# C# Notes

Table of Contents

[C# Notes 1](#_Toc115984418)

[Section 1: Your First C# Program and Overview of Visual Studio 1](#_Toc115984419)

[5. Difference between the .NET5 and .NET6 Console Template 1](#_Toc115984420)

[6. Hello World – First Program 2](#_Toc115984421)

[8. Visual Studio Interface 3](#_Toc115984422)

[Section 2: DataTypes And Variables 4](#_Toc115984423)

[14. High Level Overview of Variables and Datatypes 4](#_Toc115984424)

[15. More Datatypes and Their Limits 4](#_Toc115984425)

## Section 1: Your First C# Program and Overview of Visual Studio

### 5. Difference between the .NET5 and .NET6 Console Template

In Visual Studio code we have different templates to select for a project. We will select the Console App that has the C# logo at the top.

We will call this project "Net5".

|  |
| --- |
| Net5/Program.cs |
| using System;  namespace Net5  {  internal class Program  {  static void Main(string[] args)  {  Console.WriteLine("Hello World!");  }  }  } |

What is the .NET platform?

They wanted to unify the development of all different environments into one, so now they put everything together cloud development, desktop, web, gaming, mobile, IoT, and AI together under the dotnet hood, and there is no dotnet framework as well as dotnet core separated. It all comes together. So the .Net platform is basically just the entire framework or the package of a bunch of different pieces of code that is put together so that we can write code very easily and build applications very quickly without having to build everything from scratch.

We can now create a new project called "Net6."

|  |
| --- |
| Net6/Program.cs |
| // See https://aka.ms/new-console-template for more information  Console.WriteLine("Hello, World!"); |

The .Net 5 template and the .Net 6 template are basically the same. The difference is for .Net6 we are by default inside of the Main method block.

### 6. Hello World – First Program

In "bin/Debug/net5.0" there is a "\*.exe" file. It is closing right away. If you click on the "\*.exe" file, it will also open the console. Unfortunately, it's closing right away. If you want to make sure that the console stays open you need to add Console.Read();This will read the next character from the standard input stream.

If you run this code,

|  |
| --- |
| HelloWorld/Program.cs |
| using System;  namespace HelloWorld  {  internal class Program  {  // Entry point of our program  static void Main(string[] args)  {  Console.WriteLine("Hello World!");  Console.WriteLine("Hello Jeremy");  Console.Read();  }  }  } |

It will run as it did before but will stay open and let me press a key and then it will close so to speak even though in Visual Studio, by default, it leaves this window open for us for debugging.

Text

Description automatically generated with medium confidence

### 8. Visual Studio Interface

In Visual Studio, in the Solution Explorer, you will see all of the files that you have. Once you have multiple classes, it will be useful to jump to different files in the solution explorer.

HelloWorld is the namespace in the solution explorer.

Graphical user interface, text, application

Description automatically generated

When you click on the different fields in the solution explorer, Properties comes up and changes its appearance.

We have the Output at the bottom where we will see output about where our files are created.

A computer screen capture

Description automatically generated with medium confidence

You can change the interface if you want by dragging different boxes around like the solutions explorer, you could drag around the screen, changing its position.

If you go to Window and then select Reset Window Layout, the layout will be set to its default layout. If you like a certain layout that you have created, you can go to Window and click on "Save Window Layout." You can also go to Window and select Apply Window Layout to apply a specific window layout.

## Section 2: DataTypes And Variables

### 14. High Level Overview of Variables and Datatypes

A variable is a container that can take a value. A variable must have a type, which tells us which type of data it can store. A variable must have a value. We will need to distinguish between our variables by giving them a name.

int iAmANumber = 5;

int is a data type

iAmANumber is a variable name

5 is the value

Datatypes in C#:

int – a whole number like 1, 2, 3, …

int iAmANumber = 5;

float – used to store float numbers, meaning numbers that have decimal points

float pi = 3.1415;

bool – is used to store Boolean values which can either be true or false

bool isGPSEnabled = true;

string – used to store text; we need to use the double quotation marks to specify that these characters are actually text and not code

string myName = "Jeremy";

char – used to store a single character like an at sign or a dollar sign; we can use them to store special characters because if all we need is one character, then a string is going to be an overcoat since it's taking more memory than the character.

char at = '@' ;

The more variables we declare, the larger memory our app is going to require.

### 15. More Datatypes and Their Limits

#### Variable example with default value

Variables can be declared outside of a method as well as inside of a method. The Main method is the starting point of the program where everything begins.

public class Lecture {

int age = 15; // This is a variable of type integer

pubic static void Main(string[] args){

Console.WriteLine(age); // Output will be 15

}

}

#### Variable example with new assigned value

public class Lecture {

int age = 15; // This is a variable of type integer

pubic static void Main(string[] args){

age = 20; // New value gets assigned

Console.WriteLine(age); // Output will be 20

}

}

#### Variable example with no value

public class Lecture {

int age; // default value assigned = 0

public static void Main (string[] args) {

Console.WriteLine(age); // Output will be 0

}

}

#### Variable declared in Method

public class Lecture {

public static void Main (string[] args) {

// Creating the variable inside of the method

// The variable can only get used inside of this method

int age = 15;

Console.WriteLine(age); // Output will be 15

}

}

There is more than just the int data type.

#### Primitive Data Type: sbyte

sbyte x = 120;

sbyte stores whole numbers from -128 to 127; in total sbyte stores 256 values; sbyte stands for signed byte; sbyte does not need a lot of storage, which is great if you want to write very performant software.

#### Primitive Data Type: short

short x = 30000;

A short stores whole numbers from -32767 to 32767

#### Primitive Data Type: integer (data type written as int)

int x = 2000000000;

An integer stores whole numbers between -2,147,483,648 to 2,147,483,647.

#### Primitive Data Type: long

long x = 9000000000000000000;

A long consists of whole numbers from -9,223,327,036,854,775,808 to 9,223,372,036,854,775,807

#### When to use sbyte, short, integer or long?

Use the smallest data type your value fits into. So the smaller the data that you want to store, so the smaller the value, the smaller the data type should be that you're using. So, sbyte for verry small numbers, short for longer numbers, integers for quite long numbers, and long for super long numbers.

What about floating point values? Well, there are three data types you can use for that.

#### Primitive Data Type: float

float x = 99.99f;

float allows decimals and a range from 1.5 x 10 ^-45 to 3.4 x 10^38; floats have 7-digit precision. We have to add an f to our floating point numbers, which tells C# that we're talking about a float value. If we don't use the f after 99.99 in float x = 99.99, it will be considered a double value and we will receive an error.

#### Primitive Data Type: double

double x = 1.5;

Double allows decimals and an even higher range than the float; doubles have 15-digit precision, so if you need a higher precision and a higher range, you would use double. And as you can see, there is no F here and no D at the end.

double x = 1.5;

On the right hand side of the above statement we have a double and on the left hand side we declare a double.

If you added an f like this double x = 1.5f, then you would have the same problem as if you had left out f when creating a float.

#### Primitive Data Type: decimal

decimal x = 1.5;

The decimal data type allows decimals and an even higher range than the double; decimals have 28-digit precision.

#### When to use float, double or decimal?

The smaller the value is, the smaller the data type should be. It also has to do with performance.

float is mostly used in graphics libraries (high demands for processing powers)

double is mostly used for real world values (except money calculations)

decimal is mostly used in financial applications (high level of accuracy)

#### Primitive Data Type: boolean (data type when declared is bool)

bool switch = false;

The boolean data type only allows two states: true / false

#### Primitive Data Type: char

char singleLetter = 'A';

The char data type allows a single character literal or unicode. A char must have single quotations.

#### (not a Primitive) Data Type: string

string username = "Jason1995";

The string data type allows multiple letters and unicodes. A string has double quotes.