

# THE ULTIMATE CODING ENVIRONMENT TUTORIAL

---

So we're all basically remote now, you're a computer science/engineering major whose labs use Netbeans, or you just want a solid, reliable coding environment. But there's so many options!!! What's recommended??? Online compiler?? Netbeans IDE?? Command line? Well, that's up to you. **I highly recommend taking the time to set up a good coding environment** because not only will it make life easier the rest of the semester, but your entire career possibly, and you'll probably have to go through the trouble of a tedious environment setup at some point in your life (whether that be for a class, internship, job, whatever). And you get experience with some cool IT skills!!

This PDF is divided into sections, which you can directly go to by clicking on the table of contents below. **The difficulty is a function of how many steps/how tedious the setup is,** NOT how difficult it is to necessarily use the said coding environment.

for any comments, concerns, or suggestions for this pdf, contact [rghosh@uark.edu](mailto:rghosh@uark.edu).

---

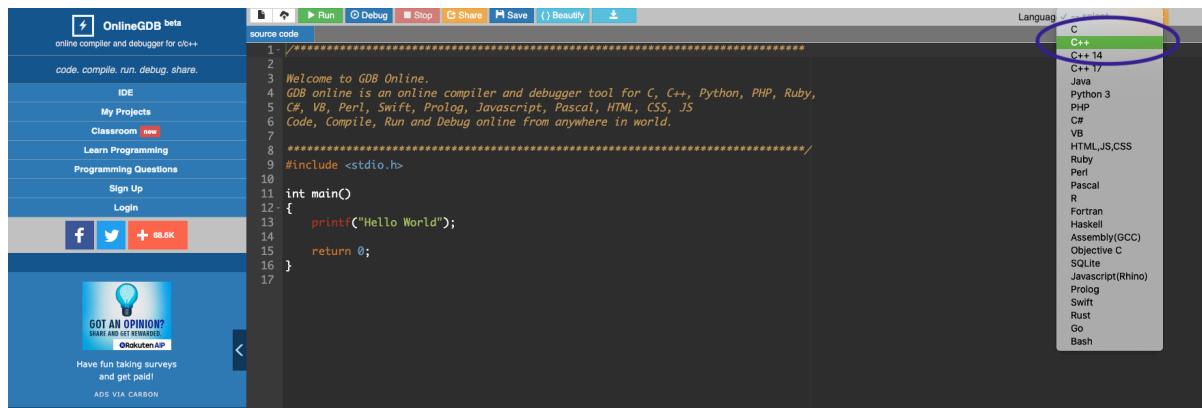
ONLINEGDB (EASY).	2
CITRIX WORKSPACE (MEDIUM).	4
NETBEANS IDE (HARDER).	6
mac os installation.	6
windows os installation.	8
using netbeans.	11
COMMAND LINE (HARDER).	14
for mac os.	14
for windows os.	16

# onlinergdb (easy).

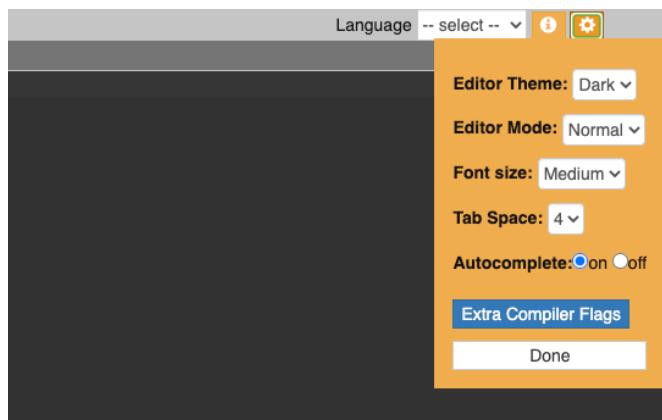
an online IDE basically that requires no installation at all.

## steps.

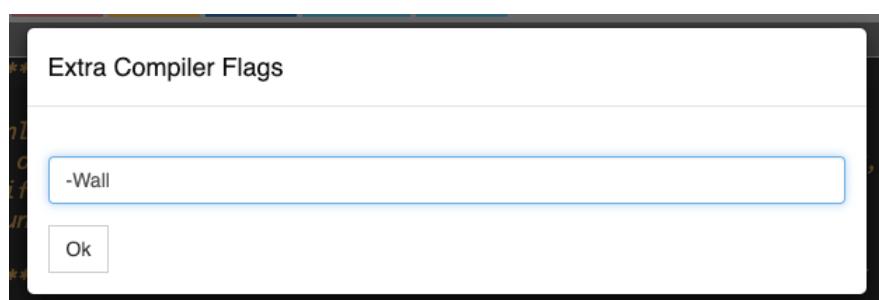
1. navigate to <https://www.onlinergdb.com/>.
2. click the selection box beside “languages” in the top right corner, and select “c++” as shown below:



3. **IMPORTANT!!!** before beginning to code, you must set an extra warning flag to make sure all warnings are caught!!! click the little settings gear to the right of where you selected:



4. click “extra compiler flags”. type in “-Wall” in the little pop up box, and click “ok”.



5. and that is it! now you are all set to use onlinegdb. you can create an account to save files online, or just download them and re-upload them every time you get to coding.

## **extras.**

for those of you curious, the extra -Wall compiler flag catches issues such as a non-void function that doesn't have a return value, and prints the warning to the console. if you did not have this flag, that error could possibly go unnoticed and ultimately break your program!! not good ):

# citrix workspace (medium).

---

a windows virtual desktop provided by our department that has all our csce applications! it has netbeans, quartus (used in digital design), vs code, a compiler installed — the whole deal. you can easily transfer files between your actual physical computer and the virtual machine even! once you get this installed, you have everrryyyything!

## steps.

1. download the citrix receiver. this page has direct links, as well as helpful videos demonstrating all the steps covered in this section:

<https://industrial-engineering.uark.edu/technical-support/it-faq.php>

2. when installing/downloading, **accept all defaults** & do **NOT** add an account or anything!
3. once successfully downloaded & installed, **navigate to this address** in a web browser (i use chrome):

<https://ctx-storefront.uark.edu>

★ **note:** bookmark/favorite this page if you will be using the remote desktop frequently!  
You will be bring it up from this page every time

4. log in with uark credentials (note: for username, do **NOT** include the @uark.edu part)  
★ **note:** this took me a couple tries to get it to load the page once i logged in; i had to clear my cache and navigate to the link in step 2 again, and log in again in order to work
  5. click "**detect receiver**"
  6. allow your web browser to open/detect the application in the citrix application you installed in step 1
  7. in your browser, at the top, click "**Desktops**"
  8. click **CSCE computer lab**, and it should launch the citrix virtual desktop, automatically log you in, and you should be in!
- ★ to continue to access the virtual desktop, only steps 3-8 need to be repeated

## important info!

**Note:** be sure to allow read & write access – that way, you can save/transfer things between the virtual desktop & your actual computer! if you open the file explorer, you will see your physical machine's local disks now appear.

**Note:** do NOT save things to the C drive (it gets wiped every time u log off the remote desktop!) save it directly under a This PC location (for example, Desktop, or Documents) in order to access it again the next time u log on

## extras.

for a more visual walkthrough, visit:

<https://industrial-engineering.uark.edu/technical-support/it-faq.php>

# netbeans ide (harder).

a free IDE you can install for c++ projects — installation can be tricky!. but it's always a good time to learn how to install one and get used to how they work!

## MAC OS INSTALLATION.

### preliminary step (jdk download).

**note:** you can skip this step if you already have the java developer kit installed.

1. download the java developer kit (jdk 8) from the following site:

<https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>

★ note: for some reason on mac os, i had to download jdk version 8 specifically!! the newest release did not work for me!!

2. you will have to create an account for the jdk 8 ): i just entered some random inputs for the technical details:

**Create Your Oracle Account**

Already have an Oracle Account? [Sign In](#)

Email Address \*  Your email address is your username.  
✓ Confirmation: We will email a confirmation to you

Password \*  Passwords must have upper and lower case letters, at least 1 number, not match or contain email, and be at least 8 characters long.  
✓ Confirmation: Password meets requirements

Retype password \*  ✓

Country \*  ✓

Name \*   ✓

Job Title \*  ✓

Work Phone \*  ✓

Company Name \*  ✓

Address \*  ✓

City \*  ✓

State/Province \*  ✓

ZIP/Postal Code \*  ✓

You may opt-out of all marketing communications: [Unsubscribe](#).

3. the download should then automatically start. install it.

# netbeans steps.

1. download netbeans 8.2 from the following link:

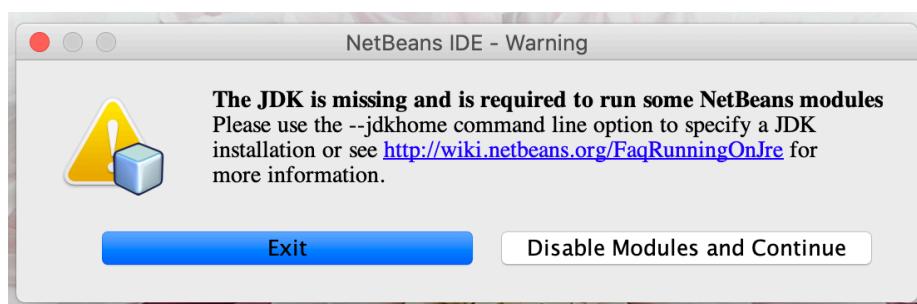
<https://netbeans.org/downloads/8.2/rc/>

★ note: it MUST be netbeans 8.2, as this is the easiest version to download that supports C++!

2. download from the C++ column, as shown below:

Supported technologies *	Java SE	Java EE	HTML5/JavaScript	PHP	C/C++	All
NetBeans Platform SDK	•	•				•
Java SE	•	•				•
Java FX	•	•				•
Java EE		•				•
Java ME						—
HTML5/JavaScript		•	•	•		•
PHP			•	•		•
C/C++					•	•
Groovy						•
Java Card™ 3 Connected						—
Bundled servers						•
GlassFish Server Open Source Edition 4.1.1		•				•
Apache Tomcat 8.0.27	•					•

3. (YOU ARE DONE IF YOU GOT NO ERRORS! stop here.) if your installation is seamless, great! you're good to go. but i got an error message:



4. click "exit." the installation should be successful. now comes the fun part!!!
5. the issue is netbeans is not finding the correct location of our jdk. we will need to go in and redirect it.
6. so first step: locate the location of that jdk 8! go to mac's **finder**.

7. in the top toolbar, click "**go**" > "**go to folder...**"
8. type "**/Library**" (the L needs to be capital!!!)
9. now go to "**Java**" > "**JavaVirtualMachines**" > "**jdk1.8.0\_261.jdk**" > "**Contents**"
10. right-click the "**Home**" folder
11. in the right-click menu, **press and hold the option key** on your mac and click "**Copy "Home" as Pathname**"
12. now go to your applications. find the netbeans folder, and **right-click the netbeans app**
13. click "**Show Package Contents**"
14. now go to "**Contents**" > "**Resources**" > "**NetBeans**" > "**etc**"
15. right-click "**netbeans.conf**" and open withTextEdit
16. scroll down to the bottom. here you will see something like:  

```
netbeans_jdkhome=" [ there's some file path here ] "
```

**replace (paste) the jdk's path we copied in step 11** inside the quotations. the correct version should look something like:

```
netbeans_jdkhome="/Library/Java/JavaVirtualMachines/jdk1.8.0_261.jdk/Contents/Home"
```
17. and exit. now start netbeans – it should be working!!!

## WINDOWS OS INSTALLATION.

---

### preliminary steps (jdk + mingw).

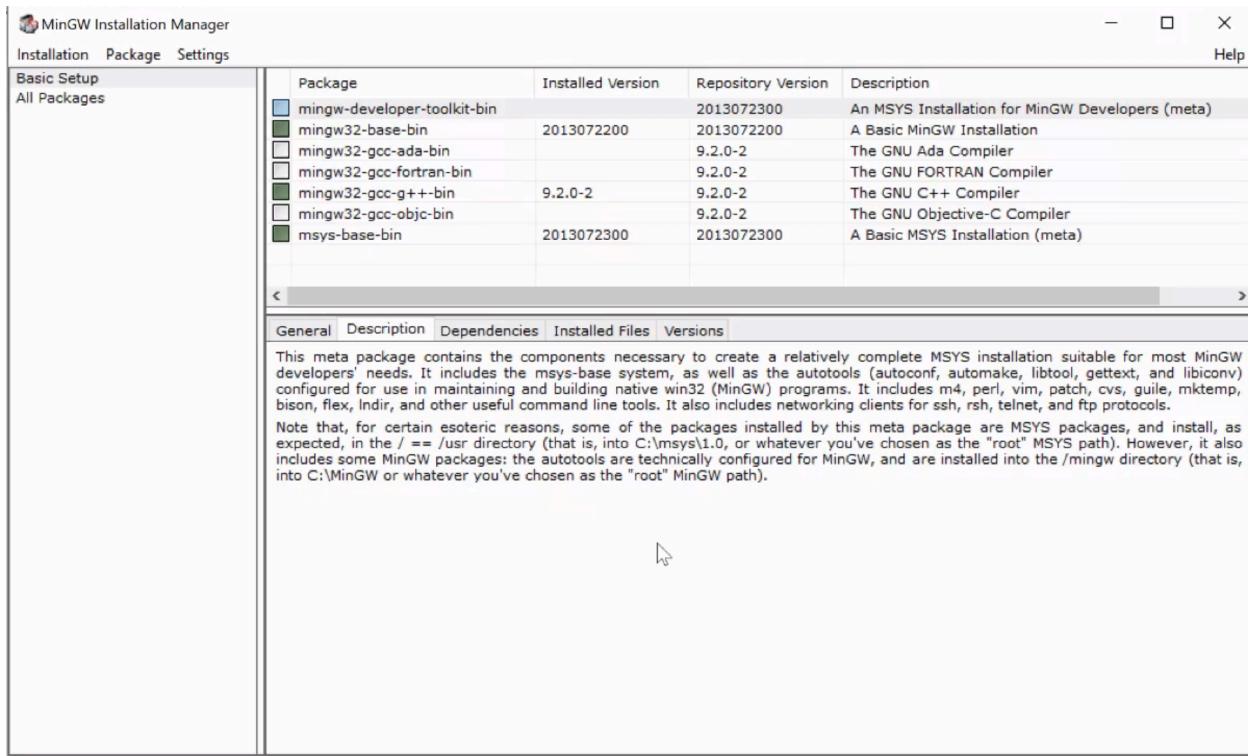
**note:** you can skip this step if you already have the java developer kit and mingw installed.

1. download and install the latest java developer kit from the following site:  
<https://www.oracle.com/java/technologies/javase-jdk14-downloads.html>
2. for windows, we also have to download & install a c++ compiler called mingw. you can do so from here:  
<https://osdn.net/projects/mingw/releases/>
3. click the blue button with the windows icon that has "**mingw-get-setup.exe**" next to it (about halfway down the page):



**mingw-get-setup.exe** (Date: 2017-09-06, Size: 91.00 KB)

4. download & install.
5. once installed, a window like this pops up:



you want to mark the boxes as shown above for installation. specifically:

- ★ mingw32-base-bin
- ★ mingw32-gcc-g++-bin
- ★ msys-base-bin

18. now click "**installation**" (in the top left corner) > "**apply changes**"

19. it might take a minute, but once it's done, you're good to go!

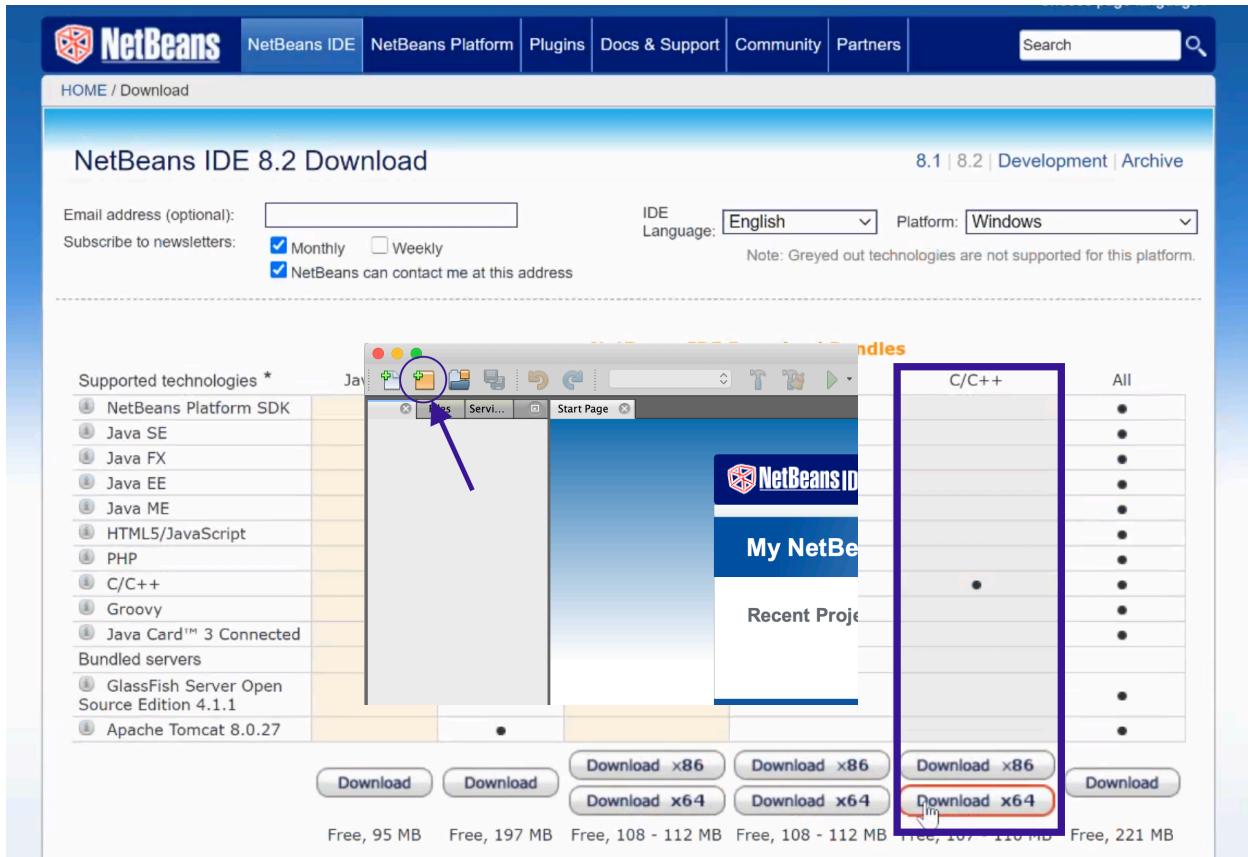
## netbeans steps.

1. download netbeans 8.2 from the following link:

<https://netbeans.org/downloads/8.2/rc/>

★ **note:** it MUST be netbeans 8.2, as this is the easiest version to download that supports c++!

2. download from the c++ column, as shown below:

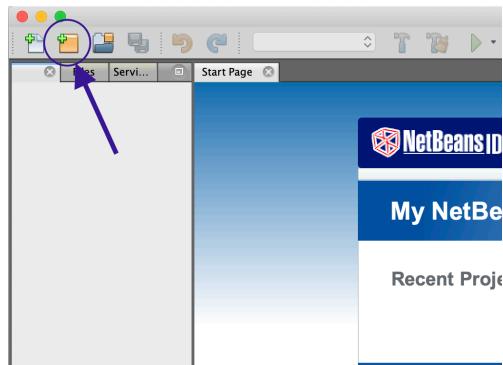


3. and once that is installed & downloaded, you should be good to go!

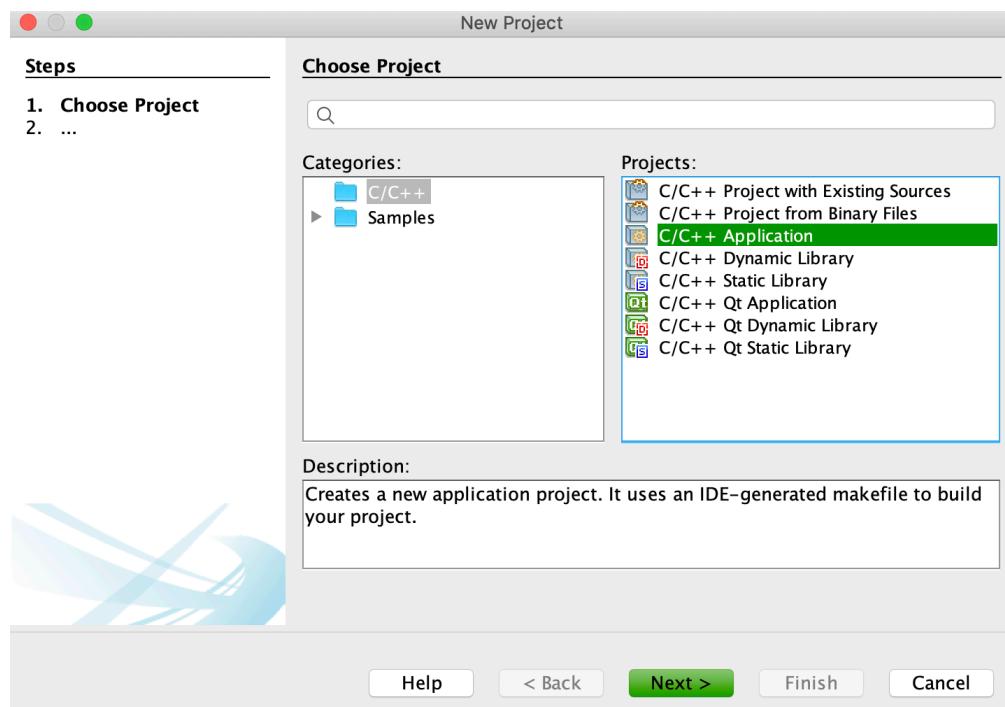
# USING NETBEANS.

## steps.

1. launch netbeans.
2. click the orange folder at the top right with the little plus sign (create new project):

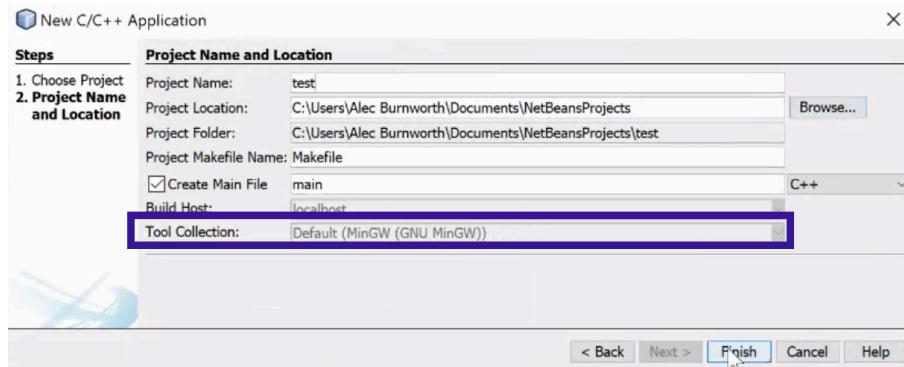


3. under categories, select "**c/c++**". under projects, select "**c/c++ application**"

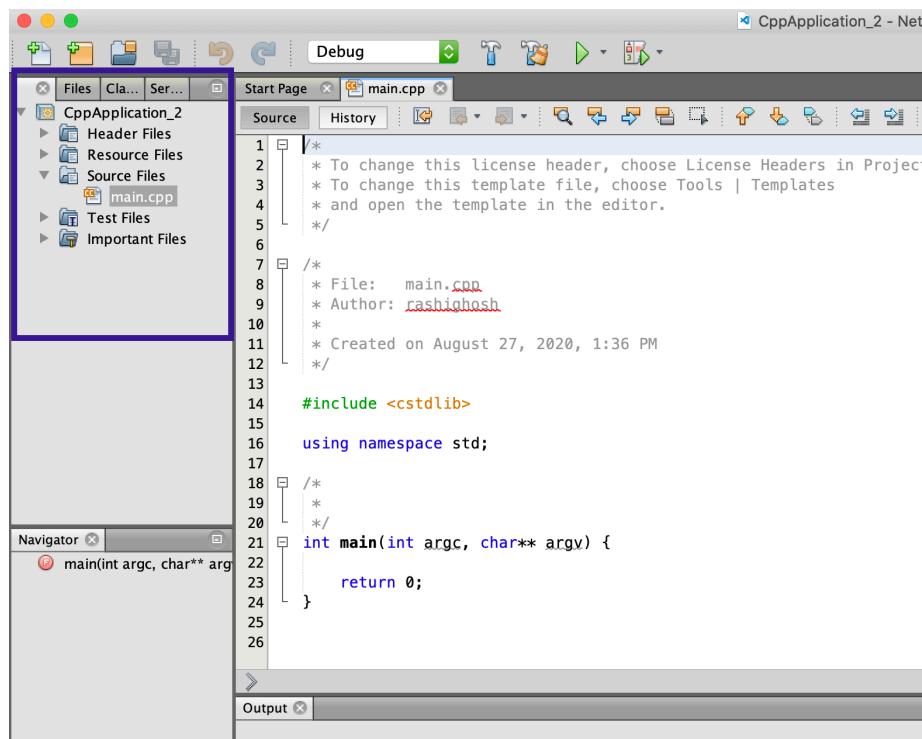


4. click next. on the following window, be sure to **change project location** to place on your computer where you can easily find it, and click finish

★ note: for windows, make sure the Tool Collection displays the following (if not, something went wrong in the mingw installation):



5. in the left panel, you should see your application. expand "source files", and you should see a "main.cpp". double click that, and start coding! (shown below)



6. once you have written your program, click the green play button (to the right of the "Debug" dropdown) to compile & run the program. (if you get any errors about a debugger tool missing at any point, just exit and don't worry about it)



# command line (harder).

---

depending on your os, either terminal or command prompt. you learn some nifty linux commands from this, which is something you could even put on a resume. this requires a nice clean text editor (of your choice, although i do have a recommendation).

FOR MAC OS.

---

## preliminary step (a good text editor).

**note:** you can skip this step if you already have a preferred text editor

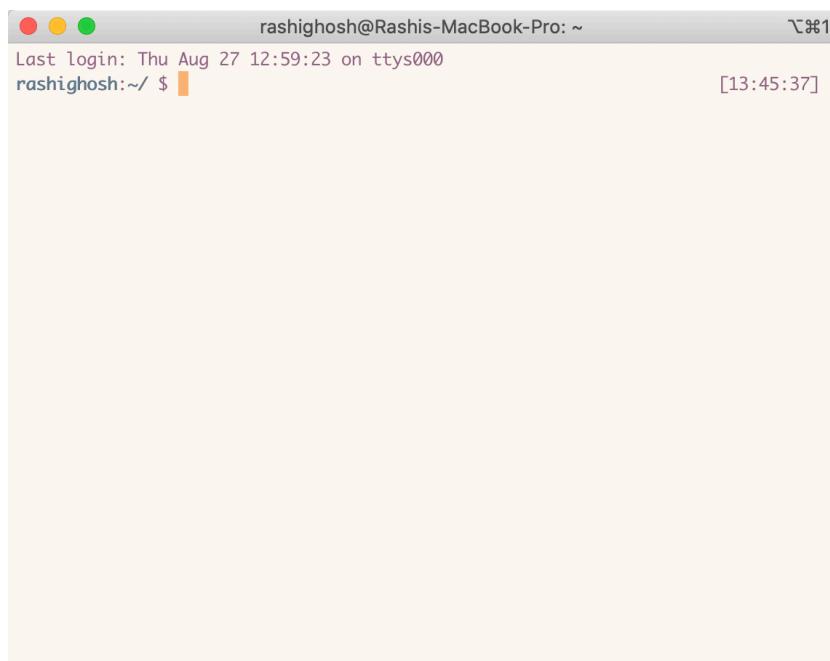
1. i personally looooveeee VS Code. a lot of my friends and even people i worked with love it too! you can download it from here:

<https://code.visualstudio.com/>

★ note: do NOT get just regular visual studio (it's purple). that's the IDE version, and has some serious weird drawbacks.

## steps.

1. create a program in your text editor. save it somewhere easily accessible.
2. open up the terminal. (can command+spacebar and search for it). you get an interface that looks something like this:



A screenshot of a Mac OS terminal window. At the top, there are three colored dots (red, yellow, green) followed by the text "rashighosh@Rashis-MacBook-Pro: ~". On the right side, there is a small icon and the text "TextEdit". Below this, the terminal displays the following text:  
Last login: Thu Aug 27 12:59:23 on ttys000  
rashighosh:~/ \$ [13:45:37]

3. now we are going to learn some linux commands! here are the most helpful ones. below is me navigating the terminal.

- o cd
  - ▶ used to **change directories**, and navigate around your computer. so if i wanted to access my Desktop, i use "**cd Desktop**"
  - ▶ if i want to go up a directory (like say i go into a folder on my desktop, and want to go back up to my desktop), i use "**cd ..**"
- o ls
  - ▶ used to **list** everything in the current directory. so if you were in Desktop, and used ls, it would list all the files on your desktop!

The screenshot shows a terminal window on a Mac. The title bar says "rashighosh@Rashis-MacBook-Pro: ~/D". The command history and output are as follows:

```
Last login: Thu Aug 27 13:45:36 on ttys000
rashighosh:~/ $ ls
Applications      Library
Creative Cloud Files Movies
Desktop          Music
Documents         NetBeansProjects
Downloads         Pictures
rashighosh:~/ $ cd Desktop
rashighosh:Desktop/ $ ls
Citrix Workspace and box setup.pdf
Resume.pdf
Screen Shot 2020-08-27 at 1.45.54 PM.png
arraysndptrs.cpp
codingenviro.pages
gre
h2p2
hooks
jbht235.png
jbht236.png
labconcept1.cpp
labconcept2.cpp
modal.js
output
output.exe
portfolio
programming foundations
rashighosh:Desktop/ $ cd hooks
rashighosh:hooks/ (master) $ cd ..
rashighosh:Desktop/ $
```

Annotations with arrows and boxes explain the commands:

- An arrow points from the first "ls" command to the list of files, with the text "listing everything in current directory".
- An arrow points from the "cd Desktop" command to the list of files in the Desktop directory, with the text "changing directory to Desktop".
- An arrow points from the second "ls" command to the list of files in the Desktop directory, with the text "listing everything in my Desktop".
- An arrow points from the "cd hooks" command to the text "going into a folder on my Desktop".
- An arrow points from the final "cd .." command to the text "returning back up to my Desktop".

- now, to compile and run a c++ file, make sure you are in the same directory as the file. then run the following two commands:

```
g++ -Wall [filename].cpp -o output  
. /output
```

the first command compiles the program. this is where you will get errors & warnings (if any). the second runs the program. here's me compiling & running a c++ file below:



The screenshot shows a terminal window with three colored icons (red, yellow, green) at the top left. The user is logged in as `rashighosh@Ra`. The terminal history shows the following commands and their output:

```
rashighosh:Desktop/ $ g++ -Wall arraysndptrs.cpp -o output  
rashighosh:Desktop/ $ ./output  
arr: 0x7ffeedd0c65c  
&arr: 0x7ffeedd0c65c  
arr[0]: 1  
&arr[1]: 0x7ffeedd0c660  
ptr: 0x7ffeedd0c65c  
*ptr: 1  
*ptr: 2  
ptr: 0x7ffeedd0c65c  
arr[0]: 2  
arr[1]: 3  
ptr: 0x7ffeedd0c664  
*ptr: 3  
2 3 3  
0x0  
rashighosh:Desktop/ $
```

as you can see on the first line, i compiled a file called "arraysndptrs.cpp". i had no warnings or errors. then on the second line i ran the program, and it outputted in the terminal!

## FOR WINDOWS OS.

### **preliminary step (text editor + mingw).**

**note:** you can skip this step if you already have a preferred text editor and mingw installed.

- i personally looooveeee VS Code. a lot of my friends and even people i worked with love it too! you can download it from here:

<https://code.visualstudio.com/>

★ note: do NOT get just regular visual studio (it's purple). that's the IDE version, and has some serious weird drawbacks.

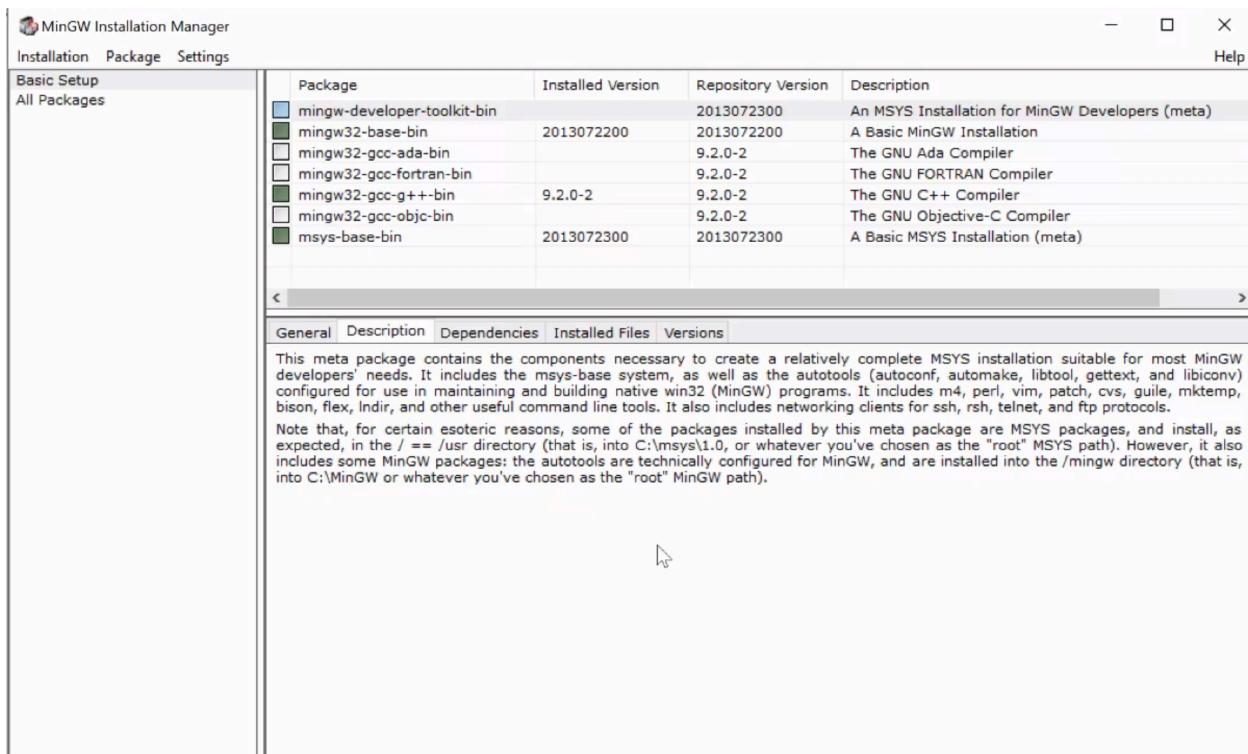
- for windows, we have to download & install a c++ compiler called mingw. you can do so from here:

<https://osdn.net/projects/mingw/releases/>

- click the blue button with the windows icon that has "**mingw-get-setup.exe**" next to it (about halfway down the page):



- download & install.
- once installed, a window like this pops up:



you want to mark the boxes as shown above for installation. specifically:

- ★ mingw32-base-bin
- ★ mingw32-gcc-g++-bin
- ★ msys-base-bin

7. now click "**installation**" (in the top left corner) > "**apply changes**"

8. it might take a minute, but once it's done, you're good to go!

## steps.

1. create a program in your text editor. save it somewhere easily accessible.

2. open up the command prompt.

3. now we are going to learn some linux-ish commands! here are the most helpful ones.

- cd

- ▶ used to **change directories**, and navigate around your computer. so if i wanted to access my Desktop, i use "**cd Desktop**"

- ▶ if i want to go up a directory (like say i go into a folder on my desktop, and want to go back up to my desktop), i use "**cd ..**"

- dir

- ▶ used to list everything in the **current directory**. so if you were in Desktop, and used dir, it would list all the files on your desktop!

4. now, to compile and run a c++ file, make sure you are in the same directory as the file. then run the following two commands:

```
g++ -Wall [filename].cpp -o output.exe  
output.exe
```

the first command compiles the program. this is where you will get errors & warnings (if any). the second runs the program. here's someone compiling & running a c++ file below:

```
C:\Users\Alec Burnworth\Downloads>g++ -Wall arraysndptrs.cpp -o arraysndptrs.exe  
arraysndptrs.cpp: In function 'int main()':  
arraysndptrs.cpp:39:11: warning: 'ptr2' may be used uninitialized in this function [-Wmaybe-uninitialized]  
39 |     cout << ptr2 << endl;  
|         ^~~~  
  
C:\Users\Alec Burnworth\Downloads>arraysndptrs.exe  
arr: 0x61fef8  
&arr: 0x61fef8  
arr[0]: 1  
&arr[1]: 0x61fefc  
ptr: 0x61fef8  
*ptr: 1  
*ptr: 2  
ptr: 0x61fef8  
arr[0]: 2  
arr[1]: 3  
ptr: 0x61ff00  
*ptr: 3  
2 3 3  
0x401cc0  
  
C:\Users\Alec Burnworth\Downloads>
```

as you can see on the first line, a file called "arraysndptrs.cpp" was compiled. there was a warning, which is displayed. then, the program was ran, and it outputted in the terminal!