CS 210 **November 3, 2016**

Notes on Separately Compile Programs

These notes are adapted from http://www.zealfortechnology.com/2013/08/create-c-program-static-library-visual-studio-2012.html.

Creating a static library using Visual Studio 2015.

Begin by creating a normal C program:

```c
#include<stdio.h>
#include<math.h>
double FindDistance(double x1, double y1, double x2, double y2);
double FindSlope(double x1, double y1, double x2, double y2);
int main()
{
    double x1, y1, x2, y2;
    double dist, slope;
    x1 = 0; y1 = 0;
    x2 = 1.0; y2 = 1.0;
    dist = FindDistance(x1, y1, x2, y2);
    slope = FindSlope(x1, y1, x2, y2);
    printf("Line from (%4.1f, %4.1f) to (%4.1f, %4.1f)\n",
            x1, y1, x2, y2);
    printf("has a length of %6.2f\n", dist);
    printf("and a slope of %6.2f\n", slope);
    return 0;
}
double FindDistance(double x1,double y1,double x2,double y2)
{return sqrt(pow(x1-x2, 2)+pow(y1-y2, 2));
}
double FindSlope(double x1,double y1,double x2,double y2)
{return (y2 - y1)/(x2 - x1);
}
```

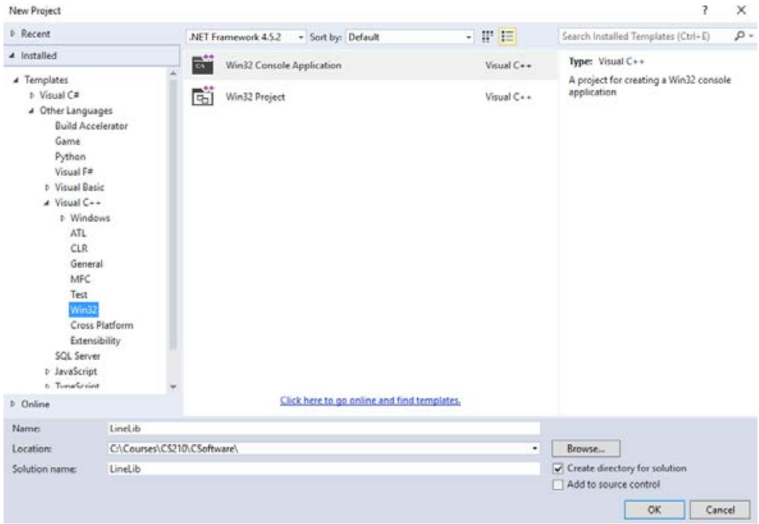
Build your program and run it to get:

```
Line from ( 0.0,  0.0) to ( 1.0,  1.0)
has a length of   1.41
and a slope of   1.00
Press any key to continue . . .
```

We will redo this project and use a static library to contain the two functions and their prototypes. The two functions and their prototypes will be contained in a *static* library and will be usable by other programs.

We will create the static library first:

Create a new project. Name it `LineLib` as shown below:



.NET Framework 4.5.2

Sort by: Default

Type: Visual C++

A project for creating a Win32 console application

Name: LineLib

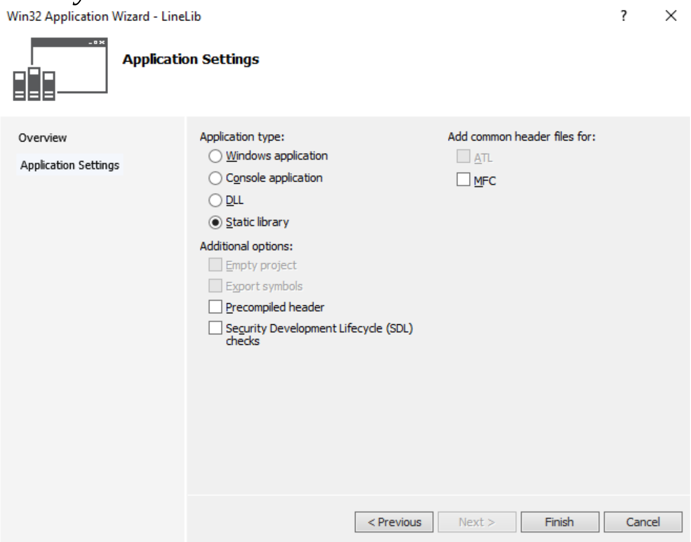
Location: C:\Courses\CS210\CSoftware\

Solution name: LineLib

☑ Create directory for solution

□ Add to source control

Click on *Next* and deselect *Precompiled header* and *Security Development*. Select the *Static library* button as shown:



Application type:

○ Windows application

○ Console application

○ DLL

● Static library

Additional options:

□ Empty project

□ Export symbols

□ Precompiled header

□ Security Development Lifecycle (SDL) checks

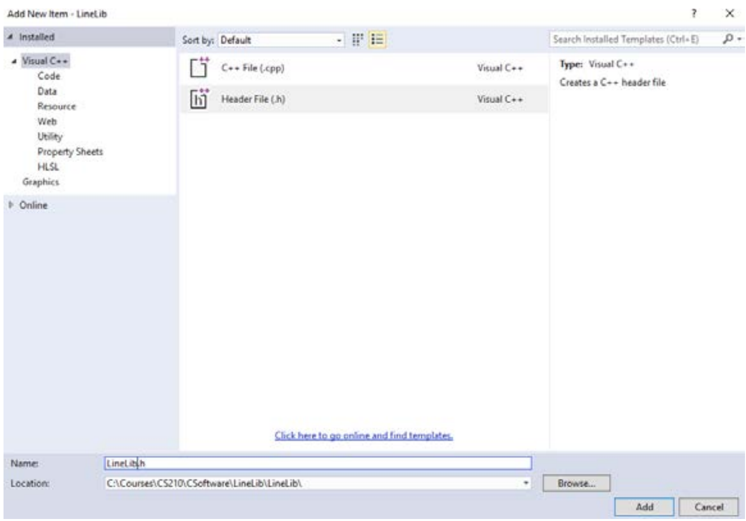
Add common header files for:

□ ATL

□ MFC

Click *Finish*.

In the Solution Explorer right click on *Header files* → *Add* → *New Item* and add a header file named `LineLib.h`


Installed
Visual C++
Code
Data
Resource
Web
Utility
Property Sheets
HLSL
Graphics
Online
C++ File (.cpp) Visual C++
Header File (.h) Visual C++
Search Installed Templates (Ctrl+E)
Click here to go online and find templates.
Browse...
Add
Cancel

Add New Item - LineLib

Sort by: Default

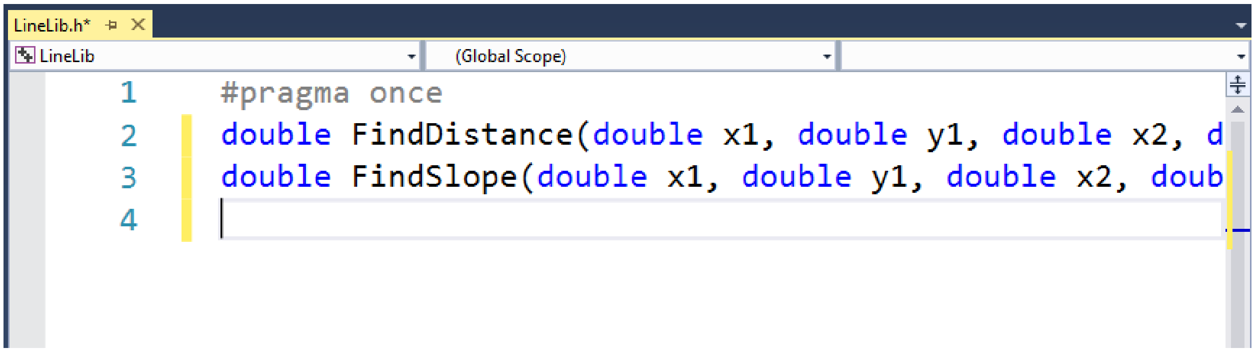
Type: Visual C++

Creates a C++ header file

Name: LineLib.h

Location: C:\Courses\CS210\CSoftware\LineLib\LineLib\

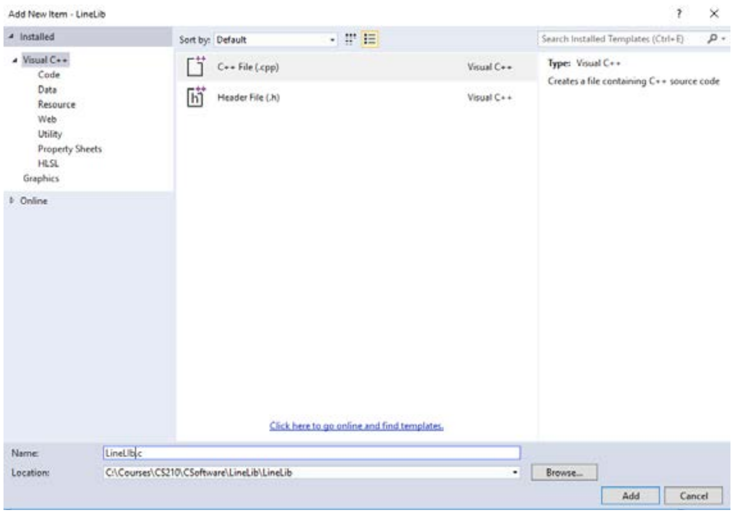
Click on *Add* and in the new header file add the following code (which you can copy from `LibDemoNormal`.



```
#pragma once
double FindDistance(double x1, double y1, double x2, d
double FindSlope(double x1, double y1, double x2, doub

```

Next, in the Solution Explorer, right click on *Source Files* → *Add* → *New Item* and add a .c file named `LineLib.c`


Installed
Visual C++
Code
Data
Resource
Web
Utility
Property Sheets
HLSL
Graphics
Online
C++ File (.cpp) Visual C++
Header File (.h) Visual C++
Search Installed Templates (Ctrl+E)
Click here to go online and find templates.
Browse...
Add
Cancel

Add New Item - LineLib

Sort by: Default

Type: Visual C++

Creates a file containing C++ source code

Name: LineLib.c

Location: C:\Courses\CS210\CSoftware\LineLib\LineLib

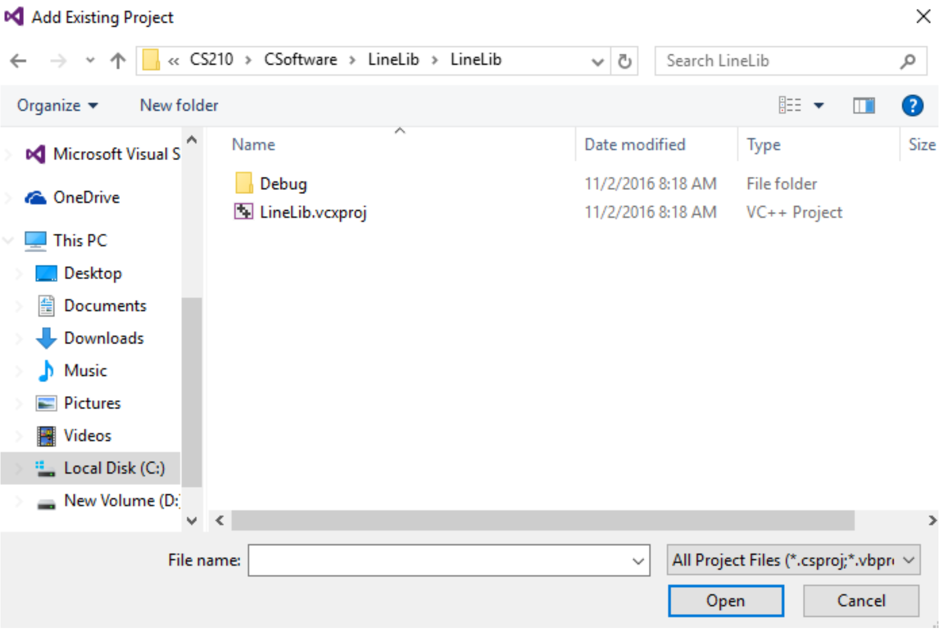
Click on *Add* and in the c-file add the following code:

```
#include "LineLib.h"
#include <math.h>
double FindDistance(double x1,double y1,double x2,double y2)
{return sqrt(pow(x1-x2, 2)+pow(y1-y2, 2));
}
double FindSlope(double x1,double y1,double x2,double y2)
{return (y2 - y1)/(x2 - x1);
}
```

At this point you can build the solution.
If you have been living a virtuous life there will be no errors. You can close the project and it can now be used with other programs.

To test our library we will create another project. Create a new project named `LineApp.` Make this an empty console application project and make sure that precompiled headers are not included.

As a first step we will add the library project to our new project. Right click on the Solution Explorer and click on *Add→Existing Projects.* Browse to the LineLib project and click on the file `LineLib.vcxproj.`

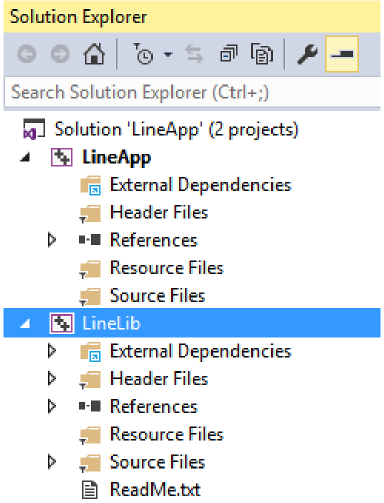


Name	Date modified	Type	Size
Debug	11/2/2016 8:18 AM	File folder	
LineLib.vcxproj	11/2/2016 8:18 AM	VC++ Project	

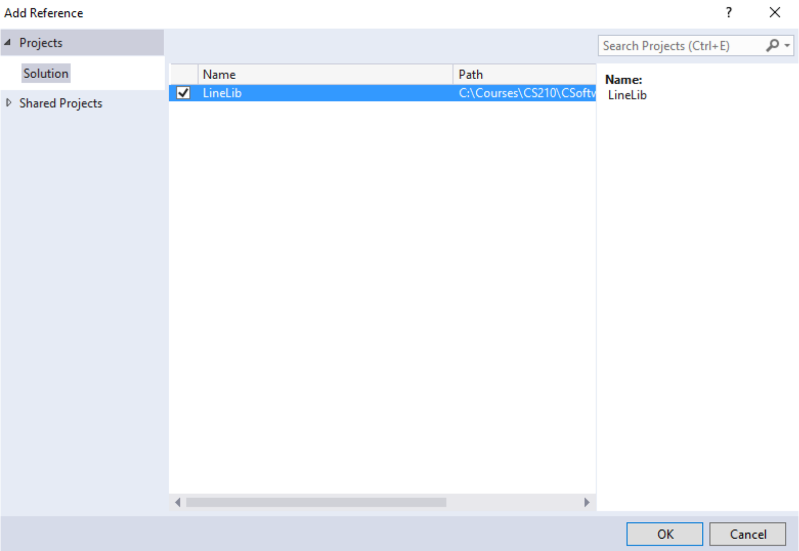
File name:

All Project Files (*.csproj;*.vbpr

Click on Open to add it to the solution. The solution should look like this:



Next we need to add a reference to the project. In the Solution Explorer right click on *References → Add New Reference*.

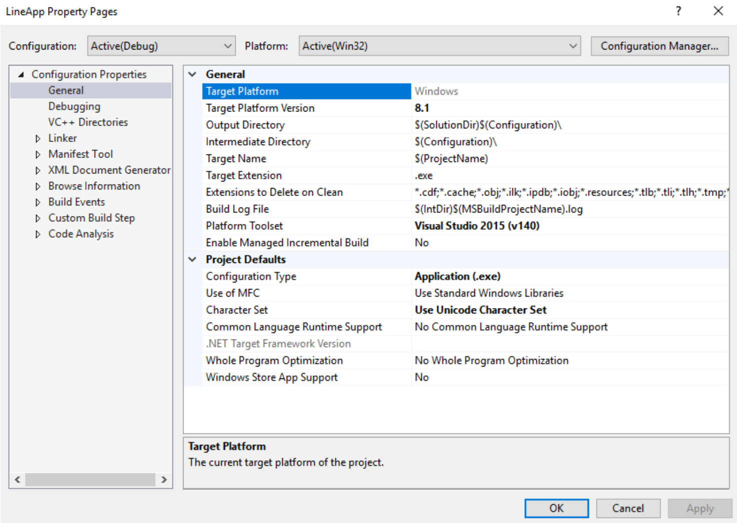


	Name	Path
☑	LineLib	C:\Courses\CS210\CSoftv

Name: LineLib

Check the box for the LineLib reference and click on OK.

In addition to adding the reference to the library we need to locate the reference so that the compiler can find it. Right click on LineApp and select Properties at the bottom of the menu.

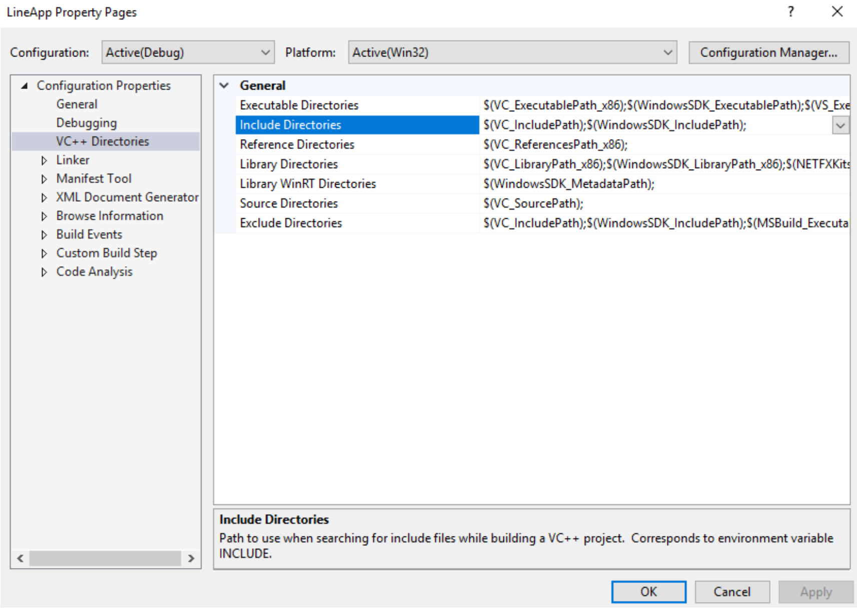


Configuration: Active(Debug) Platform: Active(Win32)

General	
Target Platform	Windows
Target Platform Version	**8.1**
Output Directory	$(SolutionDir)$(Configuration)\
Intermediate Directory	$(Configuration)\
Target Name	$(ProjectName)
Target Extension	.exe
Extensions to Delete on Clean	*.cdf;*.cache;*.obj;*.ilk;*.ipdb;*.iobj;*.resources;*.tlb;*.tli;*.tlh;*.tmp;*
Build Log File	$(IntDir)$(MSBuildProjectName).log
Platform Toolset	**Visual Studio 2015 (v140)**
Enable Managed Incremental Build	No
Project Defaults	
Configuration Type	**Application (.exe)**
Use of MFC	Use Standard Windows Libraries
Character Set	**Use Unicode Character Set**
Common Language Runtime Support	No Common Language Runtime Support
.NET Target Framework Version	
Whole Program Optimization	No Whole Program Optimization
Windows Store App Support	No

Target Platform
The current target platform of the project.

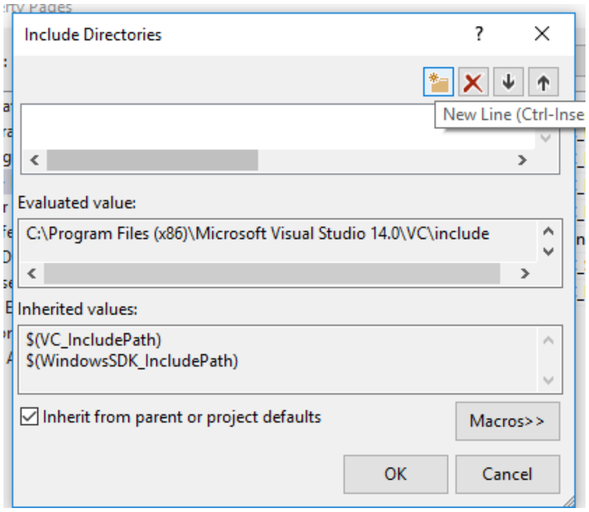
In the properties menu select *VC++ Directories* → *Include Directories*. In the pull down window click on Edit.



Configuration: Active(Debug) Platform: Active(Win32)

General	
Executable Directories	$(VC_ExecutablePath_x86);$(WindowsSDK_ExecutablePath);$(VS_Exe
Include Directories	$(VC_IncludePath);$(WindowsSDK_IncludePath);
Reference Directories	$(VC_ReferencesPath_x86);
Library Directories	$(VC_LibraryPath_x86);$(WindowsSDK_LibraryPath_x86);$(NETFXKits
Library WinRT Directories	$(WindowsSDK_MetadataPath);
Source Directories	$(VC_SourcePath);
Exclude Directories	$(VC_IncludePath);$(WindowsSDK_IncludePath);$(MSBuild_Executa

Include Directories
Path to use when searching for include files while building a VC++ project. Corresponds to environment variable INCLUDE.



Include Directories

New Line (Ctrl-Inse

Evaluated value:

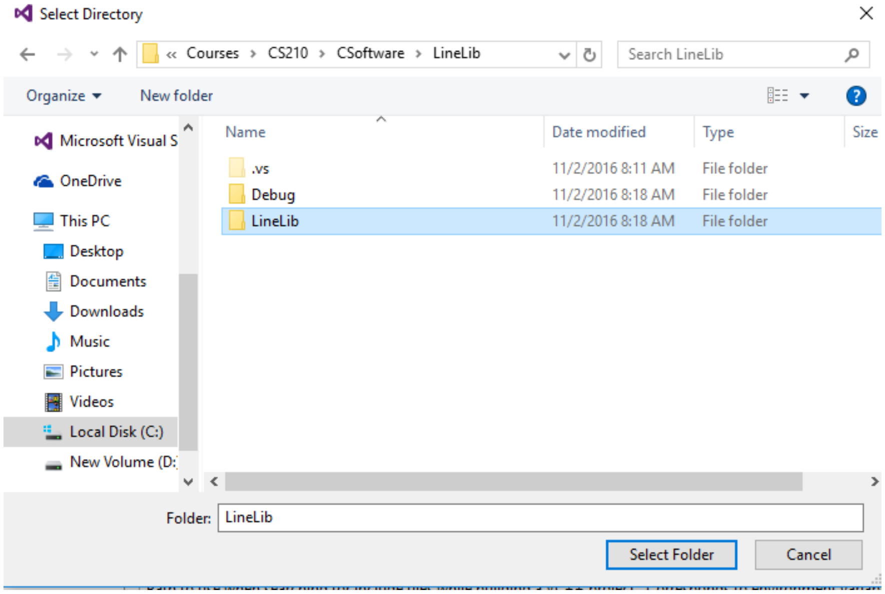
C:\Program Files (x86)\Microsoft Visual Studio 14.0\VC\include

Inherited values:

$(VC_IncludePath)
$(WindowsSDK_IncludePath)

☑ Inherit from parent or project defaults

After you click on the New Line you can click on a button with … on it to browse for a file.



Select Directory

« Courses › CS210 › CSoftware › LineLib

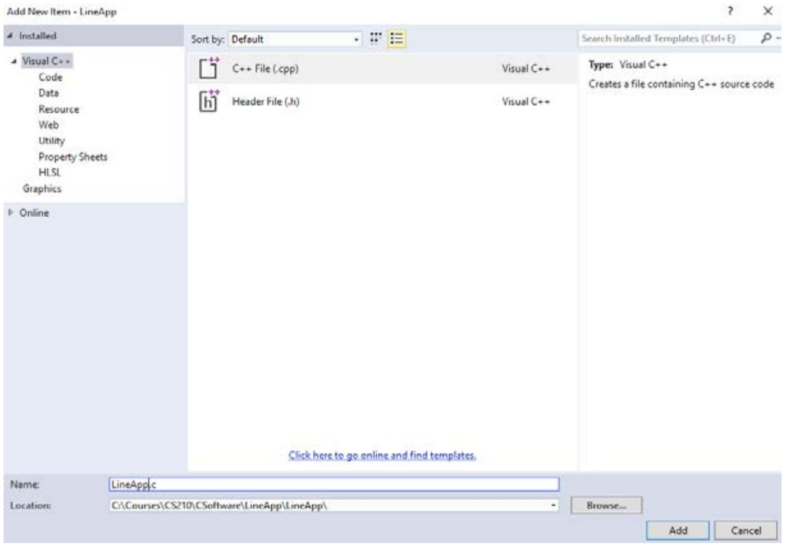
Name	Date modified	Type	Size
.vs	11/2/2016 8:11 AM	File folder	
Debug	11/2/2016 8:18 AM	File folder	
LineLib	11/2/2016 8:18 AM	File folder	

Folder: LineLib

Locate the library file in the LineLib folder. You want to drill down one more level to the second LineLib folder as shown above. Click on *Select Folder*.
Click on *OK* and then on *Apply* and then on *OK* once more.

At this point the library has been added and a path to it has been established. We can add the main program.

Right click on *Source Files* → *Add* → *New Item*. Add a new .c file named LineApp.c as shown below.



Sort by: Default

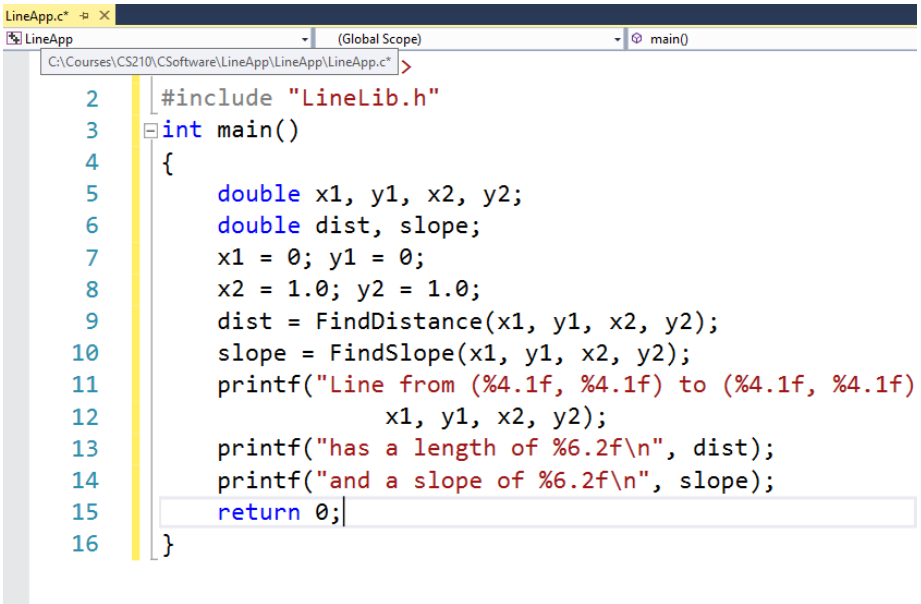
Type: Visual C++

Creates a file containing C++ source code

Name: LineApp.c

Location: C:\Courses\CS210\CSoftware\LineApp\LineApp\

In LineApp.c add the following code which you can copy from the normal program.



```
#include "LineLib.h"
int main()
{
    double x1, y1, x2, y2;
    double dist, slope;
    x1 = 0; y1 = 0;
    x2 = 1.0; y2 = 1.0;
    dist = FindDistance(x1, y1, x2, y2);
    slope = FindSlope(x1, y1, x2, y2);
    printf("Line from (%4.1f, %4.1f) to (%4.1f, %4.1f)
                x1, y1, x2, y2);
    printf("has a length of %6.2f\n", dist);
    printf("and a slope of %6.2f\n", slope);
    return 0;
}
```

You can build your program and if all goes well the build will succeed and you can run it.

The advantage of the static library is that it can be added to many other programs.