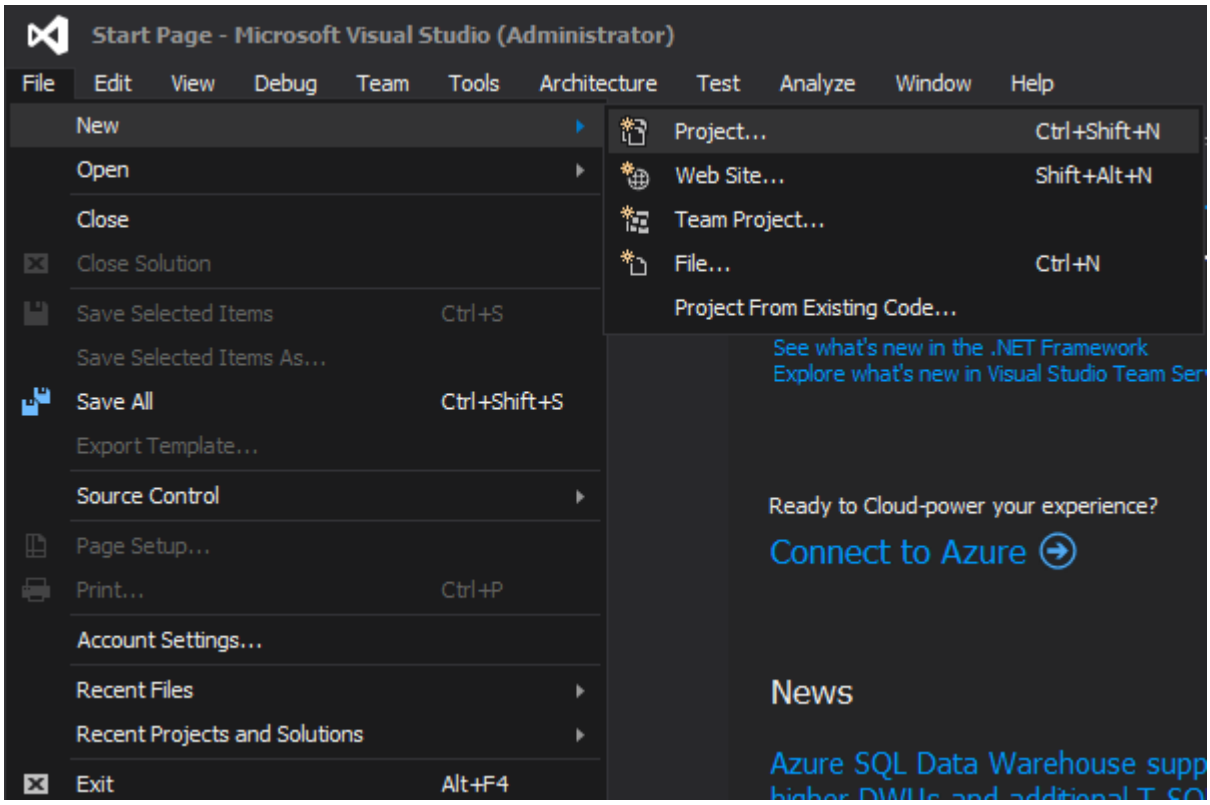
Creating a new project in Visual Studio (console application) and Running it in Debug mode

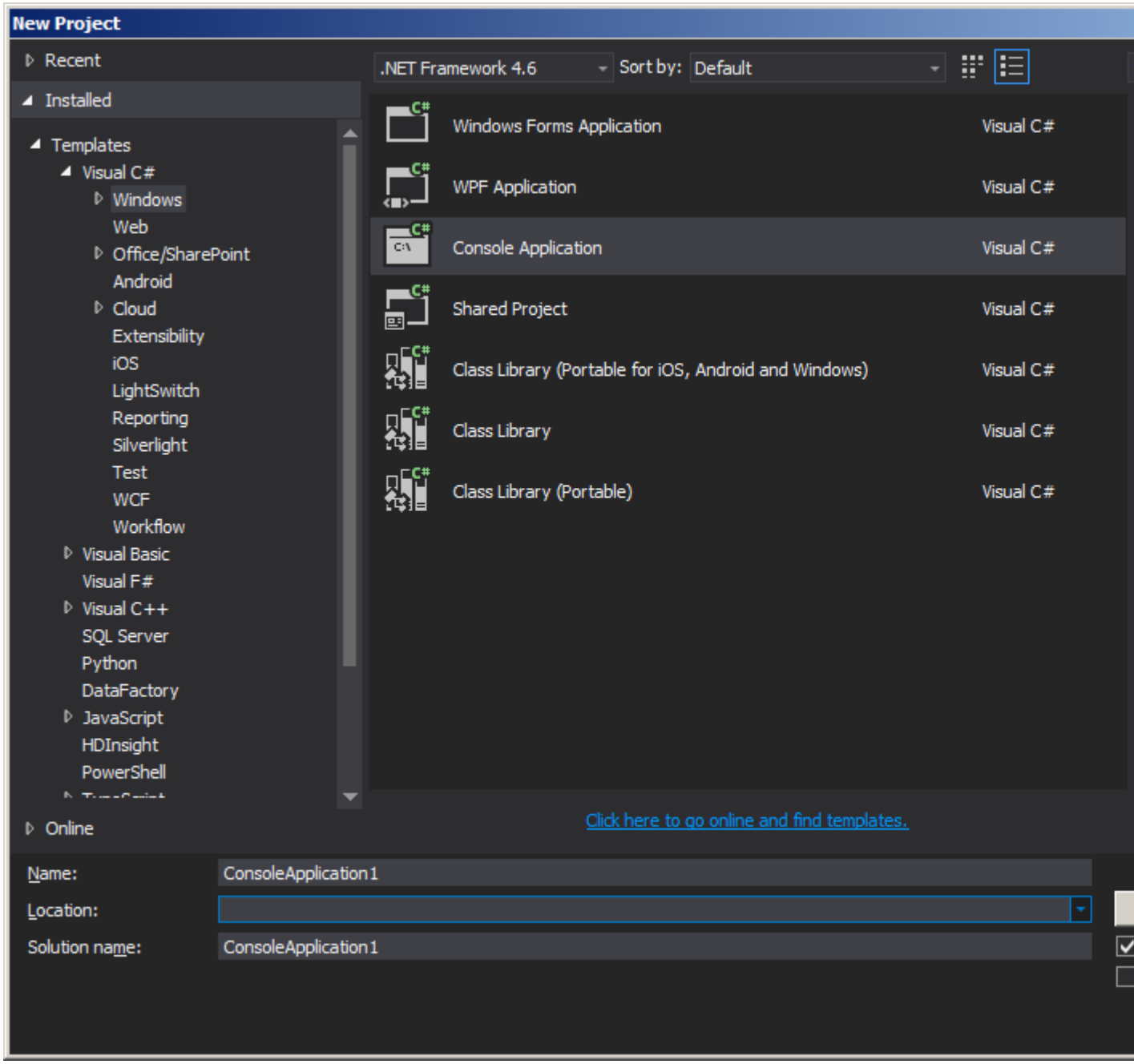
1. **Download and install Visual Studio**. Visual Studio can be downloaded from VisualStudio.com. The Community edition is suggested, first because it is free, and second because it involves all the general features and can be extended further.

2. **Open Visual Studio.**

3. **Welcome.** Go to **File → New → Project**.

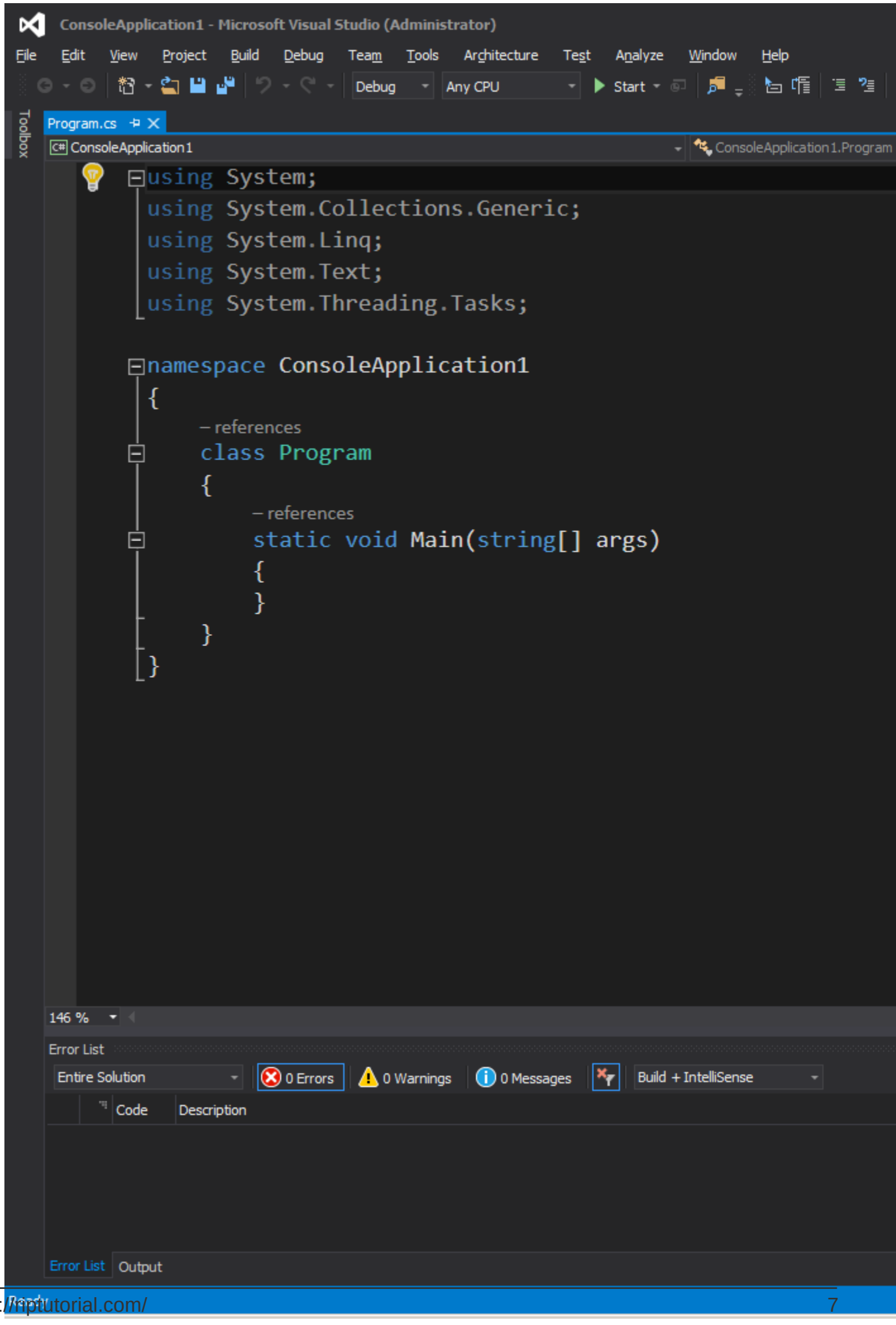


4. Click **Templates** → **Visual C#** → **Console Application**



5. **After selecting Console Application,** Enter a name for your project, and a location to save and press `OK`. Don't worry about the Solution name.

6. **Project created**. The newly created project will look similar to:


ConsoleApplication1 - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Tools Architecture Test Analyze Window Help
Debug
Any CPU
Start
Toolbox
Program.cs
ConsoleApplication1
ConsoleApplication1.Program
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApplication1
{
0 references
class Program
{
0 references
static void Main(string[] args)
{
}
}
}
146 %
Error List
Entire Solution
0 Errors
0 Warnings
0 Messages
Build + IntelliSense
Code
Description
Error List
Output
Ready

(Always use descriptive names for projects so that they can easily be distinguished from other projects. It is recommended not to use spaces in project or class name.)

7. **Write code.** You can now update your `Program.cs` to present "Hello world!" to the user.

```
using System;

namespace ConsoleApplication1
{
    public class Program
    {
        public static void Main(string[] args)
        {
        }
    }
}
```

Add the following two lines to the `public static void Main(string[] args)` object in `Program.cs`: (make sure it's inside the braces)

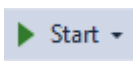
: (make sure it's inside the braces)

```
Console.WriteLine("Hello world!");
Console.Read();
```

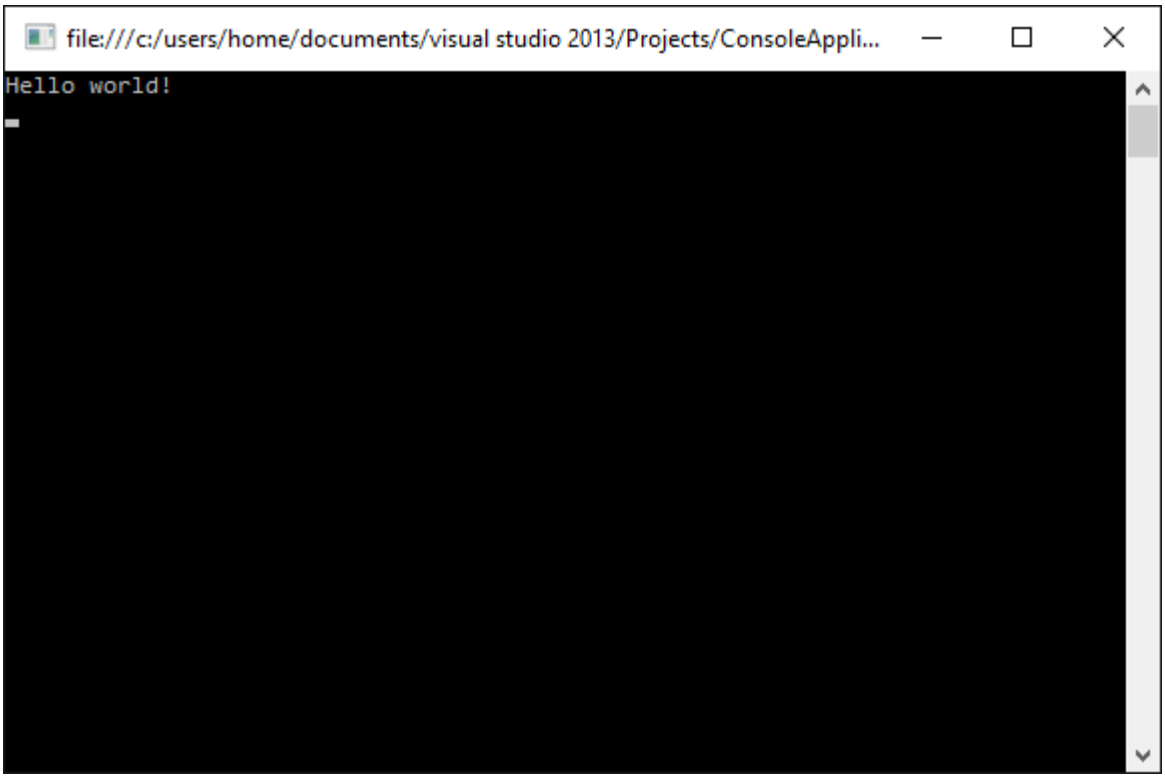
Why `Console.Read()`? The first line prints out the text "Hello world!" to the console, and the second line waits for a single character to be entered; in effect, this causes the program to pause execution so that you're able to see the output while debugging. Without `Console.Read();`, when you start debugging the application it will just print "Hello world!" to the console and then immediately close. Your code window should now look like the following:

```
using System;

namespace ConsoleApplication1
{
    public class Program
    {
        public static void Main(string[] args)
        {
            Console.WriteLine("Hello world!");
            Console.Read();
        }
    }
}
```

8. **Debug your program.** Press the Start Button on the toolbar near the top of the window 

or press `F5` on your keyboard to run your application. If the button is not present, you can run the program from the top menu: **Debug → Start Debugging**. The program will compile and then open a console window. It should look similar to the following screenshot:



9. **Stop the program.** To close the program, just press any key on your keyboard. The `Console.Read()` we added was for this same purpose. Another way to close the program is by going to the menu where the `Start` button was, and clicking on the `Stop` button.

Creating a new program using Mono

First install Mono by going through the install instructions for the platform of your choice as described in their installation section.

Mono is available for Mac OS X, Windows and Linux.

After installation is done, create a text file, name it `HelloWorld.cs` and copy the following content into it:

```
public class Program
{
    public static void Main()
    {
        System.Console.WriteLine("Hello, world!");
        System.Console.WriteLine("Press any key to exit..");
        System.Console.Read();
    }
}
```

If you are using Windows, run the Mono Command Prompt which is included in the Mono installation and ensures that the necessary environment variables are set. If on Mac or Linux, open a new terminal.

To compile the newly created file, run the following command in the directory containing `HelloWorld.cs`: