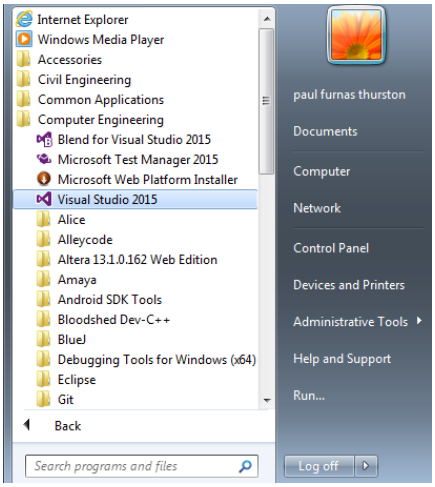
COEN 148: Lab 1
Intro to OpenGL

TA: Paul Thurston
Email: pthurston@scu.edu

Part 1: Opening Visual Studio

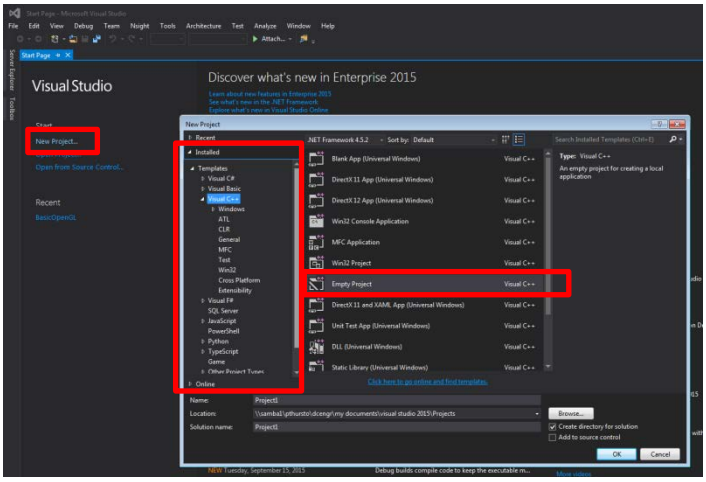
1. Open Visual Studio 2015 (Start->All Programs->Computer Engineering->Visual Studio 2015



2. You will be asked to sign in, this is optional
3. Choose a visual theme

Note: Initial setup might take a few minutes and require sometimes require you to restart Visual Studio

4. Once on the start page click ‘New Project’
5. Choose the ‘Project Template’ under ‘Installed->Templates->Visual C++’
6. Under ‘Name’ enter the name of your project (i.e. ‘OpenGL_Lab1’)
7. If desired change the location that your project will be saved (I recommend leaving it with the location).



Note: If you do decide to use a new location make sure that it is on your dc drive, not the local machine that you are working on.

8. Make sure that the ‘create directory for solution’ option is checked.
9. Click ‘OK’ to create the project

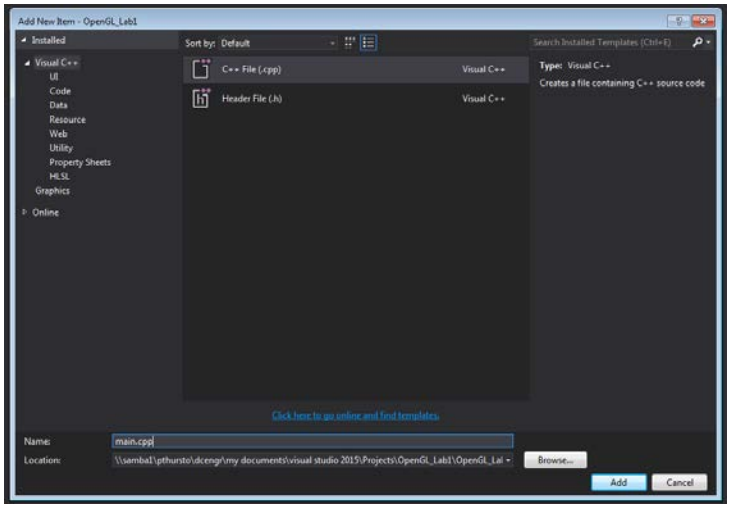
Note: If you receive a message saying ‘The project location is not fully trusted…’ ignore it and hit ‘OK’

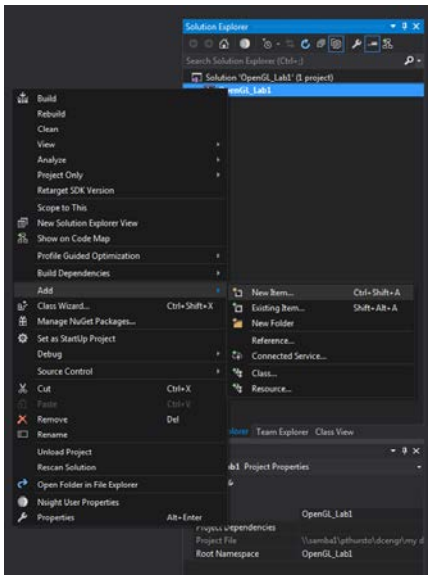
General Info: Projects vs Solutions

In Visual Studio a Project is a collection of source files (.c, .cpp, .h, etc.) that are used to build an application. A Solution is a collection of Projects that can reference each other’s code. We will not need to make complicated applications that use this feature so every lab will be a Solution with 1 Project.

10. On the right you will see the Solution Explorer. This shows all of the files in your project in a hierarchy with the Solution at the top. Right click on the Project you just made (not the Solution) and click on ‘Add->New Item’.

11. Choose to add a ‘C++ File’ and call it main.cpp





12. main.cpp should now be open in the editor. Add the following code to the file:

```
#include <iostream>

int main() {
	using namespace std;

	cout << "Hello World" << endl;

	system("PAUSE");
}
```

13. Build the project by going to ‘Build->Build Solution’ or pressing ‘Ctrl-Shift-B’
14. The Output window at the bottom of the editor will tell you if the Build was successful or if there were compile errors.
15. Once you have a successful build you can run the project by going to ‘Debug->Start Debugging’ or pressing ‘F5’
16. A terminal should open saying “Hello World” and prompting you to press a key to continue

DEMO TO TA

COEN 148: Lab 1
Intro to OpenGL

Part 2: Setting Up OpenGL

Code based on program by Dr. David Kao

1. You can delete the code in main.cpp from the previous section
2. At the top of the file include the OpenGL Utility Toolkit

```
#include <gl\glut.h>
```

3. Add a main function that takes command line arguments

```
int main(int argc, char** argv){

	return 0;
}
```

4. Initialize and Create a window: Add this to the main function

```
glutInit(&argc, argv);

//Set Display Mode
glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);

//Set the window size
glutInitWindowSize(720, 720);

//Set the window position
glutInitWindowPosition(100, 100);

//Create the window
glutCreateWindow("Lab1: Intro to OpenGL");
```

5. Set display additional settings: Add this to the main function

```
//Set background to white
glClearColor(1.0, 1.0, 1.0, 0.0);

//initialize viewing values
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
glOrtho(0.0, 1.0, 0.0, 1.0, -1.0, 1.0);
```

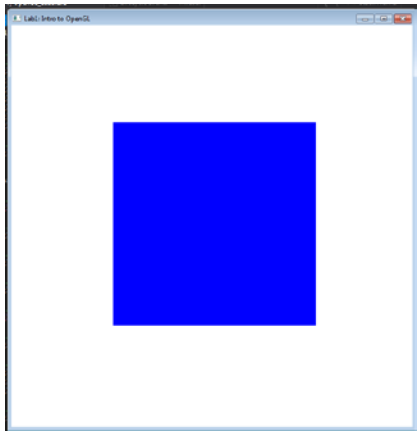
6. Displaying to the Window: Add this function to main.cpp

```
void display(void){

	//Clear all pixels
	glClear(GL_COLOR_BUFFER_BIT);

	//draw red polygon (rectangle) with corners at
	// (0.25, 0.25, 0.0) and (0.75, 0.75, 0.0)
	glColor3f(0.0, 0.0, 1.0);
	glBegin(GL_POLYGON);
	glVertex3f(0.25, 0.25, 0.0);
	glVertex3f(0.75, 0.25, 0.0);
	glVertex3f(0.75, 0.75, 0.0);
	glVertex3f(0.25, 0.75, 0.0);
	glEnd();

	// flush render objects to the screen as soon as possible
	glFlush();
}
```

7. Setup render loop: Add this to the end of the main function (before returning)

```
//Call "display" function
glutDisplayFunc(display);

//Enter the GLUT event loop
glutMainLoop();
```

8. Build and Run the program: Expected output shown below



DEMO