**Walkthrough 1 : Network Manager**

**Objectives:**

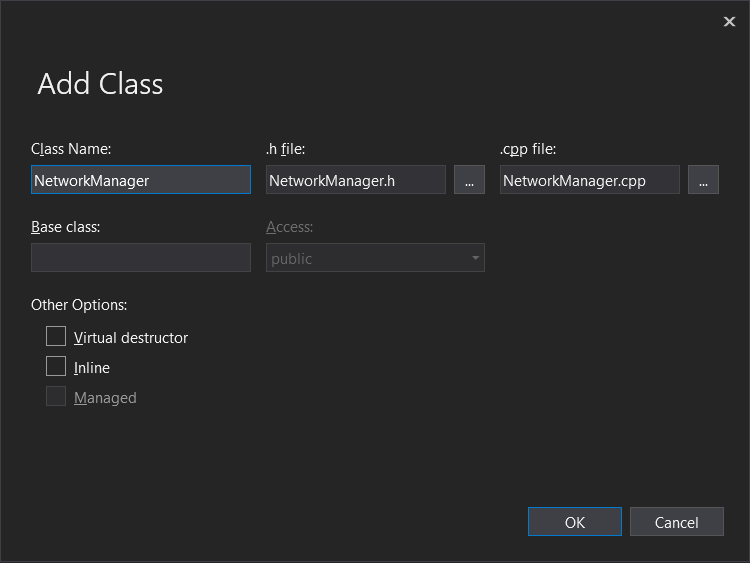
The intention of this walkthrough is to introduce students to the C++ Singleton Class Design Pattern and develop a base manager class for the networking code moving forward.

**Getting Started:**

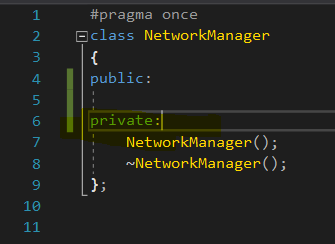
1. Download the Visual Studio SDL Simple Pong game available on blackboard under week 3 Walkthrough.
2. Extract to your C: somewhere.
3. Open up the .sln file.
4. Change the x64 dropdown to x86.
5. Examine and review the code.
6. Run the game.

**Adding a Singleton Class:**

1. Right click on the Source folder in the solution explorer of Visual Studio.
2. Click the Add -> Class… option.
3. Name your new class **NetworkManager** and **ensure it is placing it inside your source directory** by clicking the ”…” ellipses button for the cpp and header files. Then hit OK.

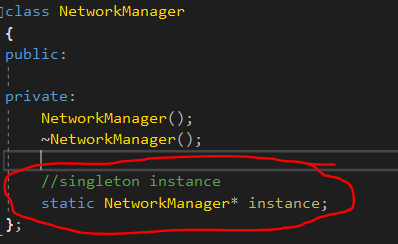


1. The first modification we must do is to change our class constructor from public to **private.**

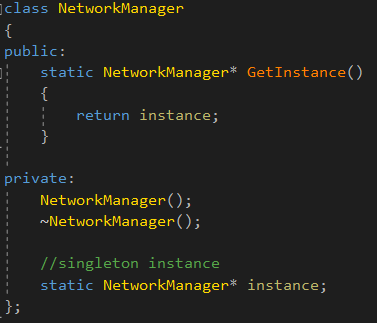


NOTE: Singletons are, as the name implies a class which has only a single instantiated copy of it in existence.

The Next thing a singleton requires is to contain its own instance.

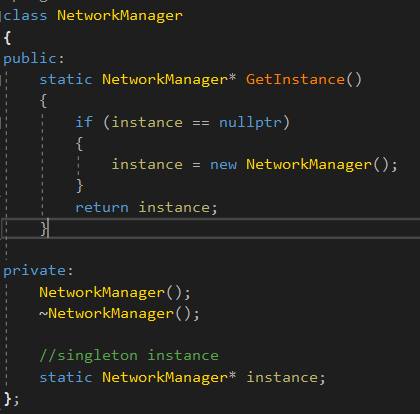
1. We create a static instance of a singleton Class like so:  
   

Since the class itself has its own instance privately, we need a way to access this class external to the class.

1. To do this, we will create a public accessor function:

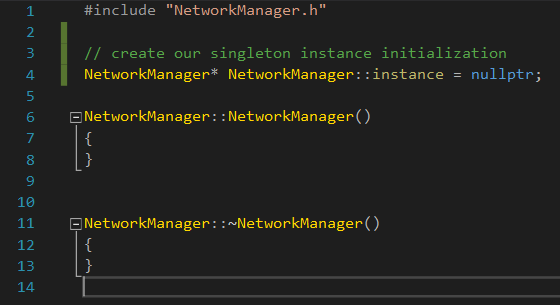
This will now give us access to the static instance from outside of the class, however, we never initialize the instance yet.

1. We can initialize a Singleton Class directly within the first call to GetInstance(). Add an if statement to check if the instance is null and if so, we instantiate it.

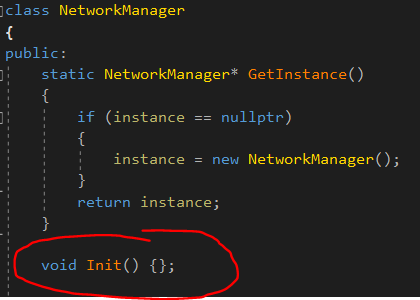
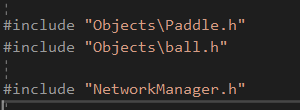


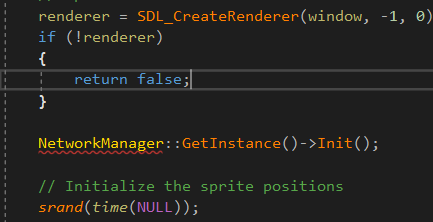
This will ensure that if we attempt to access the instance and it does not exist it creates it. This also ensures that we only ever allocate the memory once.

Note that we use the new keyword here meaning that we are manually allocating the memory for the manager here. This is because typically in games, singletons are meant to be alive throughout the games entire existence.

1. The final Step to Creating a singleton is to ensure that the static instance is assigned a nullptr to begin with. Without this our program will experience a compile error. This should be done in the cpp file of the Manager class. 

**Using our Singleton:**

1. In order to allow our singleton something to call, lets add an Init() function to our manager class which for now will do nothing. ( *we will use this in the next Walkthough* ) 
2. Now inside of our main.cpp we can include our NetworkManager.h file:
3. We can now access our NetworkManager within our main function:



**Demo your finished code to the instructor for signoff.**