# CS 150 Lab 01
Introduction to Visual Studio

The purpose of today's lab session is to familiarize you with

1. Connecting to the Computer Science server 'grace' which you will use to turn in your solution to a given program assignment / lab assignment
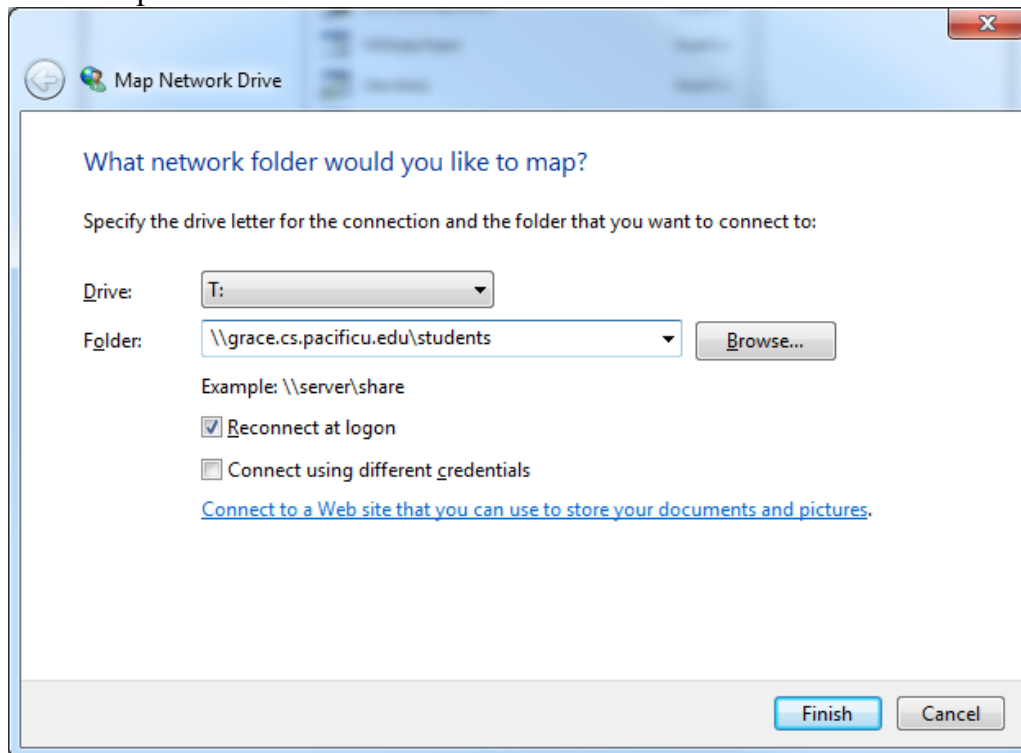2. Visual Studio 2010 Ultimate
3. Building and Running a Program

►If you have trouble with any piece of this, please ask the instructor for help.

► **Connect to the Computer Science server grace.cs.pacificu.edu**
   1. Go to Start and right-click on Computer
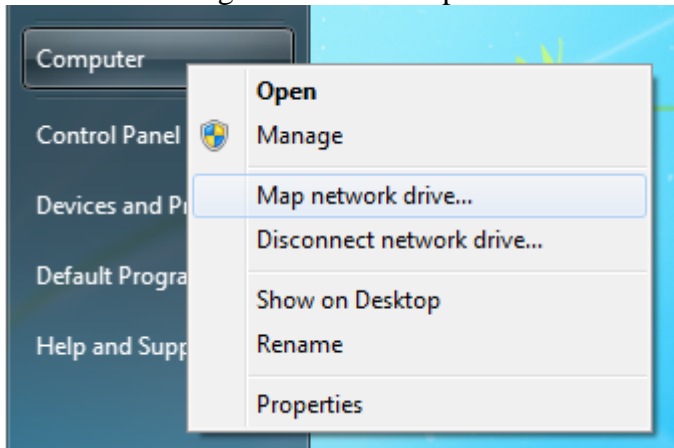
   2. Select Map network drive

   3. Type in **\\grace.cs.pacificu.edu\students** in the Folder textbox. Do not worry about what Drive letter is showing. Could be W: or Z: or anything else. Notice the slashes are \ not /.
   4. When you are prompted for a name and password, type in **eniac\yourloginname** where yourloginname is your PUNetID and the password is your **grace password**.

5. If all goes well, you will see something like the following:
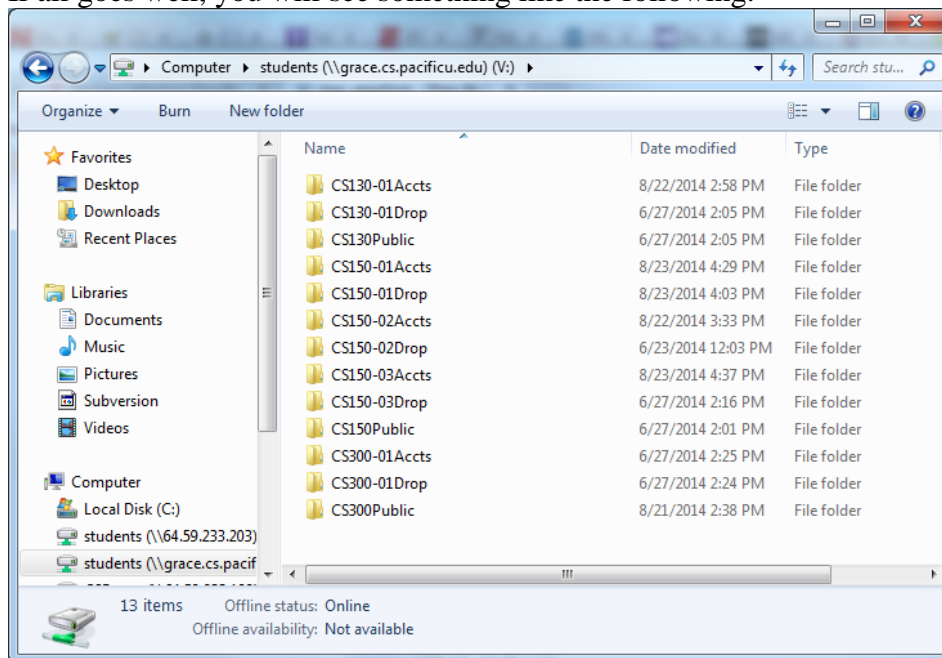
6. Inside of CS150-02Accts you will find your individual folder to be used for saving your projects. No other student has access to this folder, but faculty members do. Do not store any illegal material on grace.
7. We will come back to grace later in the lab.


► **Introduction to Visual Studio 2010 Ultimate**
1. Log on to the computer if you have not already done so.
2. Start Visual Studio 2010 Ultimate by using the **Start** button to navigate to **Programs** and then **Microsoft Visual Studio 2010**
3. We are going to go through creating a solution **PUNetIDLabs** and your first project **01_1_HelloWorld**. Create a new project by going to File→ New→Project. For **Name:** use **01_1_HelloWorld** and for **Solution name:** use PUNetIDLabs where PUNetID is your PUNetID. As an example, for a PUNet of ryandj, the window below would look like the following:
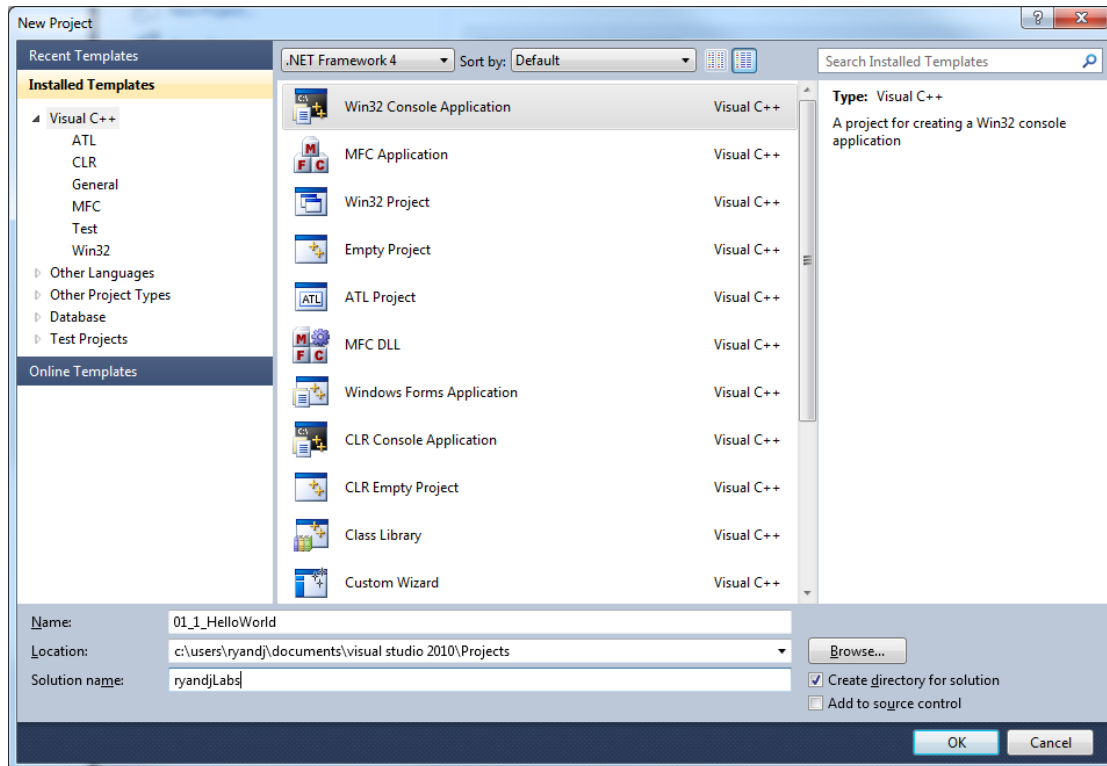
4. Once the Name, Location, and Solution name are correct AND Win32 Console Application is selected, you can click OK, which takes you to the following screen. Make sure the Application type is Console application.

5. The following window will be displayed. You need to see **Console application**.
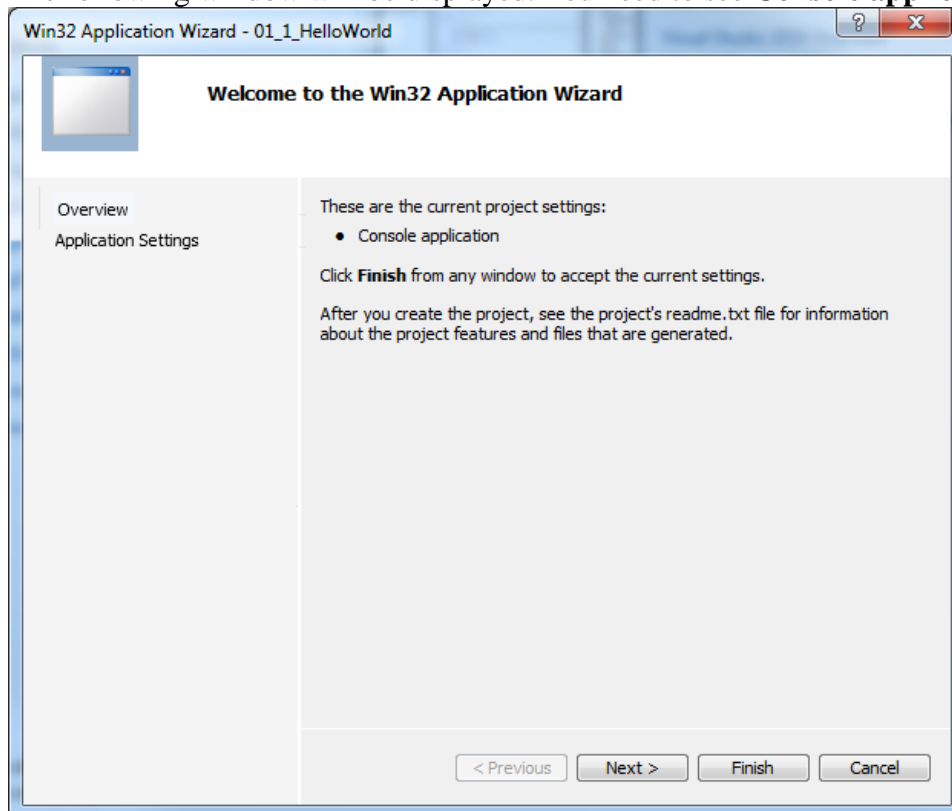
6. Clicking Next results in the following: You are to uncheck Precompiled header and check Empty Project **BEFORE** clicking Finish as shown below.
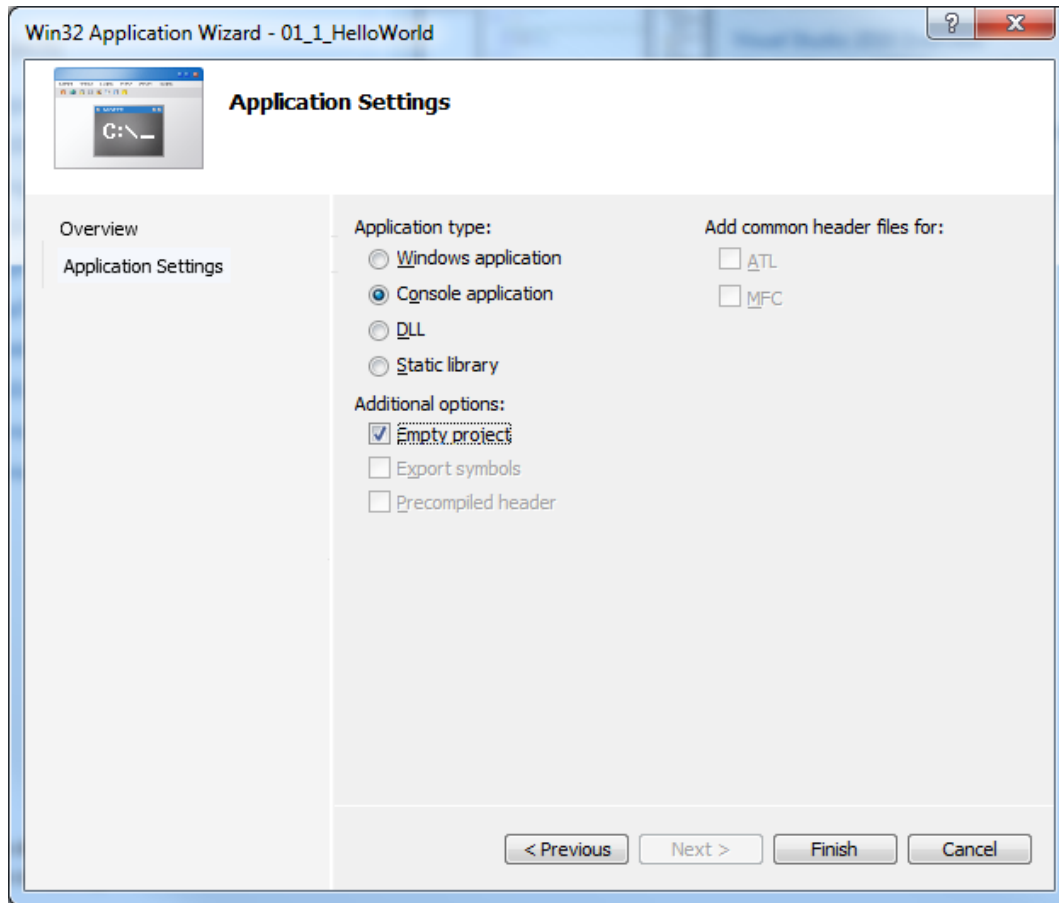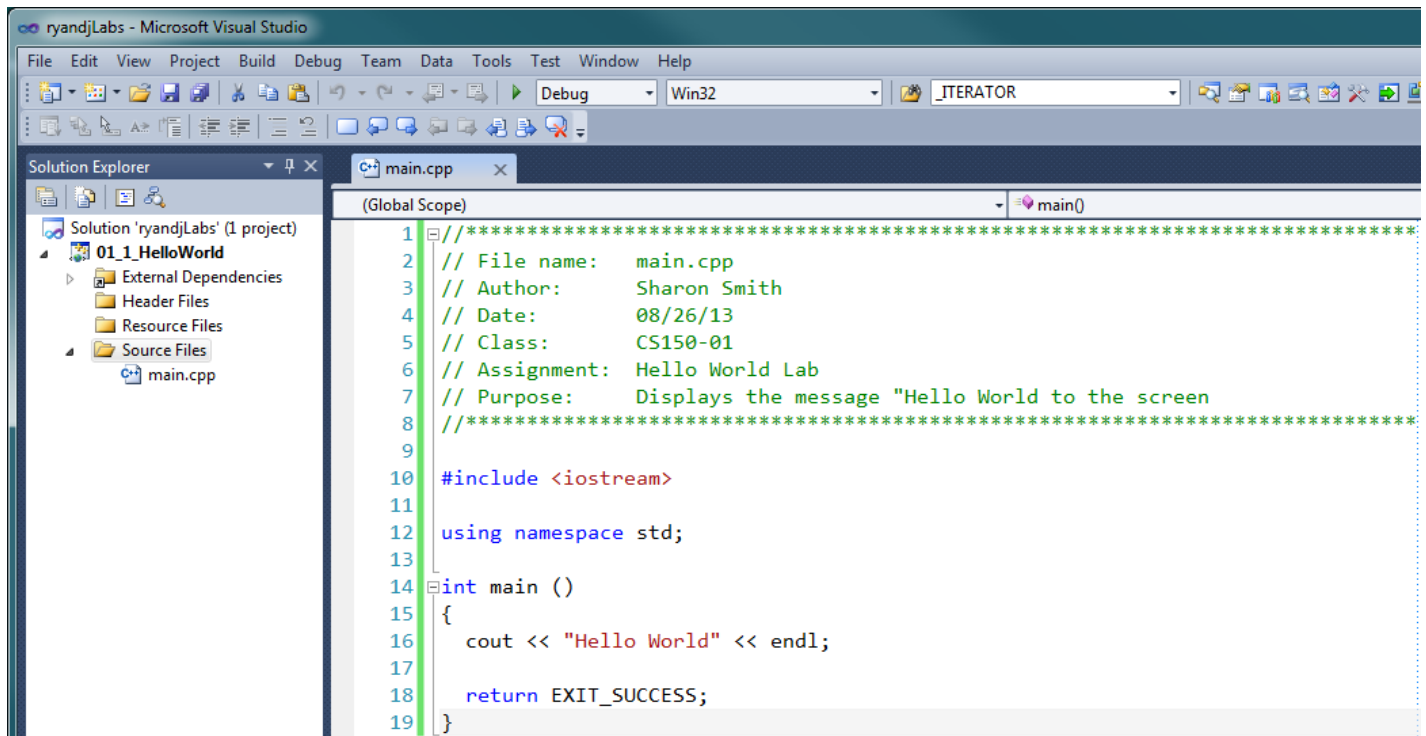
Now you can click Finish!!!!!!!

7. Once you have completed step 6, <u>stop and ask the instructor or TA to review your work</u>.

8. Add a New Item to the Project. Right click on the folder Source Files→Add→New Item…. You will see the window below. Select C++ File (.cpp) and name it **main.cpp**.
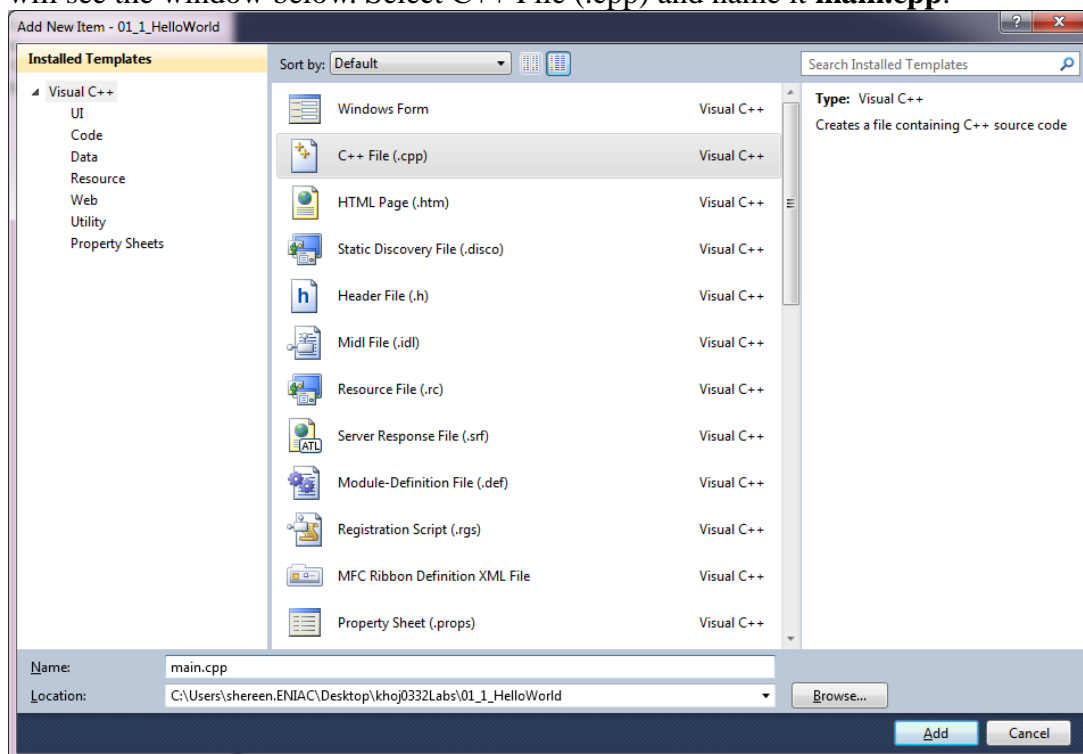
9. Type in the following program to **main.cpp**. Be sure to copy the program **exactly** as shown below. Use your name as the Author and use today's date. Complete step 10. before starting to type your program.

10. Turn on line numbers and setup the tab size to tab 2:
    1. **Show Line Numbers**
       - Tools→Options→Text Editor→All Languages→General
       - Check Line numbers
    2. **Set Tab Size**
       - Tools→Options→Text Editor→All Languages→Tabs
         1. Tab Size: 2
         2. Indent Size: 2
         3. Check Keep Tabs

11. Once you have completed step 9, <u>stop and ask the instructor or TA to review your work</u>.

► **Let's examine the program you just wrote.**
    1. Lines 1-8 are comments. Note that comments are used to explain to the human reader what this program does. Visual Studio's compiler ignores comment lines. Note that comments are green.
    2. Line 10 includes the **iostream** (input/output stream) library required to display messages to the screen. The iostream library is also required when reading input from the user.
    3. Line 12 specifies the **namespace,** which is required for using **cout** to display messages to the screen. More on this later.
    4. Line 14 declares the **main** function required by every C++ program. The main function is where every C++ program starts when you begin program execution.
    5. Line 15 and 19 contain { and }, respectively. These mark the beginning and ending of ALL lines that are included in the main function.
    6. Line 16 displays a message on the screen.
    7. Line 18 ends the program where EXIT_SUCCESS denotes that no runtime errors were encountered.

► Building the Program
Before you can run your program, you will need to build the program which checks your program for syntax errors. A program that builds with no syntax errors causes your C++ code to be translated into machine code (zeros and ones) that is understood by the computer.

1. You are ready to build and run your first C++ Program! Make sure you've saved your work using **File** | **Save All** or the blue disk icon on the tool bar.
2. Build your program using the **Build** menu and the **Build Solution** option. Or press the F7 key. Building the program checks for syntax errors.
   The program will then compile and link. The output window below the source window will display messages indicating what is going on. You can expand this window or scroll through so that you are able to read all the messages.

If you have typed everything correctly you will have 0 errors and 0 warnings. If you do get errors, then check your code and make sure you've typed everything correctly. Let me know if you can't get things to work.

► Run the Program
1. Run your program by using the **Debug** menu and the **Start Without Debugging** option.
2. A black window should appear.
3. Which lines in your program do you think produces each line in the black output window?
   Answer_____

   _____

4. What happens after you press enter? Answer _____

   _____

5. Once you have completed step 4, <u>stop and ask the instructor or TA to review your work</u>.

Congratulations! You have run your first C++ program.

What happens if we type **cot** instead of **cout**? Try it now, and build and run your program again.

Answer _____

_____

What happens if we delete **<< endl** from line 16? Try it now, and build and run your program again.

Answer _____

_____

► Once you have completed all of the above, then create a new project in your solution **PUNetIDLabs**. Name this project **01_2_PayCalculator**.

Add a file named main.cpp to the newly created project and type in the following code.

```
//****************************************************************************
// File name:    main.cpp
// Author:       Sharon Smith
// Date:         08/27/13
// Class:        CS150-01
// Assignment:   Pay Calculator
// Purpose:      Calculated the user's pay based on hourly
//               wage and the number of hours worked.
//****************************************************************************
#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
   cout << "How much do you get paid per hour? ";
   cin >> rate;

   // Calculate the pay
   pay = hours * rate;

   // Display the pay
   cout << "You have earned $" << pay << endl;

   return EXIT_SUCCESS;
}
```

Build and run your program.

Write a brief description of what the program does.

A. _____

_____

Show your completed work to the TA or the professor.

Save the project to your folder on grace and then place a copy of your project in the CS150-01 Drop folder!

8. Copy your solution PUNetIDLabs from Computer->Local Disk (C:)->Users->PUNetID->My Documents->Visual Studio 2010->Projects into your individual folder on grace.
9. Copy your solution PUNetIDLabs into the CS150-02 Drop folder. **This is how you will submit your programming assignments in this course.**
10. Once you have copied both projects to the proper location, stop and show your instructor or TA.

► After each lab session, be sure to copy your work to your folder on grace.

► As you work on your programming assignments, be sure to copy your work to your folder on grace.

Why is it so important to copy your work to grace? _____

_____

_____

►Optional Challenge (you do not need to submit this to CS150-02 Drop)

Once you have followed all of the above instructions and submitted your lab project, modify your pay calculator so that it displays a message asking for a name. Once the user types in the name, the program should display "Hello" followed by the name typed in by the user and then continue on with its previous work. **HINT**: Look at the program at the end of the handout for Monday's Lecture.

The updated input and output of the pay calculator should be as follows:

What is your name? Bob
Hello Bob!

How many hours did you work? 10
…..