(i);
}
```

- Nested loop

	○ Example:
	
	//Outer loop
	for (int i = ; i <= 2; i ++)
	{
	    Console.WriteLine(“Outer loop: ” + i); //executes 2 times
	}
	    //Inner loop
	    for (int j = 1; j<= 3; j++)
	    {
	        Console.WriteLine(“Inner: ” + j); //executes 6 times (2 * 3)
	    }

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.*
	• Integer Array
	○ Example:
	
	int[] myNumbers = {12, 13, 14, 15, 16};
	Console.WriteLine(myNumbers.Leght); //get length of array
	
	• String Array
	○ Example:
	
	string[] candy = {“Twix”, “Snickers”, “Skittles”, “Starburst”);
	candy[0] = “Gummy bears”;
	Console.WriteLine(candy[0]); //change twix to gummy bear and output that
	
	• Array using “new” Keyword (Use when you declare an array and initialize it later)
	○ Example:
	
	//Declare the array
	string[] candy;
	
	//Add values to array – use new
	candy = new string[] {“Twix”, “Snickers”, “Skittles”, “Starburst”};

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).*	• What the complexity represents: ○ O(1) – Constant time ▪ Operation will take the same amount of time regardless of the size of the data structure. ○ O(n) – Linear time

        - The time grows at the same rate as the size of the data structure.
    - O(log n) – Logarithmic time
        - The time it takes is longer as the size of the data structure grows.
- Array
    - Access – O(1)
    - Search, Insertion, and Deletion – O(n)
- Linked List
    - Access and Search – O(n)
    - Insertion and Deletion – O(1)
- Dictionary
    - Access, Search, Insertion, and Deletion – O(1)
- Queue
    - Enqueue and Dequeue – O(1)
    - Access – O(n)
- Stack
    - Push and Pop – O(1)
    - Access – O(n)
- Sorted List
    - Access and Search – O(log n)
    - Insertion and Deletion – O(n)
- Sorted Dictionary
    - Access, Search, Insertion, and Deletion – O(log 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 Car
{
        public string color {get; set;}

        //Default constructor

        public Car()
        {
                Color = “silver”; //set default color
        }

        static void Main (string[] args)
        {
                Car myObj = new Car(); //Create an instance of the car class
                Console.WriteLine(myObj.color); //print color of car
        } //end main
```

	} //end class
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# runtime environment is the common language runtime (CLR), specially .Net • Other languages that use .Net for runtime environment are F# and Visual basic.
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 helps to build web applications and APIs. It is used for developing dynamic websites and RESTful services. It enables a clean separation of concerns through the MVC architecture. • AutoMapper is a library that simplifies object-to-object mapping in .NET application. It automates the conversion of data between domain models and DTO's. It reduces the boilerplate code and enhancing maintainability. • LINQ is a querying syntax used for working with collections and data sources. It allows you to perform queries on in-memory collections which creates more organized code.
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****. E.g. Company X uses C# for its line of business applications.***	• C# is used across numerous industries because of its versatility and support for software development practice. • Company Examples that Use C# ○ Microsoft uses C# for web and game development. C# helps Microsoft improve their own application's productivity. ○ Stack Overflow uses C# for app development and web services. C# helps provide a community-driven software development with tools like their popular Q&A platform for developers. ○ Intuit uses C# for financial management tools. They make products like TurboTax and QuickBooks. C# helps manage accounting process and integrate with third-party services.

Resource Websites:

https://www.codecademy.com/resources/docs/c-sharp/loops

https://csharp-book.softuni.org/Content/Chapter-1-first-steps-in-programming/how-to-write-console-app/runtime-environments.html#

https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/builtin-types/built-in-types

https://levelup.gitconnected.com/unlocking-coding-interview-success-mastering-big-o-notation-net-c-73b4ef1554c5

https://levelup.gitconnected.com/10-essential-c-libraries-and-frameworks-for-developers-c325a4bae917

https://www.testgorilla.com/blog/what-is-c-sharp-used-for/

https://www.w3schools.com/cs/cs_data_types.php