

CS 150 – Lab

9/30/05

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.

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

Connecting to Turing:

Turing is the computer science server where all of your work will be stored. Anything you save locally on these machines will be deleted once you log off. In order not to lose your documents you must save them to your directory on the CS server (Turing).

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 .NET:

Visual Studio .NET 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 .NET 2003 then Microsoft Visual Studio .NET 2003.

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) Select Visual C++ Projects from the left window and Win32 Console Project 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. My project would be ‘01khoj0332’

6) Click on the drop down box below the Name textbox, select ‘My Network Places’ then double click ‘Turing’, then double click the folder with your PU net ID on it. This will save your project directly to ‘Turing’. If you miss or skip this step then your projects will be saved by default on the local machine in the ‘My Documents’ folder.

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. My file is called "01khoj0332.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 be:

```
//*****  
// File name: 01khoj0332.cpp  
// Author: Shereen Khoja  
// Date: 08/30/2005  
// Class: CS150  
// Assignment: Hello World  
// Purpose: Displays the message "Hello World" to the  
// screen.  
//*****  
  
#include "stdafx.h"  
#include <iostream>  
  
using namespace std;  
  
int main()  
{  
    cout << "Hello World!" << endl;  
    return 0;  
}
```

Don't worry too much about the first three lines for now. Just know that all your programs must have them.

The important part for us is the stuff between the curly braces.

The first line we come across contains 'cout <<'. This indicates that we want to write out 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.

The last line 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 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.

Note: Even if your program built successfully it is a good idea to introduce an error into the code so you can see for yourself the error messages produced by Visual Studio .NET. Introduce an error by typing `cot` instead of `cout`.

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.

Exiting VisualStudio:

17) Exit Visual Studio by clicking on the x on the top right hand corner of the screen, or by selecting 'file', 'exit' from the Visual Studio menu.

Opening your 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) To open your project, start up Visual Studio like you did in step 3.

19) Click on 'File' from the menu, then 'Open' and 'Project'.

20) Click on the drop down box and select 'My Network Places'. Double click on (Turing) then double click your folder.

21) Double click the folder called '01xxxxxxxx'. Where xxxxxxxx is replaced by your PU net ID.

You will see a list of files.

22) Select the one called '01xxxxxxxx.vcproj'. Click 'Open'.

This will open your project.

Modifying the program:

We are now going to modify the program so that it writes 'Hello, my name is xxxx' to the screen, instead of writing 'Hello World!'.

24) Select the text between double quotes and replace it with 'Hello, my name is xxxx'.

Don't forget to leave the double quotes as they are!

To test the program, you need to build it again and run it. Refer to steps 13 and 14 to see how this is done.

Saving the project:

Now that you have modified the program, you need to save it.

25) Click on the C++ file to make sure that it is selected.

26) Select 'File' from the Visual Studio menu, then select 'Save'.

Remember to always save your work! The project is saved every time you successfully build or run it.

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 Turing and your assignments to the CS150-01 folder on Turing. 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.

27) Exit Visual Studio.

28) From the desktop, double click on ‘My Network Places’. Double click ‘Turing, then double click your folder.

Don’t close that window!

29) Double click on ‘My Network Places’, and double click on ‘Turing. This should open a new window.

30) Locate the folder called ‘CS150-01 Lab’.

31) Now go back to the other window and drag the folder called ‘01xxxxxxxx’ 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`.