0.7 — Compiling your first program

BY ALEX ON MAY 28TH, 2007 | LAST MODIFIED BY NASCARDRIVER ON DECEMBER 22ND, 2019

Before we can write our first program, we need to learn how to create new programs within our Integrated Development Environment (IDE). In this lesson, we'll cover how to do that, and you'll also compile and execute your first program!

Projects

To write a C++ program inside an IDE, we typically start by creating a new project (we'll show you how to do this in a bit). A **project** is a container that holds all of your source code files, images, data files, etc… that are needed to produce an executable (or library, website, etc…) that you can run or use. The project also saves various IDE, compiler, and linker settings, as well as remembering where you left off, so that when you reopen the project later, the state of the IDE can be restored to wherever you left off. When you choose to compile your program, all of the .cpp files in the project will get compiled and linked.

Each project corresponds to one program. When you're ready to create a second program, you'll either need to create a new project, or overwrite the code in an existing project (if you don't want to keep it). Project files are generally IDE specific, so a project created for one IDE will need to be recreated in a different IDE.

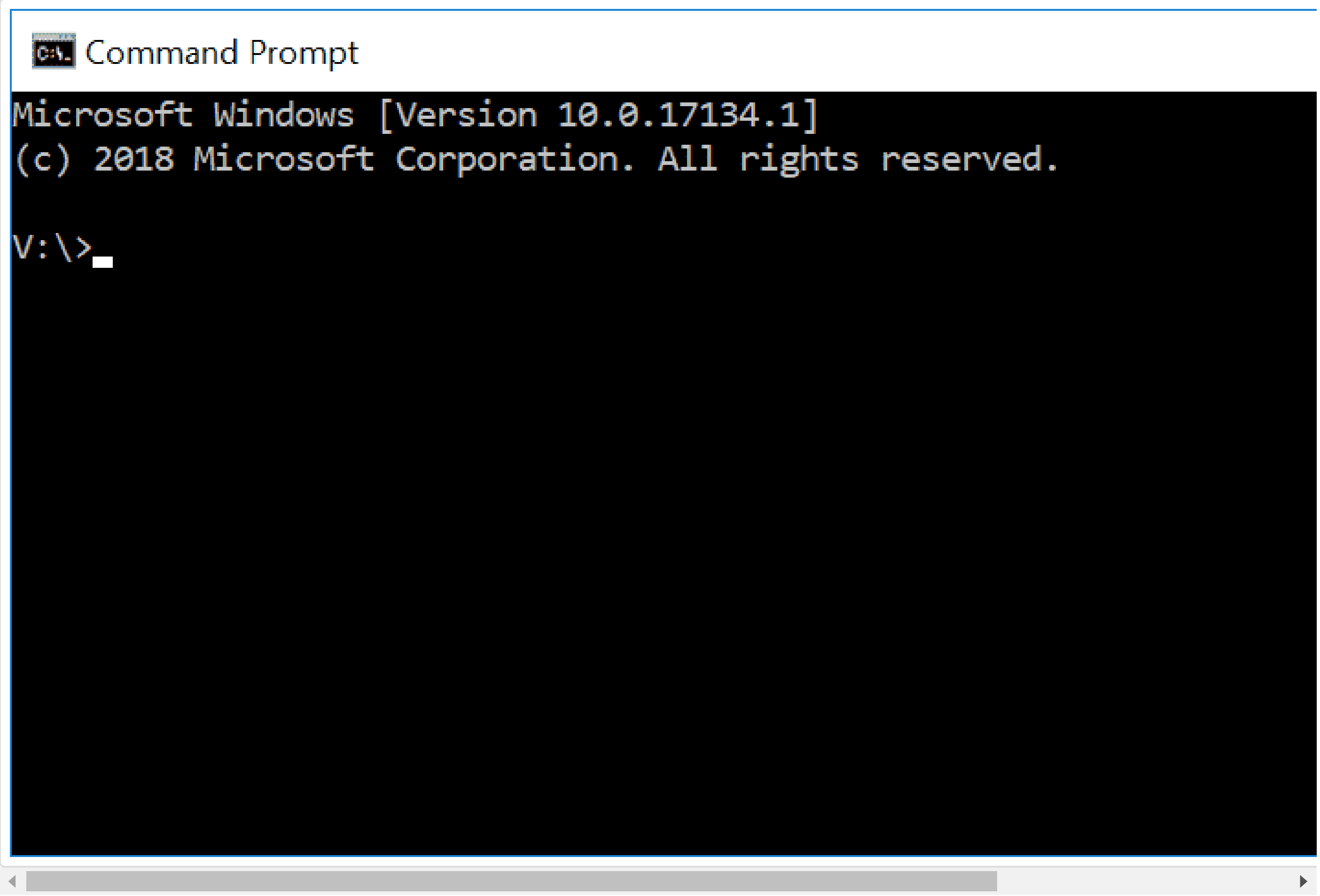
Best practice

Create a new project for each new program you write.

Console projects

When you create a new project, you'll generally be asked what type of project you want to create. All of the projects that we will create in this tutorial will be console projects. A **console project** means that we are going to create programs that can be run from the Windows, Linux, or Mac console.

Here's a screenshot of the Windows console:



```
Microsoft Windows [Version 10.0.17134.1]
(c) 2018 Microsoft Corporation. All rights reserved.

V:\>_
```

By default, console applications have no graphical user interface (GUI), they print text to the console, read input from the keyboard, and are compiled into stand-alone executable files. This is perfect for learning C++, because it keeps the complexity to a minimum, and ensures things work on a wide variety of systems.

Don't worry if you've never used a console before, or don't know how to access it. We'll compile and launch our programs through our IDEs (which will invoke the console when necessary).

Workspaces / solutions

When you create a new project for your program, many IDEs will automatically add your project to a "workspace" or a "solution" (the term varies by IDE). A workspace or solution is a container that can hold one or more related projects. For example, if you were writing a game and wanted to have a separate executable for single player and multiplayer, you'd need to create two projects. It wouldn't make sense for both of these projects to be completely independent -- after all, they are part of the same game. Mostly likely, each would be configured as a separate project within a single workspace/solution.

Although you can add multiple projects to a single solution, we generally recommend creating a new workspace or solution for each program, especially while learning. It's simpler and there's less chance of something going wrong.

Writing your first program

Traditionally, the first program programmers write in a new language is the infamous **hello world program**, and we aren't going to deprive you of that experience! You'll thank us later. Maybe.

Warning

Starting with this lesson, you will see many examples of C++ code presented. Most of these examples will look something like this:

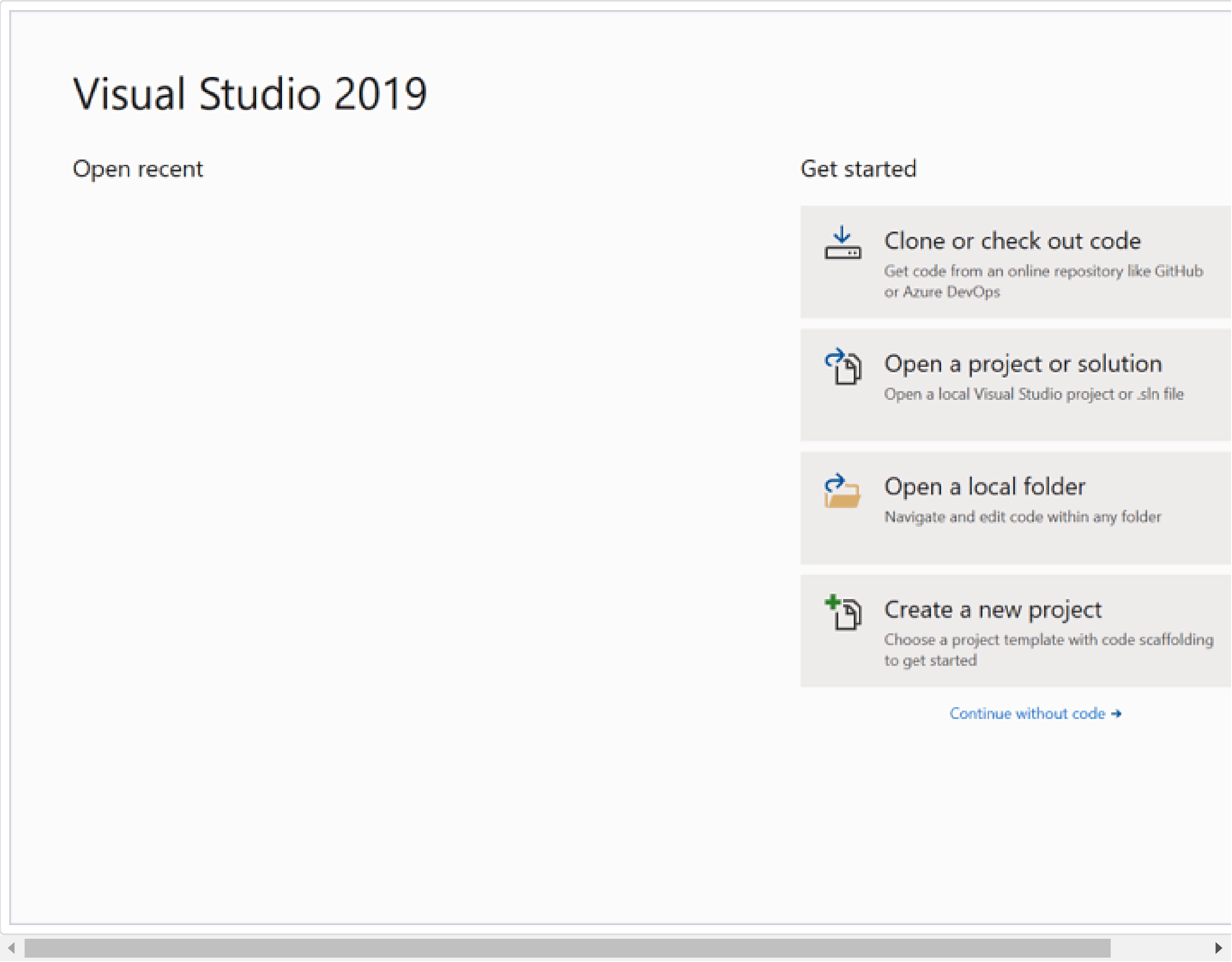
```
#include <iostream>

int main()
{
    std::cout << "Hello, world!";
    return 0;
}
```

If you select the code from these examples with your mouse and then copy/paste it into your IDE, you may also get the line numbers (depending on how you made the selection). If so, you'll need to remove the line numbers manually before you compile.

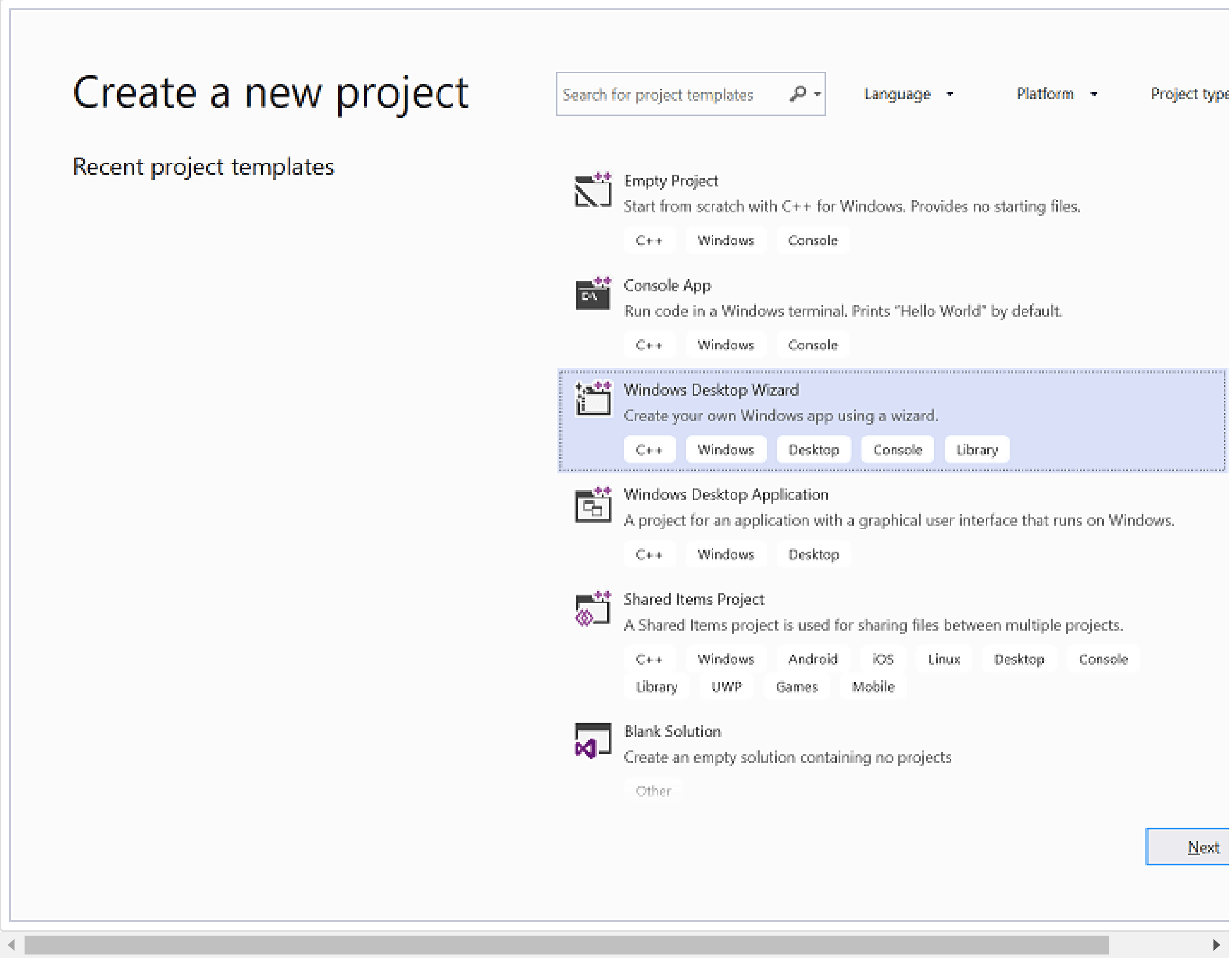
Creating a project in Visual Studio 2019

When you run Visual Studio 2019, you should see a dialog that looks like this:


Visual Studio 2019
Open recent
Get started
Clone or check out code
Get code from an online repository like GitHub or Azure DevOps
Open a project or solution
Open a local Visual Studio project or .sln file
Open a local folder
Navigate and edit code within any folder
Create a new project
Choose a project template with code scaffolding to get started
Continue without code →

Select *Create a new project*.

You'll then see a dialog that looks like this:



Create a new project

Recent project templates

Empty Project
Start from scratch with C++ for Windows. Provides no starting files.

Console App
Run code in a Windows terminal. Prints "Hello World" by default.

Windows Desktop Wizard
Create your own Windows app using a wizard.

Windows Desktop Application
A project for an application with a graphical user interface that runs on Windows.

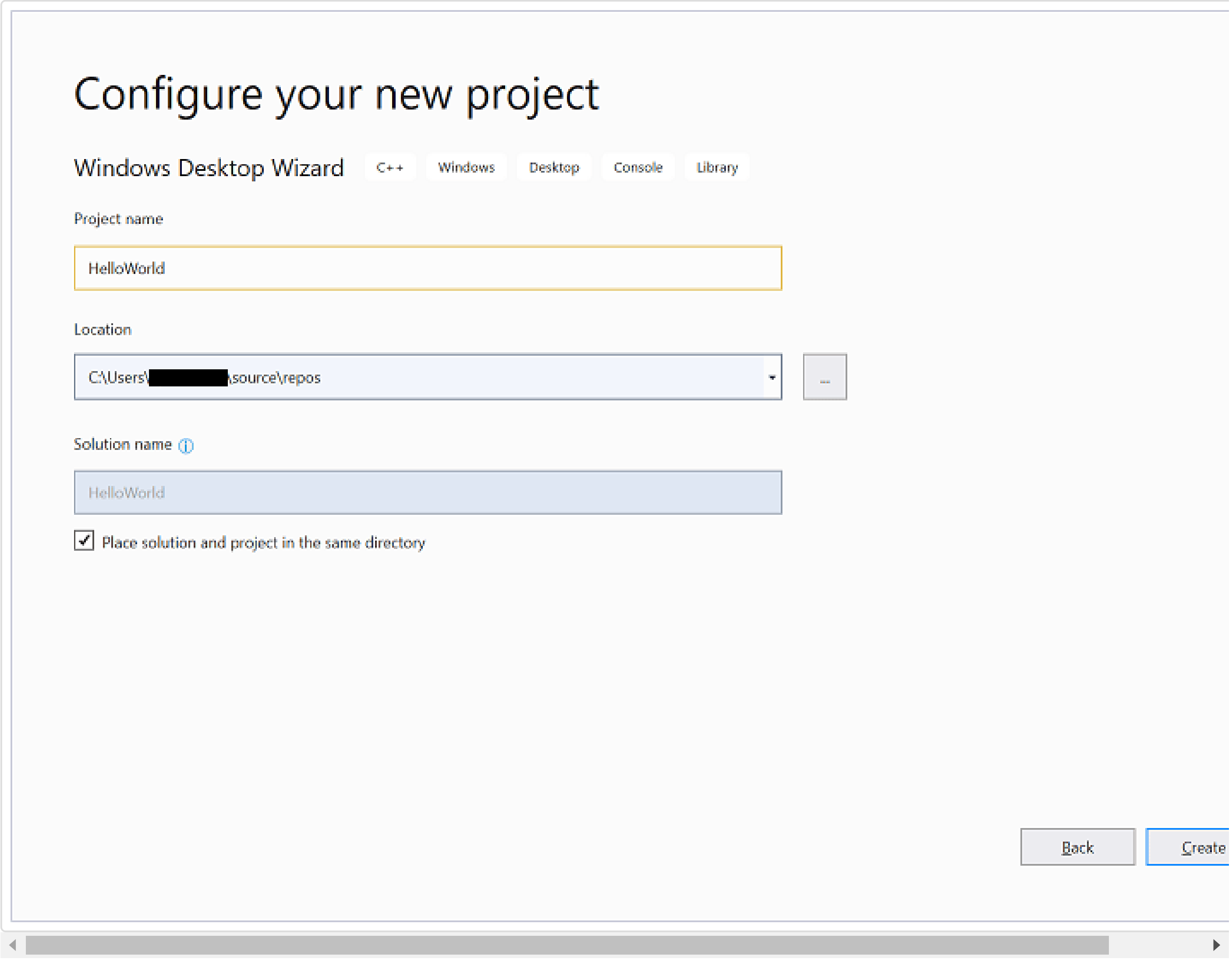
Shared Items Project
A Shared Items project is used for sharing files between multiple projects.

Blank Solution
Create an empty solution containing no projects

If you’ve already opened a prior project, you can access this dialog via the *File menu* > *New* > *Project*.

Select *Windows Desktop Wizard* and click *Next*. If you don’t see this, then you probably forgot to choose to install the *Desktop development with C++* workload when you installed Visual Studio. In that case, go back to lesson **0.6 -- Installing an Integrated Development Environment (IDE)** and reinstall your Visual Studio as indicated (note: rather than doing a full reinstall, you can run the Visual Studio installer and modify your existing installation to add the C++ workload).

Next, you’ll see a dialog that looks like this:



Configure your new project

Windows Desktop Wizard

Project name: HelloWorld

Location: C:\Users\[redacted]\source\repos

Solution name: HelloWorld

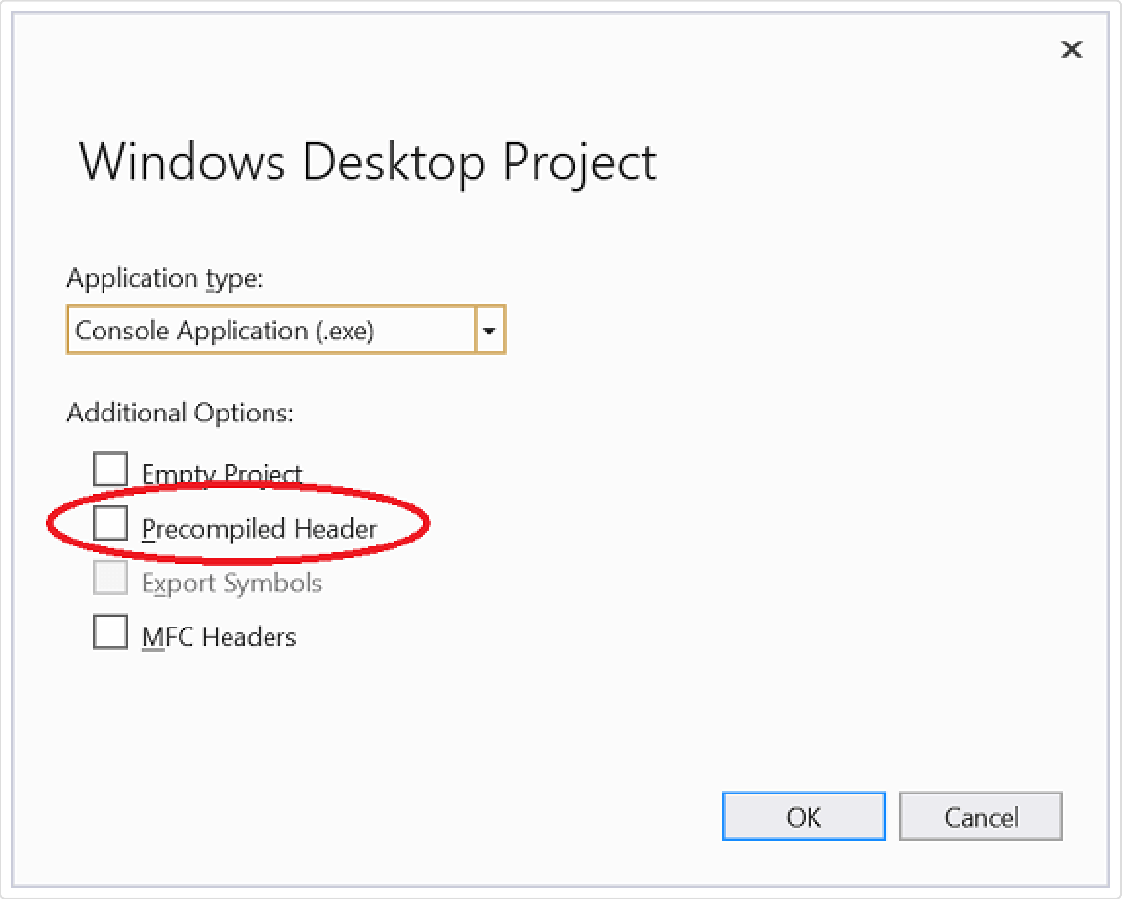
☑ Place solution and project in the same directory

Replace the existing project name with `HelloWorld`.

It’s recommended that you also check the *Place solution and project in the same directory*, as this reduces the number of subdirectories that get created with each project.

Click *Create* to continue.

Finally, you’ll see one last dialog:



Windows Desktop Project

Application type:

Console Application (.exe)

Additional Options:

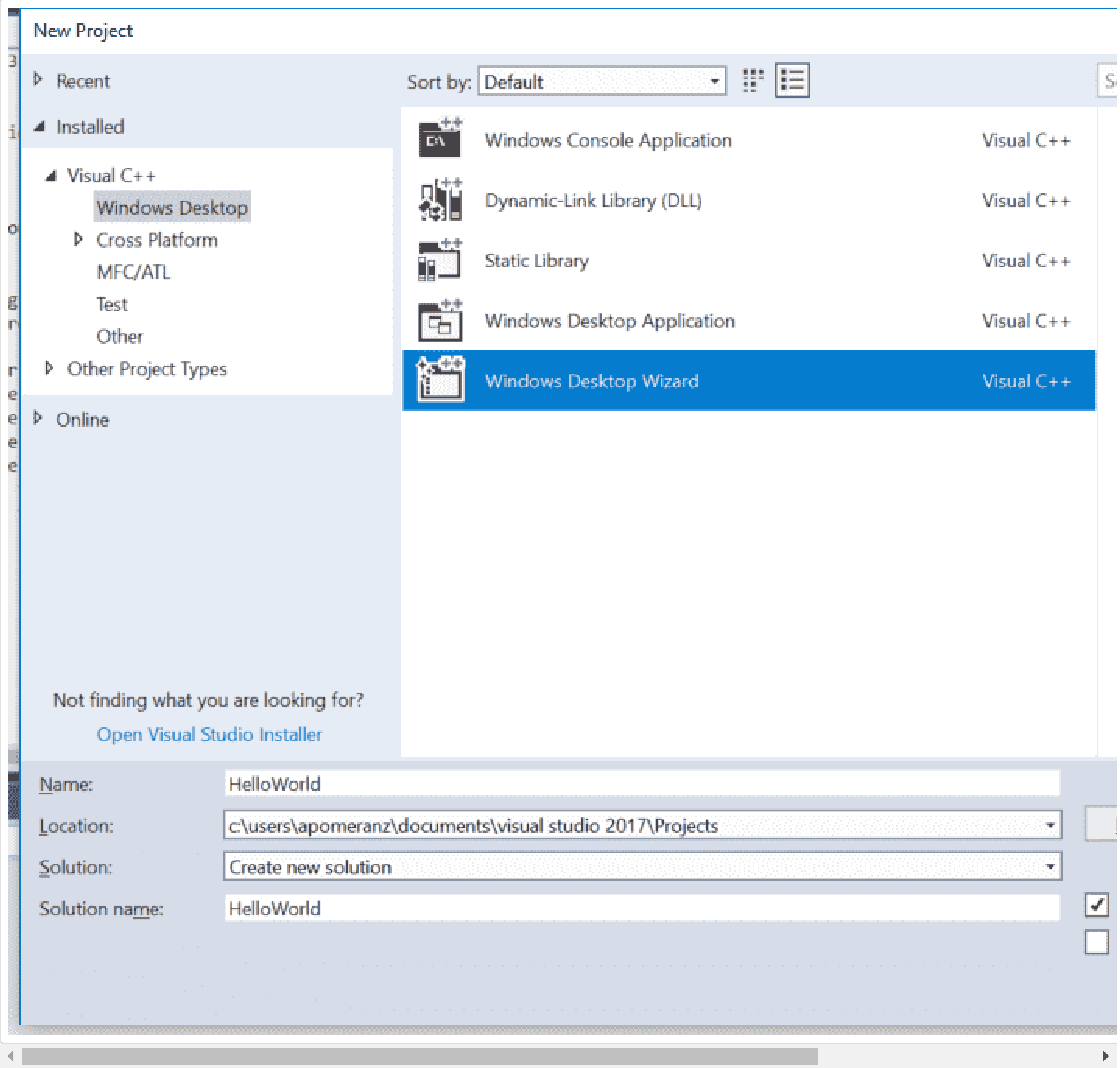
□ Empty Project
□ Precompiled Header
□ Export Symbols
□ MFC Headers

Make sure the *Application type* is set as *Console Application (.exe)* and that the *Precompiled Header* option is unselected. Then click *OK*.

You've now created a project! Jump down to the **Visual Studio Solution Explorer** section below to continue.

Creating a project in Visual Studio 2017 or older

To create a new project in Visual Studio 2017 or older, go to the *File menu* > *New* > *Project*. A dialog box will pop up that looks something like this:



New Project

Name: HelloWorld

Location: c:\users\apomeranz\documents\visual studio 2017\Projects

Solution: Create new solution

Solution name: HelloWorld

Recent
Installed
Visual C++
Windows Desktop
Cross Platform
MFC/ATL
Test
Other
Other Project Types
Online
Not finding what you are looking for?
Open Visual Studio Installer
Sort by: Default
Windows Console Application Visual C++
Dynamic-Link Library (DLL) Visual C++
Static Library Visual C++
Windows Desktop Application Visual C++
Windows Desktop Wizard Visual C++

First, make sure *Visual C++* is listed on the left side. If you don't see *Visual C++*, then you probably forgot to choose to install the *Desktop development with C++* workload when you installed Visual Studio. In that case, go back to lesson **0.6 -- Installing an Integrated Development Environment (IDE)** and reinstall your Visual Studio as indicated (note: rather doing a full reinstall, you can run the Visual Studio installer and modify your existing install to add the C++ workload).

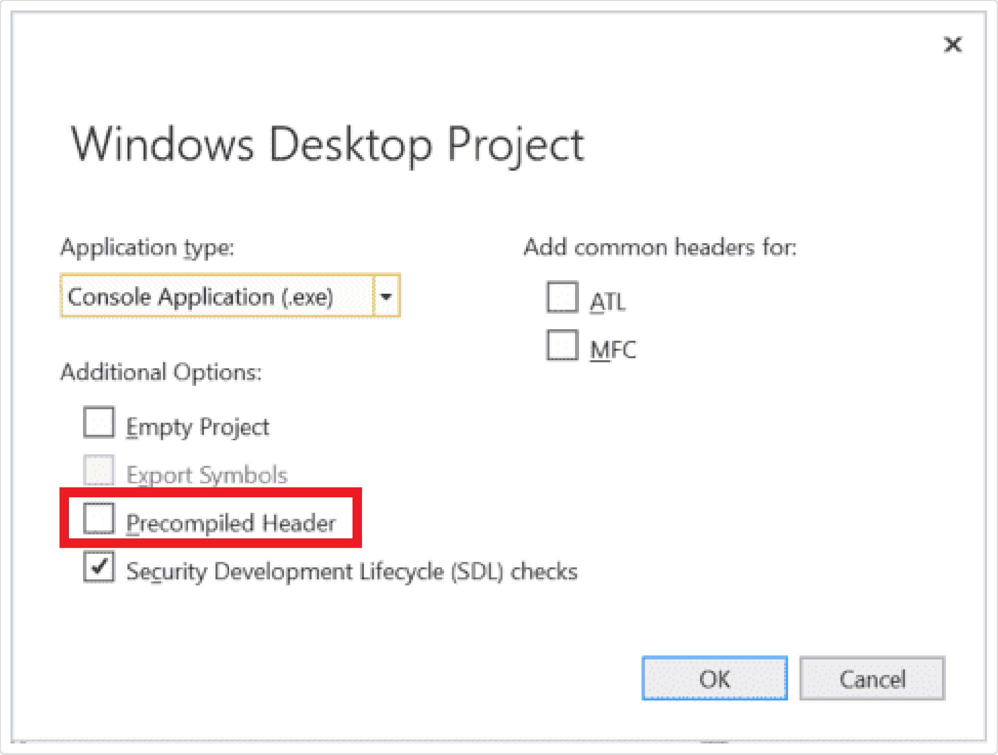
If you're using Visual Studio 2017 v15.3 or newer, underneath *Visual C++*, select *Windows Desktop* and then select *Windows Desktop Wizard* in the main window.

If you don't see *Windows Desktop* as an option, you're probably using an older version of Visual Studio. That's fine. Instead, choose *Win32* and then *Win32 Console Application* in the main window.

Down below, in the *Name* field, enter the name of your program (replace the existing name with `HelloWorld`). In the Location field, you can optionally select a different location for your project to be placed into. The default is fine for now.

Click *OK*. If you're using an older version of Visual Studio, the Win32 Application Wizard will launch. Press *Next*.

At this point, you should see a wizard dialog that looks something like this (older versions of Visual Studio use a different style, but have most of the same options):



Windows Desktop Project

Application type:
Console Application (.exe)

Add common headers for:
□ ATL
□ MFC

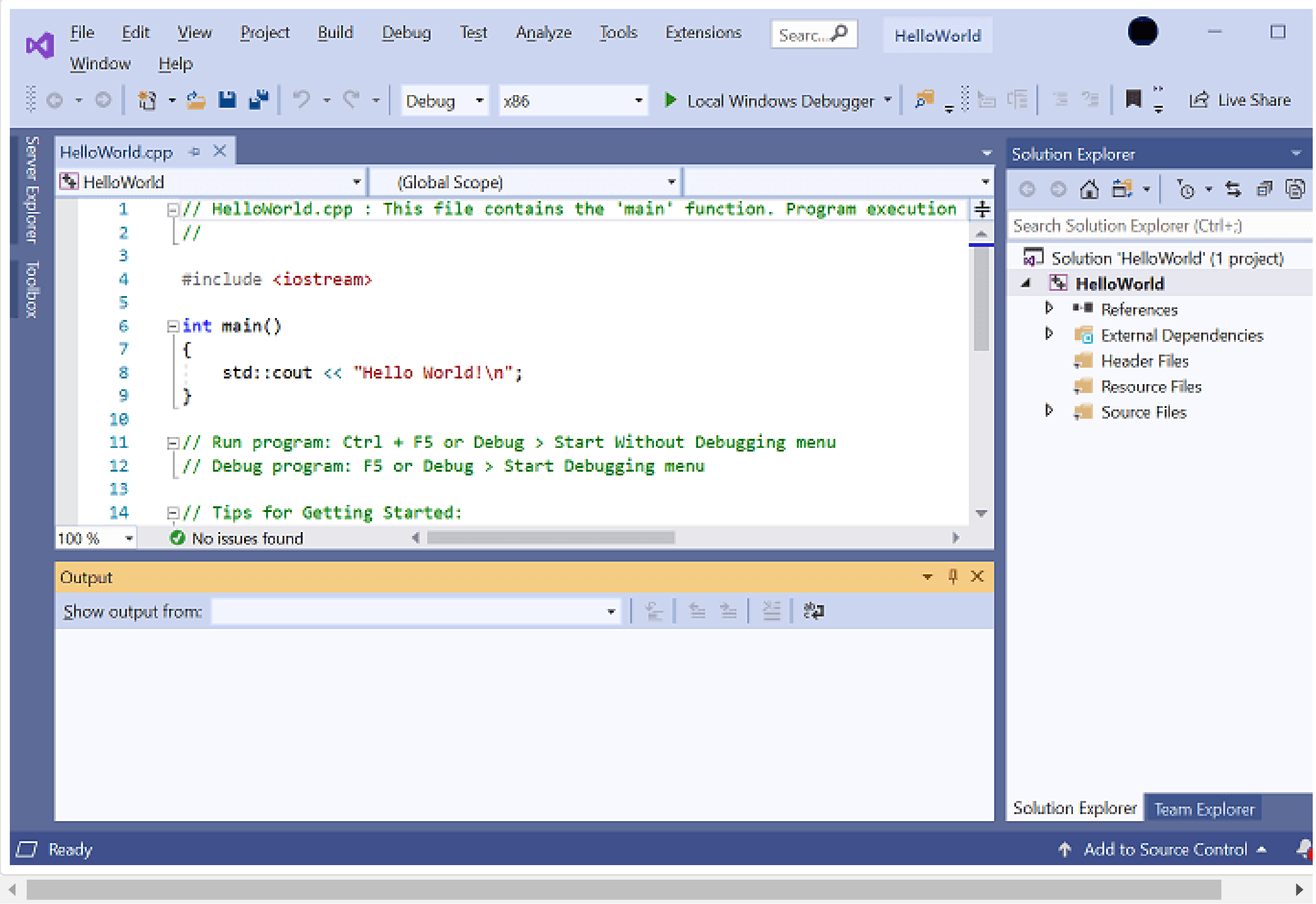
Additional Options:
□ Empty Project
□ Export Symbols
□ Precompiled Header
☑ Security Development Lifecycle (SDL) checks

Make sure you uncheck *Precompiled Header*.

Then click *Ok* or *Finish*. Now your project is created!

Visual Studio Solution Explorer

On the left or right side of the window, you should see a window titled *Solution Explorer*. Inside this window, Visual Studio has created a solution for you (*Solution 'HelloWorld'*). Within that, with the name in bold, is your new project (*HelloWorld*). Within the project, Visual Studio has created a number of files for you, including *HelloWorld.cpp* (underneath the *Source Files* tree item). You may also see some other .cpp or .h files, which you can ignore for now.



```cpp
// HelloWorld.cpp : This file contains the 'main' function. Program execution
//

#include <iostream>

int main()
{
    std::cout << "Hello World!\n";
}

// Run program: Ctrl + F5 or Debug > Start Without Debugging menu
// Debug program: F5 or Debug > Start Debugging menu

// Tips for Getting Started:
```

In the text editor, you will see that Visual Studio has already opened *HelloWorld.cpp* and created some code for you. Select and delete all of the code, and type/copy the following into your IDE:

```cpp
#include <iostream>

int main()
{
    std::cout << "Hello, world!";
    return 0;
}
```

To compile your program, either press *F7* (if this doesn't work, try *Ctrl-Shift-B*) or go to the *Build menu* > *Build Solution*. If all goes well, you should see the following appear in the Output window:

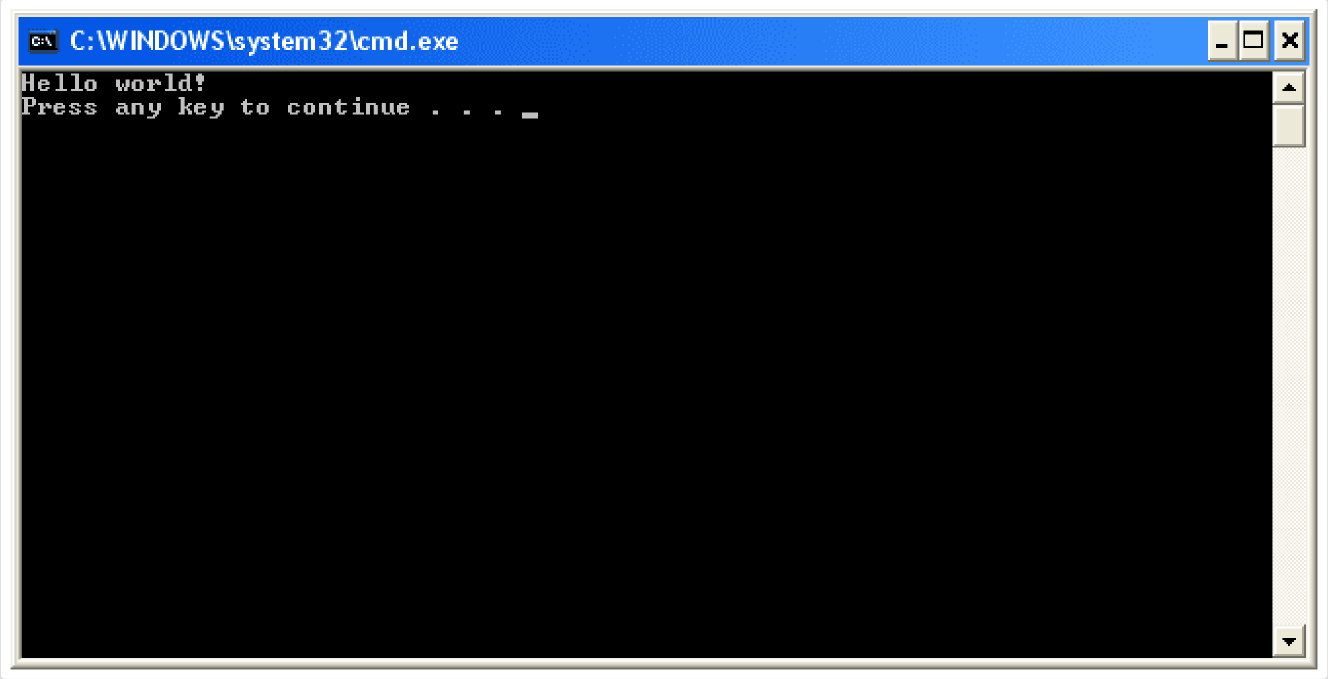
```
1>------ Build started: Project: HelloWorld, Configuration: Debug Win32 ------
1>HelloWorld.cpp
1>HelloWorld.vcxproj -> c:\users\alex\documents\visual studio 2017\Projects\HelloWorld\De
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

This means your compile was successful!

Q: I got error C1010 ("fatal error C1010: unexpected end of file while looking for precompiled header. Did you forget to add '#include "stdafx.h"' to your source?"). What now?

You forgot to turn off precompiled headers when you created your project. Recreate your project (as per the **instructions above**) and make sure to disable precompiled headers.

To run your compiled program, press *Ctrl-F5*, or go to the *Debug menu* and choose *Start Without Debugging*. You will see the following:

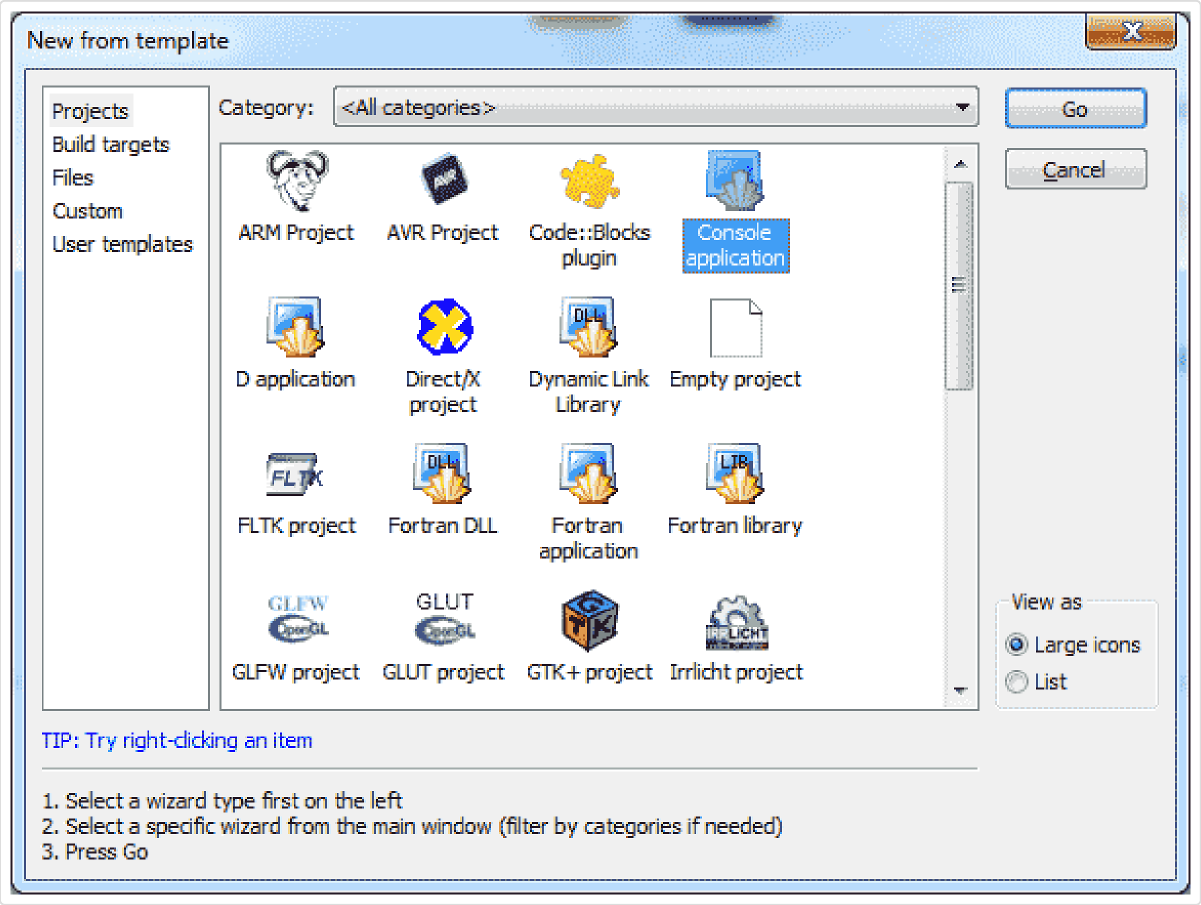


```
Hello world!
Press any key to continue . . . _
```

That is the result of your program! Congratulations, you've compiled and run your first program!

Creating a project in Code::Blocks

To create a new project, go to *File menu* > *New* > *Project*. A dialog box will pop up that looks like this:



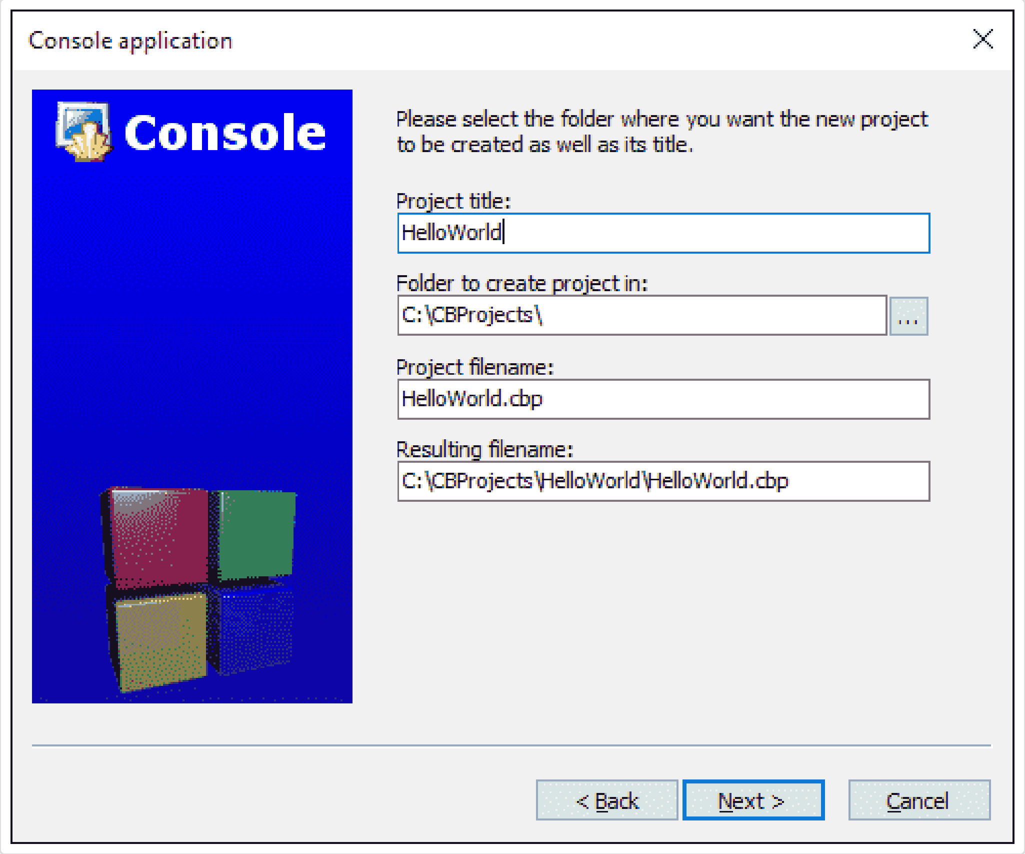
TIP: Try right-clicking an item

1. Select a wizard type first on the left
2. Select a specific wizard from the main window (filter by categories if needed)
3. Press Go

Select *Console application* and press the *Go (or Create)* button.

If you see a console application wizard dialog, press *Next*, make sure C++ is selected and press *Next* again.

Now you will be asked to name your project. Title the project `HelloWorld`. You can save it wherever you wish. On Windows, we recommend you to save it in a subdirectory of the C drive, such as `C:\CBProjects`.



Please select the folder where you want the new project to be created as well as its title.

Project title: HelloWorld

Folder to create project in: C:\CBProjects\

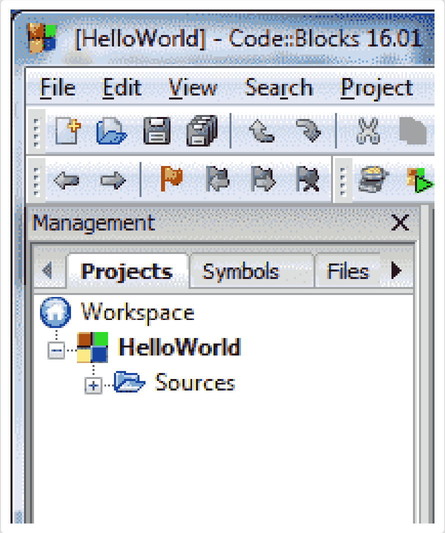
Project filename: HelloWorld.cbp

Resulting filename: C:\CBProjects\HelloWorld\HelloWorld.cbp

You may see another dialog asking you which configurations you want enabled. The defaults should be fine here, so select *Finish*.

Now your new project has been created.

On the left side of the screen, you should see a *Management* window, with the *Projects* tab selected. Inside that window, you'll see a *Workspace* folder, with your *HelloWorld* project inside of it:



Inside the *HelloWorld* project, expand the *Sources* folder, and double click on "main.cpp". You will see that a hello world program has already been written for you!

Replace that one with the following:

```
#include <iostream>

int main()
```

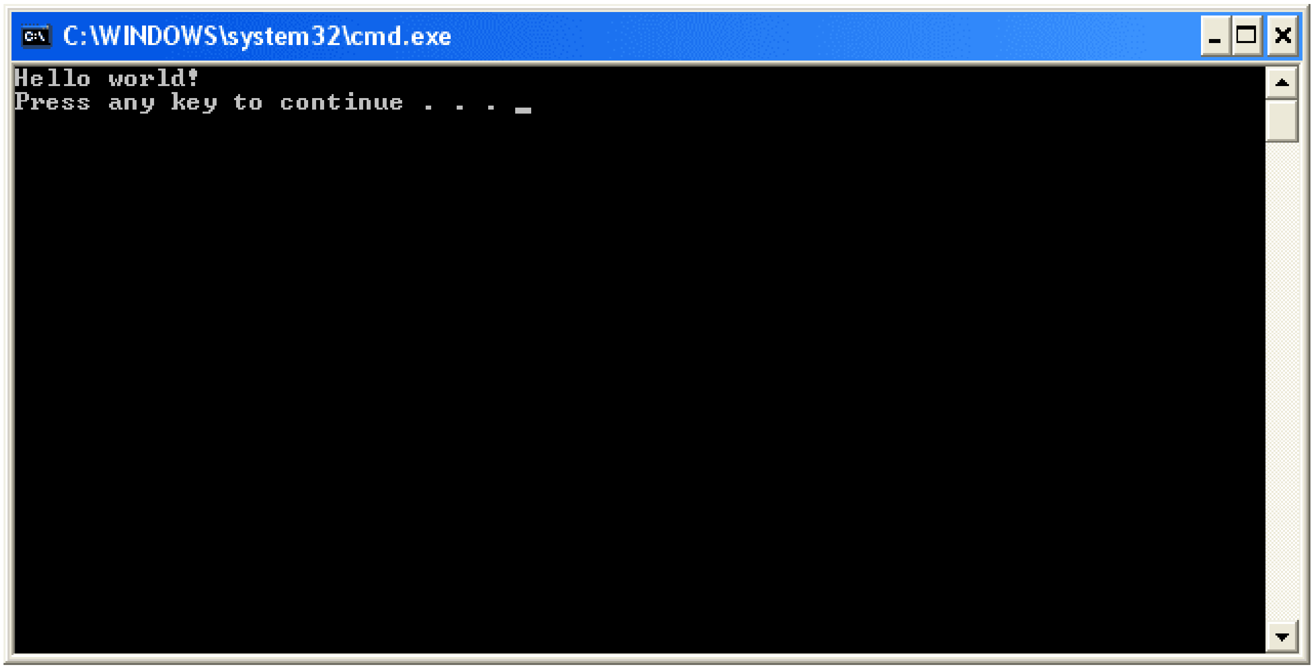
```
{
    std::cout << "Hello, world!";
    return 0;
}
```

To build your project, press *Ctrl-F9*, or go to *Build menu* > *Build*. If all goes well, you should see the following appear in the Build log window:

```
-------------- Build: Debug in HelloWorld (compiler: GNU GCC Compiler)---------------
mingw32-g++.exe -Wall -fexceptions -g -std=c++14  -c C:\CBProjects\HelloWorld\main.cpp -o
mingw32-g++.exe  -o bin\Debug\HelloWorld.exe obj\Debug\main.o
Output file is bin\Debug\HelloWorld.exe with size 1.51 MB
Process terminated with status 0 (0 minute(s), 0 second(s))
0 error(s), 0 warning(s) (0 minute(s), 0 second(s))
```

This means your compile was successful!

To run your compiled program, press *Ctrl-F10*, or go to *Build menu* > *Run*. You will see something similar to the following:



```
Hello world!
Press any key to continue . . . _
```

That is the result of your program!

For Linux users

Linux users may need to install the additional packages before Code::Blocks will compile. Please see the Code::Blocks installation instructions in lesson **0.6 -- Installing an Integrated Development Environment (IDE)** for more info.

If you’re using g++ on the command line

In this case, you don’t need to create a project. Simply paste the following into a text file named HelloWorld.cpp and save your file:

```
#include <iostream>

int main()
{
    std::cout << "Hello, world!";
    return 0;
}
```

From the command line, type:

```
g++ -o HelloWorld HelloWorld.cpp
```

This will compile and link HelloWorld.cpp. To run it, type:

`HelloWorld` (or possibly `./HelloWorld`), and you will see the output of your program.

If you’re using other IDEs or a web-based compiler

You will have to figure out how to do the following on your own:
1) Create a console project (IDEs only)
2) Add a .cpp file to the project (IDEs only, if one isn’t auto-created for you)
3) Paste the following code into the file:

```
#include <iostream>

int main()
{
    std::cout << "Hello, world!";
    return 0;
}
```

4) Compile the project
5) Run the project

If compiling fails

It’s okay, take a deep breath. We can probably fix it. :)

First, look at the error message that the compiler gave you. Most often, it will contain a line number indicating which line was in error. Examine both that line and the lines around it, and make sure there are no typos or misspellings. Also make sure you’re not including line numbers in your code.

Second, look at the Q&A in lesson **0.8 -- A few common C++ problems**, as your issue may be covered there.

Third, read the comments below -- someone may have encountered the same issue.

Finally, if all of the above fail, try searching for your error message on Google. It’s likely someone else has encountered this before and figured out how to fix it.

If your program runs but the window flashes and closes immediately

Some IDEs don’t automatically pause the console screen after the program has finished executing. Your program is running, but the output window is closing before you can view the results.

If this is the case with your IDE, the following two steps will fix your problem:

First, add or ensure the following lines are near the top of your program (Visual Studio users, make sure these lines appear after #include “pch.h” or #include “stdafx.h”, if those exist):

```
#include <iostream>
#include <limits>
```

Second, add the following code at the end of the main() function (just before the return statement):

```
std::cin.clear(); // reset any error flags
std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n'); // ignore any characters in
std::cin.get(); // get one more char from the user
```

This will cause your program to wait for the user to press a key before continuing, which will give you time to examine your program's output before your IDE closes the console window.

Other solutions, such as the commonly suggested `system("pause")` solution may only work on certain operating systems and should be avoided.

Your anti-virus or anti-malware may also be blocking execution of the program. If that's the case, try temporarily disabling your scanners and see if the problem resolves.

For Visual Studio users

Visual Studio will not pause at the end of a console application if it is run *with* debugging (Debug Menu > Start Debugging). If you want it to pause, you can either use the code solution above, or run your program without debugging (Debug Menu > Start Without Debugging).

Conclusion

Congratulations, you made it through the hardest part of this tutorial (installing the IDE and compiling your first program)!

Don't worry if you don't understand what all the different lines in the Hello World program do. We'll look at and explain each line in detail at the start of the next chapter.

0.8 -- A few common C++ problems

Index

0.6 -- Installing an Integrated Development Environment (IDE)

C++ TUTORIAL | PRINT THIS POST

784 comments to 0.7 — Compiling your first program

« Older Comments 1 ... 10 11 12

Joshua
February 7, 2020 at 6:50 pm · Reply

Hi I'm completely new to programming I'm using visual studio 2019 so this is probably a very stupid question. Anyways when I run the hello world it works fine but underneath the hello world it shows the c:/users/ thing so I was wondering how do I stop it displaying that? Again I know this is a very stupid question

Suyash
January 10, 2020 at 2:26 pm · Reply

In Visual Studio 2019, if you wish to prevent your console window from closing automatically, you can go to Tools > Options... > Debugging > General > uncheck 'Automatically close the console when debugging stops'.

By default, this option is unchecked on Visual Studio 2019... I believe the aforementioned solution should also work for earlier versions as well...

Vitaliy Sh.
December 28, 2019 at 10:18 am · Reply

Hi Sires! (i'd re-read this)

Possibly omitted:

"Within that, with the name in bold, is your new project (HelloWorld)." -- "... bold: it is ..."?

In VS 2019 there was: "Make sure ... is set as Console Application ...". It is not present in `2017 or older`'s `Make sure part`: "Make sure you uncheck Precompiled Header."

Vitaliy Sh.
December 22, 2019 at 1:23 am · Reply

HTML: " window, with the Projects tab selected. Inside that window, you’ll see a "
Possibly <em>Projects</em> ?

Vitaliy Sh.
December 22, 2019 at 1:15 am · Reply

"You can save it wherever you wish, though we recommend you save it in a subdirectory of the C drive, such as C:\CBProjects."

… though for the Windows platform we … ?
https://www.learncpp.com/cpp-tutorial/writing-your-first-program/comment-page-12/#comment-435762

nascardriver
December 22, 2019 at 4:06 am · Reply

I added that the recommendation is for Windows users. Projects emphasized. Thanks!

Vitaliy Sh.
December 21, 2019 at 3:09 am · Reply

"You forgot to turn off precompiled headers … Recreate your project (as per the instructions above) … "

```
<!-- <img id="pchOFF"> -->

<a href="#pchOFF">above</a>
```

?

Vitaliy Sh.
December 20, 2019 at 9:15 pm · Reply

"Jump down to the Visual Studio Solution Explorer section below to continue."

Can you, please, add a jump?

```
<a href="#VSSE">section</a>

<!-- ... -->

<p id="VSSE" class="cpp-section cpp-topline" style="clear: both">Visual Studio Solution Expl
```

?

nascardriver
December 21, 2019 at 4:21 am · Reply

Anchors added.

Thomas Barlow
January 4, 2020 at 12:34 am · Reply

hey there my name is Tom and I'm working with this tutorial and find it very interesting... Thanks for all the helpful tips! I have one question though, which version of of Visual Studio 2019 do I need to download in order to move along with the tutorial? I know it's late so answer at your earliest convenience! Thanks again

nascardriver
January 4, 2020 at 2:51 am · Reply

VS is for Windows, the Mac version is for C#. If you're on Mac, get something else.

Visual Studio 2019 Community (Free) is all you need. The Professional and Enterprise versions have more functions, you don't need those.

Vitaliy Sh.
December 20, 2019 at 6:42 pm · Reply

Hi Sires!
...
(note: rather doing a full reinstall, you can run the Visual Studio installer and modify your existing install to add the C++ workload)
...

If i understand, "rather" is used to make "expression less assertive" there. But the part after a comma have no "instead".

Is that is better, or worse: "instead of doing a full reinstall, you rather can run ..."?

nascardriver
December 21, 2019 at 4:17 am · Reply

Sentence changed to "rather than doing a full reinstall, you can run the Visual Studio installer and modify your existing installation to add the C++ workload" (Added "than" and changed "install" to "installation"). That should fix it.

Vitaliy Sh.
December 20, 2019 at 5:05 pm · Reply

...
<a href="http://en.wikipedia.org/wiki/Hello_world">hello world program</a>
...

Please change to "https", as it redirects there anyway.

nascardriver
December 21, 2019 at 4:11 am · Reply

Updated, thanks!

jeff
November 28, 2019 at 11:37 am · Reply

Hi, I have a c: SSD and a d: HDD but my SSD is small so I try to download/save stuff on my HDD. Whilst installing Microsoft visual studios I selected d: and it said `This drive will affect the

performance of Visual Studio and cause it to run slower.`
Is this significant? if I do not need to save my projects in the same file as my ide then I can download MVS on the SSD and save projects in the HDD.
Could you tell me what I should do in this situation?

nascardriver
November 29, 2019 at 3:09 am · Reply

Install it on the HDD. If you're not happy with the performance, reinstall on the SSD. VS itself is huge (~20GB), if your ssd is small, you probably don't want to use it for VS.

Medhat Diab
November 27, 2019 at 2:56 am · Reply

Visual Studio 2019 Users: After clicking Create, You won't see the last dialog Which is called: Windows Desktop Project.
You will need this to disable (Precompiled Headers) and also to nake Application Type: Console, to do that follow this steps:

1- In Solution Explorer press right click on your project name --> Properties --> C/C++ --> Precompiled Headers --> Precompiled Header (The first one on the top) Then Choose: Not Using Precompiled Headers.

2- Linker --> System --> SubSystem --> Choose: Console.

3- Press Apply and Ok.

Georges Theodosiou
November 22, 2019 at 5:57 am · Reply

Dear Teacher,
Please let me say regarding "If you’re using g++ on the command line" section that there are not instructions for install gcc-MinGW. Then I suggest following
quote
Go to https://sourceforge.net/projects/tdm-gcc/ and during downloading
1. On "Wizard Action" click [Create].
2. On "Select Edition" select MinGW/TDM (32-bit) > click [Next>].
3. On "License Changes" click [Next>].
4. On "New Installation: Installation Directory" click [Next>].
5. On "New Installation: Download Mirror" click [Next>].
6. On "New Installation: "Choose Components" click [Install].
7. Wait till [Next>] be brighten, then click it.
8. Click [Finish].
9. Close README text file.
10. Open a command Prompt.
quote
Also let me say you that file HelloWorld.cpp should be placed in a folder, e.g. gccProjects and commands be typed at the prompt of this folder e.g.
C:\gccProjects>
Also let me point out that project's name could be any valid name e.g. project-1. So command for compile program could be
C:\gccProjects>g++ -o project-1 HelloWorld.cpp
Then command for run program could be
C:\gccProjects>project-1
Regards.

Georges Theodosiou
November 25, 2019 at 4:44 am · Reply

Dear Teacher,
Please let me add that with Word and Word-pad you can't create file .cpp. However with Bloc-notes you can.
Regards.

Vitaliy Sh.
November 18, 2019 at 1:02 am · Reply

Between the titles "If compiling fails" and "Conclusion", there is a duplicate of 0.8's "Q: When executing a program, the console window blinks and then closes immediately.".

Alex
November 19, 2019 at 11:45 am · Reply

That's intentional. It's one of the most common issues new users encounter, so I didn't want to send new users to another lesson to find a solution. But since 0.8 is a roll-up of common issues, it seemed appropriate to put it there as well.

Vitaliy Sh.
November 17, 2019 at 11:02 pm · Reply

Hi people!

Why "ctrl" is preffered here over "Ctrl"?

https://en.wikipedia.org/wiki/Control_key#Notation
https://en.wikipedia.org/wiki/Control_key#Table_of_examples
https://wiki.archlinux.org/index.php/Keyboard_shortcuts

Alex
November 19, 2019 at 11:41 am · Reply

Fixed. Thanks! There are probably more of these in future lessons. Let me know if you find any.

Vitaliy Sh.
November 11, 2019 at 10:21 pm · Reply

Hi!

… Some compilers (such as Bloodshed's Dev C++) don't automatically pause the console screen after the program has finished executing. …

Typo: "compilers" instead of "IDEs".

Diagrams in previous lessons are so beutiful, that refer compiler in connection with already compiled app isn't fit.

Vitaliy Sh.

November 11, 2019 at 5:27 am · Reply

... that can be run from the windows, linux, or mac console. ...

Minor typo: non capital letters in "Windows", "Linux", "Mac".

Vitaliy Sh.

November 11, 2019 at 4:06 am · Reply

... save it in a subdirectory off of the C drive, such as C:\CBProjects. ...

Possible Typo: "off of", while still in C:\.

nascardriver

November 11, 2019 at 4:26 am · Reply

So many mistakes in a single lesson, thanks!

Vitaliy Sh.

November 11, 2019 at 5:11 am · Reply

Don't ban me, people, please :).

Alex

November 13, 2019 at 2:38 pm · Reply

Thanks for all your feedback! Editing is hard, and you seem to have a knack for it. :)

Vitaliy Sh.

November 11, 2019 at 3:00 am · Reply

... or go the Debug menu and choose Start Without Debugging ...

Typo: no "to" between "go" and "the".

Vitaliy Sh.

November 11, 2019 at 2:54 am · Reply

Hi.

There is two different scripts of key Ctrl, three with lower case "c", one with capital "C":

{
...(if this doesn’t work, try Ctrl-Shift-B)...
...To run your compiled program, press ctrl-F5,...

...To build your project, press ctrl-F9...
...To run your compiled program, press ctrl-F10,...
}

Maybe add some unification?

Vitaliy Sh.

November 11, 2019 at 2:02 am · Reply

...It’s recommended you also check the Place solution and...

Typo: no "that" between "recommended" and "you".

Vitaliy Sh.
November 11, 2019 at 1:30 am · Reply

...By default, console applications have no graphical user interface (GUI), print text to the monitor, read input from the keyboard, and are compiled into stand-alone executable files.

...

Maybe better:
... (GUI) - they print text to the monitor, read ...

Maybe even change "monitor" to "console", for it were mentioned already: above and below of that sentence?

Sinethemba Zungu
November 7, 2019 at 8:08 am · Reply

How do you enter "()" bracket characters on Linux command line in order to create a new directory for a new project? Example this command line ("mkdir Program Files (x86)/builds") give the following error: "bash: syntax error near unexpected token `('"

nascardriver
November 7, 2019 at 8:15 am · Reply

You don't need a "Program Files" directory on linux, you can create the project wherever you want, eg. in "~/projects/".
If you really want a name with parentheses and spaces, you can write the path in quotation marks.

```
mkdir -p "Program Files (x86)/builds"
```

-p automatically creates parent directories.

Georges Theodosiou
November 5, 2019 at 6:14 am · Reply

Dear Teacher,
Please let me ask you whether confirm following: In section "If you’re using g++ on the command line" first paragraph to be replaced by "In directory C: create folder "Projects cmd" and paste the following into a text file named HelloWorld.cpp and save in C:\Projects cmd."
After program to be written:
"
From the command line
C:\Projects cmd
type:
g++ -o HelloWorld HelloWorld.cpp
"
Regards.

nascardriver
November 5, 2019 at 6:42 am · Reply

Hi!

It doesn't matter which directory you create the file in, you can place it wherever you want. Adding a path to the lesson would make it less portable. There's a path in the Code::Blocks section, because Code::Blocks needs a path to create the project at.

Georges Theodosiou
November 5, 2019 at 6:59 am · Reply

Dear NASCAR driver,
Please accept my sincere thanks for your immediate response and useful remark.
Indeed we have to type:

```
g++ -o HelloWorld HelloWorld.cpp
```

after the path were file lives. Regards.

kens
August 11, 2019 at 6:53 pm · Reply

Hello! At Visual Studio 2019, How can I zoom out/in the entire page of it? I am not able to see the (NEXT) button below when you create a new project and click the Windows Desktop Wizard. does anybody hear knows how to fix this error?

troxodev
September 8, 2019 at 4:21 am · Reply

Hello! If you mean to "zoom" the code, you must press: "ctrl + scroll (mouse wheel)".
I hope that has been of your help!

Monolog
August 8, 2019 at 11:15 pm · Reply

I have a problem, I´m from the Czech Republic and when I want to write down anything like bracket or semicolon I have to press buttons that are basicly shortcuts for steps in Visual Studio. So the only option for me is to change to US keyboard in Windows, but I´m not really used to it. For example curly brackets are in CES Ctrl+Alt+B / N etc.. Is there any chance to fix this or should I start getting used to US keyboard?

nascardriver
August 9, 2019 at 2:37 am · Reply

You can change the affected shortcuts in visual studio's settings.
I suggest learning the US layout (Or even dvorak if you're up for some fun). Pressing more than 2 keys to get 1 character is slow and might hurt your hands if you're writing a lot of code.

Orphic
August 7, 2019 at 3:45 am · Reply

```
Hello, world!
C:\Users\Orpheus\source\repos\HelloWorld\Debug\HelloWorld.exe (process 2452) exited with code 0.
Press any key to close this window . . .
```

can someone help me understand why is the 2nd line is seen in my console window unlike his?

Alex
August 8, 2019 at 7:43 pm · Reply

Your IDE is showing you the return value from your program. I think we discuss this early in the next chapter. Keep reading.

David Cane
July 20, 2019 at 9:32 pm · Reply

And I have another question.Do ALEX has other websites or blogs regarding programming.
I prefer C/C++ □but other kinds are also fine.

Alex
July 22, 2019 at 4:31 pm · Reply

Nope. This is my sole site. It already requires more maintenance than I can manage. :)

David Cane
July 20, 2019 at 9:27 pm · Reply

Hi,I am just going to be a freshman.I have already had a rough understanding of C.
But I would like to learn C++ as well.Any tips?
Moreover,what else should I read or browse if I want to learn C++ deeper.
For instance,some books,some websites?

nascardriver
July 21, 2019 at 12:37 am · Reply

Get a project going.
When you're done with the tutorials on learncpp, read about the standard libraries on cppreference.
Don't get a book, they're likely outdated, don't get updates and you can't ask questions.

David Cane
July 21, 2019 at 5:17 pm · Reply

Thanks a lot

David Cane
July 20, 2019 at 7:20 pm · Reply

This is extremely helpful,thank you for all your efforts.

Jeff
July 20, 2019 at 5:32 am · Reply

Just as an FYI,

Im assuming visual studio code came out after the initial postings. I would recommend using that instead of the full blow visual studio, it is still free but less confusing and no real need to make "projects" just create a file with the .cpp extensions and your good to go. also has a built in terminal so you can compile and run without ever leaving the editor.

« Older Comments 1 … 10 11 12