CS-181 Lab 1

Student Learning Outcomes

- Installation and setup of MS Visual Studio or MS Visual Studio Code (Mac users)
- Introduction to the Microsoft Visual Studio Integrated Development Environment (IDE)
- Introduction to creating a C++ project in MS Visual Studio
- Using the code editor
- Sequential code
- Console input and output

Overview

This lab provides an introduction to programming in C++. You will complete a guided exercise that will help you develop the skills you need for creating a C++ project in MS Visual Studio, using the code editor and running a C++ application. You are given the code but this is an opportunity to start developing skills you will need to complete the rest of the programming assignments given in this class. This lab also gives you an opportunity to apply some of the terminology and concepts presented in chapter 1 as well as a little "sneak peek" introduction to chapter 2.

In order to complete the labs in this class, you will need to download and install Microsoft Visual Studio. The Microsoft Visual Studio Community edition is free and it's sufficient for completing the labs assigned in this class. Visual Studio Link. When you install Visual Studio, ***make sure you select C++ as one of the options to install***. ***Beware***! C++ is not selected by default!

In the "real world", as a professional software developer, you will normally be responsible to install and configure a working software development environment to do your work so this lab gives you a practical opportunity to develop that skill.

Mac users: There is a version of Visual Studio available for Mac. Beware! It does ***not*** contain C++! Instead, you will need to download and install Visual Studio Code (also a free download from Microsoft). Install steps for Mac users:

1. Install Visual Studio Code for Mac. You may find this installation tutorial link helpful.

2. Open terminal and type *clang --version* . If it shows a version of clang (the Mac C++ compiler), you can continue. Otherwise, type *xcode-select --install* to install clang.

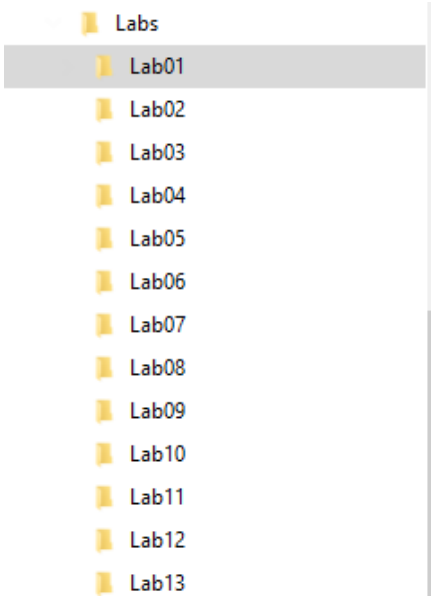
3. Open VS Code, open the "Extensions" tab, and install both the C/C++ extension and Easy C++ Projects

4. Open your folder, press the F1 key, type *easy c++* then select "Create new C++ Project"

5. Under the "src" folder, click "main.cpp" and edit your code there.

6. You can run your code at any time by pressing the F7 key.

Before starting on any lab work for this class, take a few minutes to create a folder structure for your labs like the one shown below. Being organized is a critical survival skill in software development! If you are using the classroom or campus lab computers, be sure to create this on your USB drive and be sure to save to it! Saving on the C: drive of the lab computers is a disaster waiting to happen! If you are working on your own computer, you may use the C: drive or your USB drive. Be sure you can back up your work periodically. A USB drive, or cloud storage such as Google Docs, Drop Box, One Drive, GitHub, etc. are all excellent choices.

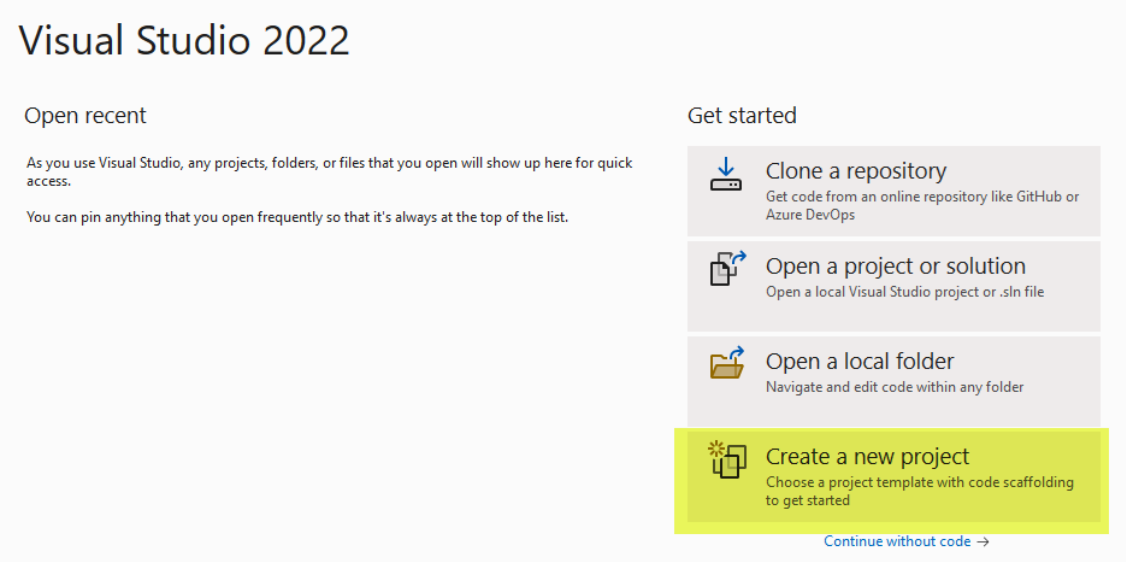


Exercise 1: Creating a C++ Project in MS Visual Studio

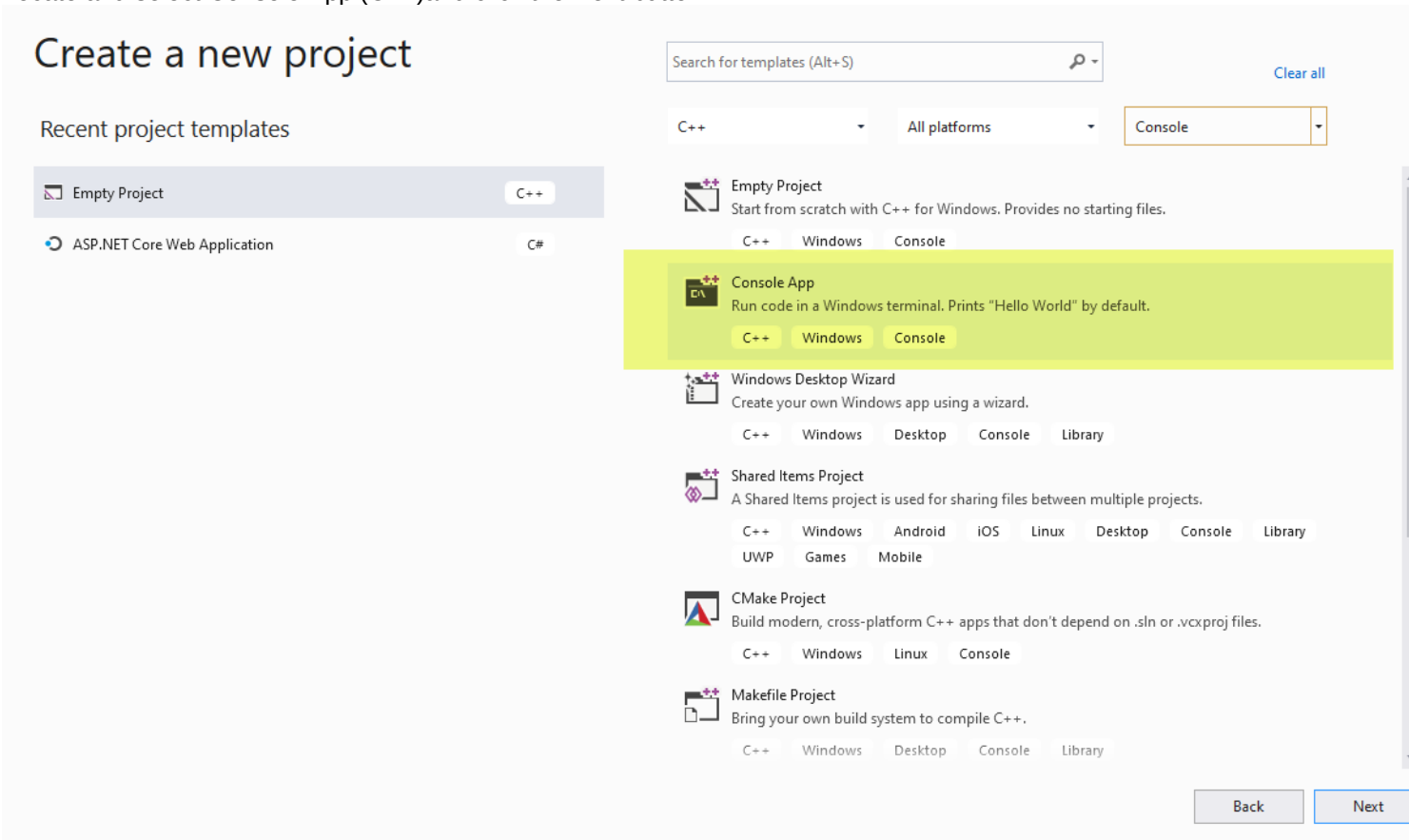
Being the very first lab, this is going to be a guided walk through of creating a C++ project in MS Visual Studio and using the code editor. If you are using a Mac and Visual Studio Code, things will be different from what is shown in this handout.

Steps

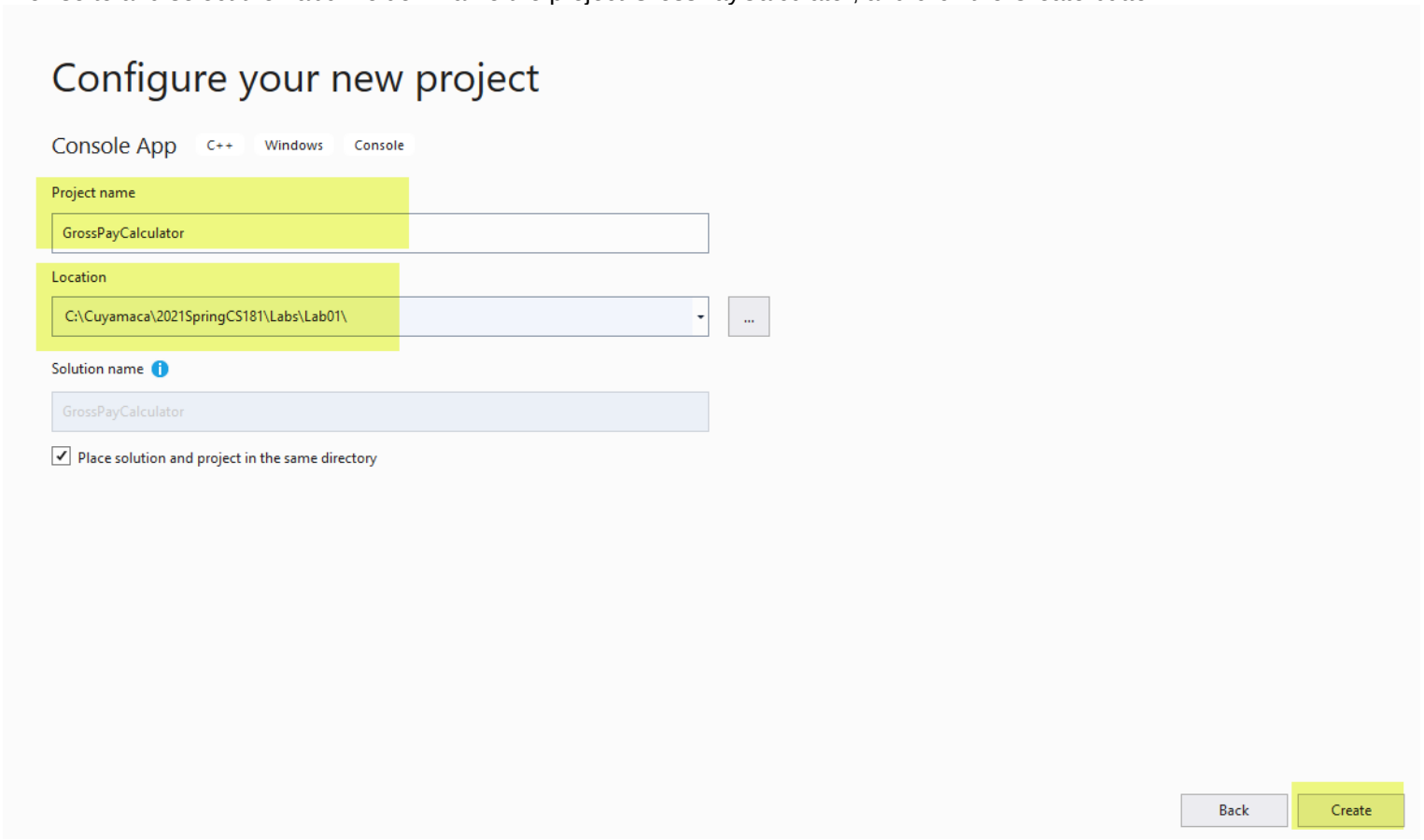
1. Start MS Visual Studio.
2. From the File menu, click Create a new project.



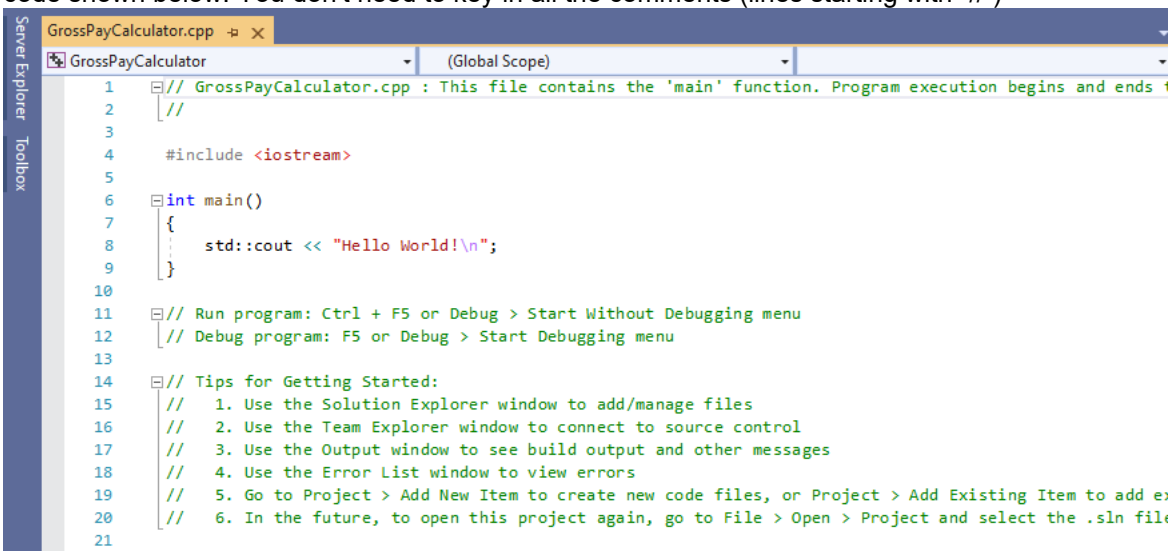
3. Locate and select Console App (C++)and click the Next button



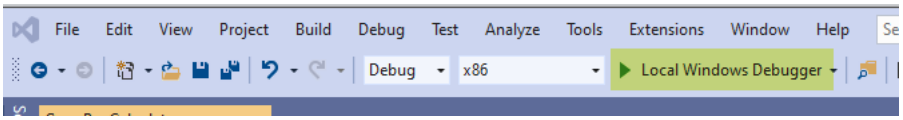
4. Browse to and select the Lab01 folder. Name the project GrossPayCalculator, and click the Create button:



5. You will get a window with the “Hello World” code below. Try running the program to make sure you get the “Hello World” output in the console window. Note: If you are on a Mac using Visual Studio code, you will need to key in the code shown below. You don’t need to key in all the comments (lines starting with “//”)



To run your program, press the F5 key or select Local Windows Debugger:

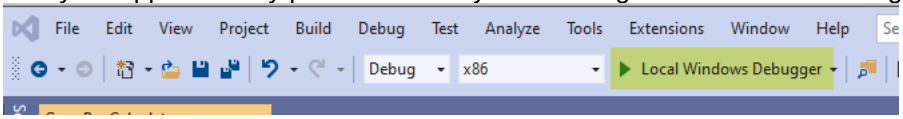


6. Replace the code on line 8 with the code in Program 1-1 on pages 14 – 15 in the text. Be sure to add the using namespace std; statement right after #include <iostream> statement on line 4. A screen shot of the code is provided below for convenience. We will be covering the code presented here in more depth in chapter 2. Note that the line numbers don't have to match exactly.

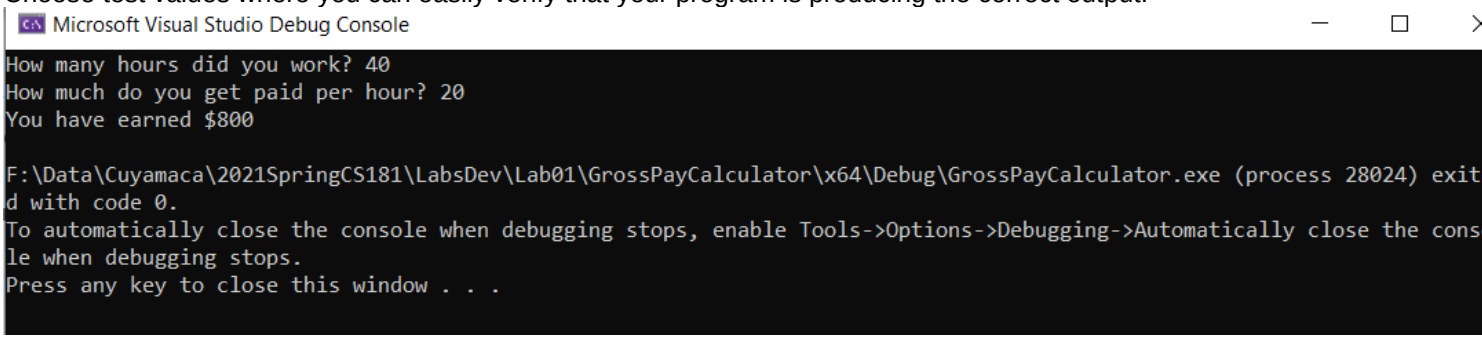
```
4    #include <iostream>
5    using namespace std;
6
7    int main()
8    {
9        // multiple variables can be declared on 1 line as shown in the text:
10       // double double hours, rate, pay;
11       // It's cleaner and better coding pratice to declare each variable on a single line by itself
12       double hours;
13       double rate;
14       double pay;
15
16       // Input - get hours worked and pay rate
17       cout << "How many hours did you work? ";
18       cin >> hours;
19
20       cout << "How much do you get paid per hour? ";
21       cin >> rate;
22
23       // process - calculate pay
24       pay = hours * rate;
25
26       // output - display pay
27       cout << "You have earned $" << pay << endl;
28       return 0;
29
30   }
31
```

7. Be sure to save your work!

8. Run your application by press the F5 key or selecting Local Windows Debugger.



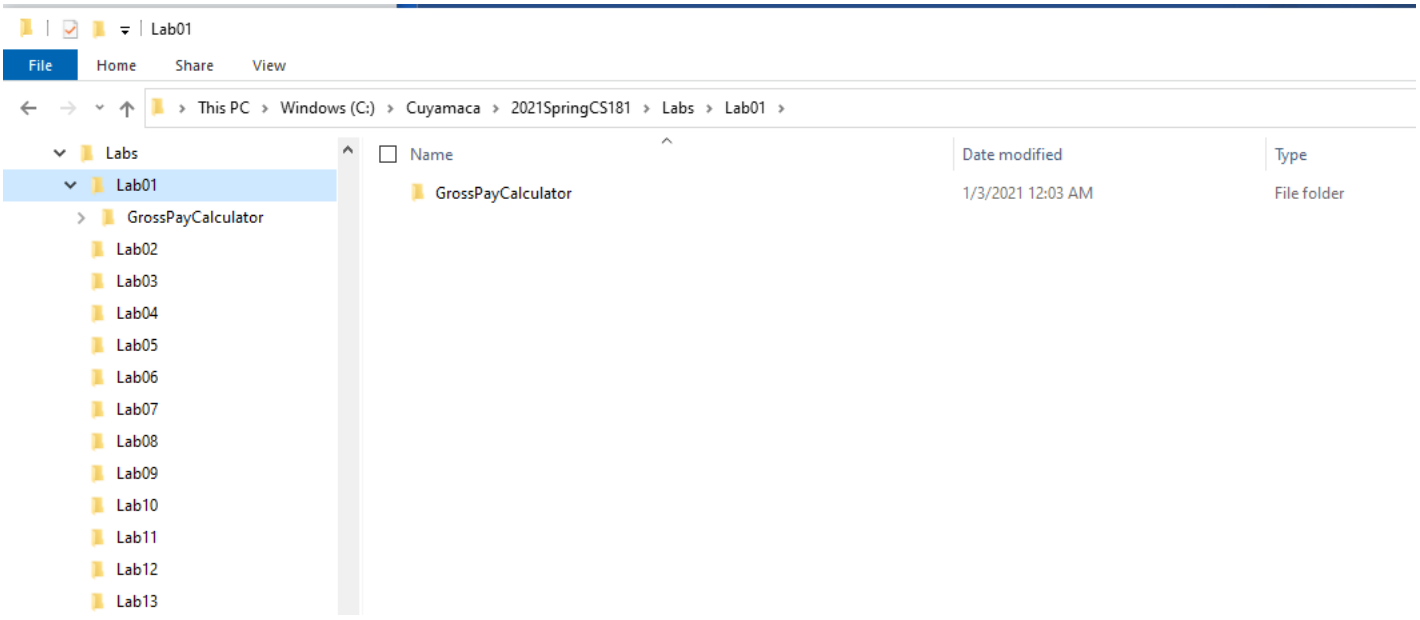
9. Choose test values where you can easily verify that your program is producing the correct output.



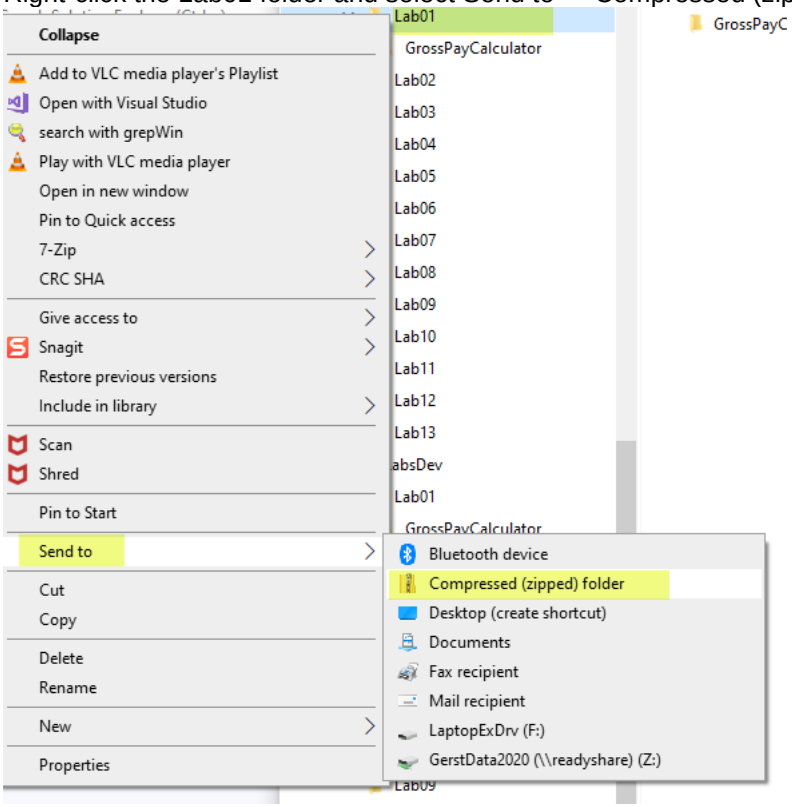
10. If you get a correct output, congratulations! You have just successfully created your first C++ application!

11. For labs 2 – 13, you will have additional coding exercises to complete.

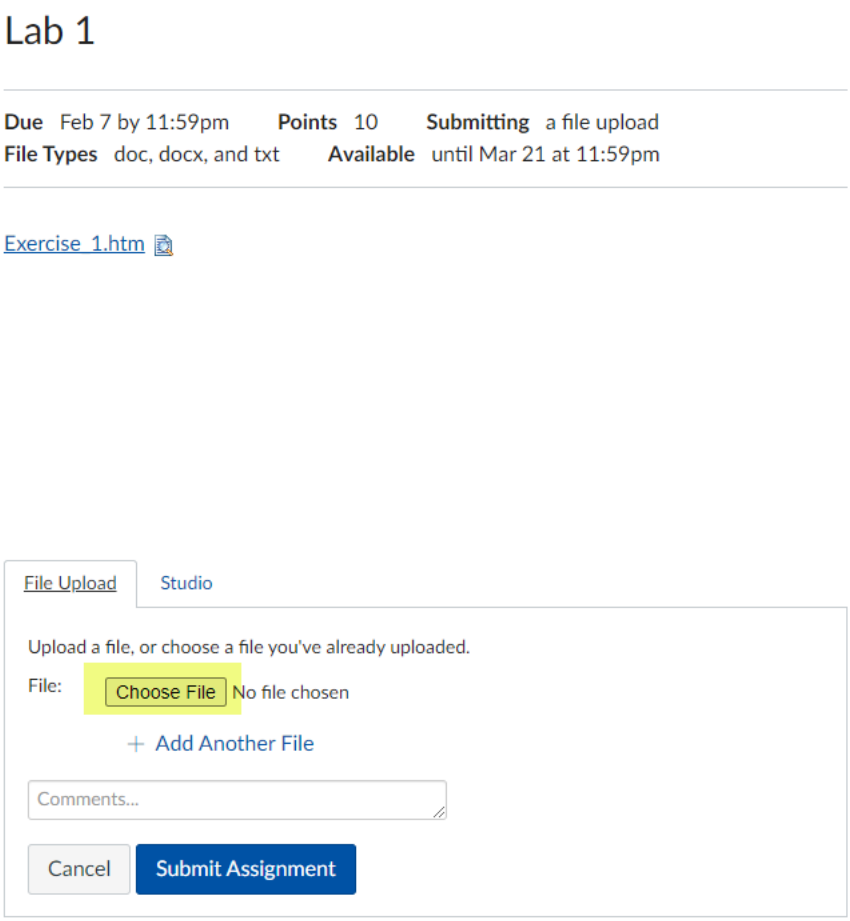
12. The next step is to zip up and submit your work. Be sure you have saved your work, tested everything, and you have exited Visual Studio. Use Windows File Explorer to locate your lab 1 folder. Make sure you have a sub-folder for each exercise:



13. Right-click the Lab01 folder and select Send to -> Compressed (zipped) folder>



14. Locate the zip file. In Canvas go to the assignment submission page, select Choose File, locate your zip file and submit it for the assignment.



Assignment Submission

Your completed assignment must be uploaded to Canvas in a zip file format.

Grading Criteria:

Deliverable	Points	Breakdown
Guided exercise C++ program code	**25**	Complete, code is clear, descriptive variable names, appropriate use of comments.
Guided exercise run	**10**	Compiles, runs, produces correct output.
Lab submission	**15**	Lab 1 folder and sub-folders are in a zip file. Program loads and runs in Visual Studio.
Lab Total	**50**	