Step-1

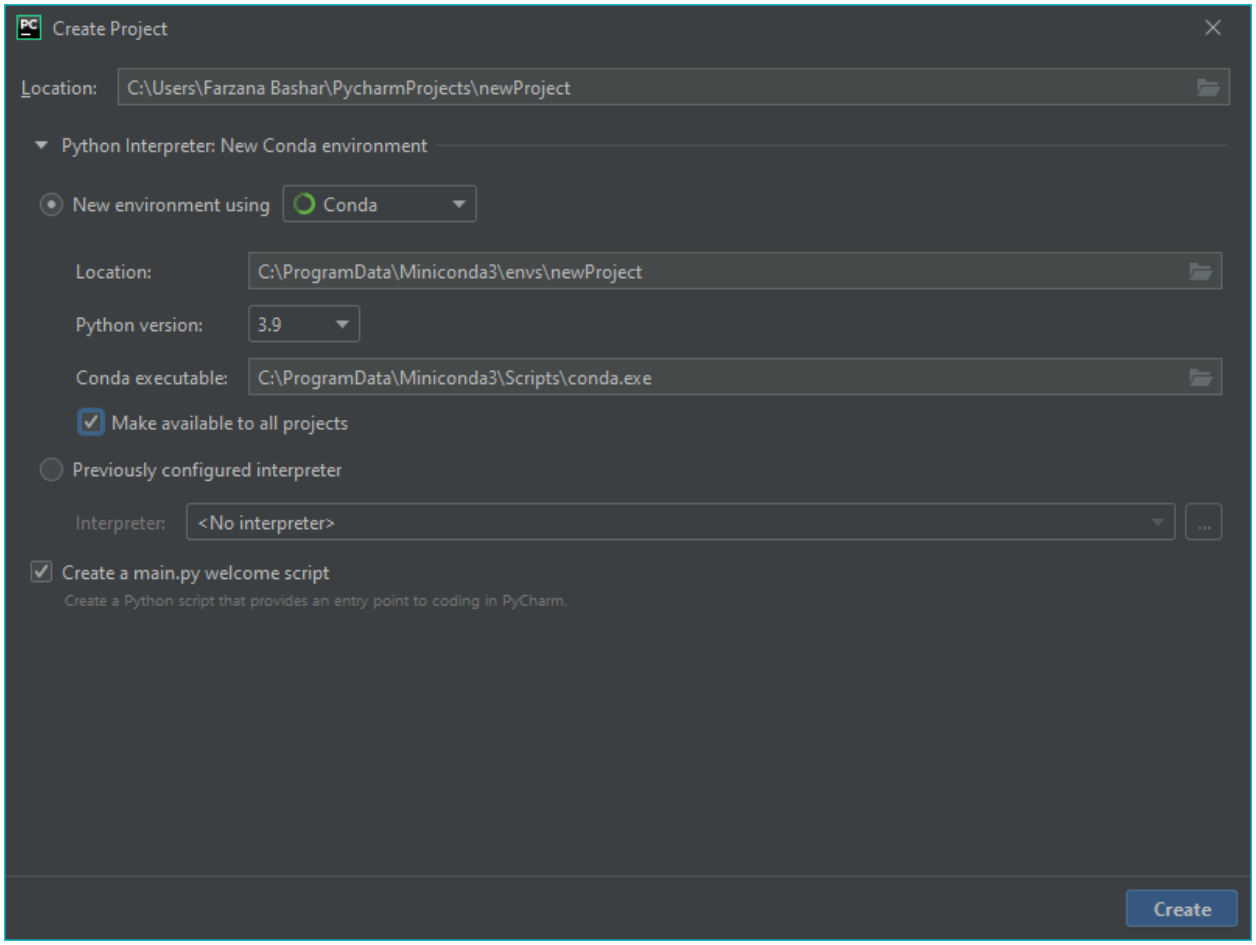
Install Python from any python environment management system (I recommend ***conda***).
If you do not have any prior experience with python environment managers then I would recommend continuing with Java. They can cause unpredictable problems that can take up a lot of time to solve.

Step-2

Install a Python IDE (I recommend ***PyCharm***)

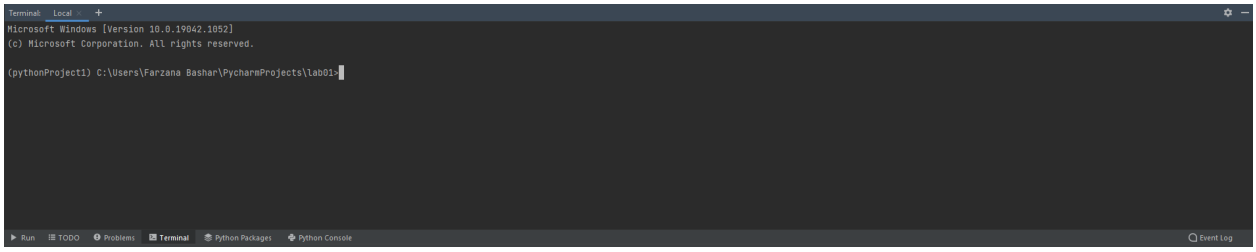
Step-3

Start a new project (for the first time)



Choose the python environment manager that is in your system.
Tick the "***Make available to all projects***" option. That way you can start from the same virtual environment in the new projects from now on.
Remember the name of the project for later use.
Press the create button.
Now in this virtual environment, we will install OpenGL using the following command.

Go to the terminal at the bottom of the screen.

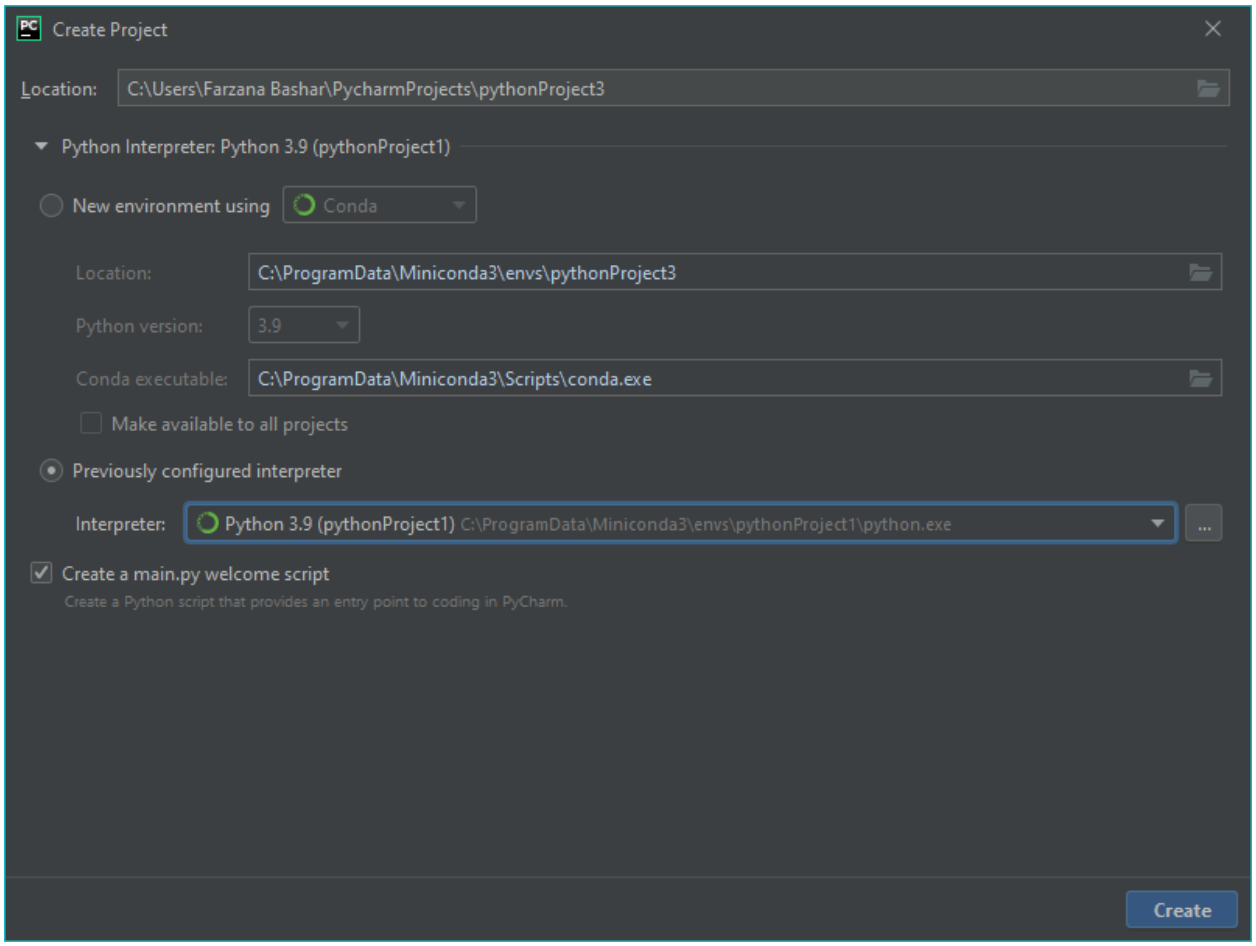


Type the following command.

```
$ pip install PyOpenGL PyOpenGL_accelerate
```

It will show that PyOpenGL was installed successfully.
So, this virtual environment is set up for OpenGL and if we use this one, we won't have to install it again.
(Note: Please make sure at least ***Microsoft Visual C++ 14*** or later version is installed or otherwise it will fail)

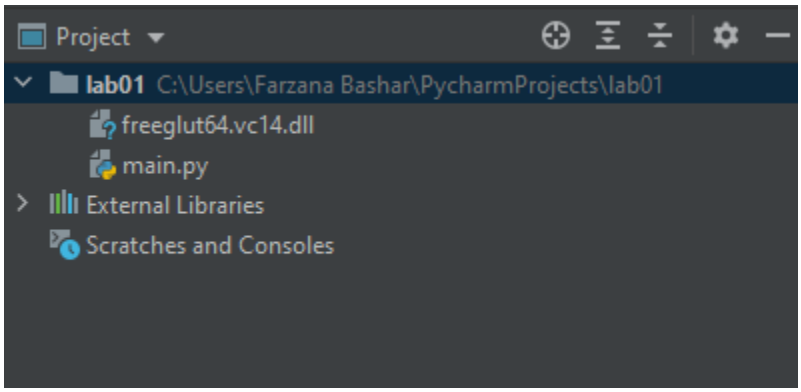
Start a new project (Using previously set up virtual environment)



Now use the "***Previously configured interpreter***" option. From the drop down you can easily choose your virtual environment.

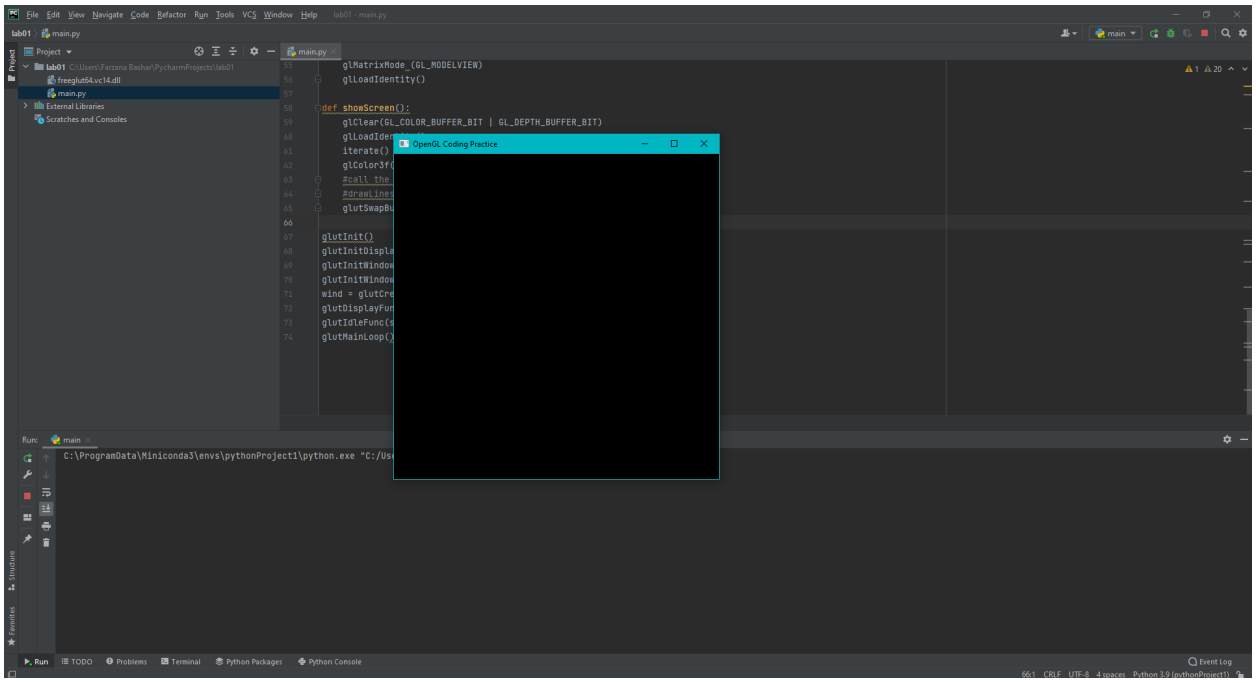
Step-4

Now we have to add the ***FreeGLUT.dll*** file
Download ***freeglut64.vc14.dll***
(If you are using x86 architecture please rename it to "***freeglut32.vc14.dll"***
Then, copy this dll to your project alongside "***main.py***"



Step-5

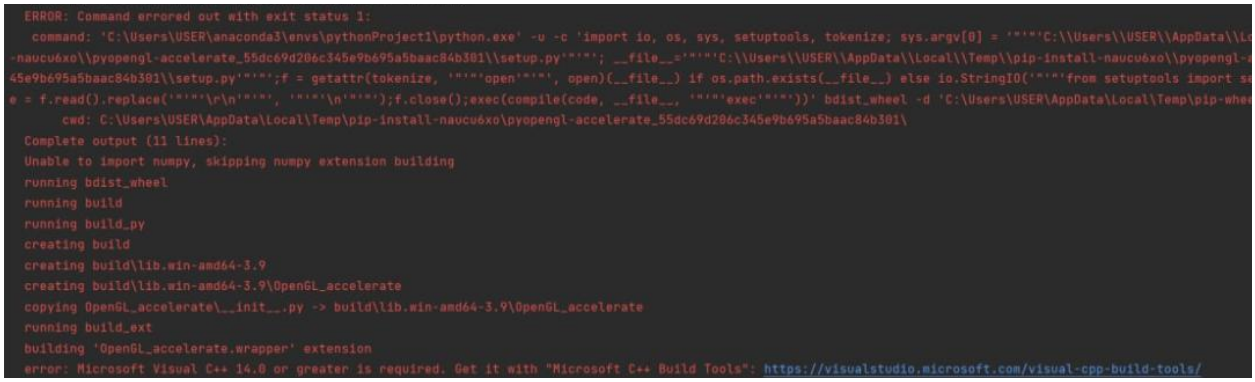
Copy the template code
And paste it into ***main.py***
Then, run and you should see a blank screen.



Now go ahead and watch the introductory video.
https://youtu.be/E09j-bRBC24

Frequently faced problems and solutions

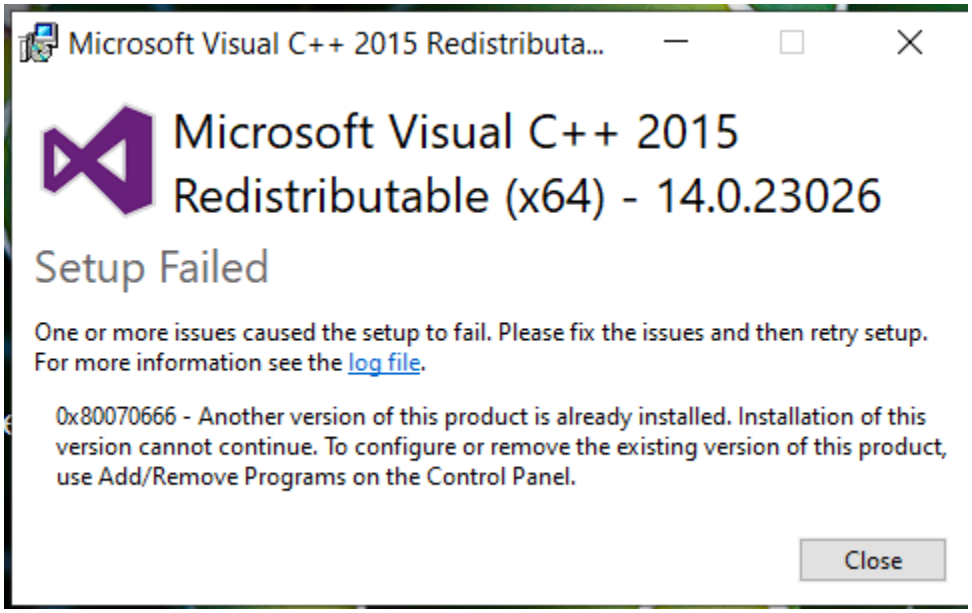
1.



MS Visual C++ error

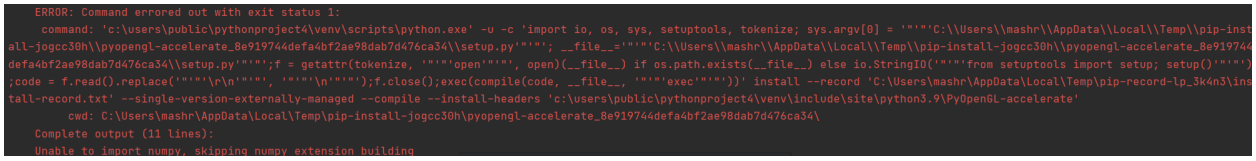
If you face this problem after running the command “pip install PyOpenGL PyOpenGL_accelerate”, then this means you have to install Microsoft Visual C++ latest version.
Please download it from the following link:
https://www.microsoft.com/en-us/download/details.aspx?id=48145

Then, you might face the following problem.



If this occurs, go to “Add/Remove Programs” in windows settings and uninstall only the version conflicting with the one you are trying to install (14.0.xxx in this case)
If it still doesn’t solve the problem, contact me asap on DM.

2.

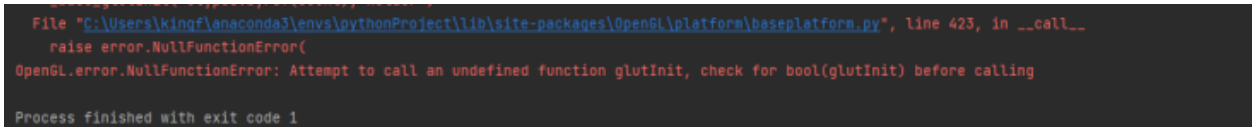


Import numpy error

Even after you installed MS Visual C++ 14, this error can show after running the command “pip install PyOpenGL PyOpenGL_accelerate”. This means numpy was missing from the base environment in python. Don’t worry. Just run the command:

pip install numpy

3.



OpenGL.error.NullFunctionError/ Attempt to call an undefined function glutInit

This is probably the most frustrating problem faced.
Follow the following steps very carefully.
ERROR: OpenGL.error.NullFunctionError: Attempt to call an undefined function glutInit, check for bool(glutInit) before calling In Python

Solution:

It is mainly because you're running 64-bit windows put pip is installing 32-bit version of PyOpenGL.

To fix this, follow these steps:

1. Uninstall existing PyOpenGL, Run pip uninstall PyOpenGL PyOpenGL_accelerate

2. Download the 64-bit builds of PyOpenGL and PyOpenGL accelerate from here: https://www.lfd.uci.edu/~gohlke/pythonlibs/#pyopengl

3. How to choose which one to download? Well, first you need to check your python version. Run python --version to determine.
Then according to your version download the whl files for PyOpenGL and PyOpenGL accelerate. For example, if you have Python 3.8, download these 2 files:

PyOpenGL-3.1.5-cp38-cp38-win_amd64.whl

PyOpenGL_accelerate-3.1.5-cp38-cp38-win_amd64.whl

Similarly, if you run Python 3.9, download these instead:

PyOpenGL-3.1.5-cp39-cp39-win_amd64.whl

PyOpenGL_accelerate-3.1.5-cp39-cp39-win_amd64.whl

Note: Must download amd64 ones, after all, you're running 64-bit windows.

4. Now go to the folder where you downloaded the files and run powershell/cmd with the administrator there.

5. Use pip to force install those files. For example:

pip install PyOpenGL-3.1.5-cp38-cp38-win_amd64.whl --force-reinstall

pip install PyOpenGL_accelerate-3.1.5-cp38-cp38-win_amd64.whl --force-reinstall

Note: Install PyOpenGL first and then PyOpenGL_accelerate

6. If the installation succeeds, you may be able now to run.

Some facts to make sure:

*All platforms must be of the same bit version.

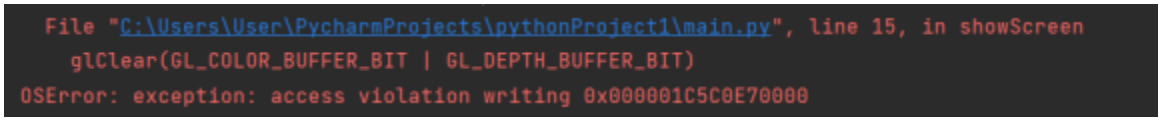
*System type, python, OpenGL needs to be of the same bit version. In my case it was x64 bit

*It is necessary to restart your desktop if you reinstall any one of them (Python, OpenGL)

*It is recommended to keep your pip version at 20.3, I downgraded it from 21.1.3

[If this doesn't solve your problem, please contact me asap on DM]

4.



glClear Error/ access violation writing

There is a temporary solution to this problem. To comment out glClear
But that is not recommended.
glClear is called by OpenGL regularly to refresh the screen. If we comment it out, whatever is drawn on the screen will likely remain. That may not seem problematic but in the

long run, it can produce some weird results. Especially in the task of drawing random pixels, OpenGL keeps producing random pixels to draw and the whole screen becomes a mess.

The root of this problem is unclear but it seems to occur only when nothing is drawn on the screen, yet glClear is called. It is a fault due to poor implementation of OpenGL in python (the implementation is basically just a python wrapper over the c/c++ implementation of OpenGL). Joyanta made some changes in the template file and the new template file doesn't cause this error (it doesn't comment out glClear). Please try running that template which also draws a pixel. Joyanta also deleted glutIdlefunc() call in the previous template, although I am not sure why that was causing the problem but I am glad to see it works. You are encouraged to try with the `"glutIdleFunc(showScreen)"` line just like in the link [1] in reference. If it causes a problem, delete it but if it works, you can keep the line.

Many many thanks to Joyanta, Abrar Mahmud, Tawhid Sir and many others for not only finding these solutions but also continuously helping out students relentlessly.

Reference links:

1. https://stackabuse.com/brief-introduction-to-opengl-in-python-with-pyopengl
2. https://stackoverflow.com/questions/27093037/python-glutcreatewindow-error-wrong-type
3. https://stackoverflow.com/questions/39181192/attempt-to-call-an-undefined-function-glutinit