

# CS 150 Lab

## Introduction to Visual Studio

The purpose of today's lab session is to familiarize you with the Computer Science server 'Turing' where you will be submitting all your assignments and lab projects. We will also use Visual Studio to compile and run basic C++ programs.

### Lab 1.1

We will work through the first problem together. We are going to connect to the Computer Science server, and write a program to display the message 'Hello World!'.

#### Logging on:

**1) Log on to the computer using your PU net ID and password.** Your PUnet ID is the first four letters of your last name and the last four letters of your id number. Your password is your seven digit ID number.

#### Connecting to Turing:

Turing is the computer science server where all of your work will be stored. Anything you save locally on the machines in the lab will be deleted once you log off, so it's important to save your files on Turing in order not to lose your documents.

Before you can save anything to the server you will need to connect to it.

#### **2) Connect to (Turing):**

- a. **Open 'My Network Places' on the desktop.**
- b. **Double click on 'Add Network Place'.**
- c. **In the text box type: [\\turing.cs.pacificu.edu\students](http://turing.cs.pacificu.edu/students). Click 'next'.**
- d. **You will be prompted to type a name and password. Type your PU net ID as your name and your student ID as your password. Click 'next'.**
- e. **You will be prompted to name this network place. Type 'Turing' in the text box. Click 'finish'.**

This will open up the Students folder, which should contain a folder with the same name as your PU net ID. 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.

## **Changing your Password:**

To change your password, write down on the piece of paper I give you:

- Your name
- Your student ID
- Your new password

**Be sure to write the new password clearly!!**

Hand them to me by the end of class and the password will be changed by next week.

## **Starting Visual Studio:**

Visual Studio is the integrated development environment (IDE) that you will use to develop your C++ programs. The software will be available for you to install on your home machines next week.

**3) Open Visual Studio by clicking the Start menu, selecting Programs, then Microsoft Visual Studio 2005 then Microsoft Visual Studio 2005.**

Visual Studio will start with no projects open.

Today, we are going to create a new C++ program and compile and run it. You are also going to copy your work to the “CS150-01 Lab” drop box, which is where you will be submitting all your future lab projects.

## **Creating a new project:**

To create a new project, **select File -> New -> Project from the menus.**

This opens a dialog box with several options. The window on the left lists the project types, while the window on the right lists the templates. All the projects that we will create this semester will be Win32 Console Projects

**4) Expand *Visual C++* from the left window and click on the *Win32* option. Select the *Win32 Console Application* from the right window.**

You will select these options for all your projects this semester since you will only be developing C++ console projects.

**5) In the Name textbox, enter a name for your project. For this project, call it ‘01xxxxxxx’ replacing the x’s with your PUNet ID.**

**6) Click on the drop down box below the Name textbox, select ‘. This will save your project to the desktop.**

7) Click the OK button.

8) Check that everything is correct in the dialog box and click Finish.

**Congratulations! You have created your first C++ project.**

The right hand side of the window lists all the files that are in your project. Double-click the file that has the same name as your project (i.e. PUNet ID) and has the extension cpp.

This will display the contents of the file in the main window.

### **Examining the program:**

The first few lines in green at the top are just comments. You can ignore these for now. When you start writing your own programs you should write your name and the date as comments in your program. You should also write down the functionality of your program in a paragraph or two. More about this later.

The next line includes the libraries that we are using in the program. Visual Studio has placed `#include "stdafx.h"` for you. You should leave this line in all of your programs.

The third line defines our 'main' function. Visual Studio's main is:

```
int _tmain(int argc, _TCHAR* argv[])
```

You could either use this one or use the 'main' that is in your book and my class notes;

```
int main()
```

Either of those will work fine.

Notice that open and close curly braces follow the 'main' line. You should never remove these. They are always needed in your main function.

After a couple of lines you will see

```
return 0;
```

This is needed at the end of your main function to signal that your program completed successfully.

The program will not do anything as it currently stands. Let us add a few lines to make it output the message "Hello World" to the screen.

Your complete program should look like the one below, but without the line numbers.

```
1. //*****
2. // File name: 01abcd1234.cpp
3. // Author: John Smith
4. // Date: 08/29/2006
5. // Class: CS150
6. // Assignment: Hello World Lab Program
7. // Purpose: Displays the message "Hello World" to the
8. // screen.
9. //*****
10.
11. #include "stdafx.h"
12. #include <iostream>
13.
14. using namespace std;
15.
16. int main()
17. {
18.     cout << "Hello World!" << endl;
19.     return 0;
20. }
```

Lines 1 through 9 are comments and will be ignored by the compiler when running the program.

Line 11 includes the `stdafx` library required by Visual Studio.

Line 12 includes the `iostream` (input/output stream) library required to display messages to the screen. It's also required when reading input from the user.

Line 14 specifies the namespace, which is required for using the keyword `cout` to display messages to the screen.

Line 16 declares the main function that is required by every C++ program.

Line 18 sends the message "hello world!" to the screen. `cout <<` indicates that we want to write to the standard output (the screen). The text between double quotes (") is what is written on the screen after running the program. The `endl` at the end of the line moves the output cursor to a new line.

Line 19 contains `'return 0'` which just tells the program to exit normally.

### **Building the program:**

Before you can run your program and see it in action you will need to build it. Building the program checks it for syntax errors and translates the C++ code into machine code that is understood by the computer.

**13) To build the program, click on 'Build' from the menu, then 'Build Solution'.**

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, then 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.

**Q 1.1.** What happens if we type `cout` instead of `cout`? Try it now, and build and run your program again.

**A.**

### **Running the program:**

Now that we have built the program, we can run it. Running the program will execute the C++ code that we have just written.

**14) To run the program, click 'Debug' from the menu then click 'Start Without Debugging'. You can also hit ctrl-F5.**

What happens next?

A new window should open with the text 'Hello World!' displayed. There will also be a line that says 'Press any key to continue'. Pressing any key will close the window.

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

**15) Press any key to close the window.**

**Q 1.2.** What happens if we delete `<< endl` from line 18? Try it now, and build and run your program again.

**A.**

### **Saving your projects onto Turing**

Before you log off of the local machines, you should save your work to your folder on Turing so that it is not lost. You do this by opening quitting Visual Studio and going to the desktop. Open your folder on Turing, and drag the project folder to the folder on Turing. You must copy the entire folder to save everything correctly.

## **Lab 1.2: Opening and fixing C++ programs**

For this next problem, you will be opening some existing programs and running them in Visual Studio. These programs contain errors that you will need to identify and fix.

### ***Lab 1.2.a: Syntax errors***

#### **Opening a project:**

The following instructions are for opening an existing C++ project. This could be one that I give you or one that you have created.

**18) You will need to copy the projects to your folder before opening them in Visual Studio. Go to the 'CS150-01 Public' folder on Turing. You will find the folder '01CompileError' in the '01Lab' folder. Copy the folder '01CompileError' to the desktop.**

**19) To open your project, start up Visual Studio like you did in step 3. Click on 'File' from the menu, then 'Open' and 'Project/Solution'.**

**20) Click on the drop down box and select 'Desktop'. Double click on the folder '01CompileError'.**

You will see a list of files.

**21) Select the Microsoft Visual Studio Solution file called '01CompileError'. It's the one with this icon.**



**Click 'Open'.**

This will open the project.

Build the program. It will give you an error, which is an example of a syntax error. You are likely to encounter many of these errors as you progress through the course, and it's important for you to learn to recognize them.

Try to find the error and fix it. Note that the line number given to you by the compiler could be misleading.

Once you have fixed the error and your program builds, run the program and input the number 9. Write down the output, and run the program again with another number. Do you feel that you are getting valid output?

## **Lab 1.2.b Run-Time Errors**

**22) Copy the '02RunTimeError' project located in the 'CS150-01 Public' folder under 'Lab Projects' to your folder on Turing, then open it as you did in the previous example.**

**23) Try and build the program and run it.**

Input any number into the program. What is the result? Is this the result that you expected?

The problem here is that we are asking the computer to break the laws of Math by dividing by zero. It cannot be done, so the compiler outputs a nonsense answer.

### **Submitting your finished project:**

Once you have completed your lab projects or assignment, you will need to submit it for grading. You will submit your projects to the 'CS150-01 Lab' folder on. You may only submit your projects once, so make sure that they are in their final form. Also, make sure that you submit the correct project.

**24) Exit Visual Studio.**

**25) From the desktop, double click on 'My Network Places'. Double click 'Turing, then double click your folder. Create a folder called '01xxxxxxx', where you replace the x's with your PU Net ID. Place your two projects '01CompileError' and '02RunTimeError' into this folder.**

Don't close that window!

**26) Double click on 'My Network Places', and double click on 'Turing. This should open a new window.**

**27) Locate the folder called 'CS150-01 Lab'.**

**28) Now go back to the other window and drag the folder called '01xxxxxxx' and place it in the folder called 'CS150-01 Lab'.**

This is how you will hand in all of your lab programming assignments.

### Optional Challenge:

Once you have followed all of the above instructions and submitted your lab project, modify your project 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.

For example, if the user types in ‘Bob’, the program output should be:

```
What is your name? Bob  
Hello Bob!
```

Refer to the book on how to get information from the user. Hint: you will need to use `cin`.