**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, meaning that a data type must be declared, and attempts to pass in the wrong parameter will result in an error.    string pet = "Mina";  int x = 2;  List<double> myList = new List<double>(); |
| **Data Types**  *List all of the data types (and ranges) supported by this language.* | sbyte: -128 to 127  byte: 0 to 255  short: -32,768 to 32,767  ushort: 0 to 65,535  int: -2,147,483,648 to 2,147,483,647  uint: 0 to 4,294,967,295  long: 9,223,372,036,854,775,808 to 9,223,372,036,854,775,807  ulong: 0 to 18,446,744,073,709,551,615  float: -3.402823e38 to 3.402823e38  double: -1.79769313486232e308 to 1.79769313486232e308  decimal: -7.9e28 to 7.9e28  char: Unicode character  string: string of Unicode characters  bool: True or False  object: an object |
| **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 (thisInteger > 23) {  Console.WriteLine("This integer is greater than 23.");  }  If-else statement  if (thisInteger > 23) {  Console.WriteLine("This integer is greater than 23.");  }  else {  Console.WriteLine(“This integer is not greater than 23.”);  }  If-else-if statement  if (thisInteger > 23) {  Console.WriteLine("This integer is greater than 23.");  }  else if (thisInteger < 0) {  Console.WriteLine(“This integer is negative.”);  }  else {  Console.WriteLine(“This positive integer is not greater than 23.”);  }  Nested if statements  if (thisInteger > 23) {  if (otherInteger > 23) {  Console.WriteLine("Both integers are greater than 23.");  }  Switch statements  switch (thisInteger) {  case 1:  Console.WriteLine("This integer is equal to 1.");  break;  case 2:  Console.WriteLine("This integer is equal to 2.");  break;  default:  Console.WriteLine("This integer is not equal to 1 or 2.");  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 < 5; i++) {  Console.WriteLine("Hello, World!");  }  While loop  while (i < 10) {  Console.WriteLine("This is my output.");  i++;  }  Do-while loop  do {  Console.WriteLine("This is my output.");  i++;  } while (i < 5);  Foreach loop  int[] theseIntegers = {2, 5, 9};  foreach (int thisInteger in theseIntegers) {  Console.WriteLine(thisInteger);  } |
| **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.* | int[] intArray = new int[3] {1, 2, 3};  int[] intArray = new int[10];  string[] strArray = new string[] {"string1", "string2"};  string[] names= {"George", "Elton John", "Queen"}; |
| **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).* | Array  - Access: O(1)  - Search: O(n)  - Insertion: O(n)  - Deletion: O(n)  List  - Access: O(1)  - Search: O(n)  - Insertion: O(n)  - Deletion: O(n)  LinkedList  - Access: O(n)  - Search: O(n)  - Insertion: O(1)  - Deletion: O(1)  Queue  - Enqueue: O(1)  - Dequeue: O(1)  - Peek: O(1)  Stack  - Push: O(1)  - Pop: O(1)  - Peek: O(1)  HashSet  - Search: O(1)  - Insertion: O(1)  - Deletion: O(1)  Dictionary  - Access: O(1)  - Search: O(1)  - Insertion: O(1)  - Deletion: O(1) |
| **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.* | Creation:  public class AnObject {  public string name;  public SimpleObject() {  name = "default";  }  }  Instantiation:  AnObject thisObj = new AnObject(); |
| **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# compiles to Common Language Runtime (CLR), which is part of the .NET Framework.  Other languages that compile to CLR: Visual Basic, F#, C++, IronPython, Eiffel, Component Pascal, and more. |
| **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.* | **Entity Framework Core (EF Core)**: lets developers work with databases using C# objects instead of SQL queries & support database providers.  **ASP.NET Core**: web framework that provides features for building web apps & APIs.  **AutoMapper**: mapping library that simplifies mapping objects between types & eliminates repetitive mapping code/errors. |
| **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.*** | Microsoft (created C#): uses C# to develop many applications and services.  Stack Overflow: uses for web services and app development.  ServiceTitan: uses for Android app development and web services. |