**1.** Install and make yourself familiar with Microsoft Visual Studio and Microsoft Developer Network (MSDN) Library Documentation.

- Done

**2.** Find the description of the System.Console class in the standard .NET API documentation (MSDN Library).

- Represents the standard input, output, and error streams for console applications. This class cannot be inherited.

**3.** Find the description of the System.Console.WriteLine() method and its different possible parameters in the MSDN Library.

**-** Concsole.WriteLine Method - Writes the specified data, followed by the current line terminator, to the standard output stream.

|  |  |
| --- | --- |
| [WriteLine(String, Object, Object)](https://docs.microsoft.com/en-us/dotnet/api/system.console.writeline?view=net-5.0#System_Console_WriteLine_System_String_System_Object_System_Object_) | Writes the text representation of the specified objects, followed by the current line terminator, to the standard output stream using the specified format information. |
| [WriteLine(String)](https://docs.microsoft.com/en-us/dotnet/api/system.console.writeline?view=net-5.0#System_Console_WriteLine_System_String_) | Writes the specified string value, followed by the current line terminator, to the standard output stream. |
| [WriteLine(Char[], Int32, Int32)](https://docs.microsoft.com/en-us/dotnet/api/system.console.writeline?view=net-5.0#System_Console_WriteLine_System_Char___System_Int32_System_Int32_) | Writes the specified subarray of Unicode characters, followed by the current line terminator, to the standard output stream. |
| [WriteLine(String, Object[])](https://docs.microsoft.com/en-us/dotnet/api/system.console.writeline?view=net-5.0#System_Console_WriteLine_System_String_System_Object___) | Writes the text representation of the specified array of objects, followed by the current line terminator, to the standard output stream using the specified format information. |
| [WriteLine(String, Object)](https://docs.microsoft.com/en-us/dotnet/api/system.console.writeline?view=net-5.0#System_Console_WriteLine_System_String_System_Object_) | Writes the text representation of the specified object, followed by the current line terminator, to the standard output stream using the specified format information. |
| [WriteLine(UInt64)](https://docs.microsoft.com/en-us/dotnet/api/system.console.writeline?view=net-5.0#System_Console_WriteLine_System_UInt64_) | Writes the text representation of the specified 64-bit unsigned integer value, followed by the current line terminator, to the standard output stream. |
| [WriteLine(UInt32)](https://docs.microsoft.com/en-us/dotnet/api/system.console.writeline?view=net-5.0#System_Console_WriteLine_System_UInt32_) | Writes the text representation of the specified 32-bit unsigned integer value, followed by the current line terminator, to the standard output stream. |
| [WriteLine(Single)](https://docs.microsoft.com/en-us/dotnet/api/system.console.writeline?view=net-5.0#System_Console_WriteLine_System_Single_) | Writes the text representation of the specified single-precision floating-point value, followed by the current line terminator, to the standard output stream. |
| [WriteLine(Decimal)](https://docs.microsoft.com/en-us/dotnet/api/system.console.writeline?view=net-5.0#System_Console_WriteLine_System_Decimal_) | Writes the text representation of the specified [Decimal](https://docs.microsoft.com/en-us/dotnet/api/system.decimal?view=net-5.0) value, followed by the current line terminator, to the standard output stream. |
| [WriteLine(Int64)](https://docs.microsoft.com/en-us/dotnet/api/system.console.writeline?view=net-5.0#System_Console_WriteLine_System_Int64_) | Writes the text representation of the specified 64-bit signed integer value, followed by the current line terminator, to the standard output stream. |
| [WriteLine(Int32)](https://docs.microsoft.com/en-us/dotnet/api/system.console.writeline?view=net-5.0#System_Console_WriteLine_System_Int32_) | Writes the text representation of the specified 32-bit signed integer value, followed by the current line terminator, to the standard output stream. |
| [WriteLine(Double)](https://docs.microsoft.com/en-us/dotnet/api/system.console.writeline?view=net-5.0#System_Console_WriteLine_System_Double_) | Writes the text representation of the specified double-precision floating-point value, followed by the current line terminator, to the standard output stream. |
| [WriteLine(Char[])](https://docs.microsoft.com/en-us/dotnet/api/system.console.writeline?view=net-5.0#System_Console_WriteLine_System_Char___) | Writes the specified array of Unicode characters, followed by the current line terminator, to the standard output stream. |
| [WriteLine(Char)](https://docs.microsoft.com/en-us/dotnet/api/system.console.writeline?view=net-5.0#System_Console_WriteLine_System_Char_) | Writes the specified Unicode character, followed by the current line terminator, value to the standard output stream. |
| [WriteLine(Boolean)](https://docs.microsoft.com/en-us/dotnet/api/system.console.writeline?view=net-5.0#System_Console_WriteLine_System_Boolean_) | Writes the text representation of the specified Boolean value, followed by the current line terminator, to the standard output stream. |
| [WriteLine()](https://docs.microsoft.com/en-us/dotnet/api/system.console.writeline?view=net-5.0#System_Console_WriteLine) | Writes the current line terminator to the standard output stream. |
| [WriteLine(String, Object, Object, Object)](https://docs.microsoft.com/en-us/dotnet/api/system.console.writeline?view=net-5.0#System_Console_WriteLine_System_String_System_Object_System_Object_System_Object_) | Writes the text representation of the specified objects, followed by the current line terminator, to the standard output stream using the specified format information. |
| [WriteLine(Object)](https://docs.microsoft.com/en-us/dotnet/api/system.console.writeline?view=net-5.0#System_Console_WriteLine_System_Object_) | Writes the text representation of the specified object, followed by the current line terminator, to the standard output stream. |

4. Compile and execute the sample program from this chapter using the command prompt (the console) and Visual Studio.

- using System;

namespace detyra4

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Programming with C#");

}

}

}

5. Modify the sample program to print a different greeting, for example "Good Day!".

-using System;

namespace detyra5

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Good Day!");

}

}

}

6.  Write a console application that **prints your first and last name** on the console.

using System;

namespace detyra6

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Elita");

Console.WriteLine("Rexha");

}

}

}

7.  Write a program that **prints the following numbers** on the console 1, 101, 1001, each on a new line.

using System;

namespace detyra7

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("1");

Console.WriteLine("101");

Console.WriteLine("1001");

}

}

}

8.  Write a program that prints on the console the **current date and time**.

using System;

namespace Detrya

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine(" The current date and time is " + DateTime.Now);

}

}

}

9.  Write a program that prints the **square root of 12345**.

using System;

namespace detyra9

{

class Program

{

static void Main(string[] args)

{

System.Console.WriteLine(Math.Sqrt(12345));

}

}

}

10. Write a program that prints the first 100 members of the **sequence** 2, -3, 4, -5, 6, -7, 8.

using System;

namespace detyra10

{

class Program

{

static void Main(string[] args)

{

for (int i = 2; i < 101; i++)

{

if (i % 2 == 0)

{

Console.WriteLine(i);

}

else

{

Console.WriteLine(-i);

}

}

}

}

}

11. Write a program that reads your age from the console and prints your **age after 10 years**.

using System;

namespace Detyra11

{

class Program

{

static void Main(string[] args)

{

int age = 19;

Console.WriteLine("My age after 10 years is " + (age + 10));

}

}

}

12.   Describe the difference between **C#** and the **.NET Framework**.

- In simple terms, **C# is** a programming language, whereas . **NET is** the **framework** on which the language **is** built. ... **NET** supports many programming languages, and defines the rules and associated libraries those languages will use.

13.   Make a list of the **most popular programming** languages. How are they different from C#?

- Python, JavaScript, Java, C, C++, PHP….

As a much more modern programming **language**, **C#** was designed to work with the current Microsoft . NET framework in both client and web-based applications. While C++ is an object-oriented **language**, **C#** is considered a component-oriented programming **language**. ... **C#** does not use pointers, while C

++ can use pointers anywhere