This lab exercise is a chance to gain familiarity (or refresh your memory) regarding the Microsoft *Visual Studio* Toolset, and to create a simple C++ Program.

Due Date

You must *demonstrate* the solution to this lab exercise to the instructor during class by

Saturday, September 19, 2020,

in order to receive full credit for this work.

Design the MilesPerGallon Program

Before we begin any programming task, it is a good idea to think about what we will need our program do:

- What will be the inputs to our program?
- What will be the outputs from our program?
- How would we solve this problem if we were doing it *without* a computer (that is, using **only** pencil and paper)?

What does it mean to calculate "miles per gallon" when driving a car?
A more detailed description is: "miles travelled" for some amount of "gallons of fuel used".

Miles travelled is easy enough to calculate: write down the odometer value at the beginning of your measurement, and the odometer value at the end. The distance travelled is the difference between the two.

Measuring "gallons of fuel used" is a bit tricky, because we do not have a precise value for how much fuel is remaining in the tank at any particular moment. A simple way to address this issue is:

- When purchasing fuel, always fill the tank to the top.
- Write down the odometer value each time you fill the tank.
- Also write down how many gallons you purchased each time you fill the tank.

Miles travelled between fill-ups is

$$MilesTravelled = OdometerAtLatestFillup\ - OdometerAtPreviousFillup$$

Miles-per-gallon between fill-ups is

$$MilesPerGallon = \frac{MilesTravelled}{GallonsPurchasedAtLatestFillup}$$

WARNING

The most important detail to remember about any computer work you do is to DOUBLE CHECK that your work is saved on a backup device, such as a flash drive.

In case you do not have a flash drive with you: another technique for saving your work is to compose an EMail message to yourself, and attach the source file as an attachment to that EMail. Do this *before* you shut down or reboot the PC.

Software Development Tools

If you have not already installed a working development toolset for C++ on your computer, please refer to the **Appendix** on the last page of this lab document. Most of this document assumes you are running *Visual Studio Community* on a Windows 10 computer.

REMEMBER: The *Primary Rule* for Coding and Debugging:

Always develop and test your program incrementally! That is:

1. Start by making a copy of the source code for some known, working program.
2. Copy the source code for the known, working program into the source code directory of the new project.
3. Rename the source code to the desired name for the new program. (Always use an appropriate descriptive name, to avoid confusion regarding what this program is about.)
4. Compile and run the program to be sure it works correctly.
5. Make a small change in the source code .
6. Compile and run the program to make sure it works correctly.
7. Repeat steps 5 and 6 again and again, always making *small* changes, and always testing that the program still works.

Eventually, you will have implemented the new program. The advantage of using this incremental approach, as opposed to typing in lots of new code all at once, is that it makes it easier for you to correct your mistakes as you go.

Know where your files are!

Be sure you are familiar with using the **File Explorer** tool on a Windows computer. On ***Windows 10***, you can easily start the **File Explorer** tool by clicking the small icon (on the task bar) that looks like a file folder. Experiment with the menus until you feel that you can “find your way around”. (If you are using a **Mac** or a **Linux** computer, then you should become familiar with whatever file management tools that system provides.)

It is important to know the **actual location** of your *Visual Studio* project. Very often students just save their project on the “desktop”, or blindly trust *Visual Studio* to choose a default location. Then, when they need to access the actual Windows files for their project, they have difficulty finding the correct folder.

It is also important to give a unique name to each *Visual Studio* project that you create. For example, you might name the project for this lab exercise “**MilesPerGallon**”. However, if you have difficulty and need to start over for whatever reason, it would be smart to name the *second* attempt "**MilesPerGallon2**". This will help avoid confusion.

Software Tools that You Will Need to Complete this Lab Exercise:

- **File Explorer**: to create directories, as well as to copy and rename the source code file.
- A **browser**: to login to the *Moodle* Learning Management System and download the sample code.
- **Visual Studio**: to edit, compile and execute your program.

Prepare a Folder on Your Hard Drive

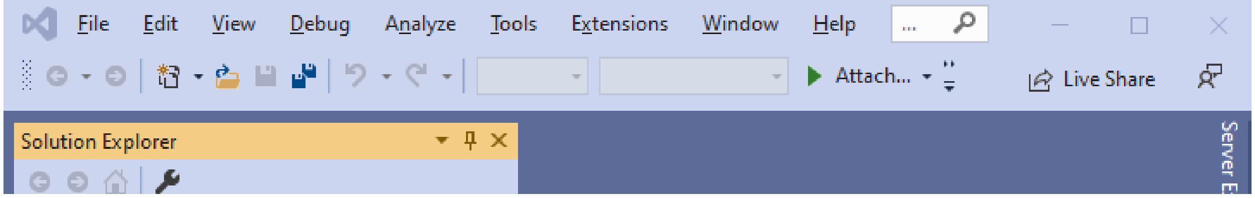
8. In the **File Explorer** window, navigate to the root directory of the C-drive (**C:**\) and create a folder named "**CSC237**".
9. Inside the "**CSC237**" folder, create a folder named "**Lab**", and open that folder.
10. Inside the "**C:\CSC237\Lab**" folder, create another folder named "**Lab02a_MilesPerGallon**". (In this course, we follow the convention of numbering Lab exercises to correspond to the relevant chapter in the textbook.)

Basic Mechanics of Using Microsoft *Visual Studio*

11. Start the *Visual Studio* environment **either** by double-clicking the shortcut on the desktop (if present) **OR** by following menus (more or less) like the following:

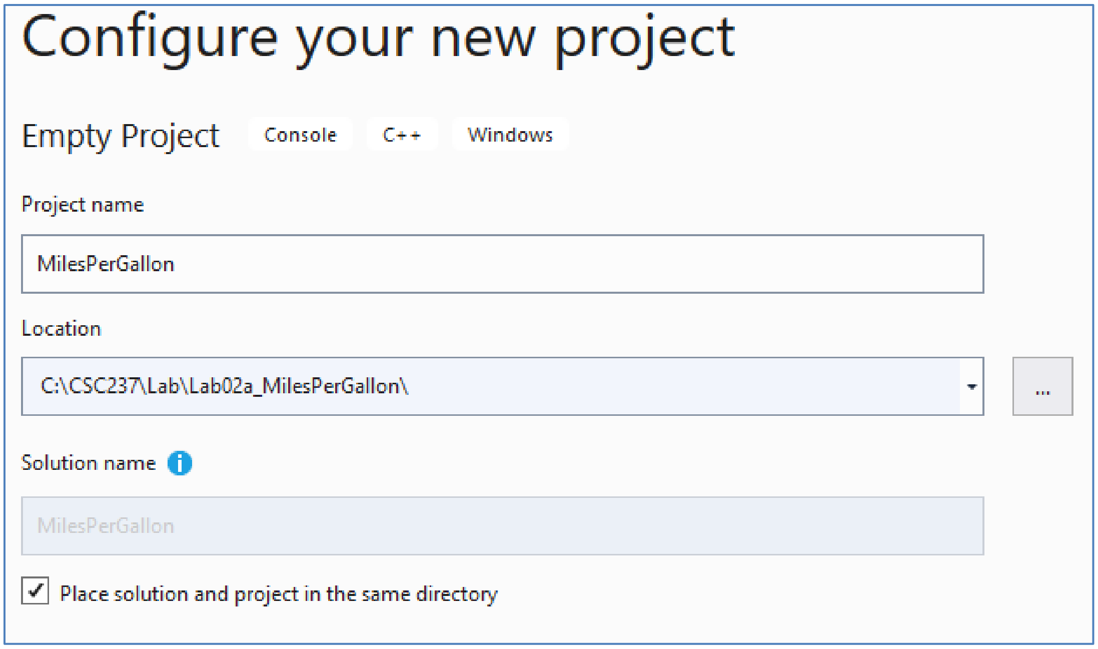
    Click on the **Start** button (lower-left corner) and scroll down to the "V" section, where you will see the icon for ***Visual Studio***. Click on that ***Visual Studio*** icon.

12. When *Visual Studio* opens, click the "**Continue Without Code→** " link near the bottom of the screen. You should see a window open that looks something like the following:



13. Click the "**File → New→Project**" menu. You will see the "**Create a new project**" screen open: Select "**Empty Project**", and click "**Next**".
14. Next you will see the "**Configure your new project**" screen open:
    - In the Project name field, enter "`MilesPerGallon`".
    - Click the "**…**" button (to the right of the Location field).
    - Navigate to the "`C:\CSC237\Lab\Lab02a_MilesPerGallon`" folder, and click "**Select Folder**".

    Your screen should now contain the following:



Configure your new project

Empty Project

Project name: MilesPerGallon

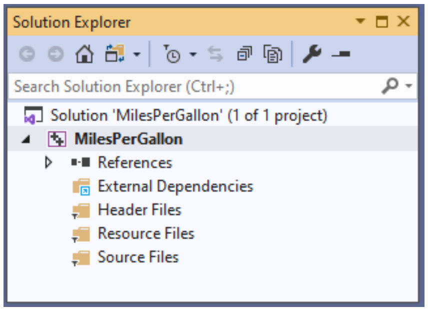
Location: C:\CSC237\Lab\Lab02a_MilesPerGallon\

Solution name: MilesPerGallon

☑ Place solution and project in the same directory

- Verify that the Project name field contains "**`MilesPerGallon`**".
- Verify that the Location field contains "**`C:\CSC237\Lab\Lab02a_MilesPerGallon`**".
- Verify that the "Place solution and project in the same directory" box **IS** checked.
- Click "**Create**". (Wait while *Visual Studio* initializes the project.)

The *Visual Studio* application should show a "**Solution Explorer**" window that looks something like:



Solution 'MilesPerGallon' (1 of 1 project)
MilesPerGallon
References
External Dependencies
Header Files
Resource Files
Source Files

(If you do not see the "Solution Explorer" window, click the **View** ⟶ **Solution Explorer** menu.)

Switch to your Browser and use *Moodle* to Download the Sample Program:

15. Login to *Moodle*, open the page for this course:
    - Click on the **Sample Code** topic.
    - Click on **Ch01_sample_code** and download the **Ch01_sample_code.zip** file.
    - If your chosen browser allows you to specify the download destination directory, choose the "**`C:\CSC237\Lab\Lab02a_MilesPerGallon\MilesPerGallon`**" directory.
    - If your browser does *not* allow you to specify the download destination directory, then open the **File Explorer** tool and move the downloaded file (**Ch01_sample_code.zip**) to the "**`C:\CSC237\Lab\Lab02a_MilesPerGallon\MilesPerGallon`**" directory.

Switch to File Explorer to Extract and Rename the Sample Source Code:

- Using the **File Explorer** program, navigate to the "`C:\CSC237\Lab\Lab02a_MilesPerGallon\MilesPerGallon`" directory and open the ZIP file.

16. Right-click on the "**Program_1-1.cpp**" file (inside the ZIP file) and select "**Copy**".
17. Click on the "`C:\CSC237\Lab\Lab02a_MilesPerGallon\MilesPerGallon`" directory, point to an "empty" portion of the window, right-click and select "**Paste**".
    (Observe that the "Program_1-1.cpp" file is now present in the "`C:\CSC237\Lab\Lab02a_MilesPerGallon\MilesPerGallon`" directory.)
18. In the "`C:\CSC237\Lab\Lab02a_MilesPerGallon\MilesPerGallon`" directory, right-click on the "**Program_1-1.cpp**" file and select **Rename**. Rename the file to "**MilesPerGallon.cpp**".

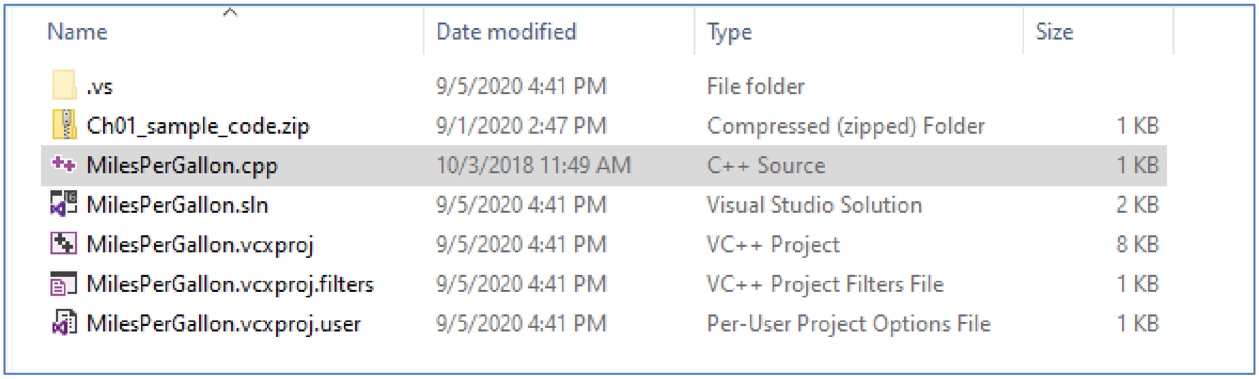
Switch to Visual Studio:

19. Inside the Visual Studio application, click the **View ⟶ Solution Explorer** menu:

    In the "**Solution Explorer**" window, there is a folder named "Source Files".

    Right-click "Source Files" and select **Add ⟶ Existing Item...**

    Click on the "MilesPerGallon" **C++ File (**`MilesPerGallon.cpp`**),** and click "**Add**".

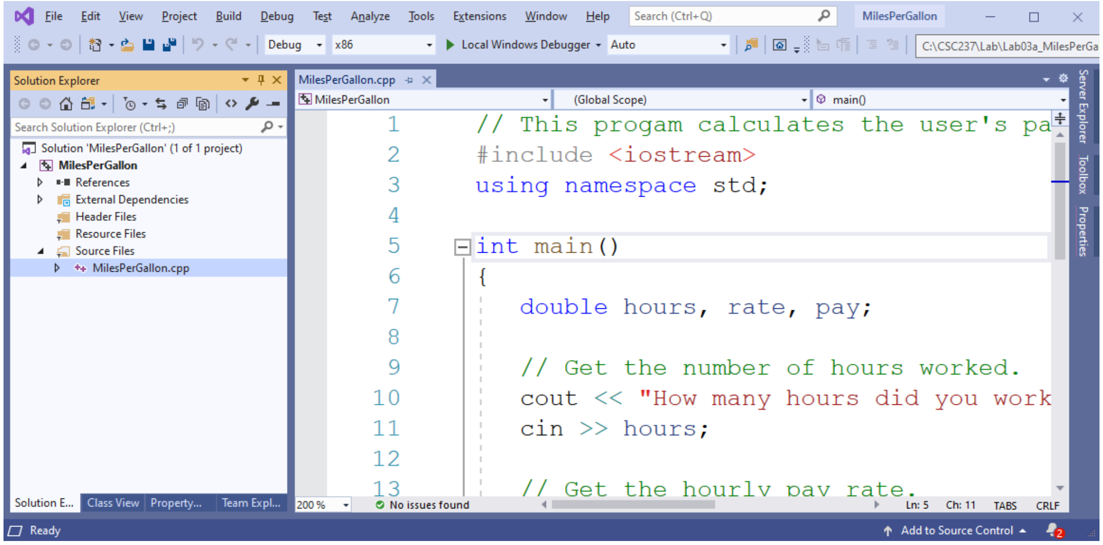


Name	Date modified	Type	Size
.vs	9/5/2020 4:41 PM	File folder	
Ch01_sample_code.zip	9/1/2020 2:47 PM	Compressed (zipped) Folder	1 KB
MilesPerGallon.cpp	10/3/2018 11:49 AM	C++ Source	1 KB
MilesPerGallon.sln	9/5/2020 4:41 PM	Visual Studio Solution	2 KB
MilesPerGallon.vcxproj	9/5/2020 4:41 PM	VC++ Project	8 KB
MilesPerGallon.vcxproj.filters	9/5/2020 4:41 PM	VC++ Project Filters File	1 KB
MilesPerGallon.vcxproj.user	9/5/2020 4:41 PM	Per-User Project Options File	1 KB

(**NOTE**: the "`.cpp`" filename suffix may not be visible on your computer, but the "Type" column will clearly indicate something like "**CPP File**" or "**C++ Source**".)

Important "**Best Practices**" note:
Whenever you are choosing the file name for a source code file, do **NOT** use the default file name "Source". Instead, choose a descriptive file name appropriate for the task at hand.

20. In the **Solution Explorer** window, right click on the new source file: **MilesPerGallon.cpp** and select "open".
21. Observe the *Visual Studio* main screen, now containing a tab labeled "`MilesPerGallon.cpp`". At this moment, the *contents* of this file are the instructions for the sample program "**Program_1-1**":



```
// This progam calculates the user's pa
#include <iostream>
using namespace std;

int main()
{
    double hours, rate, pay;

    // Get the number of hours worked.
    cout << "How many hours did you work
    cin >> hours;

    // Get the hourly pay rate.
```

Verify that the *Visual Studio* Project is Configured Correctly

Before making any changes to this program, it is a good idea to compile and run the program, just to verify that we have done everything correctly so far. (This is really a check of your ***Visual Studio*** project.) If the program does not run, then you probably made some error when you were setting up the ***Visual Studio*** project.

22. To compile and run the program, click the green triangle in the ***Visual Studio*** toolbar: ▶.
23. When the dialog box appears, asking you if it should re-build the program, click **Yes**. (The program should compile with no errors.)
24. A window will open and display the prompt, asking you to enter how many hours you worked, and what your hourly pay rate is. Enter input values as shown in the example below. Verify that the results look correct. (In the sample output shown below, text that the user types is shown in **BOLD** font. When the program actually runs, all text is shown in the same font.)

Sample output from unmodified copy of the sample program "Program_1-1":

```
How many hours did you work? 32.5
How much do you get paid per hour? 15.75
You have earned $511.875

C:\CSC237\Lab\Lab02a_MilesPerGallon\MilesPerGallon\Debug\MilesPe
rGallon.exe (process 6628) exited with code 0.
To automatically close the console when debugging stops, enable
Tools->Options->Debugging->Automatically close the console when
debugging stops.
Press any key to close this window . . .
```

Now that we have a working *Visual Studio* project that contains a known, working program, we can begin entering our new code.

Begin Writing and Testing *Your* Code

Before you begin coding, think about the structure of the sample (starting code) that you currently have in your **MilesPerGallon.cpp** file:

- It defines several variables of type **double**.
- It contains prompts for the user, and inputs values for several variables.
- It calculates the result and prints it to the screen.

This “starter” program is certainly not the final **MilesPerGallon** code that you need, but it does have a structure very similar to what you need. You can use that similar structure as a sort of “template” for your solution, and in the process avoid many of the annoying little errors that we all tend to make when we write code “from scratch”.

Always remember: make small, incremental changes. Test each small change as you go.

Sample Output

Your final working solution should produce output similar to the examples below. In the sample output shown below, text that the user types is shown in **BOLD** font. When the program actually runs, all text is shown in the same font.)

Sample Input / Output – Example 1

```
Enter Odometer at PREVIOUS fill-up: 56100
Enter Odometer at LATEST fill-up: 56600
Distance travelled = 500.0 miles.
Enter gallons purchased at LATEST fill-up: 12.2
Your fuel economy is: 41.0 MPG.

C:\CSC237\Lab\Lab02a_MilesPerGallon\MilesPerGallon\Debug\MilesPerGallon.exe
(process 2628) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options-
>Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

Sample Input / Output – Example 2

```
Enter Odometer at PREVIOUS fill-up: 95125
Enter Odometer at LATEST fill-up: 95602
Distance travelled = 477.0 miles.
Enter gallons purchased at LATEST fill-up: 11.4
Your fuel economy is: 41.8 MPG.

C:\CSC237\Lab\Lab02a_MilesPerGallon\MilesPerGallon\Debug\MilesPerGallon.exe
(process 13120) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options-
>Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

Demonstrate the Working Program to the Instructor

Demonstrate the working program to the instructor. Be sure to save a copy of the source file (**`MilesPerGallon.cpp`**) in a safe place for future reference.

Appendix: Software Development Tools

Windows 10 users: Installing Visual Studio for C++ on a Windows 10 Computer

If you have not already installed Visual Studio on your computer, go to the following web page:

https://docs.microsoft.com/en-us/visualstudio/install/install-visual-studio?view=vs-2019

Follow the instructions for installing **Visual Studio Community** on your computer.

When the instructions direct you to **Choose Workloads**, choose “Desktop Development with C++”.

MacIntosh users: Developing C++ Programs on a MacIntosh Computer

Most students who use a MacIntosh computer choose XCode:

https://developer.apple.com/xcode/

https://www.cprogramming.com/xcode.html

There is a short video on *Moodle* that will help you get started using XCode:

- Open the *Moodle* page for this course.
- Look in the **Tools** folder.
- Download the Zip file in the **XCODE_CPP_video** resource.