Visual Studio

We recommend you install **Visual Studio IDE** on your Windows computer for programming in the C language. An IDE is an Integrated Development Environment which is far more than a coding editor.

Visual Studio is a "fully-featured, extensible, free IDE for creating modern applications for Android, iOS, Windows, as well as web applications and cloud services." We use it in all our programming courses because it is one of the world's most highly rated and most used software development tools popular among both professionals and students. The C programming course lectures, workshops, and how-to instructions all assume the use of Visual Studio IDE.

Although the cross-platform Visual Studio Code is not officially supported or used in our school, it has been a viable alternative to Visual Studio IDE for many who use macOS or Linux, or even Windows running on a low or medium powered machine. ("machine" is geek speak for any computer.)

Installing Visual Studio for macOS *will waste your time – despite the name, it is not the same thing as VS IDE or VS Code.*

Operating System environment

Visual Studio IDE runs **only in a Windows environment**.

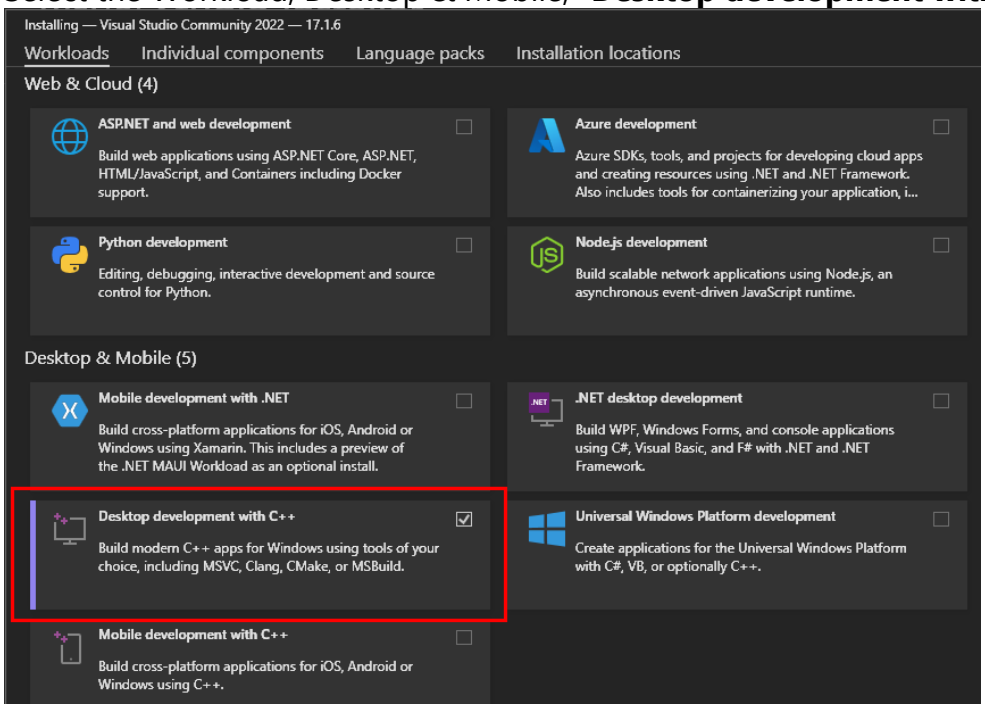
Students are no longer licensed for the Windows OS to upgrade their own PCs or to run Windows as a second OS with macOS or Linux. *Complain to your student council.*

To run VS IDE well requires a relatively powerful machine. Apple macOS users and Linux gurus must run a dual boot system (e.g. Apple Bootcamp) or run VS IDE inside a virtual Windows OS machine (e.g. Parallels for macOS which requires a *powerful* machine). When running Windows OS on an Apple Mac, it assumes a Microsoft standard keyboard. See Apple Keyboard in Windows with Boot Camp and Keyboard mappings using a PC keyboard on a Macintosh.

Visual Studio IDE Installation

Start your installation of Visual Studio Community from https://www.visualstudio.com/vs/community/. During VS Installation,

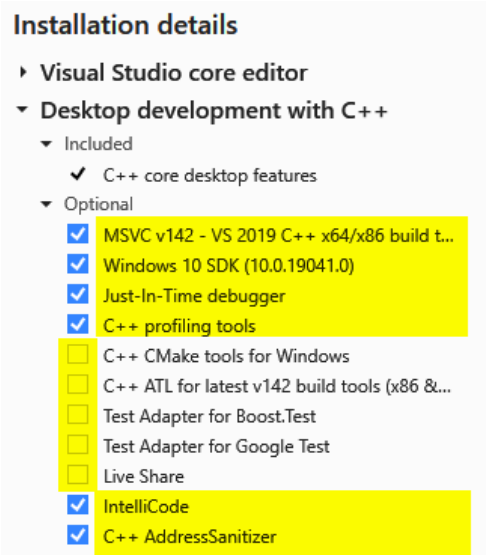
- Select the Workload, Desktop & Mobile, "**Desktop development with C++**



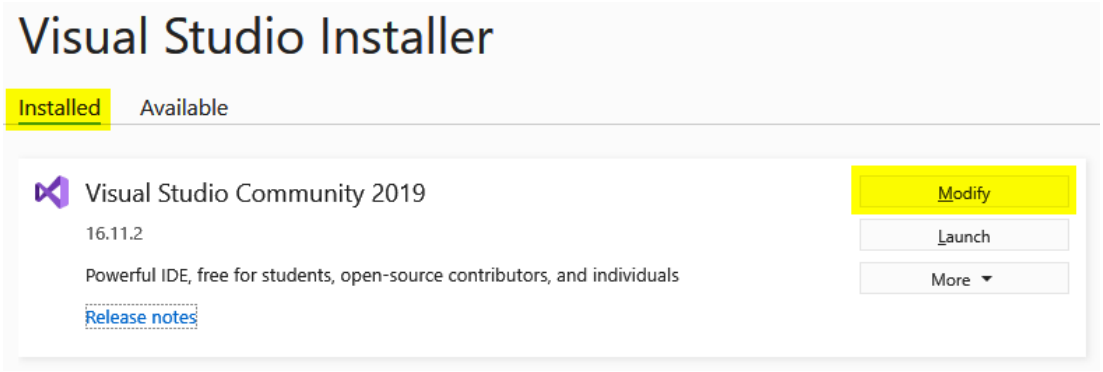
"

- Check these "Optional" components needed for C programming
  - MSVC v### [the most current version of build tools] for your processor
  - Windows 10 SDK [the most current version]
  - Just-In-Time debugger
  - C++ profiling tools
  - IntelliCode
  - C++ AddressSanitizer

- Uncheck the following to save ~832MB of storage and reduce the background load on your system.
  - C++ CMake tools for Windows
  - C++ ATL for latest v000 build tools
  - Test Adapter for Boost Test
  - Test Adapter for Google Test
  - Live Share



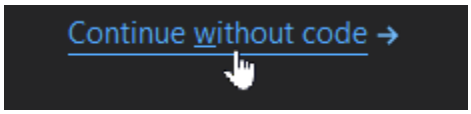
If the above was missed on the initial install, run “Visual Studio Installer” > Modify to modify Visual Studio.



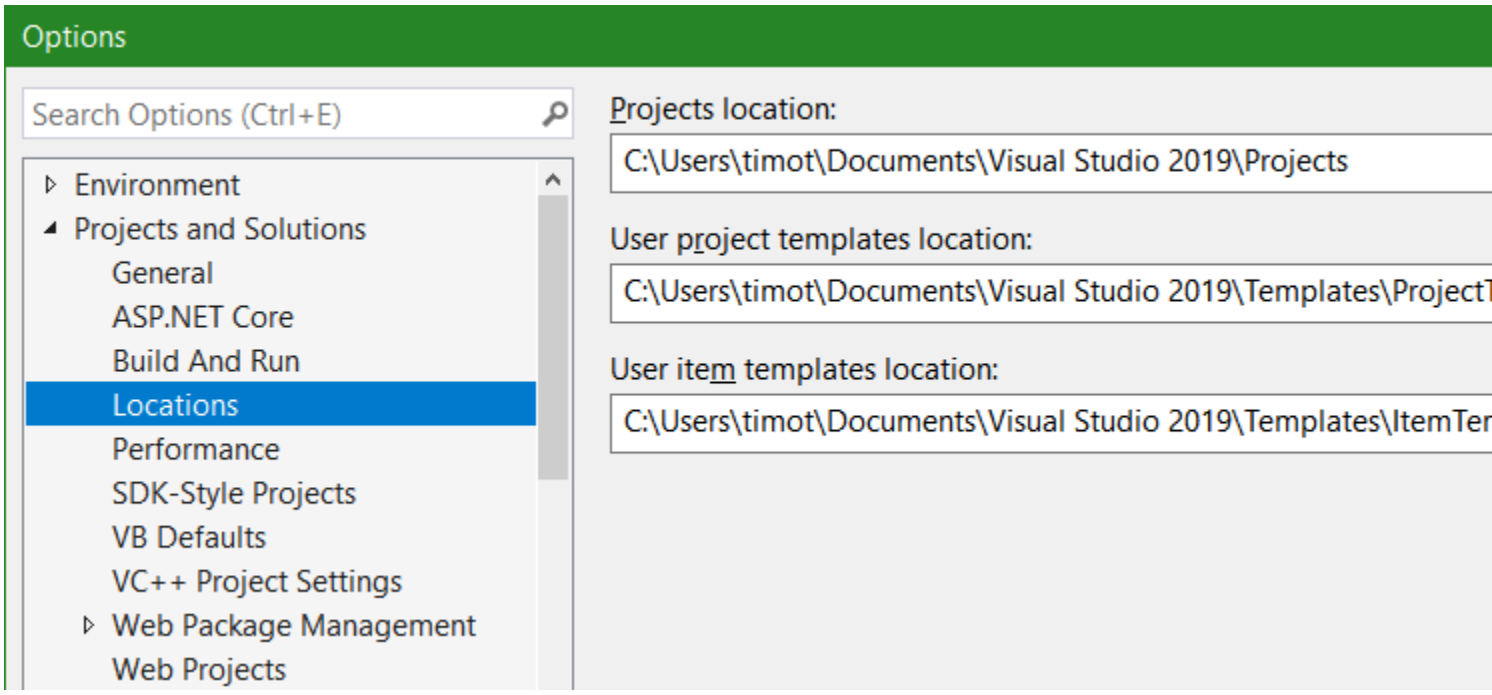
Windows Security / Microsoft Defender or other anti-malware anti-virus utilities

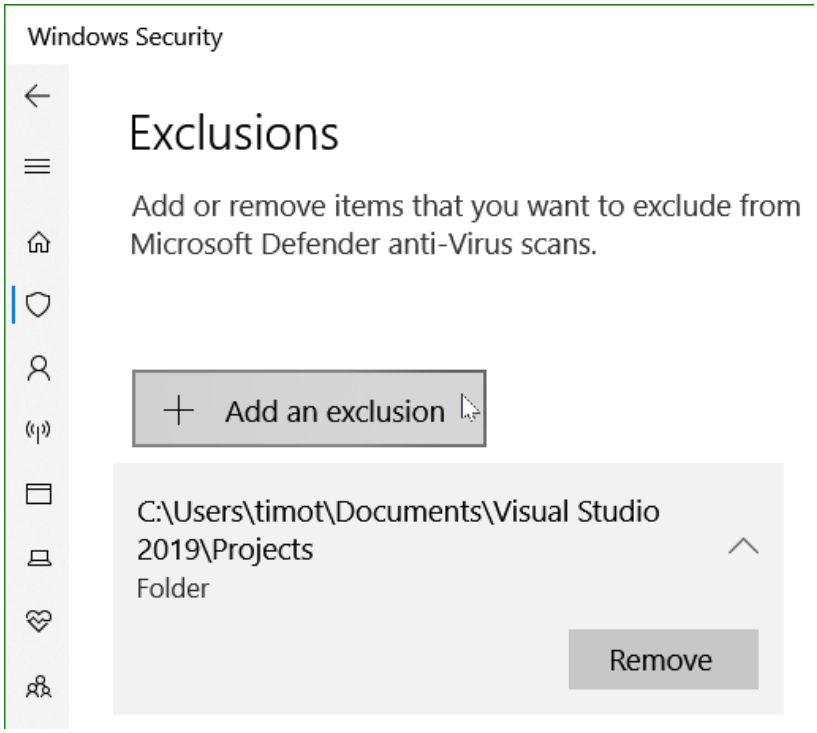
Security utilities sometimes prevent new executable files, like you create from C programming, from running on your system. Add an Exclusion for the Visual Studio projects folder.

Start Visual Studio, continue without code →



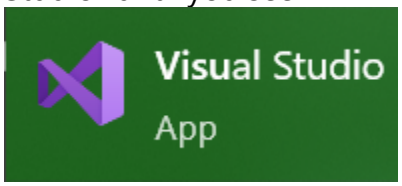
menu Tools > Options > Projects and Solutions > Locations





Running Visual Studio

***To start Visual Studio IDE*:**
Press the Windows key ⊞ or click the Windows icon in the lower left and start typing "Visual Studio" until you see



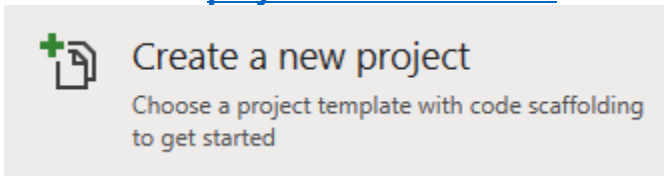
(click on this or press Enter to launch)

Show these notes on one side of your screen: Windows key ⊞ + → [right arrow]

Show Visual Studio beside these notes with Windows key ⊞ + ← [left arrow]
Because Visual Studio (VS) can manage various types of projects, it is one of the industry standards for systems development. As such, it is far more than just a programming code editor. Thus, the next few steps require your careful attention to set up the VS project for the type of program we will be creating.

From the **Visual Studio**, **Get started window …**

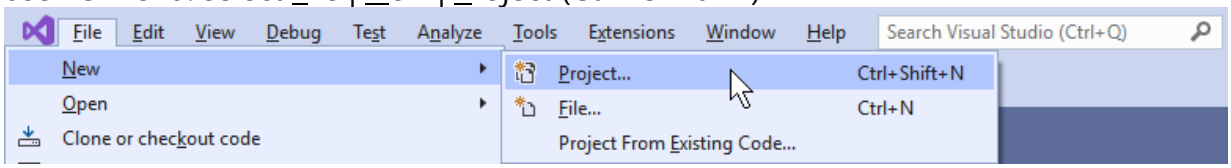
Create a new project in Visual Studio



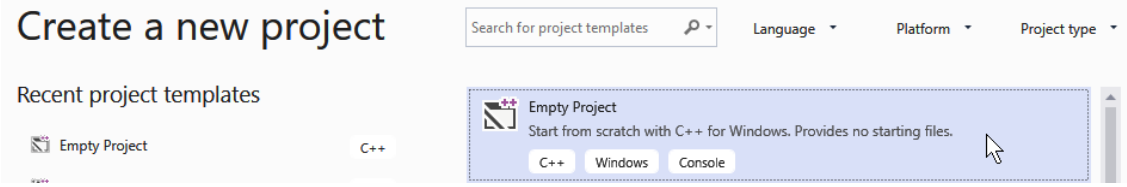
Or click on 

to get to the application menu

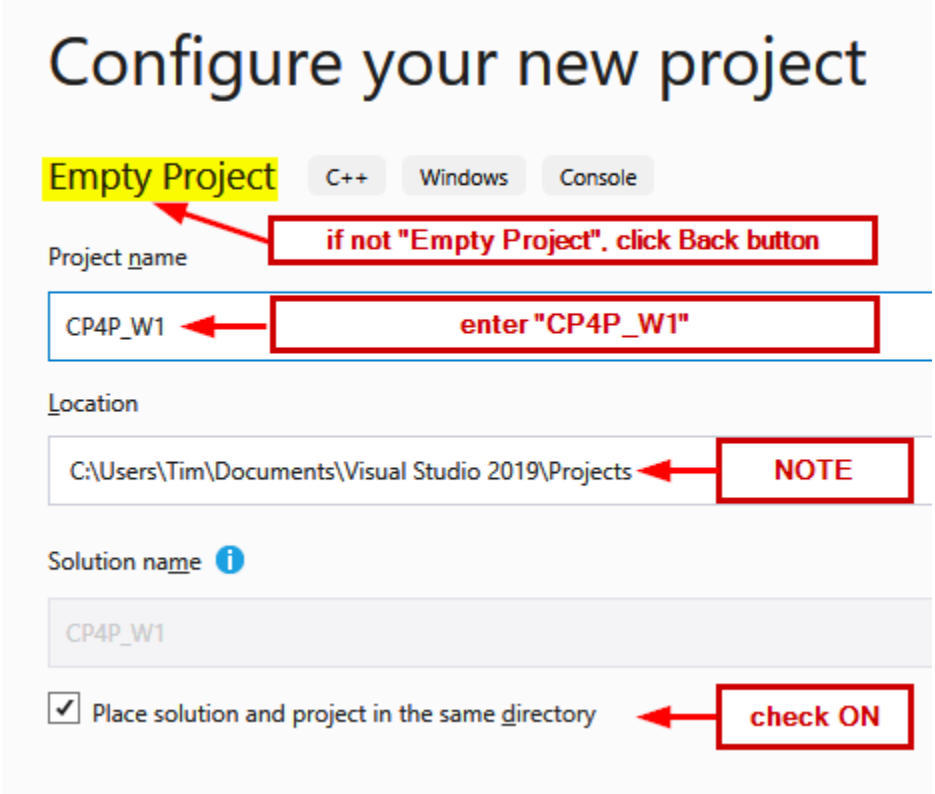
- use VS menu: select File | New | Project (Ctrl+Shift+N)



- On the **Create a new project** display, select **Empty Project** (else Visual Studio will default to C++ instead of C) and press Enter



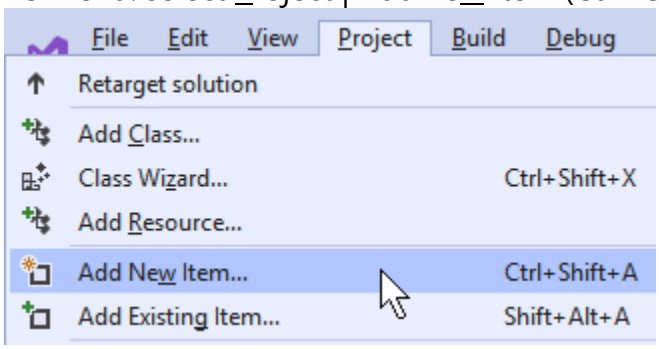
- Enter **CP4P_W1** as the Project Name
  (**C**omputer **P**rinciples ***for*** **P**rogrammers _ Week 1)
  - Note the Location (or override it) before continuing by clicking Create in the bottom right corner



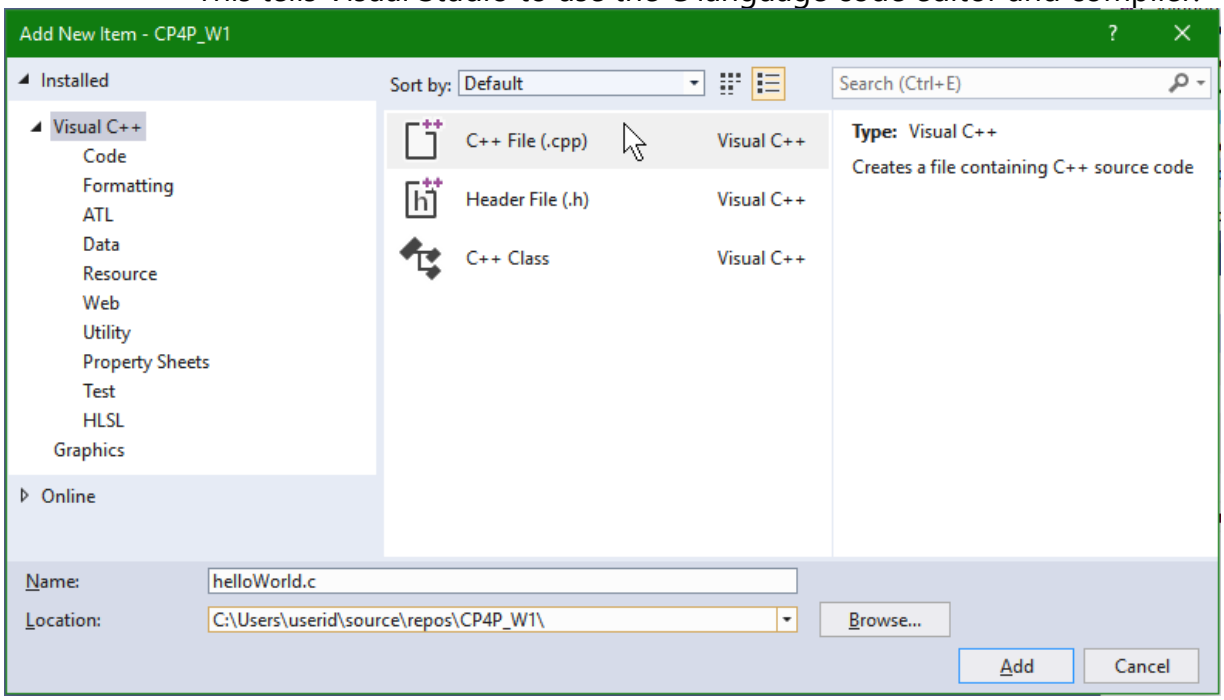
Hello World

Create a C language source code file…

- VS menu: select Project | Add new Item (Ctrl+Shift+A)



- Check that Visual C++ is selected on the left
  and **C++ File (.cpp)** in the center pane is also selected
  enter `helloWorld.c` as the file Name | press Add
  - ***Make sure the file extension is ".c", not the default .cpp*** (C++)
    This tells Visual Studio to use the C language code editor and compiler.



- Copy & paste the source code for the classic first program, "Hello World". Change `yourNameHere` to your own name.

```
/* Thanks to Brian Kernighan, 'Hello World' is the traditional first C program. It became
legendary with the publication of "The C Programming Language" by Kernighan and Ritchie (1978).
Now, Hello World is the canonical test of any programming language. */
/*
Full Name   :
Student ID# :
UserID/Email: @mySeneca.ca
*/
#include <stdio.h>// C language module providing Input/Output facilities

int main(void)    //  main() is automatically called to start a C program
{
    printf("Hello, World!\n");   // output greeting

    printf("This is yourNameHere.\n"); // output your name

    return 0; // return to operating system
}
```

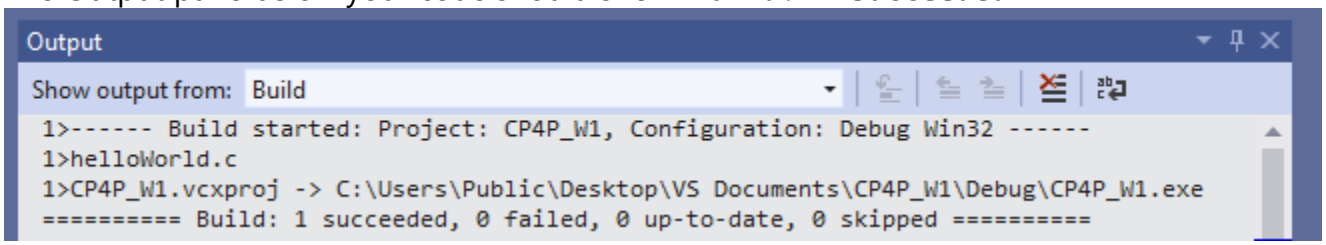
See *Hello World*, *The History of 'Hello, World'*, *The Hello World Collection*

- If the first line of comments is not wrapping,
  - VS menu: Edit | Advanced | Word Wrap (Ctrl+EW)
- Save the source file (Ctrl+S)

Compile your C program…

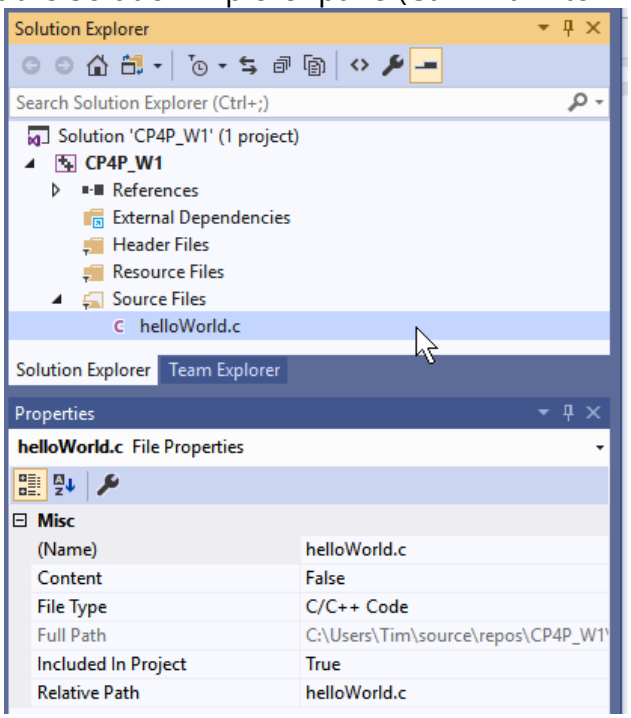
- VS menu: select **Build** | **Build Solution** (F7 or Ctrl+Shift+B)

The Output pane below your code should show `Build: 1 succeeded`



Troubleshooting

If it failed and your source code exactly matches the above,
look at the Solution Explorer pane (Ctrl+Alt+L to View it)

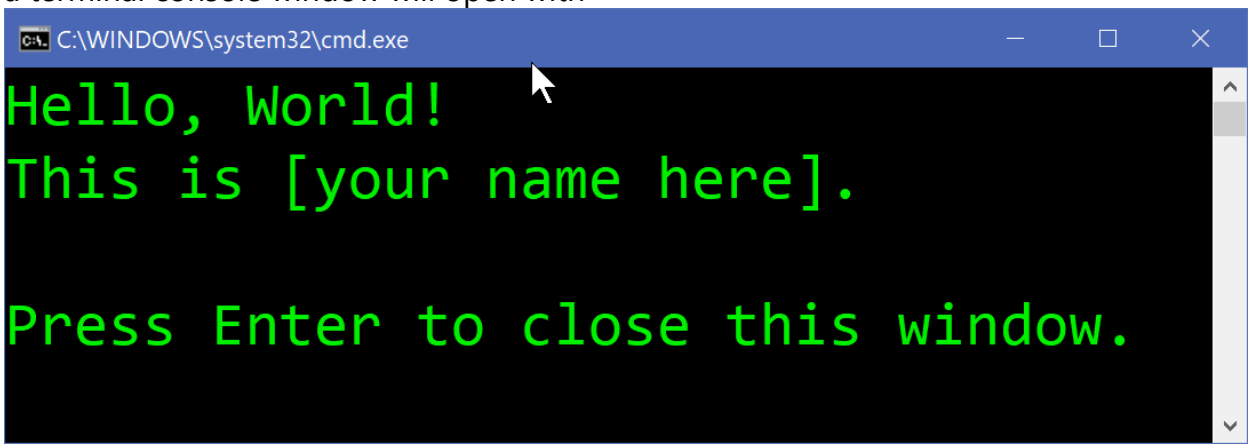


- Only one source file should exist in the project
  - delete all except **`helloWorld.c`**
- Check that **`helloWorld`** ends with the extension **`.c`** and not **`.cpp`**
  - Use the Solution Explorer / Properties pane to rename the file to end in **`.c`**
- …and run Build Solution again.
- If all else fails, it is best to start again.
  - Exit Visual Studio
  - Use File Explorer to find the CP4P_W1 folder under …\repos\
  - Delete the CP4P_W1 folder
  - Then recreate the project as per the above notes.

When the compile was successful, run your program.

- VS menu: select **Debug | Start without Debugging** (Ctrl+F5)

- a terminal console window will open with



  The appearance of the window varies.
  Customize by clicking the icon in the top left / Properties.
- Close the window when you've sufficiently admired your work.

Microsoft has notes for Getting Started with Visual Studio for C and C++ Development which you can explore.

Locating VS repository and source file

Now, where is that helloWorld.c **source file?**
You may have seen the full pathname when your program ran in the terminal window.
Source files are located under the VS Project name folder. And where is that?
There are several ways to find it.

You can start from the file's location and work your way up the folder structure:

- Press the Windows key ⊞ and type the filename **helloWorld.c**
- Right clicking on the filename, depending on your PC's configuration, may give you a list of options such as Open file location, Copy full path

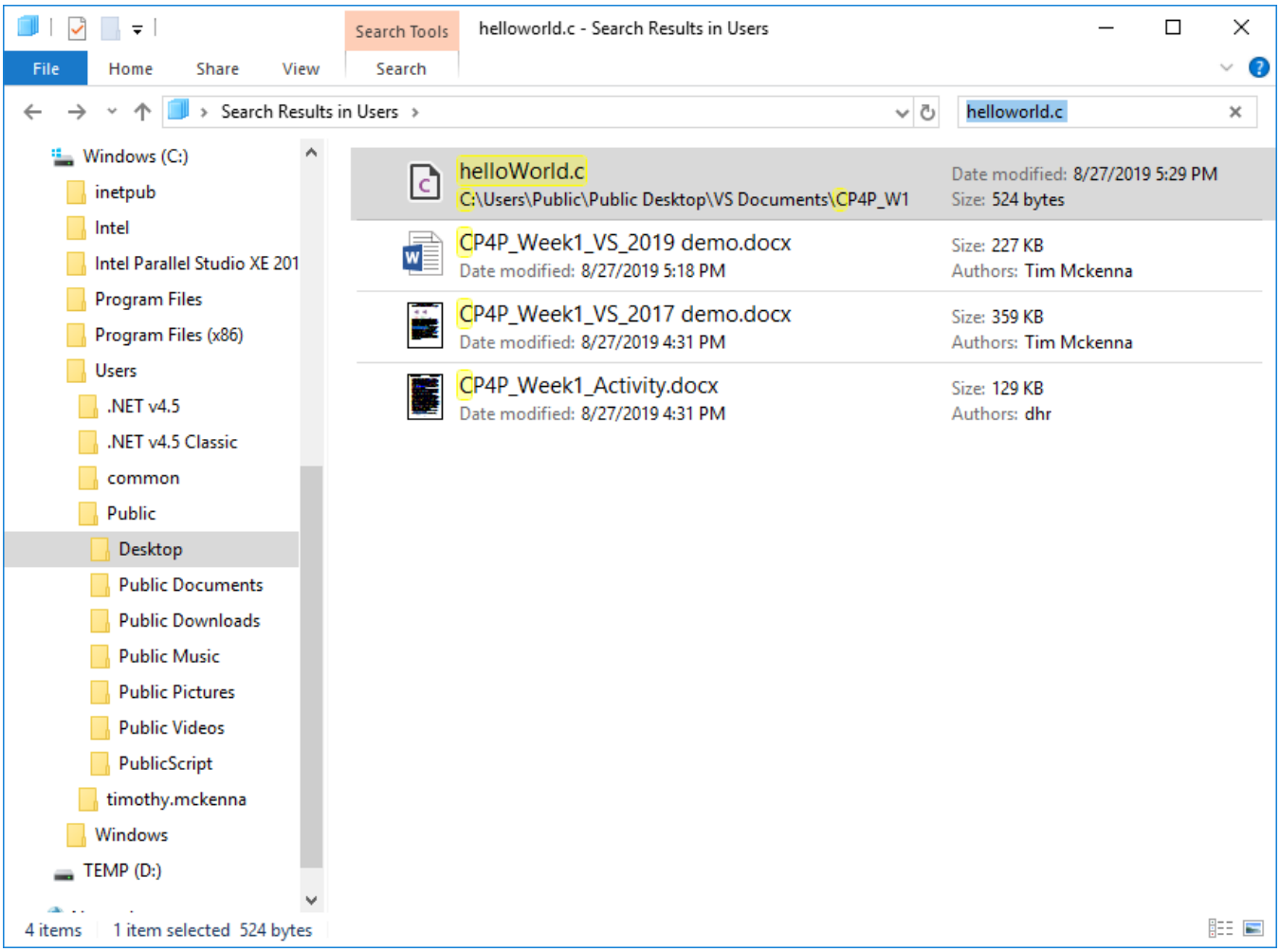
You can use Visual Studio:

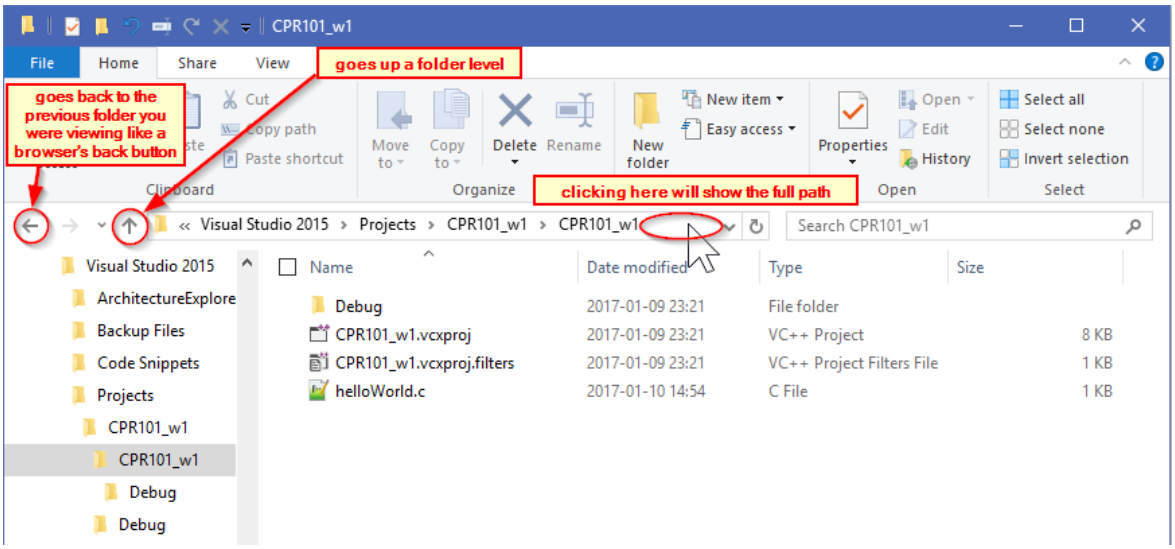
- Hover the mouse pointer over the **helloWorld.c** tab in the editing pane and the full path will appear.
- Right click on the **helloWorld.c** tab in the editing pane for more options.
- In the Solution Explorer pane, right click on the project name and select Open Folder in File Explorer and search from there.

You can use the Windows File Explorer (⊞ + E)

- File Explorer has a Search feature but if you search "This PC" for **file:helloWorld.c**, your patience will be tested. Before searching…

- Use the left pane to navigate to Windows (C:) and click on the Users folder
  - Then search for the helloWorld.c file





Test on Matrix

`matrix.senecacollege.ca` is a Linux cluster which is the host server for your completed C programs.

• First: a good place to start for most things at Seneca is at https://students.senecacollege.ca/ The Wiki Index to Pages and the search feature are useful in accessing or finding many Seneca resources.

- install GlobalProtect VPN (Virtual Private Network). Note the link for the version to use in China. Being signed on to the VPN and using multi-factor authentication is necessary for access to many systems at Seneca. The Wiki Index to Pages and the search input box at the top can be useful in finding and accessing many Seneca resources. Students must set up Microsoft Authenticator and enroll in MFA.
- *You must reconnect to the VPN every time you restart your computer*. This will likely require Multi-Factor Authentication (MFA).
- To use SSH/SFTP to matrix, users must be connected to the Seneca VPN.
- macOS users might save time by reading this first: https://tech.wayne.edu/kb/security/wsu-virtual-private-network/500214
- Troubleshooting

• Second: install a Secure SHell Telnet (SSH) client which is a terminal app to sign on and use a Unix/Linux server like Matrix.

- Windows -- install PuTTY using the MSI Package or Windows Terminal
- macOS -- use the built-in SSH Terminal (Finder, Go -> Utilities), or other options
- Linux -- use the preinstalled OpenSSHspan> terminal, or other options including PuTTY

• Third: install a Secure File Transfer Protocol (SFTP) GUI app for transferring files from your system to a server

- Windows -- install WinSCP. Its UI can launch PuTTY if it was previously installed.
- macOS / Linux / Windows -- install FileZilla for a cross-platform app very similar to WinSCP

Upload your C source code file to matrix via an SFTP application (Secure File Transfer Protocol) such as WinSCP or Filezilla. Next, compile and run the program on `matrix` using a terminal program with SSH (Secure SHell). Steps to do this follow below.

> Host name is `matrix.senecacollege.ca` on port 22 for SFTP and SSH terminal.
> Use your Seneca UserID and password as for other Seneca systems.

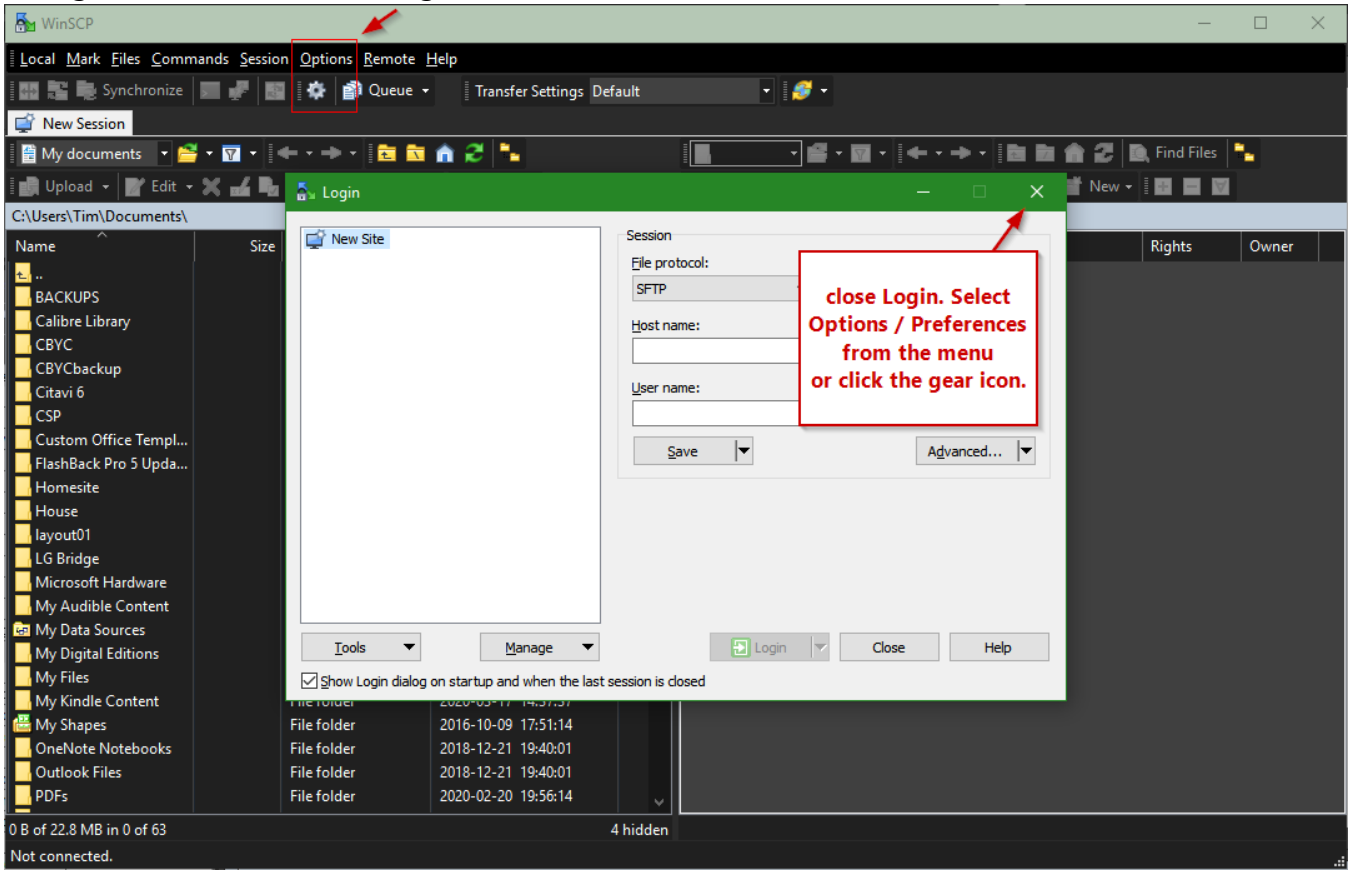
N.B. ensure you input your UserID in lower case. Unix/Linux/*nix regards 'a' and 'A' as different characters in UserIDs, directory/file names, and on the command line unlike many other Operating Systems (OS) which are case independent in those instances. Passwords are *always* case sensitive in any contemporary OS.

Transfer source via SFTP

Regardless of the SFTP client you use, it is CRITICAL that C source files are **Transferred** in **TEXT** mode. **Automatic** mode transfer works only if `*.c` and `*.cpp` file extensions are included in the FTP client's list of text file masks. There are small but important differences in the way different operating systems encode plain text files such as your .c source code files; use TEXT mode and FTP will take care of those differences. (End of Line & line termination) Transferring files in Binary mode *ignores* all differences and will cause you grief.

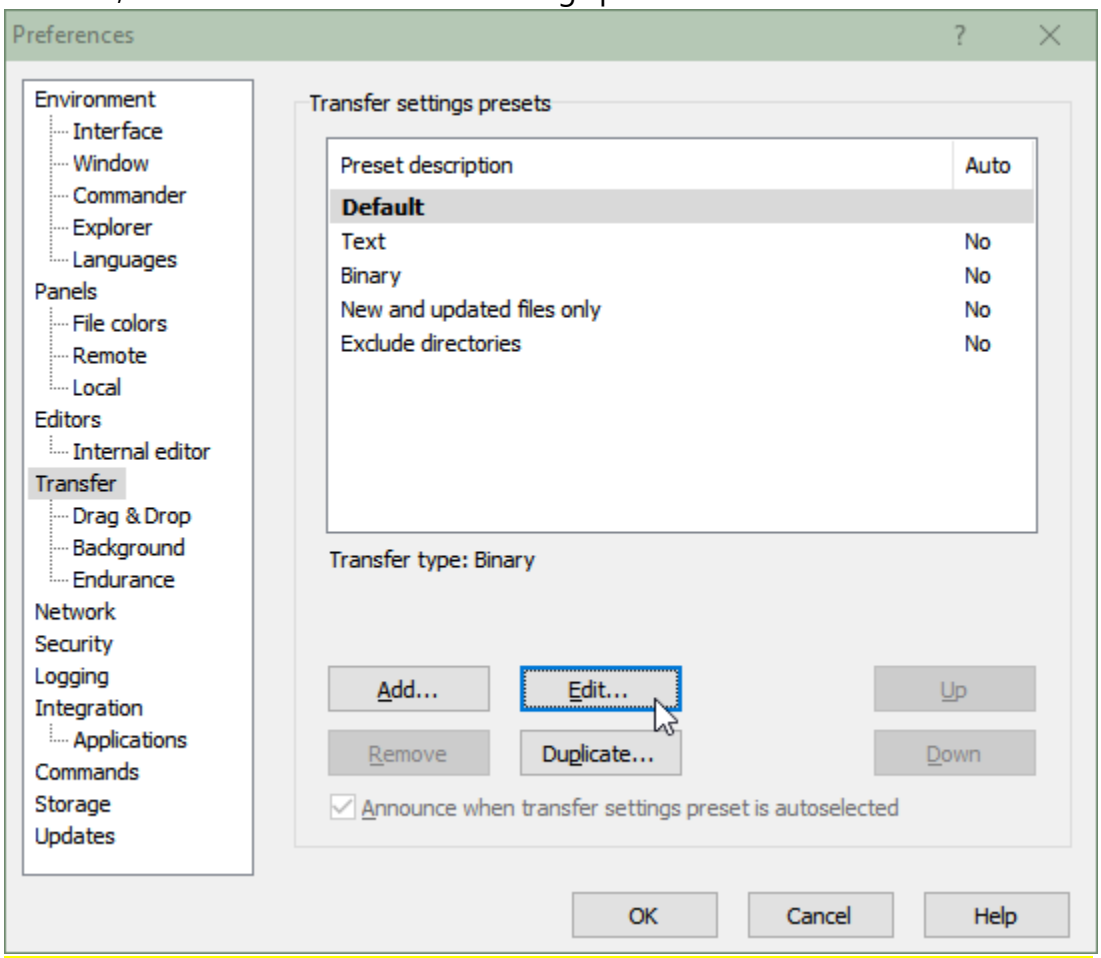
Configure WinSCP

Starting WinSCP for the first time will show this. Close the Login dialog and make the following changes to the WinSCP settings…



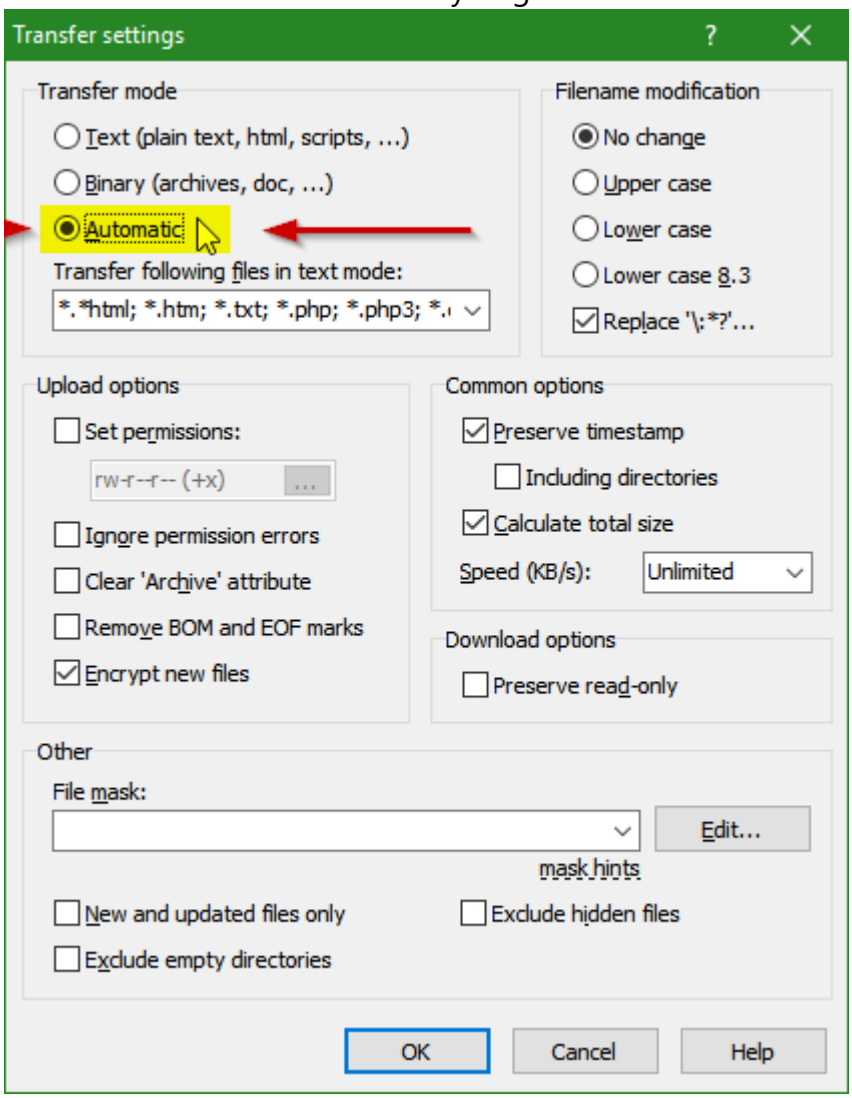
WinSCP Options / Preferences

- Transfer, Edit **Default** in 'Transfer settings presets'



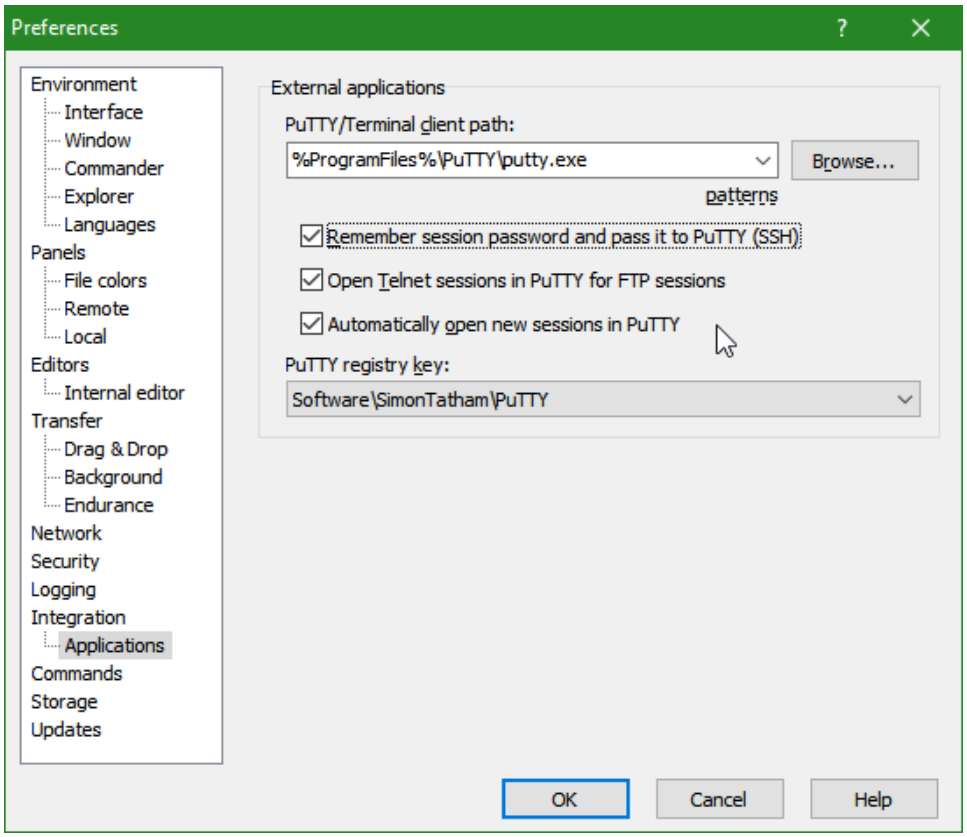
It is CRITICAL that you change the **Transfer** preset for **Default** to **Automatic**.
There are small but important differences in the way different operating systems encode plain text files such as your .c source code files. Transferring files in Binary mode *ignores*

those differences and will cause you grief.

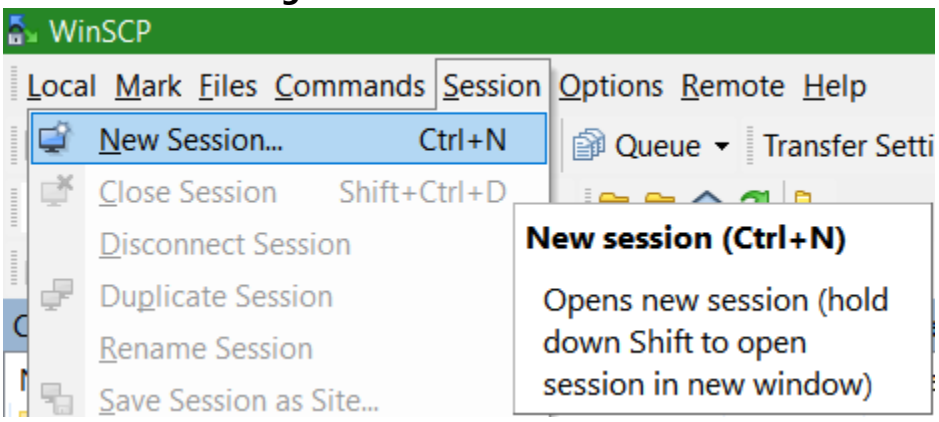


- WinSCP can open a PuTTY terminal window automatically when you open an SFTP session to transfer a source file. This will save you a step later. Change the following settings in Preferences / Integration / Applications. (WinSCP Integration with PuTTY

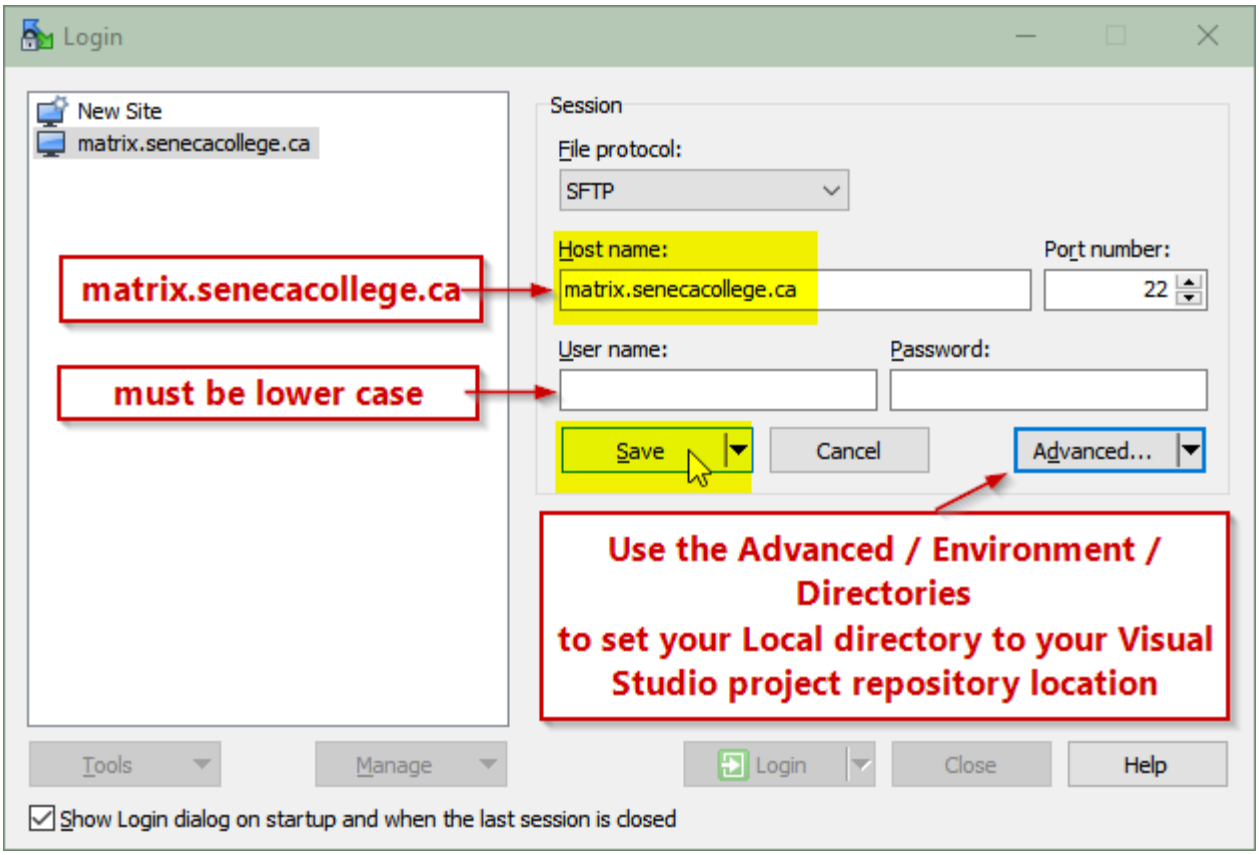
documentation)



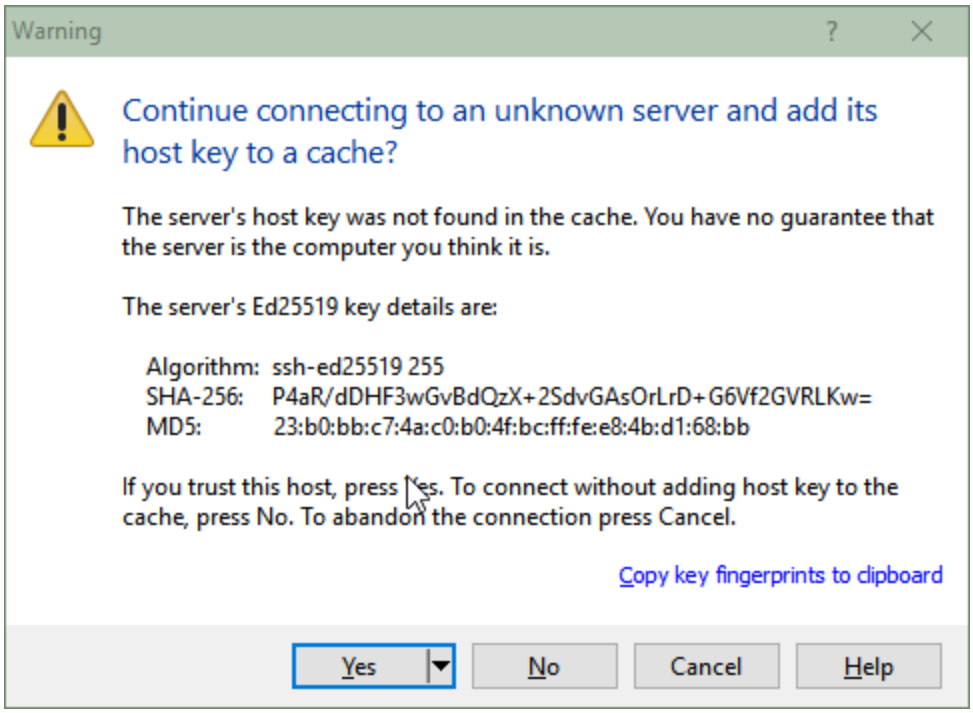
- Set up a New Session in the Site Manager for the Seneca server at **matrix.senecacollege.ca**



Ensure your Username is input in lower case.



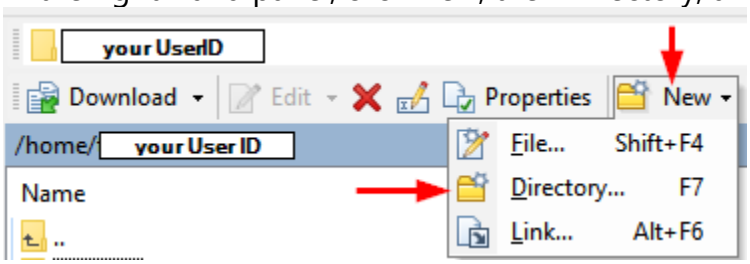
- Save the Session
- Ensure your Student VPN is installed and you are signed in.
- Select **matrix.senecacollege.ca** then click the Login button. You will see the following the first time you contact matrix. Accept the host key.



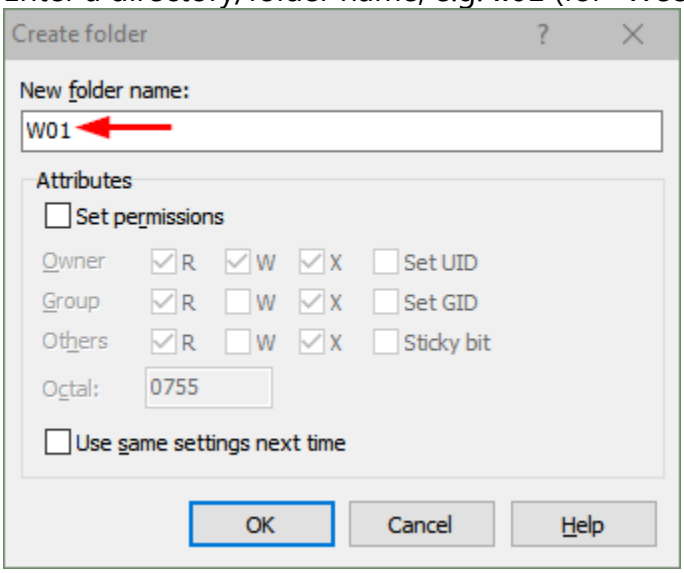
Transfer source via WinSCP

- From this point onward, you can click and drag your .c source file from the left to the right side. Consider first creating a directory on matrix to contain your .c files.

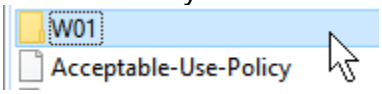
In the right-hand panel, click New, then Directory, and



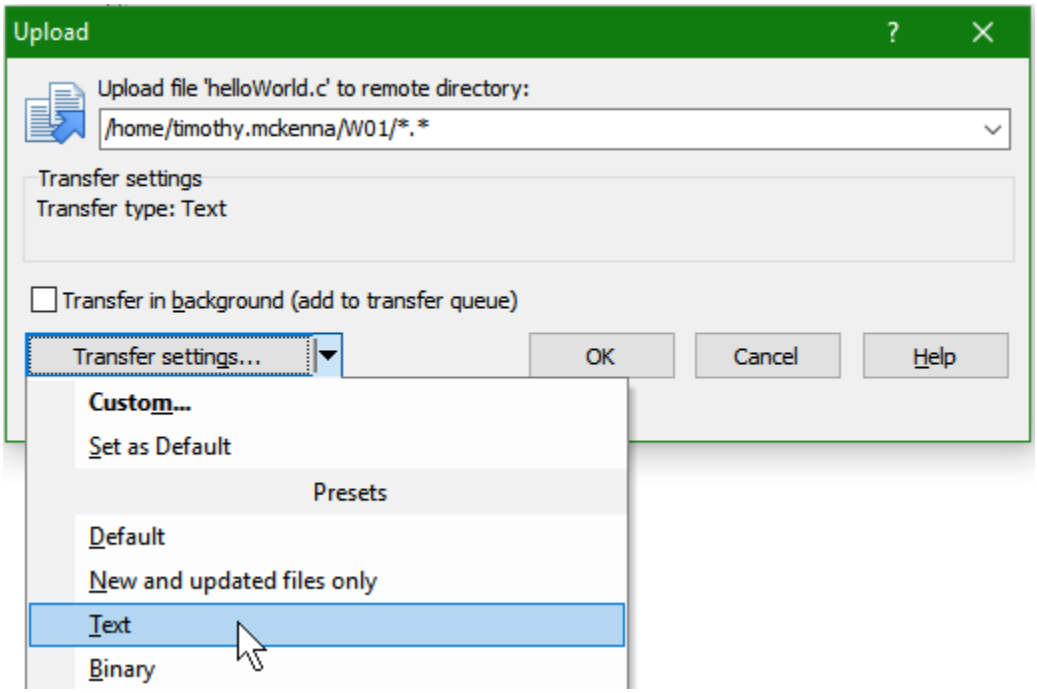
Enter a directory/folder name, e.g. `w01` (for "Week 1")



Double click your new directory/folder name to make it your current directory/folder



Find your `helloWorld.c` source file using the left panel, click on your source file and drag/drop it to the right panel. This will transfer your source file from your local computer to the directory named `w01` on the matrix server.
*Make sure the files are transferred in Text mode, **not** binary. See above notes on this.*

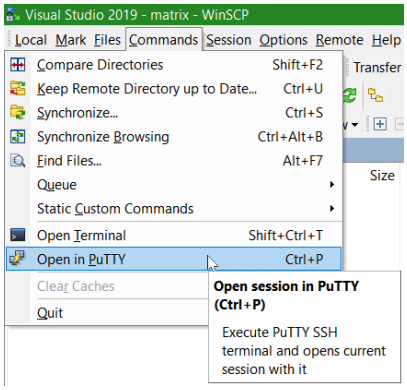


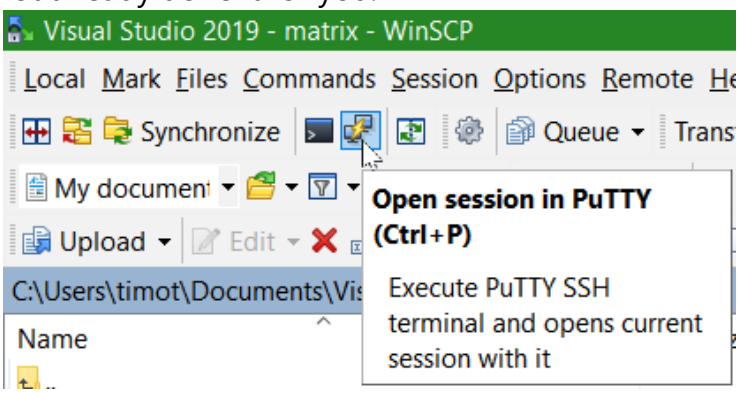
Double click on the file in the right-hand panel to see the file's contents. Close that window.

Compile and run on matrix

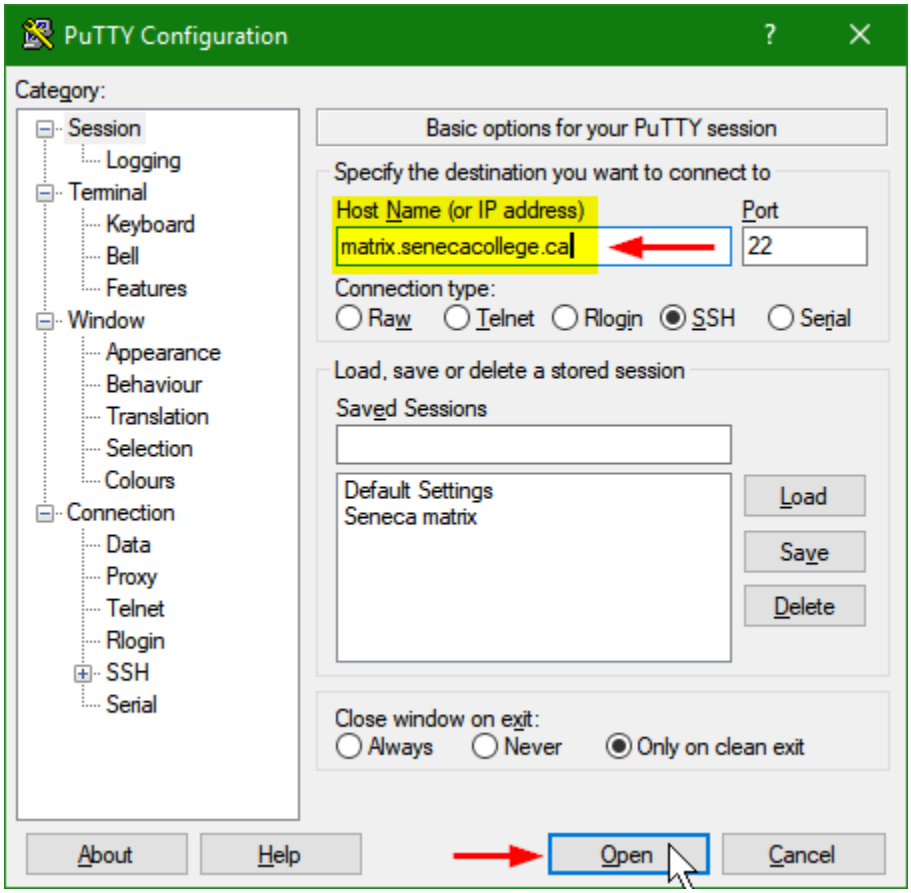
Compile and run your program on matrix using a traditional telnet terminal and command line interface.

Launch PuTTY if WinSCP has not already done it for you.





See the PuTTY documentation for details.



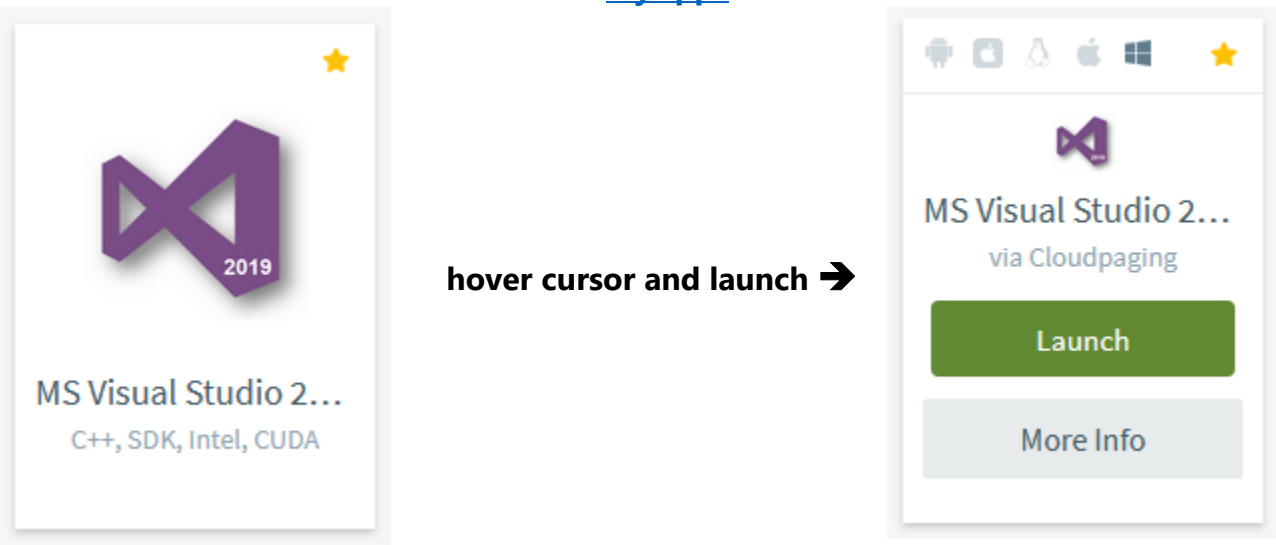
To customize PuTTY for the " WinSCP temporary session", launch PuTTY, open that Session, change settings in the Category tree (Window / Appearance / Font is a favourite to be changed), and remember to return to Session and Save.

```
login as: yourUserID
##########################################################################
# Welcome to Matrix
<snip>
# All activities on this system are governed by
# Seneca Information Technology Acceptable Use Policy
# For complete ITAU policy visit http://www.senecacollege.ca/policies/itau.html
##########################################################################
yourUserID@matrix.senecacollege.ca's password: yourPassword
Last login: Thu Sep 13 17:42:46 2018 from 174.93.86.94
[yourUserID@mtrx-node02pd ~]$ cd w01  change to the directory containing your source file
[yourUserID@mtrx-node02pd w01]$ ls    list the current directory–is your source file there?
helloWorld.c
[yourUserID@mtrx-node02pd w01]$ gcc helloWorld.c -o helloWorld    compile your source file
                                                                  with the gnu c compiler
[yourUserID@mtrx-node02pd w01]$ helloWorld                        run your program
Hello, World!
This is yourNameHere.
[yourUserID@mtrx-node02pd w01]$ logout        sign off
```

Seneca myApps

An application streaming service called MyApps is available. Some software, such as Visual Studio, requires a connection through StudentVPN to access the licensing server. (VPN = Virtual Private Network) All other software not needing the VPN is Open Source – do not bother with myApps and just install locally. A "cloudpaging player" client app must be loaded on your computer. There is significant communications overhead with this and should be used on your own PC if Visual Studio or other apps cannot be installed locally.

Someday, when the COVID-19 pandemic ends and we return to computer labs on campus, you will launch Visual Studio and WinSCP from **myApps** on a Seneca lab PC.



Troubleshooting myApps: first be patient, the app…is…being…downloaded…from the cloud. After running out of patience, try reloading the page. If your UserID will not validate or the Visual Studio item itself needs validation, visit the ITS Service Desk in the Learning Commons: there may be a problem with your user account's permissions.