**Brock Barlow**
**ID #1113**
**Assessment ADGP 201 - Graphics**

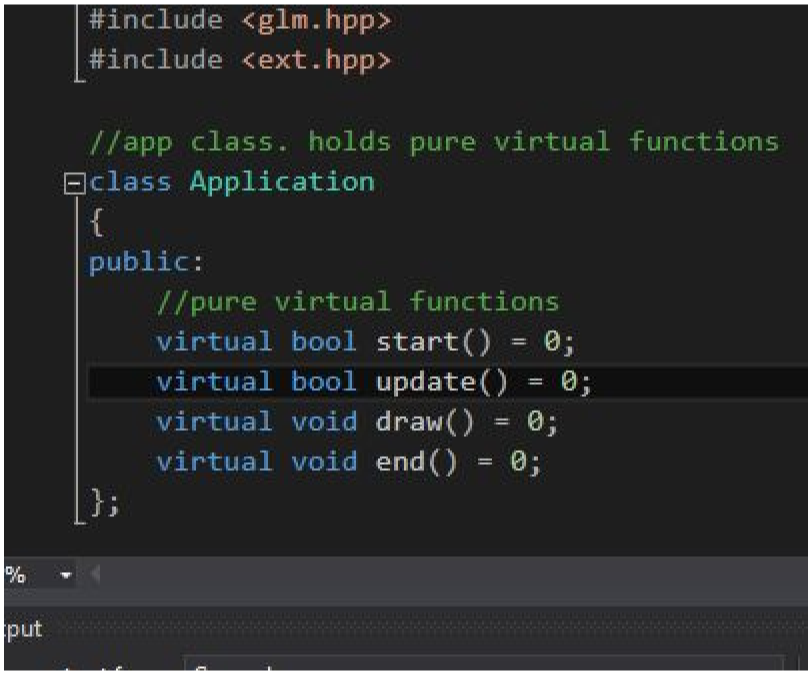
# Graphics Assessment Documentation for "Intro to OpenGL"

## Purpose:
Introduce the steps needed to setup the GLFW and OpenGL libraries using environment variables, using inheritance and using OOP design.

## Learning Outcomes:
1) Setup GLFW library.
2) Setup OpenGL library.
3) Must use environment variables ie: $(SolutionDir).
4) Must use inheritance. OOP in design involving a base class and classes that will inherit from this base class.
5) Documentation.

## Evidence:
The GLFW and OpenGL libraries were linked in visual studios by right clicking the project (Intro to OpenGL) and selecting properties. Going to "VC++ Directories" then "Library Directories", the following path was added: **$(SolutionDir)dep\glfw\lib;$(LibraryPath)**. Then under the "Linker" section and under "Input", the following items were adding: **opengl32.lib;glfw3.lib;%(AdditionalDependencies)**. This was done to both the debug and release configurations.

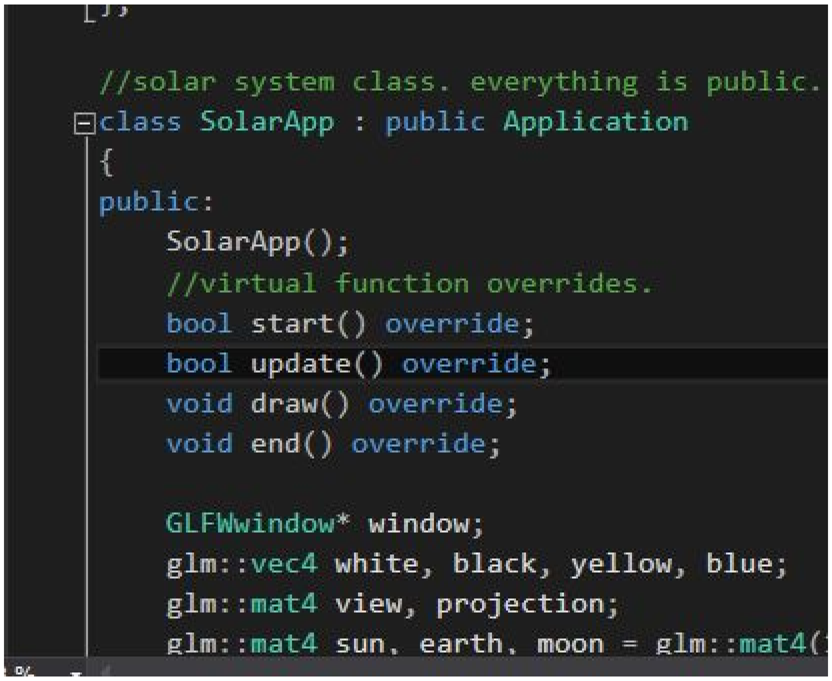
In the App.h file, a class was created called: Application. This class will be used as the "base" class that will get inherited by other classes. This class has four virtual functions under the public level. There are two bool virtual functions and two void virtual functions. Each function is equal to the value of zero. When inheriting classes use these functions, they will perform an override to them. The solar system class inheritances from the application class. The solar system class overrides the four virtual functions in the application.
In the main source file, the main function creates a new solar system. It then calls the four overridden functions.



```
#include <glm.hpp>
#include <ext.hpp>

//app class. holds pure virtual functions
class Application
{
public:
    //pure virtual functions
    virtual bool start() = 0;
    virtual bool update() = 0;
    virtual void draw() = 0;
    virtual void end() = 0;
};
```

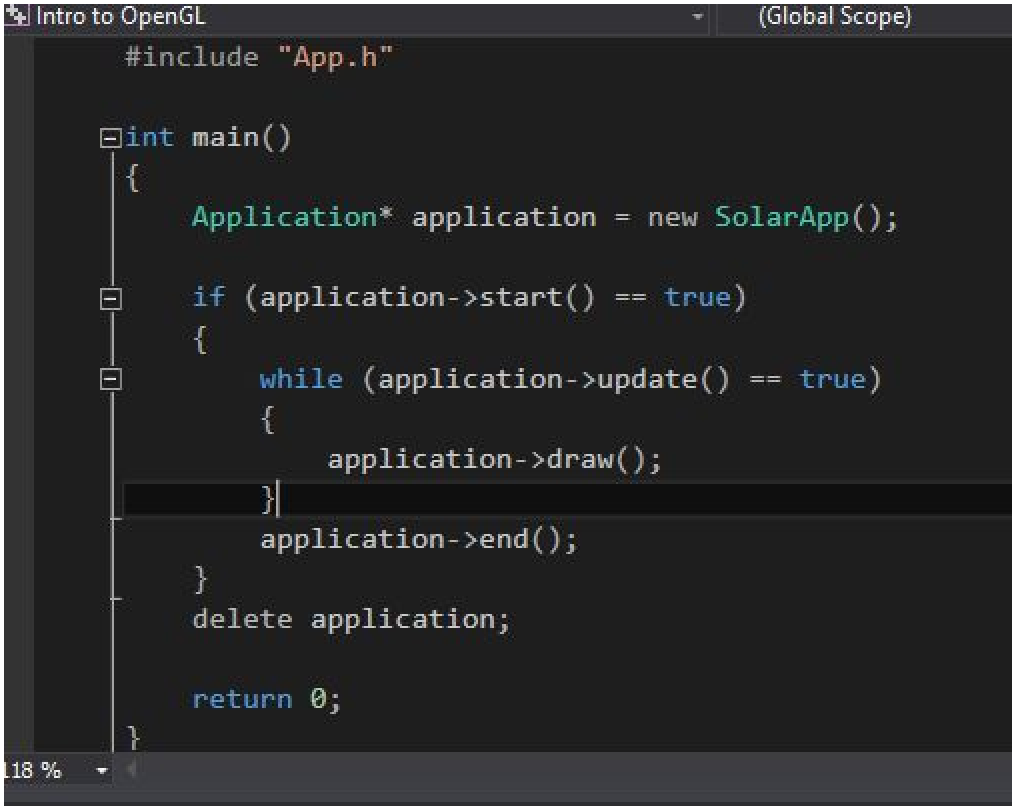
*Picture of the application class



```
//solar system class. everything is public.
class SolarApp : public Application
{
public:
    SolarApp();
    //virtual function overrides.
    bool start() override;
    bool update() override;
    void draw() override;
    void end() override;

    GLFWwindow* window;
    glm::vec4 white, black, yellow, blue;
    glm::mat4 view, projection;
    glm::mat4 sun, earth, moon = glm::mat4(
```

*Picture of the solar system class



```
#include "App.h"

int main()
{
    Application* application = new SolarApp();

    if (application->start() == true)
    {
        while (application->update() == true)
        {
            application->draw();
        }
        application->end();
    }
    delete application;

    return 0;
}
```

*Picture of the main function