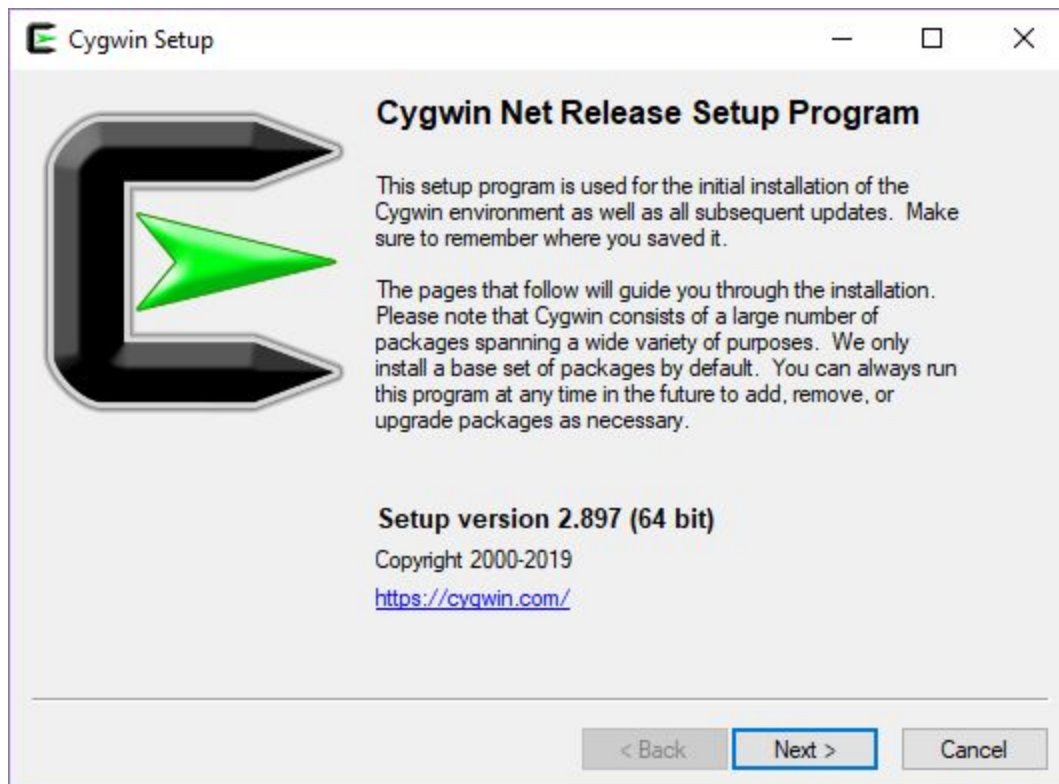


LW: Setting up your PC for CSCE 121

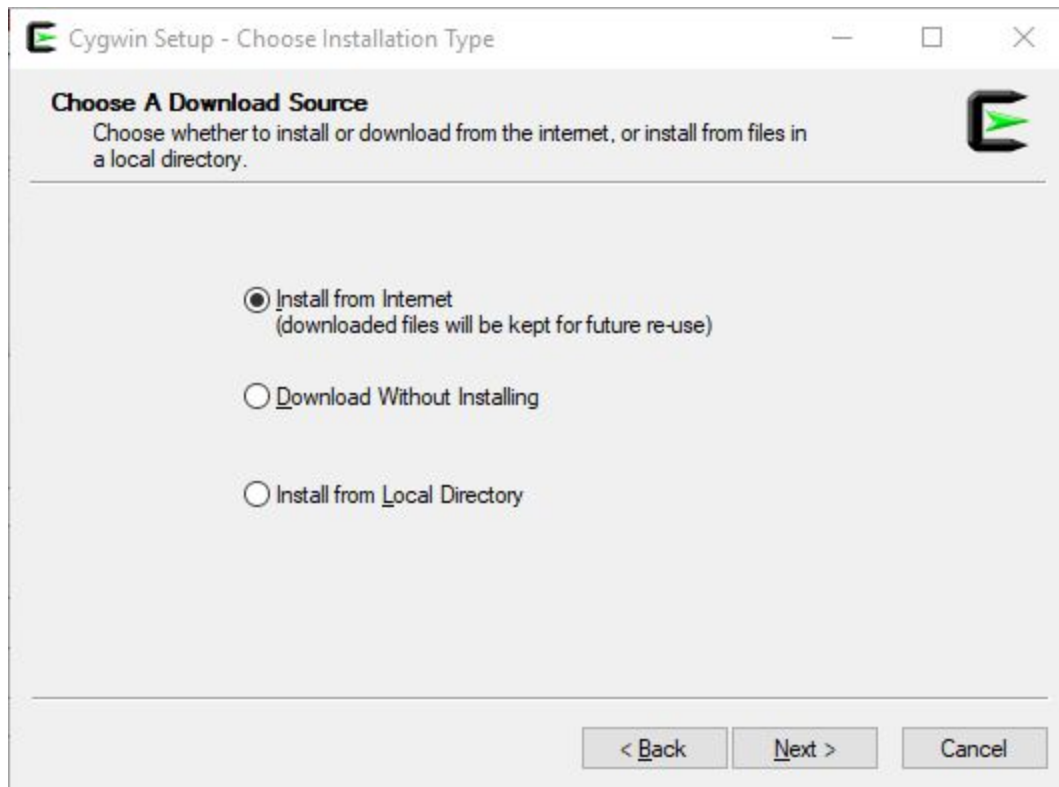
For the first week only, you can complete this labwork on your own and do not have to attend lab to get credit.

Installing the C++ Compiler

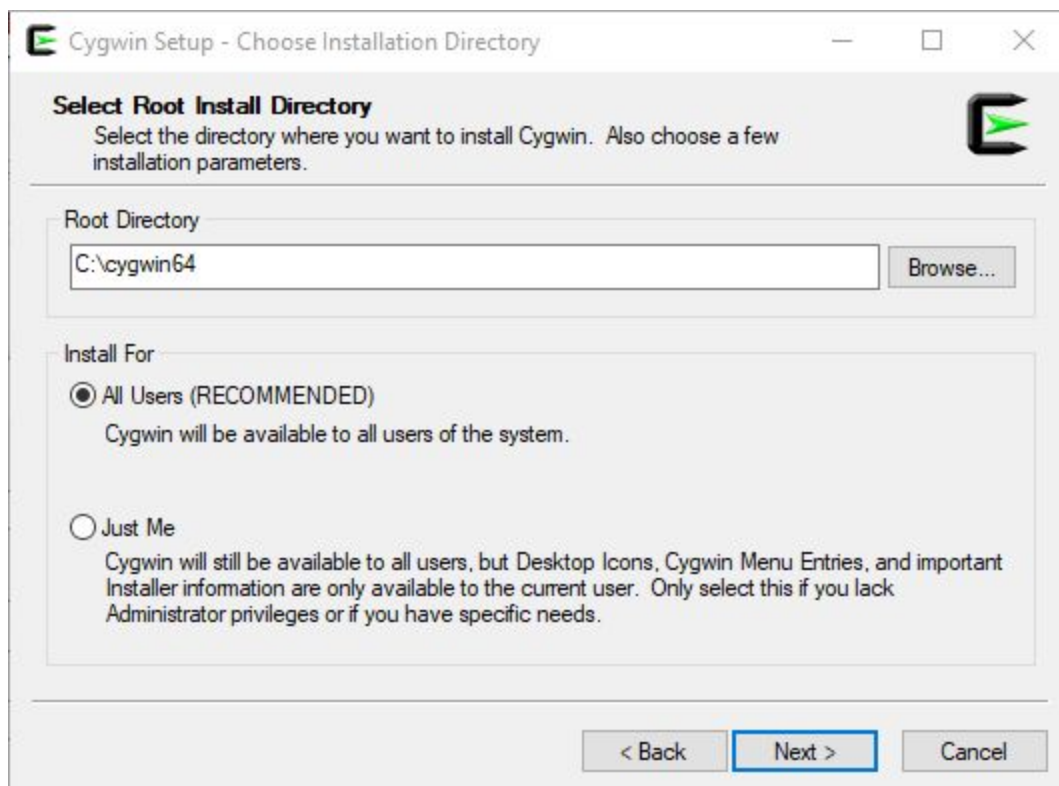
1. Download Cygwin setup
 - a. Go to <https://cygwin.com/>
 - b. Under “Installing Cygwin” follow the setup link for your computer to download the installer executable. You should try to use version (64 bit or 32 bit) that matches your computer's version of the Windows operating system.
 - i. [This link will help you figure out if you are using the 64- or 32-bit version of windows](https://support.microsoft.com/en-us/help/827218/how-to-determine-whether-a-computer-is-running-a-32-bit-version-or-64) (<https://support.microsoft.com/en-us/help/827218/how-to-determine-whether-a-computer-is-running-a-32-bit-version-or-64>).
 - c. Run the setup executable and select “Next”



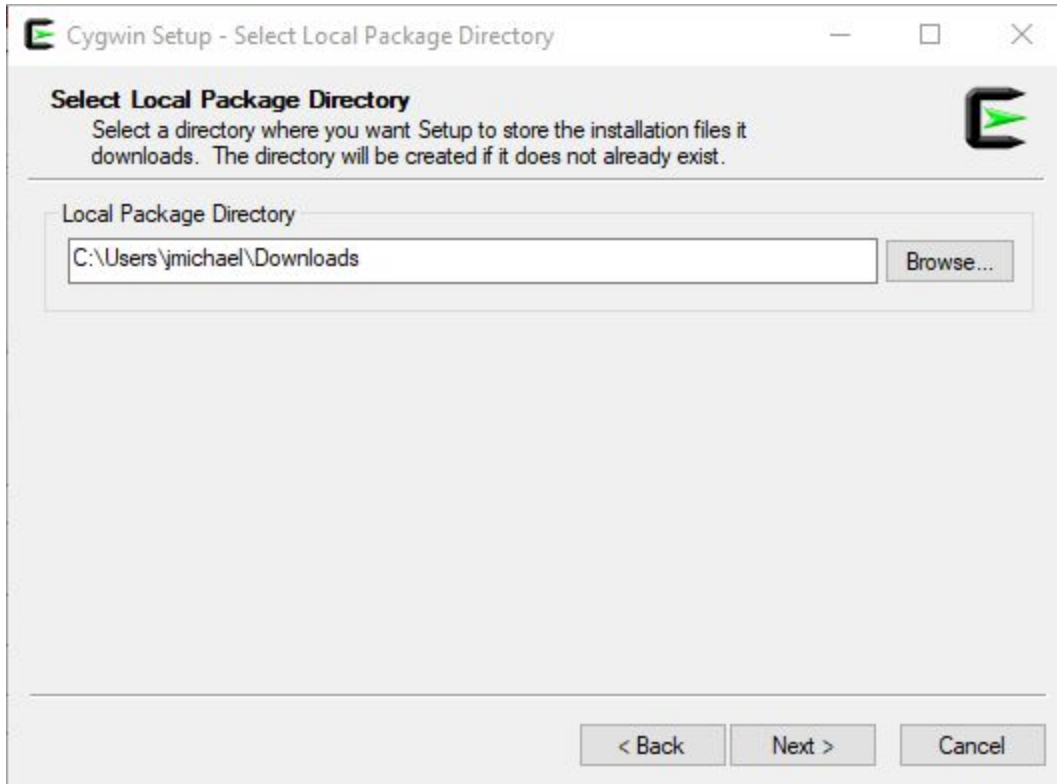
d. Select "Install from Internet" and then "Next"



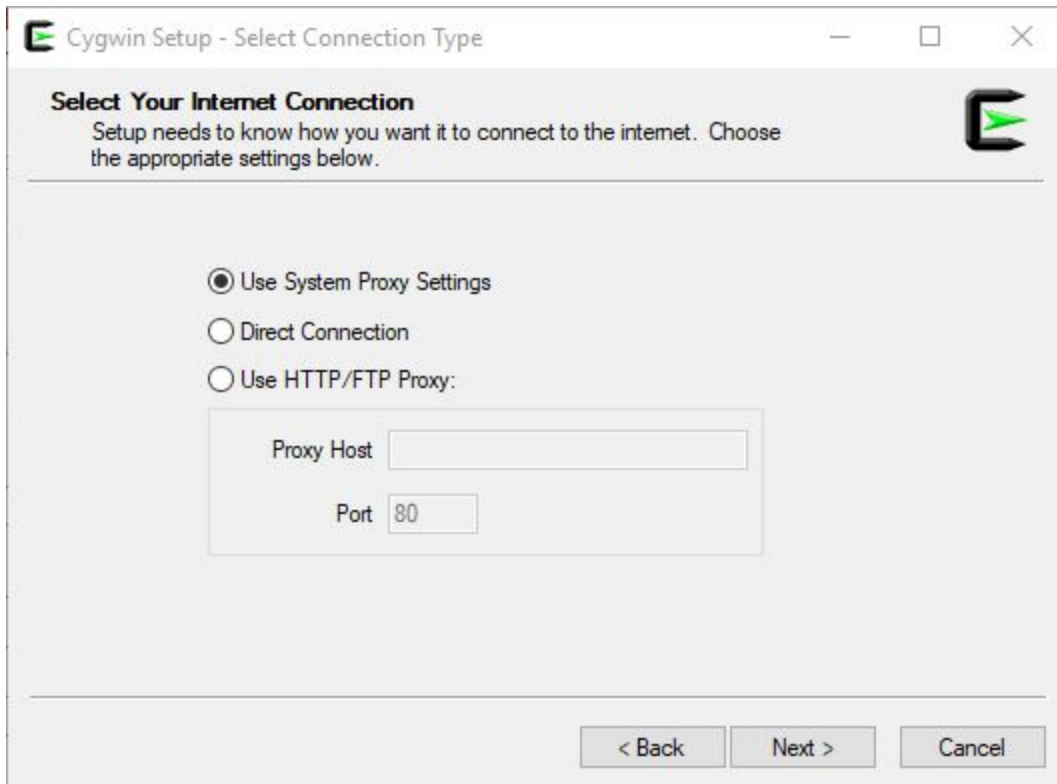
e. Accept the default directory. Choose what works for your situation under "Install For".



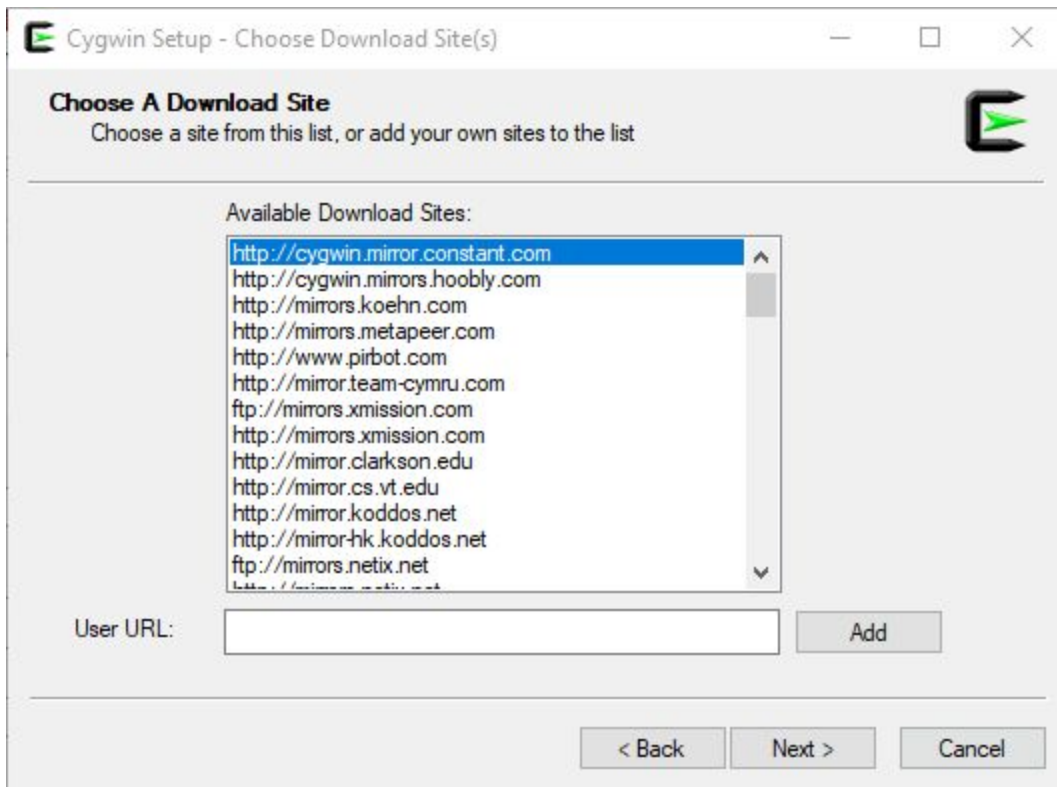
f. Use whatever directory you want and then “Next”.



g. Accept the default (Use System Proxy Settings) and then “Next”.



- h. Choose whatever download site you want. I would recommend using servers in the US (i.e. .com, .edu, and .net). Anything in other countries is likely to be slower.



- i. Search and select each of the following packages **before** clicking next.
Note: You just need bin, not src but feel free to get the src if you want to look at it.
Note: Use search to find them more easily.

i. **gcc-core**

Select Packages
Select packages to install

View Category Search gcc-core Clear Keep Best Sync Test

Package	Current	New	Bin?	Src?	Categories
All		Default			
Devel (5)		Default			
cygwin32-gcc-core		Skip	<input type="checkbox"/>	<input type="checkbox"/>	Devel
djgpp-gcc-core		Skip	<input type="checkbox"/>	<input type="checkbox"/>	Devel
gcc-core		7.4.0-1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Devel
mingw64-i686-gcc-core		Skip	<input type="checkbox"/>	<input type="checkbox"/>	Devel
mingw64-x86_64-gcc-core		Skip	<input type="checkbox"/>	<input type="checkbox"/>	Devel

ii. **gcc-g++**

Select Packages
Select packages to install

View Category Search gcc-g++ Clear Keep Best Sync Test

Package	Current	New	Bin?	Src?	Categories
All		Default			
Devel (5)		Default			
cygwin32-gcc-g++		Skip	<input type="checkbox"/>	<input type="checkbox"/>	Devel
djgpp-gcc-g++		Skip	<input type="checkbox"/>	<input type="checkbox"/>	Devel
gcc-g++		7.4.0-1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Devel
mingw64-i686-gcc-g++		Skip	<input type="checkbox"/>	<input type="checkbox"/>	Devel
mingw64-x86_64-gcc-g++		Skip	<input type="checkbox"/>	<input type="checkbox"/>	Devel

iii. **libgcc1**

Select Packages
Select packages to install

View Category Search libgcc1 Clear Keep Best Sync Test

Package	Current	New	Bin?	Src?	Categories
All		Default			
Libs (1)		Default			
libgcc1		7.4.0-1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Libs

iv. **gdb**

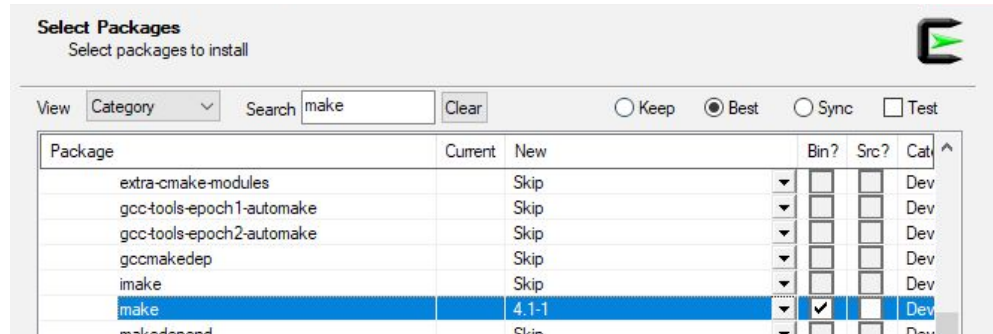
Select Packages
Select packages to install

View Category Search gdb Clear Keep Best Sync Test

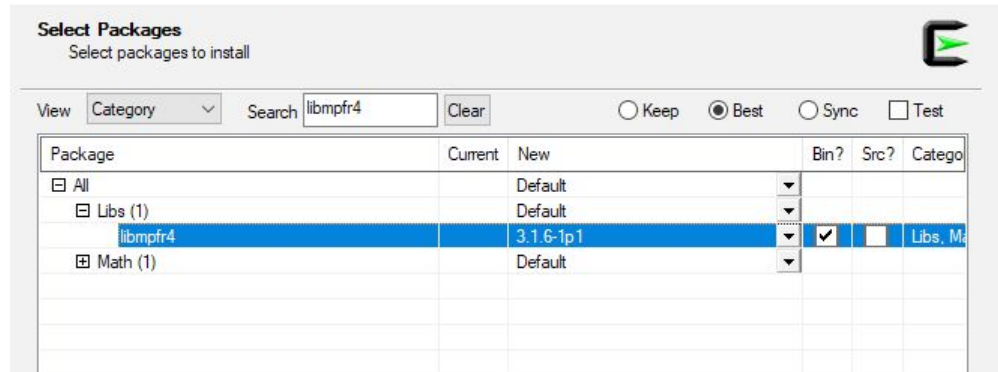
Search for this string in package names.

Package	Current	New	Bin?	Src?	Categories
All		Default			
Database (1)		Default			
Debug (3)		Default			
Devel (4)		Default			
cgdb		Skip	<input type="checkbox"/>	<input type="checkbox"/>	Devel
gdb		7.9.1-1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Devel

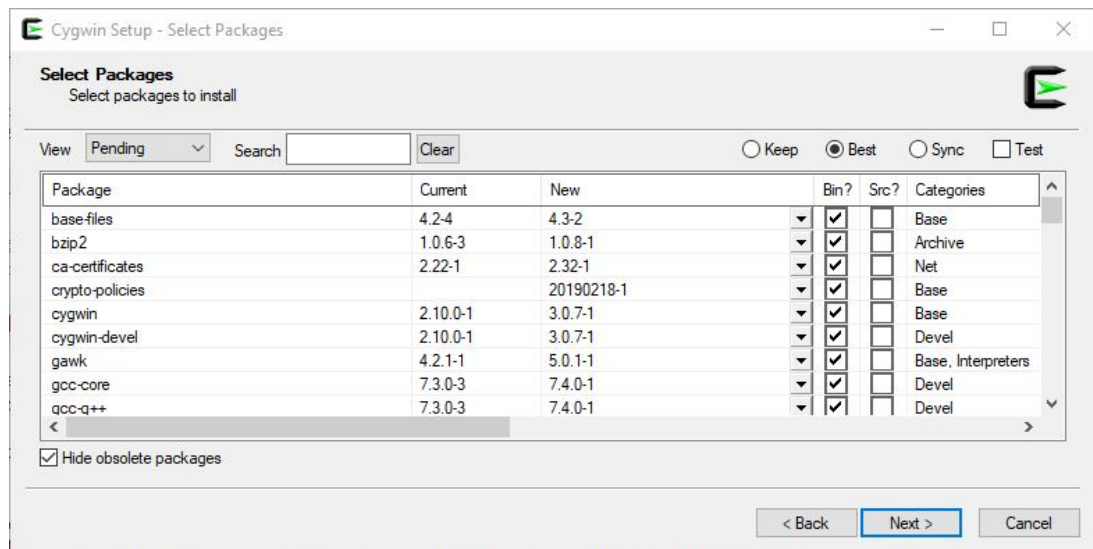
v. **make**



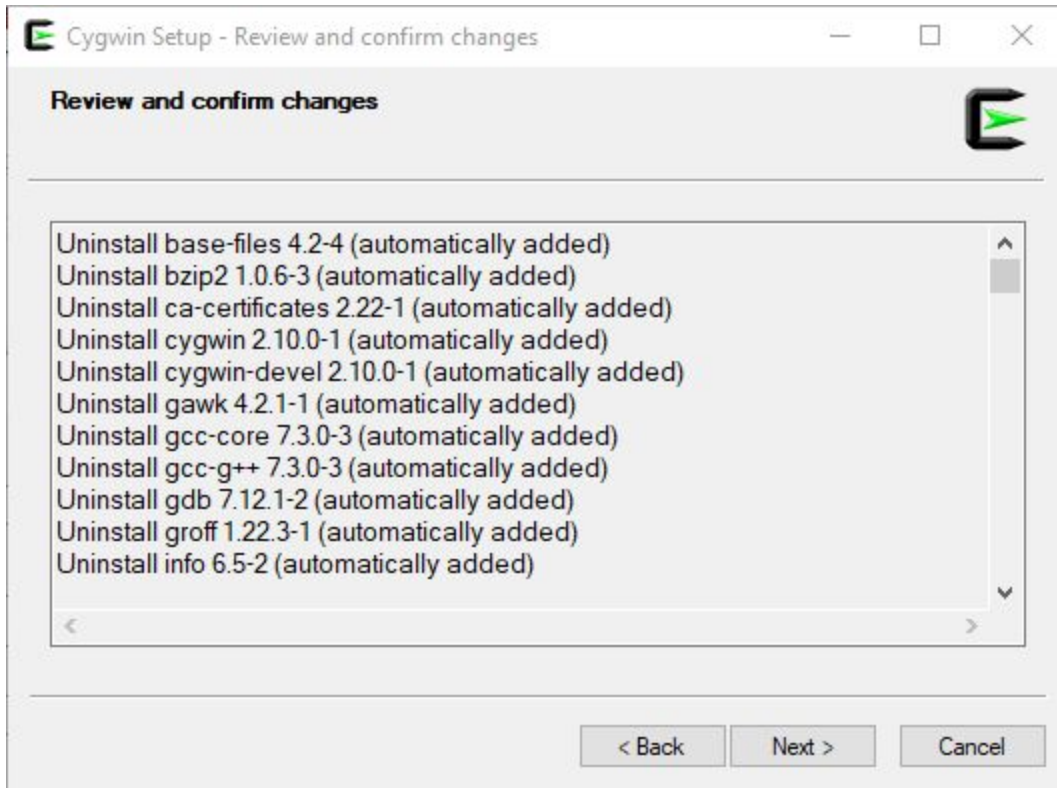
vi. **libmpfr4**



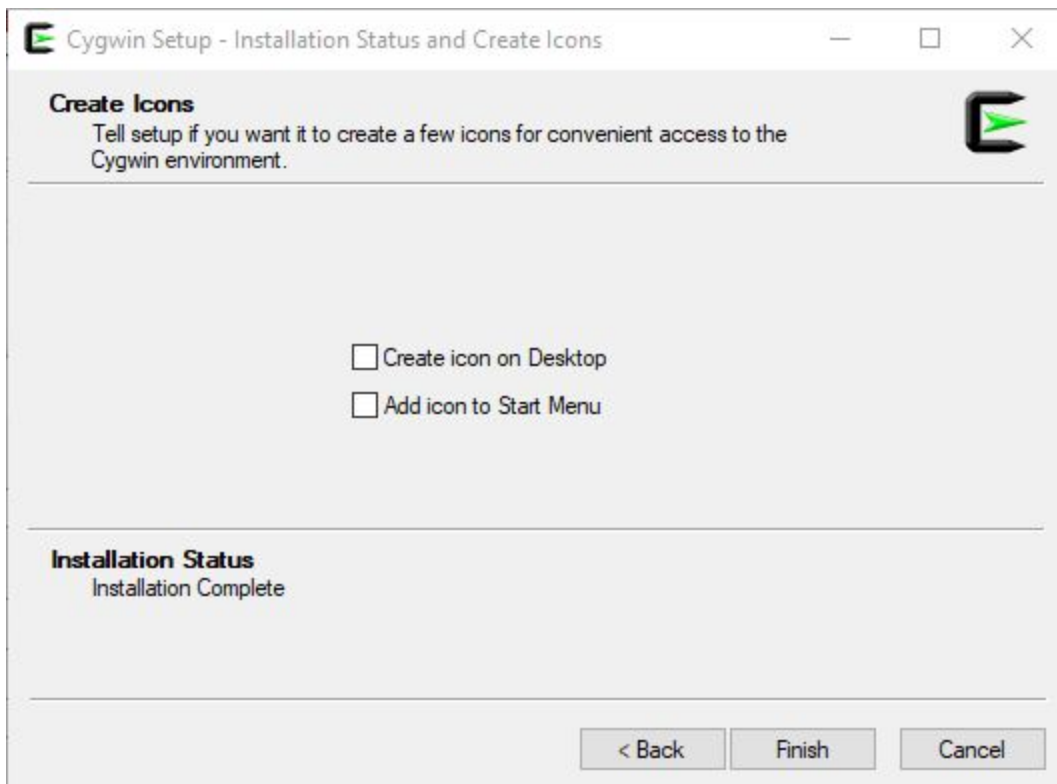
j. Then click next



- k. You probably won't get this window unless you've installed Cygwin before. This is uninstalling parts to replace with updated parts. Click "Next".



- l. Create icons if you want. Now you're done so choose "Finish"!



- m. Now you can run it. Either
 - i. Type “Cygwin” in the start bar in the lower left-hand corner of the screen.
 - ii. Open the icon you put on the desktop (if you did that.)
- n. You should now have a command-line interface ready for you to use to compile!

```
AUTH+jmichael@CSE-MOORE-NB ~  
$ |
```

Writing, Compiling, and Executing a C++ Program

1. Now that you’ve got the compiler installed, let’s go through the steps of writing a program that simply prints “Hello, World!” to the standard output.
2. Open your favorite text editor (I’m using [Notepad++](#)) and create a new file. If you don’t have a text editor installed, go to the “Text Editor Installation (optional)” section at the end of this document and install Notepad++.
3. Type the following into your new file:

```
#include <iostream>  
  
int main() {  
    std::cout << "Hello, World!" << std::endl;  
    return 0;  
}
```

4. Save this source file as `hello_world.cpp`.
 - a. Save to `C:\cygwin64\home\\lw\1\`
 - i. `<username>` is your username (i.e. mine is `pcr`)
 - ii. You will need to create the new folders “`lw`” and “`1`”.
5. Launch the Cygwin Terminal application from the Start Menu (⊞ Win, type `Cygwin`) and change directory to that containing `hello_world.cpp` (`cd lw/1/`).
 - a. The full path would be `/home/<username>/lw/1`

```
AUTH+pcr@CSE-RITCHEY-NB1 ~  
$ cd lw/1/  
  
AUTH+pcr@CSE-RITCHEY-NB1 ~/lw/1  
$ |
```


6. Once in that directory, verify that `hello_world.cpp` exists by listing the directory's contents (`ls`):

```
AUTH+pcr@CSE-RITCHEY-NB1 ~/lw/1
$ ls
hello_world.cpp

AUTH+pcr@CSE-RITCHEY-NB1 ~/lw/1
$ |
```

7. Compile and run:
 - a. Compile the code with `g++ -std=c++17 -o hello_world hello_world.cpp`

```
AUTH+pcr@CSE-RITCHEY-NB1 ~/lw/1
$ g++ -std=c++17 -o hello_world hello_world.cpp

AUTH+pcr@CSE-RITCHEY-NB1 ~/lw/1
$ |
```

- b. Run the code by executing `./hello_world.exe`
 - i. `./hello_world` also works

```
AUTH+pcr@CSE-RITCHEY-NB1 ~/lw/1
$ ./hello_world.exe
Hello, world!

AUTH+pcr@CSE-RITCHEY-NB1 ~/lw/1
$ |
```

Labwork Acknowledgment

After completing the lab, fill out and submit the [LW 1 Acknowledgment form](https://forms.gle/AsTiajrNaUh6zJePA):
<https://forms.gle/AsTiajrNaUh6zJePA> You must be logged in to your @tamU Google account.

Text Editor Installation (optional)

You will want a text editor with syntax highlighting. Although it is not strictly necessary, it really makes reading code much easier. We prefer to use [Notepad++](#). It is very easy to use. You should avoid using the Notepad editor that is built into Windows.

There are other options available that are popular with many programmers and work on multiple platforms. However, these may not be as easy to use initially.

- [Sublime](#)
- [Atom](#)
- [Visual Studio Code](#)

For the brave souls out there, you can use vim in Cygwin. If you decide to go that route, you're on your own. You can also install nano (during setup) for a gentler (but much more limited) text-editing experience.