

Learn C# - Week 1 – Hello World

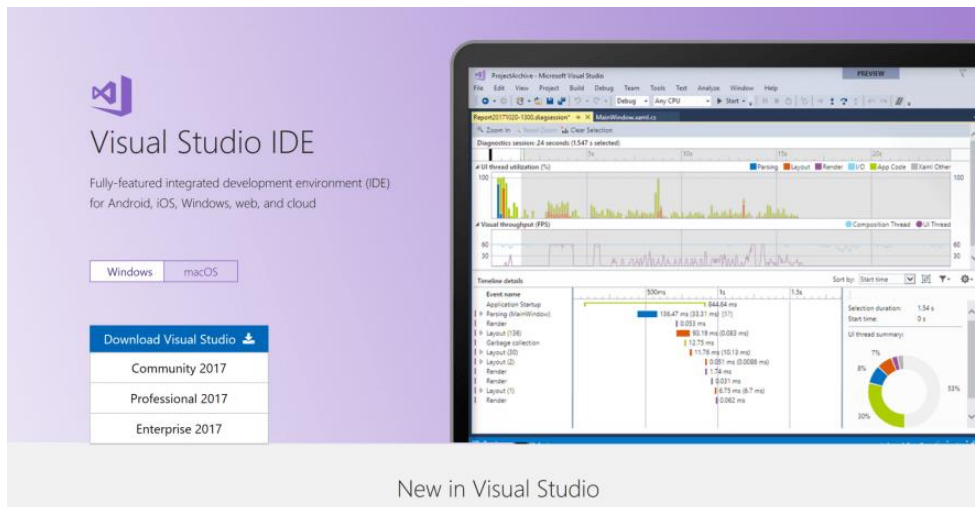
Ever want to learn programming without having the resources to take on formal training? Well we got you covered, TICGN will be giving several lessons that should introduce to you the basics of programming. Now these lessons aren't formal and I would still recommend looking into taking classes on programming or even getting an ebook on the material in order to get answers on some overlooked material. These lessons , while helpful, will not go too in detail on the specifics of programming and instead will focus on providing an easy to understand point of view in programming. By the end of these lessons you should be capable of creating text based games.

Now let me introduce the programming language C#. This programming language was chosen because as you can see, this is a gaming site and am sure most readers are gamers. With C# you could take some of this knowledge and transition to working with the game engine Unity. Another reason C# was selected is because we are going to be taking it easy with scope of programming. We don't want to dive deep into sections like the runtime stack or deallocating memory because these lessons aren't for those who are currently taking college courses on Design Patterns or anywhere near that level. We are not going to be focusing too much on math, I don't want to scare you guys away. This is for absolute beginners and as such I will invoke an old saying my college math professor would say, "Street Talk." Some of the words used when teaching these concepts are complex so I'll try to bring them down to street talk so that anyone can understand. So without a further a due, lets begin the journey that never ends.

Visual Studio

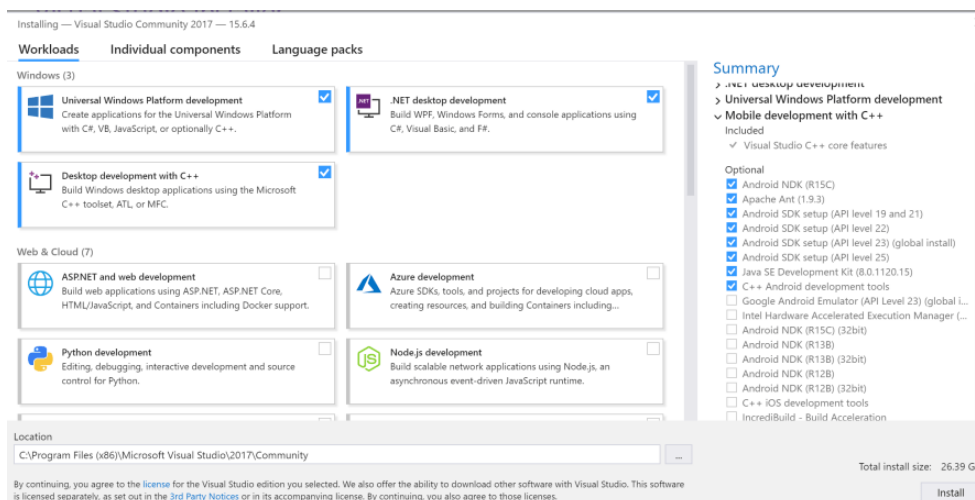
Alright so we decided on programming in C# but now how do we begin programming? Well first we need an **Integrated Development Environment aka IDE**. An IDE is just a collection of software that translates the programming languages people use to machine code. What's machine code? 1's and 0's. Yeah you have seen the Matrix right? All those 1's and 0's are basically instructions a computer takes in. So if you're wondering how all this works in the background just know that we will be writing in C# we will hit a button inside our IDE and that will turn it into 1's and 0's so the computer knows how to run the instructions we typed out. That's as far as we need to go on that subject so lets move on to downloading an IDE.

[Visual Studio Link](#)

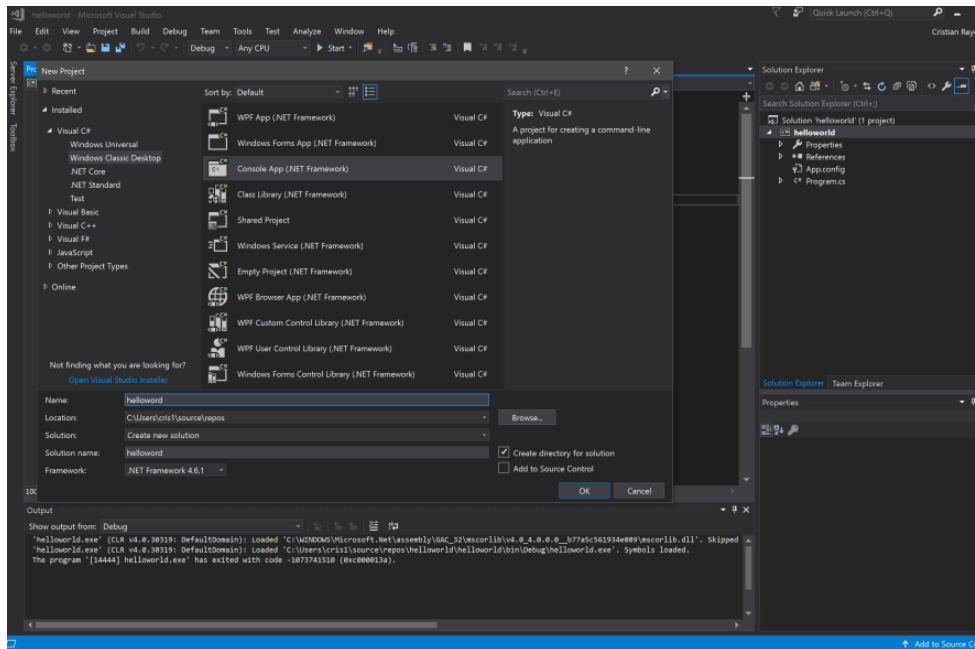


Above I provided a link to Microsoft's IDE Visual Studio. You're going to want to download the community version which is the free version. The free version allows you to build all your programs but not to sell them commercially, don't quote me on that, so basically if you ever want to sell your programs your going to need to upgrade to the professional versions. All that legal stuff aside Visual Studio is regarded as one of the best IDE's on the planet if you ever run into a bug when creating a program it's not Visual Studio that has that bug, it will be all your own fault which just means Visual Studio is a reliable place to create programs.

When you download it and install it just hit all of the defaults until it comes to the part where it tells you to select languages to include. If you want you can download them all which is the best option but will take a long time, if you want a shorter amount of time just make sure C# options are highlighted. **Make sure .NET desktop development is flagged to download.** If it ask you to download the 64 bit or 32 bit look at how many bits your PC handles. If you don't know how to find that on Windows 10 you can search on Cortana the settings->system->about. In that page you should find if your system is 64 or 32 bits.



Once you boot up Visual Studio and signed in with your Microsoft account you are ready to begin. Once Visual Studio is up and running lets begin the process of starting our first C# program. First hit **File** on the upper left corner of the screen then hit **new** then hit **project**. You'll get a pop up window showing which languages you want to use. Hit **Visual C#** and then hit Console APP(.Net Framework). Then name your project whatever you want and hit ok on the bottom. This process is no different than from creating a new file in Microsoft Word. The only difference is Visual Studio will create a folder in which all your files for your project will be kept.



My Visual Studio has a different colored theme so don't worry if it looks different.

After you have properly set up your project a default main file be automatically created for you. If you did everything correctly you should get some text already placed in the file. We won't get into the details of what every line does because as these lessons go on you'll start to see for yourself more clearly what each does. For now lets look at the bracket of code listed below.

```
static void Main(string[] args)

{

}
```

This is where your programs will start from. In between the curly braces {} is where your code starts and ends. In between the curly braces lets write the following.

```
Console.WriteLine("Hello World");
```

Something to keep in mind while coding is that punctuation counts. Whether it's lowercase or uppercase or even a missing period the smallest mistake will make an entire program that has millions of lines of code fail. So make sure you're writing is correct, a mistake of improper punctuation is called a **Syntax Error**. These are some of the most common bugs and luckily for us is usually caught by Visual Studio and highlighted by underlining your code with a redline. You can check it out by changing the word Console to consle. This is where the strength of Visual Studio comes in, it detects Syntax Errors before we build our project.

Moving on to actually building the program. Making sure you changed consle back to Console, it's time to debug. Hit the Green Start arrow in order to debug and compile the program. If you didn't write the Hello World code snippet above exactly as is then you'll get errors showing up in the output console on the bottom of Visual Studio. If you got errors and a pop up showed up saying something along the lines of your programming failing to compile then exit out of the error menus and rewrite the code. Once you fix it hit start it should properly Compile your program by debugging which makes it so Visual Studio see's if there are any issues with the **Syntax**, any issues with the logic of your code, and then turn C# or the words we typed into 1's and 0's. There are couple of steps that happen in between when you compile a program but that's not very important at our level.

Off topic: I know some of you reading our probably upset that a lot of topics are being skimmed over but to be completely honest with you this is something that is usually done in Computer Science. There is a ton of stuff going on under the hood that it would take years to explain how even the most simple of things are done. In fact there are classes dedicated just for how this one line of code we just wrote works, it's usually called Machine Language and it's an eye opener but not a lot of professional developers work with it everyday. You may be frustrated now that some things are kept secret from you but as we progress you'll start to understand that somethings should be hidden for the consumers either because they could break a program or because they just don't need to know. Just like how you don't need to know how a car works, you just know it works.

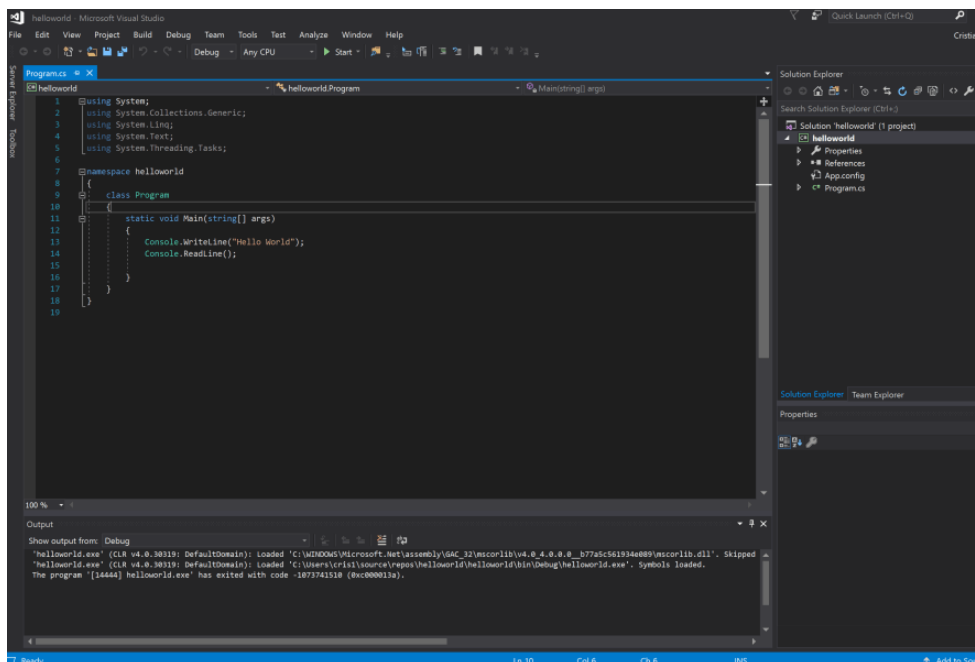
Alright going back to our program you're probably thinking "Yeah! I just build my first program. But wait... nothing happened." Here is another lesson for you programmers. A computer is the smartest stupidest thing you'll ever know. If you haven't guess it by now the code we wrote should print out the line "Hello World" on to the screen. The thing is nothing happened right? Well it turns out the computer did exactly what you told it to do. It printed out Hello World and immediately finished the program. Computers handles tasks a million times faster than we do so it just pupped it on screen for a half a sec and terminated the program. In this regard you have to take into account that you have to write every single step of a program. If you want to write a program that makes a sandwich you can't just say computer make me a sandwich. You're going to need to write code for every single step. Take out the bread, take out the cheese, the mayonnaise, the ham, and bam? No you still have to tell the computer where that content is, what order to put them in, how do you even

stack them together. In our case we have to tell the computer to print out the word Hello World and then wait and hold it on screen.

Try adding this line of code after your Hello World line.

```
Console.ReadLine();
```

Try hit the green Start Button to Debug and Compile your program. This time the words “Hello World” should stay on the screen and it’s because `Console.ReadLine();` waits for a user to enter any key from the keyboard followed by the Enter key. So if you want to exit out of the console just hit Enter.



There you have it. Your first C# program and a special one at that. Almost every student of Computer Science has built a Hello World program as their starting point. Until next weeks lesson try experimenting on this program on your own. What happens if you change the text inside the parenthesis? What if you paste a copy of the Hello World line below it. What happens if you change `Console.WriteLine` to `Console.Write`?

What You Learned:

- How to Create C# Projects in Visual Studio. It’s just like creating a Word document.
- What a Syntax Error is. It’s an error when you code is not properly typed. Sometimes Visual Studio will tell you where your error is and on which line sometimes it won’t.
- How to Compile a program. Using the Green Start Button Visual Studio first checks for bugs either by Syntax Errors, or logic errors and then Compiles the program

which is a fancy way of saying translating C# into language your computer understands which are 1's and 0's.

Until next week leave any comments or questions bellow in the comments or anything interesting that you do with your program below.