This lab exercise is a chance to review, or gain familiarity with, the Microsoft *Visual Studio* Toolset, and to add some enhancements to the LabIntro "Starter Code": **`LabIntro.cpp`**.

The "Starter Code" is a working program that simulates a very simple bank account. (As we will see, the program does have one small bug.) The assignment is to create a new Visual C++ "project" for the lab exercise, verify that the "Starter Code" works almost correctly, and then add several easy enhancements.

Due Date

You must demonstrate the solution to this lab exercise to the instructor by **Wednesday, September 11, 2019**, in order to receive full credit for this work.

WARNING

The most important detail to remember about any computer work you do during class is to DOUBLE CHECK that your work is saved on your flash drive. Any files saved on the computer hard drive will be lost when the machine reboots, which is every night.

In case you do not have a flash drive with you: another technique for saving your work is to compose an e-mail message to yourself, and attach the source file (**`LabIntro.cpp`**) as an attachment to that e-mail. Do this *before* you leave the room or reboot the PC.

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.)

If you have your flash drive with you, then the instructions in the next section will help you create a dedicated folder for the CIT237 lab exercises.

Regardless of whether or not you are using a flash drive, 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 you create. For example, you might name the project for this lab exercise "**LabIntro**". However, if you have difficulty and need to start over for whatever reason, it would be smart to name the *second* attempt "**LabIntro2**". This will help avoid confusion.

Preparing a Folder on Your Flash Drive

1. Insert your Flash Drive.
2. You may see a message asking you what to do when a removable disk is inserted: choose "**Display Files**".
   Otherwise, Right-click the Start button (lower-left corner of screen) and click **File Explorer**.
3. Look for the flash drive on the File Explorer display, for example: "**Removable Disk (F:)**".
4. Double-click the icon for the Flash Drive.
5. In the File Explorer window, create a folder named "**CIT237**" on the Flash Drive, and open that folder.
6. Inside the "**CIT237**" folder, create a folder named "**Lab**", and open that folder.
7. Inside the "**CIT237\Lab**" folder, create another folder named "**Intro**".

Downloading the "Starter Code" from Moodle

8. Open a browser and navigate to the web page: https://www.bhcc.edu/
9. There are a number of links displayed across the top of the screen. Click on the "**Moodle**" link.
10. In the upper-right corner of the display you will see a message such as:

    You are not logged in. (Log in)

11. Click on the "Log in" link, and use your Webadvisor username and password. (This is the ***same* username** and **password** that you use for your BHCC E-mail.)
12. There should be menus on the left side of the screen, including one with the label "**My courses**". Click on the link for **CIT-237**.
13. ***If*** you cannot find the "**My courses**" menu on the left side of the screen, look for a section on the right side of the screen labeled "**NAVIGATION**". Expand the menus until you see the "**My courses**" menu. Click on the link for **CIT-237**.
14. In the section for today's date , click on the link labeled "**CIT237_Lab_Intro_...**".
15. Observe a dialog box, asking you what you want to do with the file "**CIT237_Lab_Intro_....zip**".
16. Click "**Save File**" (or "**Save As**" if that is what your computer has).
17. If you are using a flash drive, navigate to the "**CIT237\Lab\Intro**" folder on your Flash drive. Click "**Save**".
18. As a safety check to prevent confusion, **write down the folder** where the downloaded file is saved:

    Download folder: _____

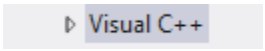
Basic Mechanics of Using Microsoft *Visual Studio*

19. 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) followed by the "**search**" button in the upper-right corner. In the search field, type **`Visual Studio 2017`**

    (Note: the "search" button is actually an icon that looks like a magnifying glass.)

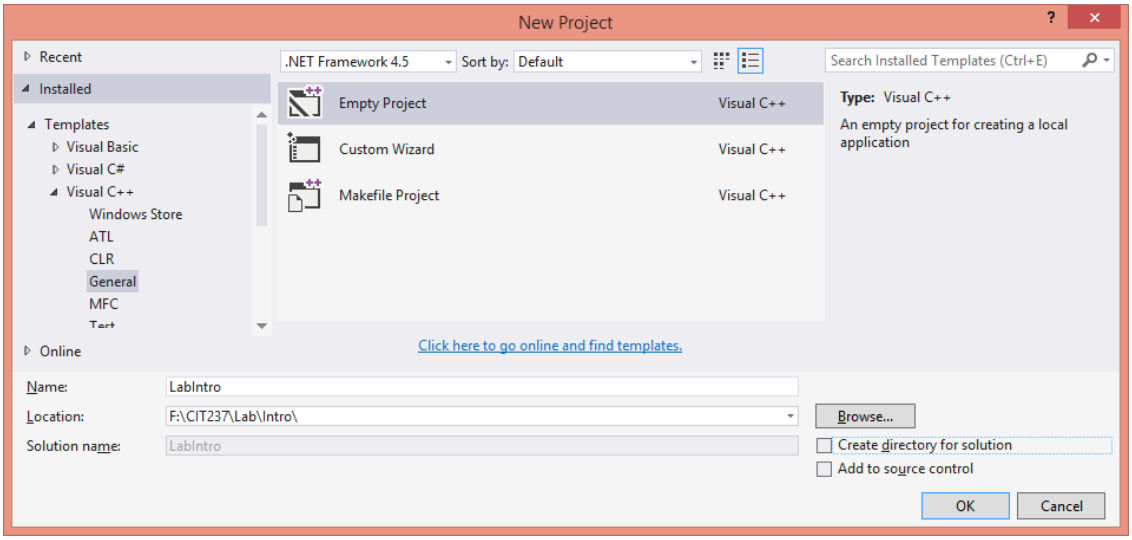
20. When *Visual Studio* opens, click "**New Project...**". A "New Project" dialog opens.
    - On the left side of the dialog screen, expand the menu for "**Visual C++**" by clicking the small triangle:

    - Click "**General**"
    - In the center portion of the screen, click "**Empty Project**":

    - Enter "**`LabIntro`**" in the Name field.
    - Click the "**Browse...**" button (to the right of the Location field) and navigate to the **drive letter** that you identified in step 3 on the previous page. (That is, be sure you are creating the project ***on your Flash Drive***.)
    - Navigate to the "**CIT237\Lab\Intro**" folder, and click "**Select Folder**"
    - Check that both the Name and Solution Name fields contain "**`LabIntro`**" and that the "Create directory for solution" box is ***not*** checked.
    - Verify that the Location field contains the path to the desired folder ***on your Flash Drive***. (If you are not using a Flash Drive, carefully note the exact location of your project.)
    - The window should look (more or less) like the following:

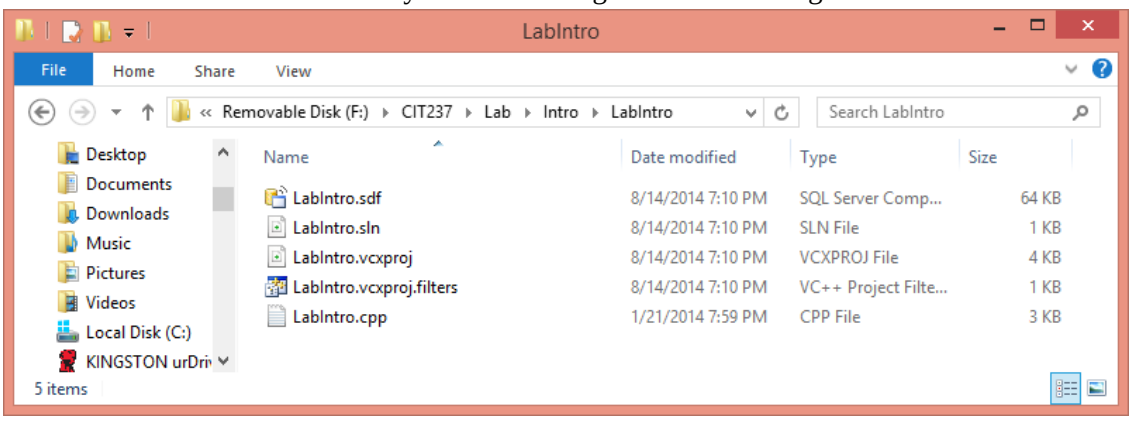
      

    - Click "OK". (Wait while *Visual Studio* initializes the project.)

21. Return to the **File Explorer** window.
22. Using **File Explorer**, copy the source file for the "starter code", **`LabIntro.cpp`** , from the ZIP file to the following folder:

    F:\CIT237\Lab\Intro\LabIntro (where "**F:**" is the drive letter for your flash drive).

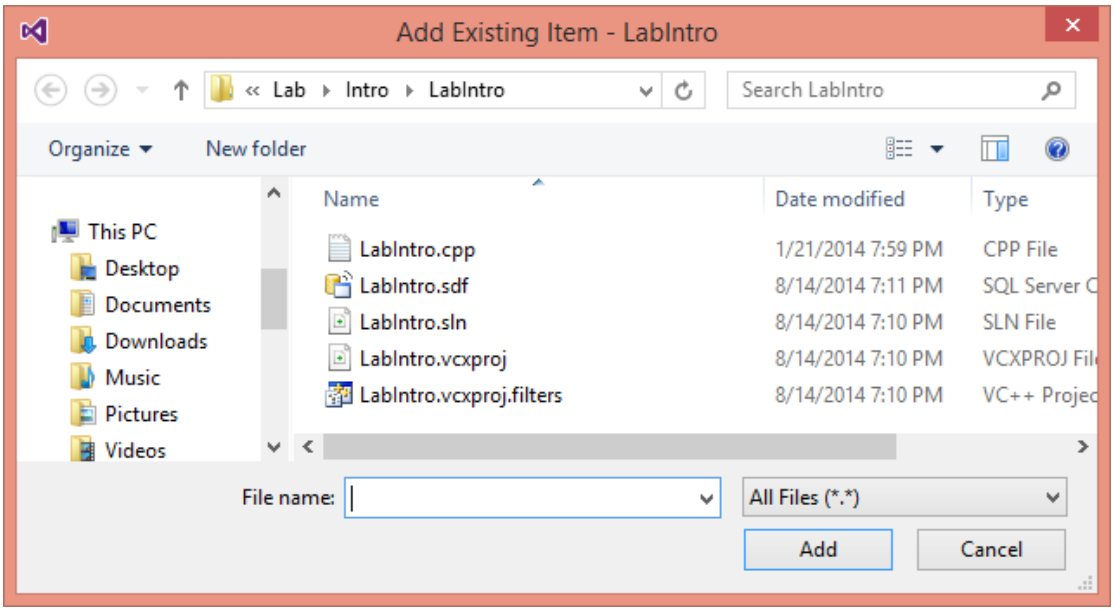
    The contents of this folder may look something like the following:



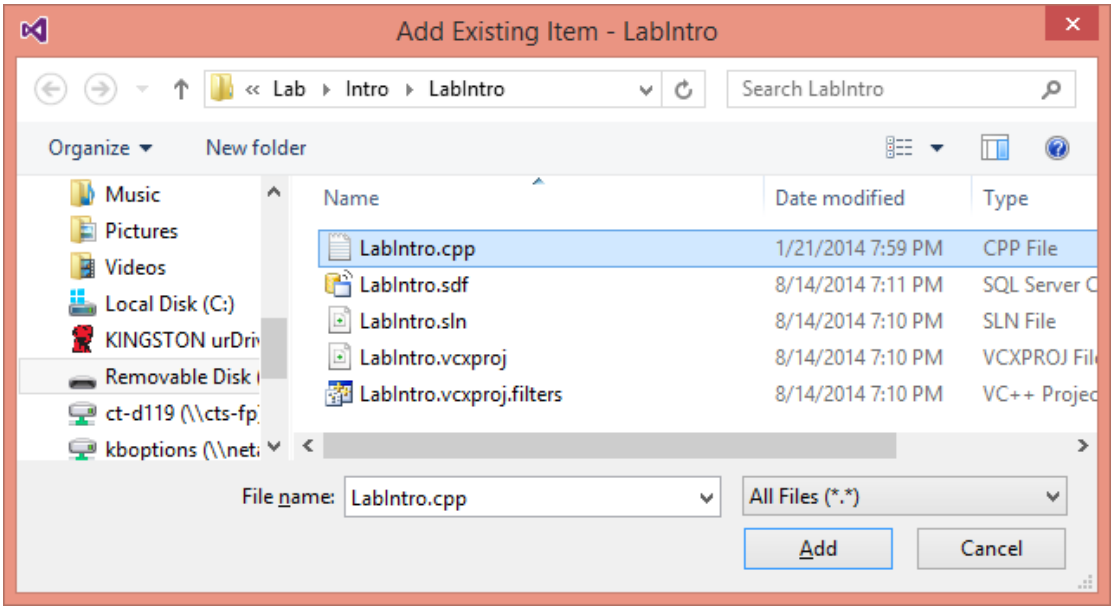
23. Return to the **Visual Studio** window:

    In the "**Solution Explorer**" window, containing the "**LabIntro**" project, there is a folder named "Source Files". **Right-click** "Source Files" and select **Add → Existing Item...**

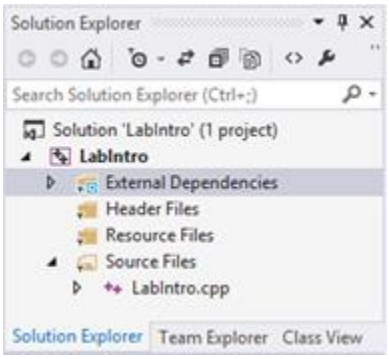
    Select the file name of the source code: `LabIntro.cpp`



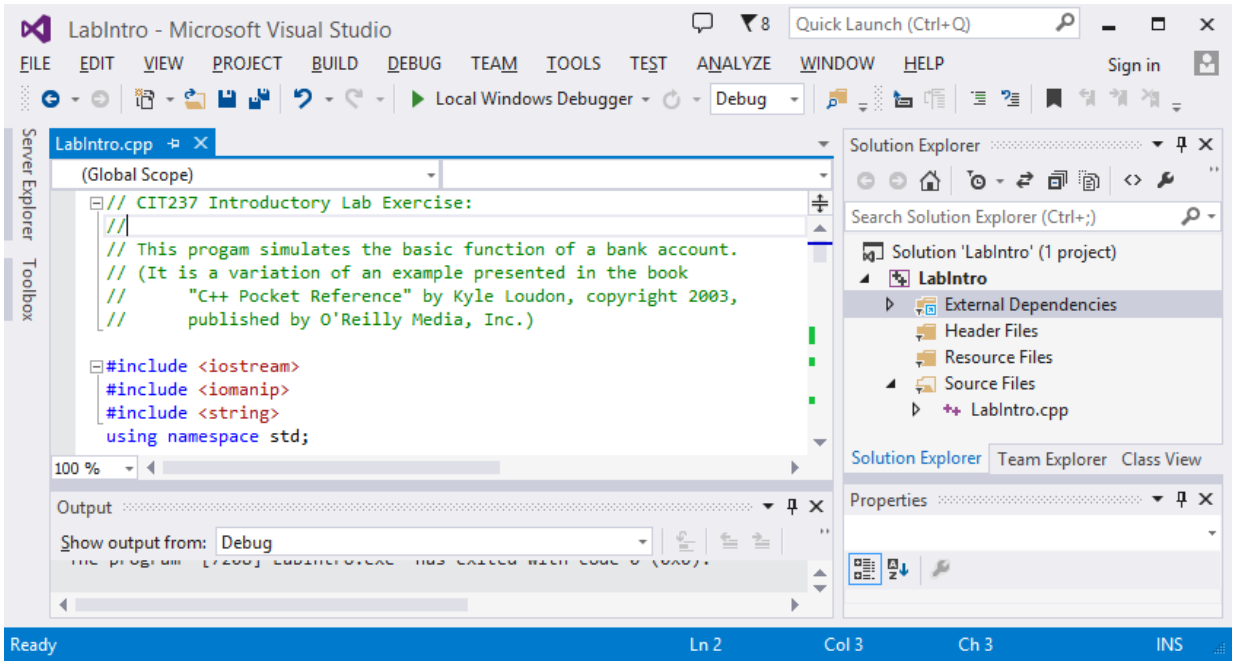
    and click "**Add**". (**NOTE**: the "**`.cpp`**" filename suffix may not be visible on your computer, but the "Type" column will clearly indicate something like "**CPP File**" or "**C++ Source File**".)



24. In the "Solution Explorer" pane of the Visual Studio window, double-click the file name "**`LabIntro.cpp`**" in order to view or edit the source file.

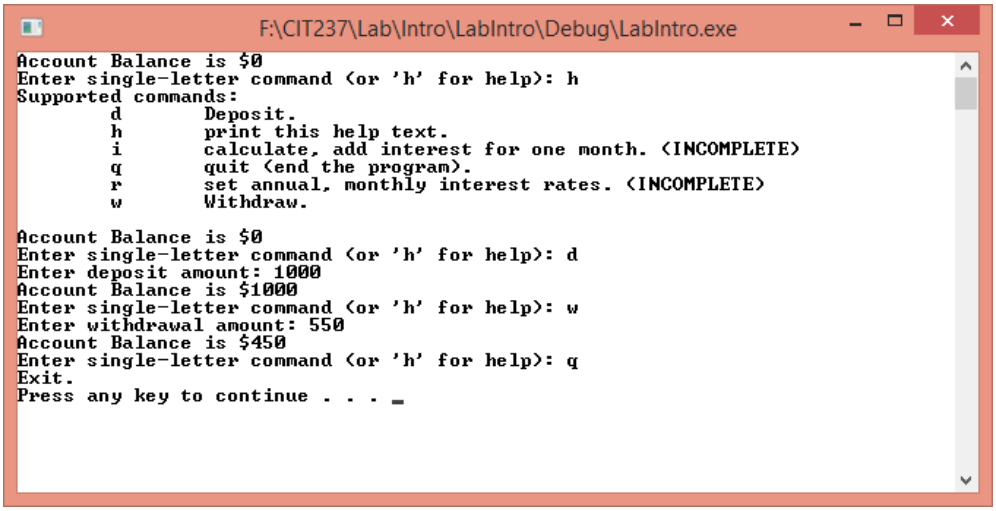


25. Observe the Visual C++ main screen, now containing a tab labeled "**`LabIntro.cpp`**":



26. Above the source code display, you should also see the words: "**(Global Scope)**". If not, then you may have made an error setting up your project. (Usually, the best way to fix this is to START OVER, and repeat all steps in this procedure, except give the new *Visual Studio* project a slightly different name, such as "**LabIntro2**". This will help avoid confusion.)
27. Build the **`LabIntro.cpp`** program by clicking the green triangle in the toolbar ( ), or by typing CTRL+F5.
28. Observe build status, including any error messages, in the "Output" window.
29. When the program builds successfully, it will run in a console window.
30. Test the various commands, to verify that they work:
    - 'h' to view the "help" text,
    - 'd' to make a deposit,
    - 'w' to make a withdrawal.
    - Try withdrawing an amount larger than the current balance. (What is wrong with the result?)

- Finally, test the 'q' command, and observe that the program exits.



Program Enhancement Exercises

Now that you have a working program, hopefully without too much trouble, let's add some enhancements to the program.

Test each enhancement after coding it. (Enhancements **1** through **3** are required. Enhancement **4** is optional.)

Enhancement 1: Prevent Overdrafts on the Bank Account (required)

As you experimented with the "starter code", you no doubt noticed that this bank account program lets you withdraw money you do not have, and run a negative balance. Fix this bug by adding a check to the "withdrawal" command, to verify that the current balance is greater than or equal to the amount requested to be withdrawn. If the current balance is too small, output an "Insufficient funds" error message, and keep running.

That is, you will need to replace the statement:

```
balance -= amount;
```

with something like:

```
if (balance >= amount)
{
     balance -= amount;
}
else
{
     cout << "Insufficient funds." << endl;
}
```

Enhancement 2: Set Annual Interest Percentage (required)

Add support for a new command (the '**r**' command): the code to check if the user has entered the '**r**' command is already there, but it does not do anything yet. You need to add code which does the following:

- Prompts the user to enter the annual interest percentage.
- Reads a double value from the keyboard, and sets the **annualInterestRate** variable.
- Calculates and sets the **monthlyInterestRate** and **monthlyInterestCoefficient** variable based on the following formula:
  monthlyInterestRate = annualInterestRate / MONTHS_PER_YEAR
  monthlyInterestCoefficient = monthlyInterestRate / 100

Enhancement 3: Calculate Interest for One Month, Add to Balance (required)

Add support for a new command (the '**i**' command): the code to check if the user has entered the '**i**' command is already there, but it does not do anything yet. You need to add code which does the following:

- Calculates a value for the **interestThisMonth** variable, based on the following formula:
  interestThisMonth = balance * monthlyInterestCoefficient
- updates the balance variable by adding **interestThisMonth**.

Enhancement 4: Format the Output (optional challenge)

Special Note: One important skill of a computer programmer is to be a good researcher: it is important to be willing and able to search for answers to questions you may have about solving a problem.

- The topic related to this enhancement is discussed in section **3.7,** "Formatting Output" of the Gaddis textbook (pages 110 – 120).
- *Another approach* is use the web site http://www.cplusplus.com/.
  Click on the **Reference** section and search for "**`setprecision`**".

Add Stream Manipulator Statements so that the balance and other floating point numbers always display with two decimal places (indicating dollars and cents).

For example, instead of seeing output like: `Account balance is 1000`

we would prefer to see: `Account balance is $1000.00`

Demonstrate the Working Program to the Instructor

Show the instructor your working code to receive credit for the lab assignment.

REMINDER: Always make sure your work is saved on your **Flash Drive**. As previously stated, any personal data files left on the classroom PC hard drives will be deleted overnight.

If you do not have a Flash Drive with you, one alternative is to compose an e-mail message to yourself, and attach the source file (**`LabIntro.cpp`**) as an attachment.