CS 150 Lab 01
Introduction to Visual Studio

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

1. Visual Studio 2010
2. Building and Running a Program
3. Connecting to the Computer Science server 'Turing'

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

► Introduction to Visual Studio 2010

1. Log on to the computer if you have not already.
2. Start Visual Studio 2010 by using the **Start** button to navigate to **Programs** and then **Microsoft Visual Studio 2010**
3. **Visual Studio** will prompt you for which language you want to use. Choose **C++**
4. Bring up the Visual Studio FAQ on the CS 150 website and follow the instructions on how to create a new project. The address is: **http://zeus.cs.pacificu.edu/chadd/VS_FAQ.html**
   Be sure to create a **Win32 Console Application.**
   Save this project to your **Desktop**.
   Name this project **Lab01_1_*PUNetID***, replacing ***PUNetID*** with your actual PUNetID. When I do this Lab, I create a project named Lab01_1_will4614.
   Add a **File** named **main.cpp.**
5. Once you have completed step 3, stop and ask the instructor to review your work.
6. Type in the following program to **main.cpp**. Be sure to copy it as exactly as you can. Use your name as the Author and use today's date.

```
//********************************************************
// File name:  main.cpp
// Author:     John Smith
// Date:       08/28/2010
// Class:      CS150
// Assignment: Hello World Lab Program
// Purpose:    Displays the message "Hello World" to the
//             screen.
//********************************************************

#include <iostream>

using namespace std;

int main()
{
  cout << "Hello World!" << endl;
  return 0;
}
```

7. Turn on line numbers and setup the tab size, as explained in the Visual Studio FAQ.
8. Once you have completed step 6, stop and ask the instructor to review your work.

► Let's examine the program you just wrote.

1. Lines 1-9 are comments. Notes that are used to explain to the human reader what this program does. Visual Studio ignores these lines. Note that they are green.
2. Line 11 includes the **iostream** (input/output stream) library required to display messages to the screen. It's also required when reading input from the user.
3. Line 13 specifies the **namespace,** which is required for using **cout** to display messages to the screen. More on this later.
4. Line 15 declares the **main** function that is required by every C++ program. The main function is where your program starts when you run it.
5. Line 16 and 19 contain { and }, respectively. These mark which lines are included in the main function.
6. Line 17 displays a message on the screen.
7. Line 18 ends the program. The 0 denotes that no errors were encountered.

► Building the Program

Before you can run your program and see it in action you will need to build it. Building means the program is checked for syntax errors and the C++ code is translated into machine code 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 should have 0 errors and 0 warnings. If you do get errors, then check your code and make sure that it matches mine. Let me know if it still doesn't 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?
   A____

   ____
4. What happens after you press enter? A ____

   ____
5. Once you have completed step 4, stop and ask the instructor to review your work.

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.
A. ____

____

What happens if we delete **<< endl** from line 17? Try it now, and build and run your program again.
A. ____

► **Connect to the Computer Science server turing.cs.pacificu.edu**

1. Open **My Network Places** which is in the **Start Menu**.
2. Double click on **Add Network Place**.
3. Select **Choose another network location**.
4. For the Internet or network address type: \\turing.cs.pacificu.edu\students.
5. Click **next**.
6. You ***may*** be prompted to type a name and password.
   If you are, type your PUNetID as your user name and the password you setup in class. Click **Ok** then click **Next.**
7. Click **Finish.**
8. You should see a window full of folders. Close this window.
9. Click on **My Network Places** again. Your connection to students on Turing should be listed. Click on that icon.
10. Click on **CS150-02 Fall2010Accts**. You should see a folder with your PUNetID!
    This is your personal folder that you can use to store anything related to CS150. No other student has access to this folder, but faculty members do. Do not store any illegal material on Turing.
11. Open the folder with your PUNetID.
12. Copy your Lab01_1_PUNetID Visual Studio project to your folder on Turing by dragging the folder from your Desktop to the window you just opened.
13. Copy your Lab01_1_PUNetID folder from the **Desktop** to **CS150-02 Drop** on Turing. Open CS150-02 Drop after you do this to confirm it contains your project. **This is how you will submit your programming assignments in this course.**

► After each lab session, be sure to copy your work to your folder on Turing.
► As you work on your programming assignments, be sure to copy your work to your folder on Turing.

Why is it so important to copy your work to Turing? ____

____

____

► Once you have completed all of the above, then create a new project following the same steps as above. Name this project "Lab01_2_*PUNetID*" replacing *PUNetID* with your PUNet ID.

Add a file named main.cpp and type in the following code. Change the header comment to reflect today's date and your name.

```
//********************************************************
// File name:  main.cpp
// Author:     John Smith
// Date:       08/29/2010
// Class:      CS150
// Assignment: Pay calculator
// Purpose:    Calculates 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 0;
}
```

Build and run your program.

Write a brief description of what the program does.

A. ____

____

Label each line on the paper above with a brief description of what that line does.

Save the project to your folder on turing and submit it in the CS150-02 Drop folder!

Show your completed work to the TA or the professor.

►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
…..
```