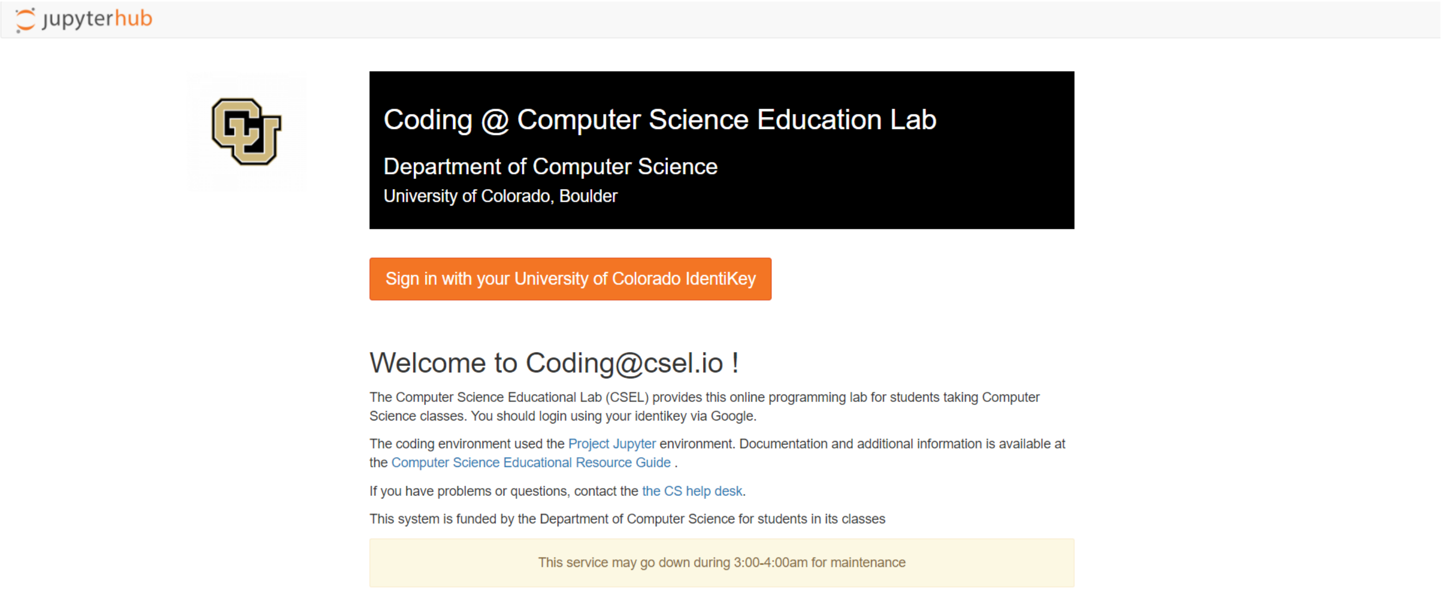
CSCI 1300 CS1: Starting Computing
Hoenigman/Naidu/Park/Ramesh - Fall 2023
Visual Studio Code - Cloud Platform (CSEL)

You may also use CU’s online coding platform CSEL (coding.csel.io) for this course, if none of the local installations work for you. This platform should work on a Chromebook.

Coding@CSEL.io

Step 1: Open your favorite browser and go to coding.csel.io. You should see the following welcome screen:



Coding @ Computer Science Education Lab

Department of Computer Science

University of Colorado, Boulder

Welcome to Coding@csel.io !

The Computer Science Educational Lab (CSEL) provides this online programming lab for students taking Computer Science classes. You should login using your identikey via Google.

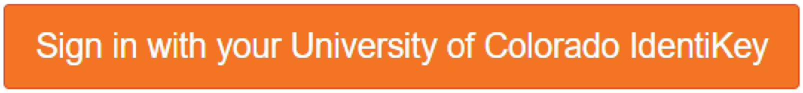
The coding environment used the Project Jupyter environment. Documentation and additional information is available at the Computer Science Educational Resource Guide .

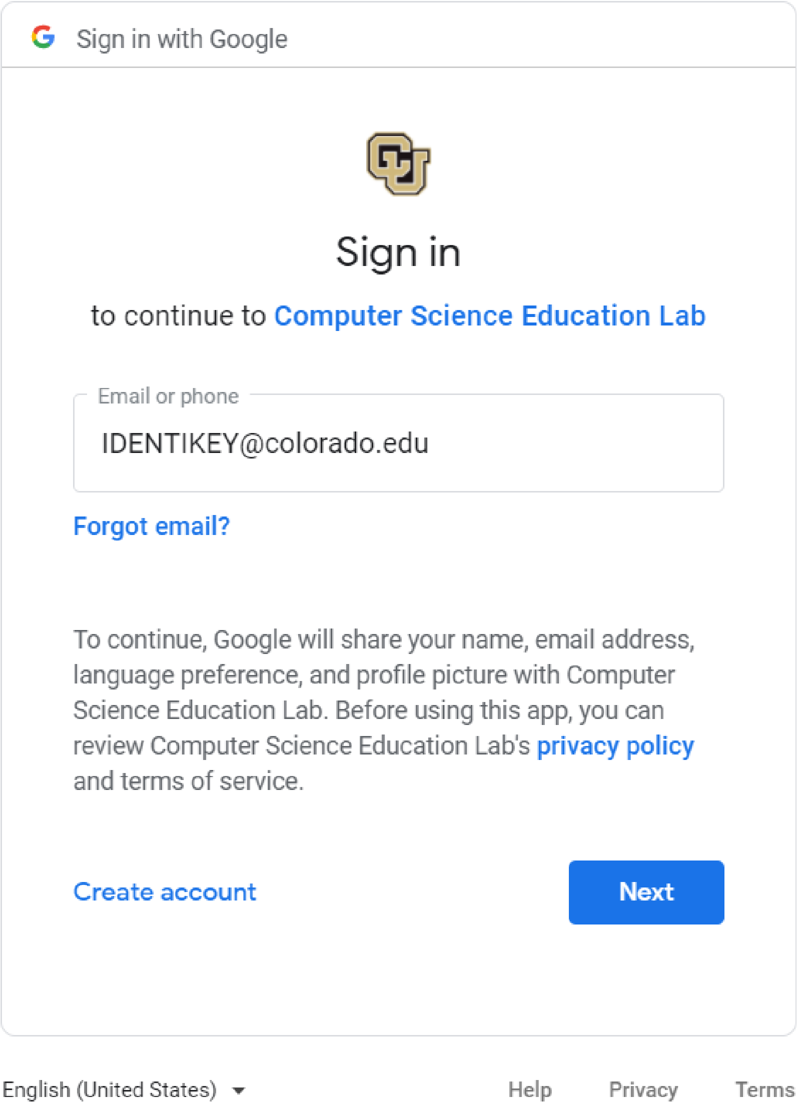
If you have problems or questions, contact the the CS help desk.

This system is funded by the Department of Computer Science for students in its classes

This service may go down during 3:00-4:00am for maintenance

Step 2: Click on the “*Sign in with your University of Colorado IdentiKey*” button. This will further open a Google sign-in page. Log-in using your identikey@colorado.edu Google account.





Sign in with Google

Sign in

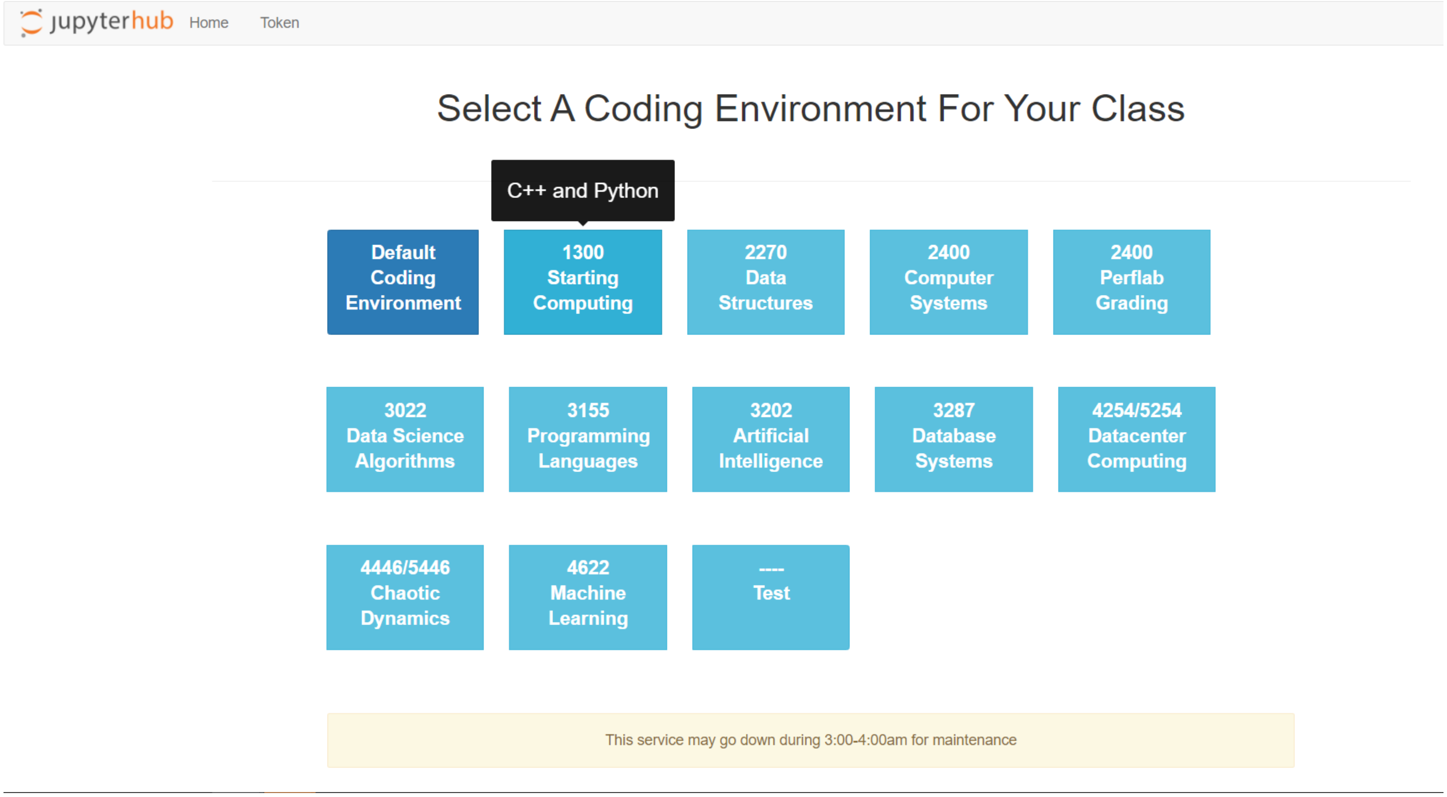
to continue to Computer Science Education Lab

Email or phone: IDENTIKEY@colorado.edu

To continue, Google will share your name, email address, language preference, and profile picture with Computer Science Education Lab. Before using this app, you can review Computer Science Education Lab's privacy policy and terms of service.

Step 3: Once you are authenticated and logged-in, you will see the following screen. Choose the "*1300 Starting Computing*" tile.

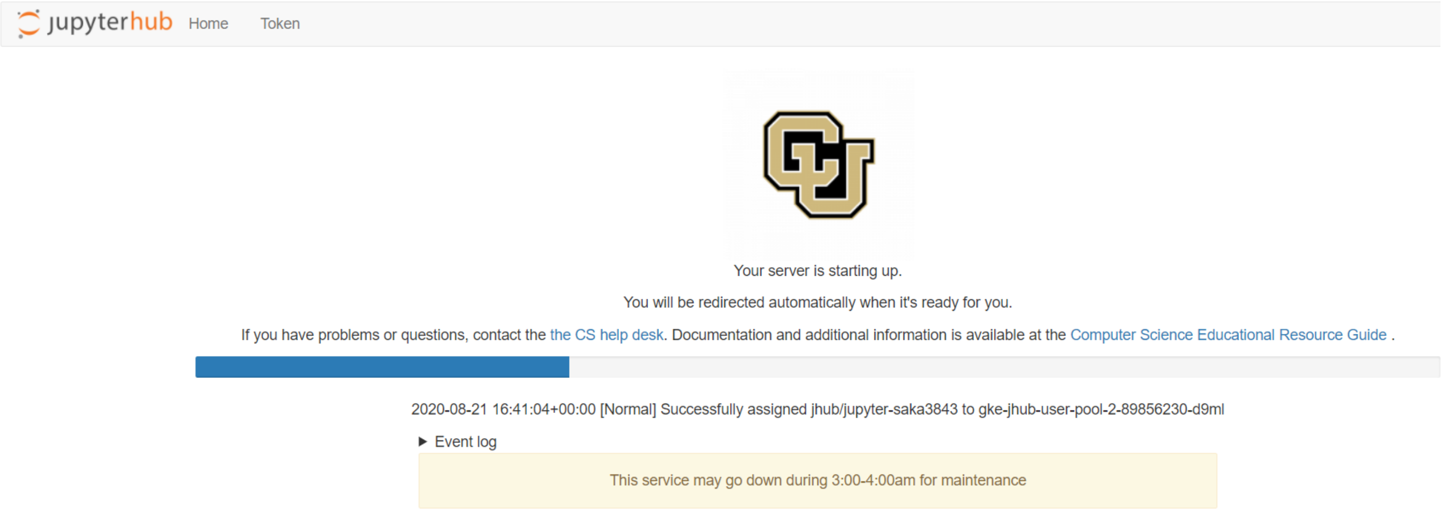
NOTE: If you are not able to log-in to the platform, email help@cs.colorado.edu with your issue.



Select A Coding Environment For Your Class

This service may go down during 3:00-4:00am for maintenance

Once selected, a container will be created for you. It may take a few seconds to boot up.



Your server is starting up.

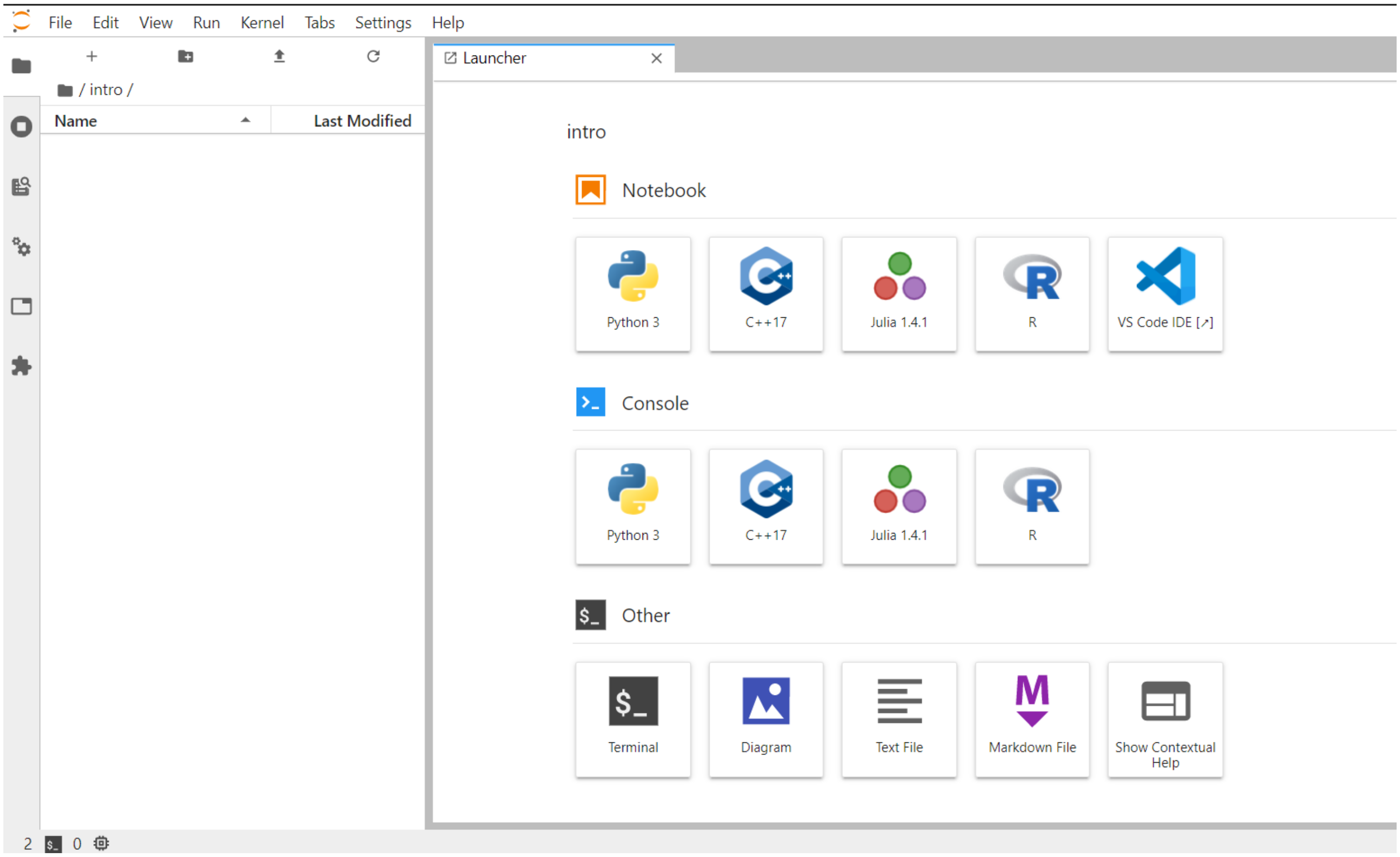
You will be redirected automatically when it's ready for you.

If you have problems or questions, contact the the CS help desk. Documentation and additional information is available at the Computer Science Educational Resource Guide .

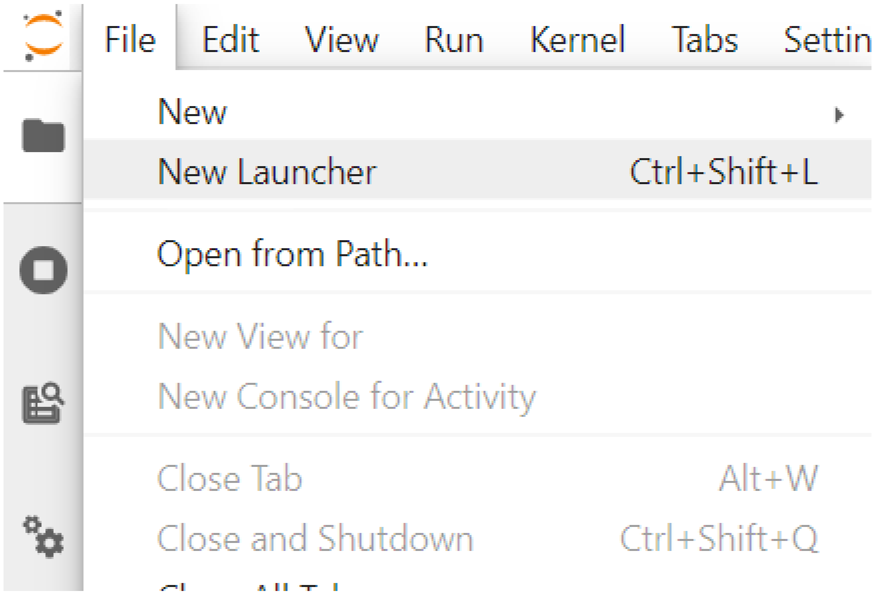
```
2020-08-21 16:41:04+00:00 [Normal] Successfully assigned jhub/jupyter-saka3843 to gke-jhub-user-pool-2-89856230-d9ml
```

This service may go down during 3:00-4:00am for maintenance

Step 4: You will see the following screen. Notice the right pane titled “Launcher”.



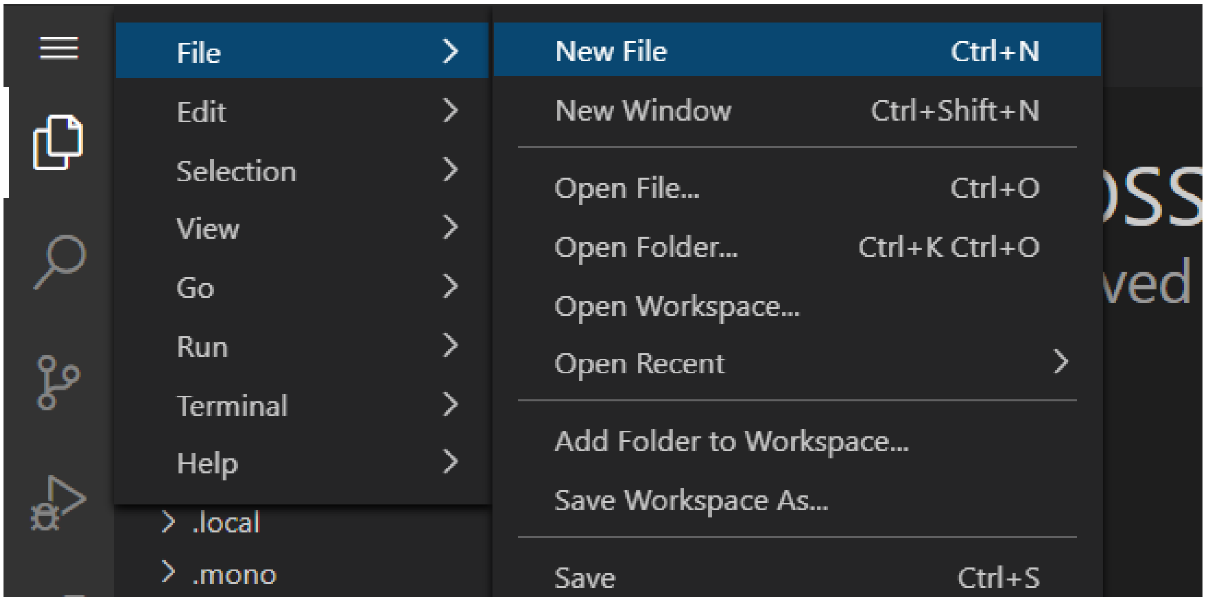
If the “Launcher” is not available, go to File > New Launcher.



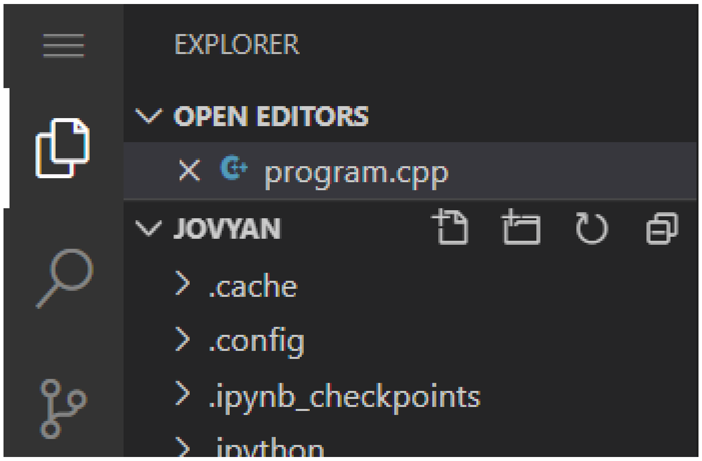
Step 5: In the “Launcher” pane, click on the “VS Code IDE” icon. This will open a New tab/window with an online version of VS Code.

Step 6: You can now use this version of VS Code as you would on a local machine. Let's go ahead and try writing a "Hello, World!" C++ program.

6.1: Start by going to the top left and clicking on the menu (≡). Then choose File > New File.



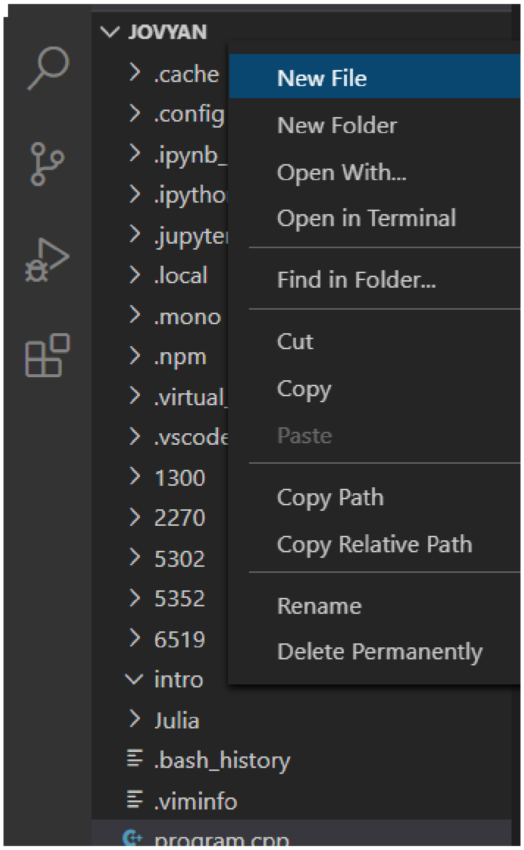
You can also create a new directory/new file using the explorer pane.



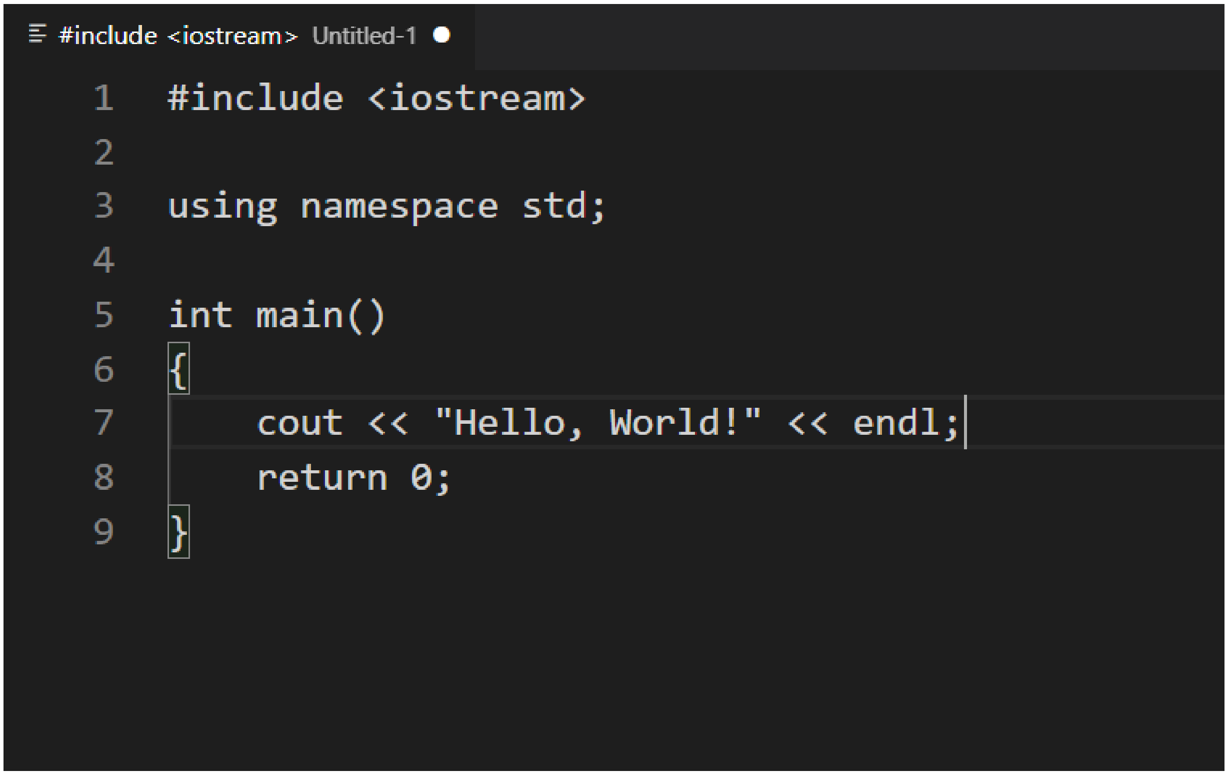
Use 🗋 for a new file and 🗀 for a new folder.

Once that file/folder is created, rename it and then you can drag it into a different folder.

You can also right-click a folder and create a New File.



6.2: Once the file is available, go ahead and type the following in your newly opened by file.

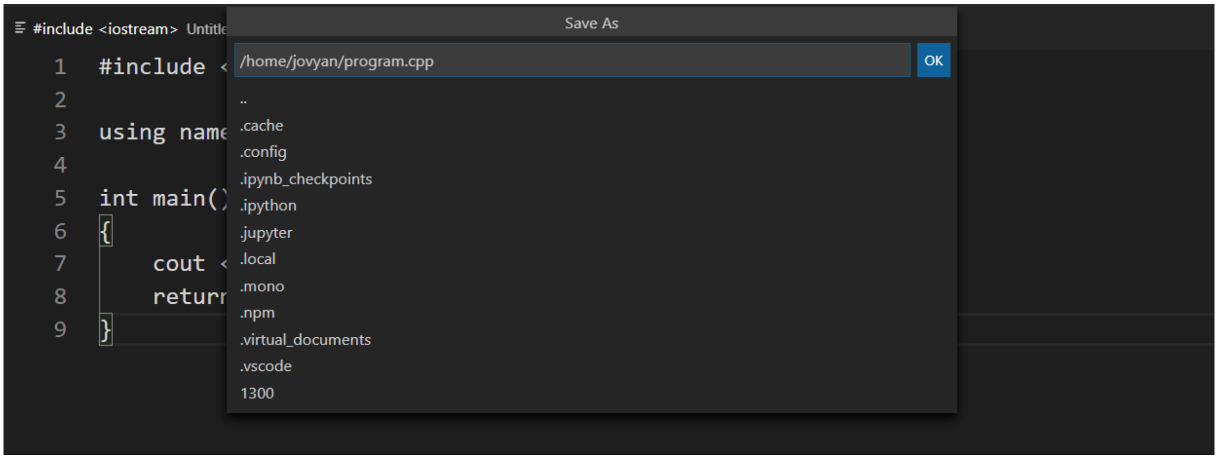


```
#include <iostream>

using namespace std;

int main()
{
    cout << "Hello, World!" << endl;
    return 0;
}
```

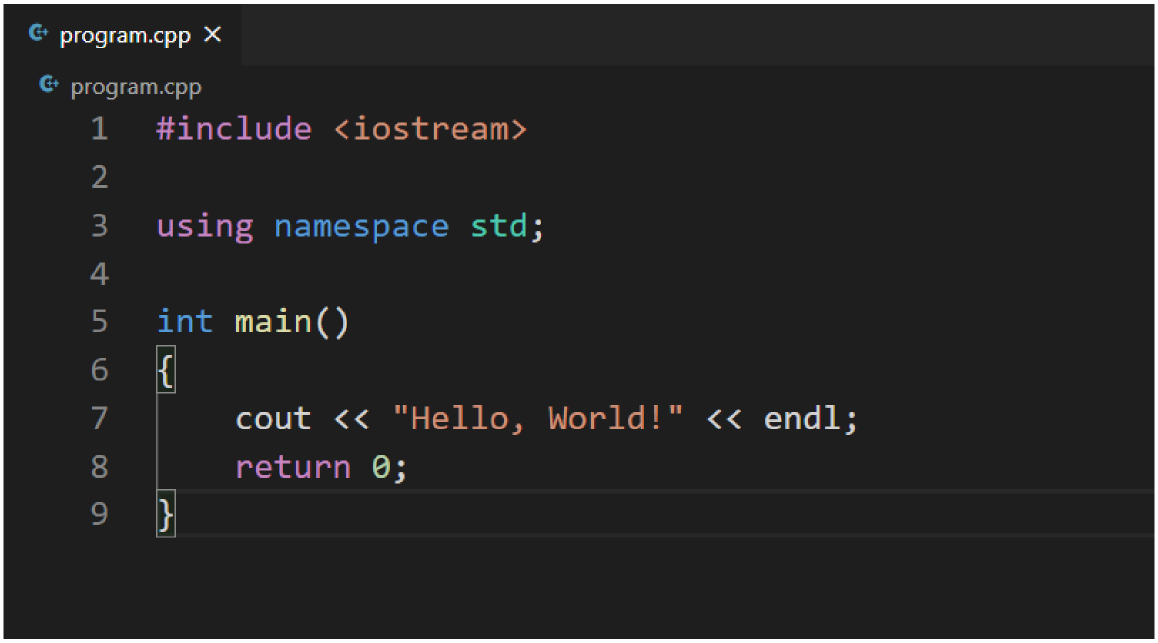
6.3: Then go ahead and save it. Use **Menu > File > Save** or press **Ctrl + S** or **Cmd + S**. If your file does not have a name already, you can specify the file path, adding the extension `.cpp` and save your file by clicking OK.



```
#include <
using name
int main()
{
    cout <
    returr
}
```

```
/home/jovyan/program.cpp
..
.cache
.config
.ipynb_checkpoints
.ipython
.jupyter
.local
.mono
.npm
.virtual_documents
.vscode
1300
```

6.4: Once you have saved the file, the syntax highlighting should come into effect.

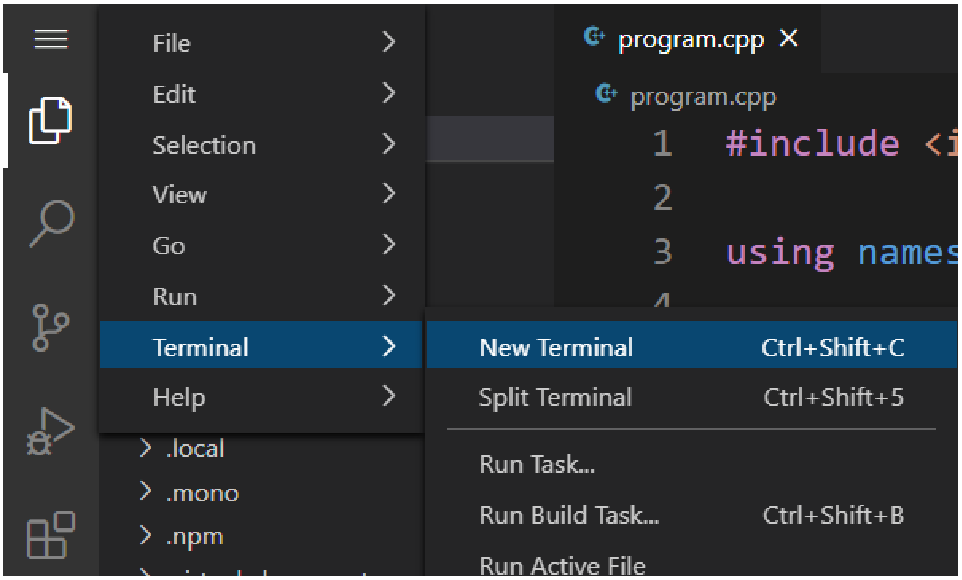


```cpp
#include <iostream>

using namespace std;

int main()
{
    cout << "Hello, World!" << endl;
    return 0;
}
```

Step 7: Now, let’s compile and run the program. Go to the application menu (≡) > Terminal > New Terminal.

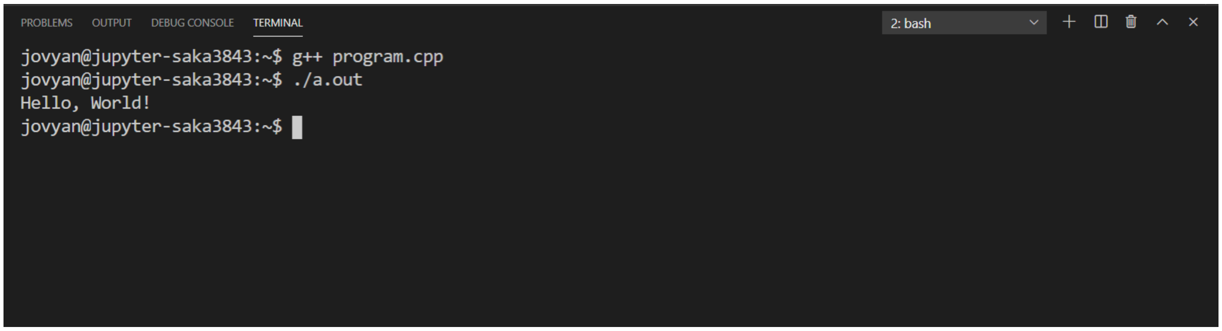


The Terminal should open up towards the bottom of the VS Code window. Go ahead and test the compile and execution commands.

Navigate to the directory of the program file that you just wrote. Use the following commands to compile and execute the program:

```
$ g++ <program_name.cpp>
$ ./a.out
```

If you get the following output, then you have successfully compiled and executed a C++ program.

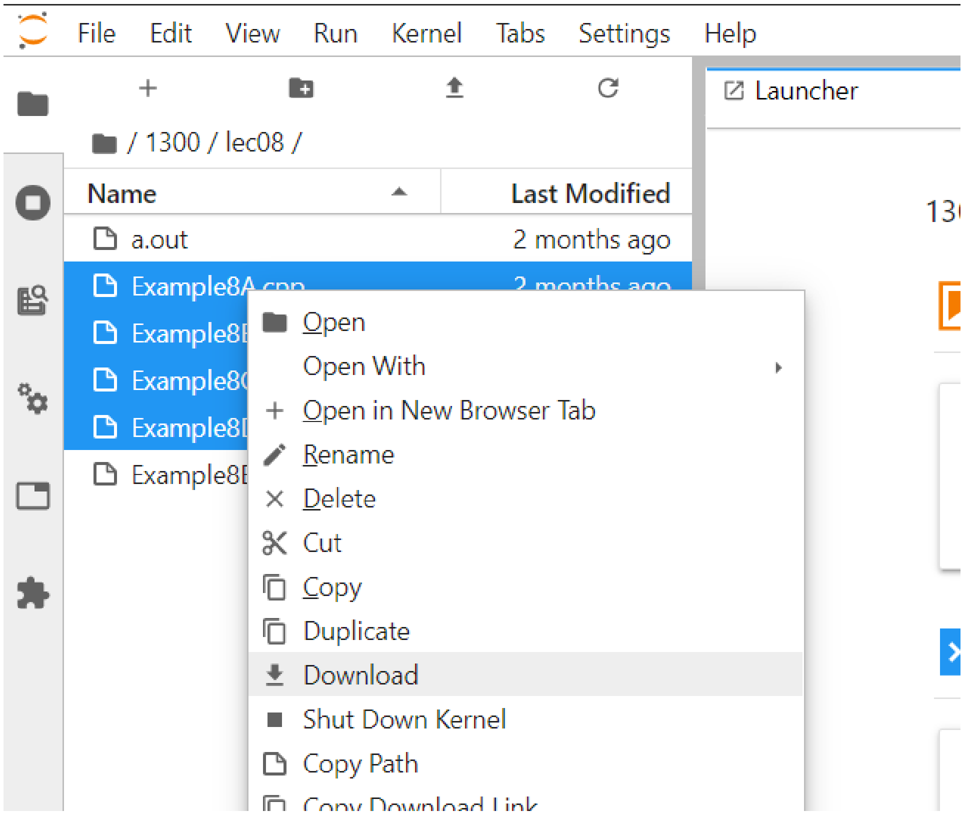


```
jovyan@jupyter-saka3843:~$ g++ program.cpp
jovyan@jupyter-saka3843:~$ ./a.out
Hello, World!
jovyan@jupyter-saka3843:~$ 
```

Note on Downloading Files from Coding@CSEL

You cannot download files directly from the VS Code window. You will need to open the JupyterLab window (coding.csel.io).

On the left pane of the JupyterLab window, open the file explorer and select all the files that you want to download. Then right-click on them and select Download.



And that’s it! You have successfully set up your coding environment on your machine. Well done!

Happy coding!