

# Assignment 1 - Introduction to Visual Studio, Subversion and C

Assigned: Wednesday, January 20<sup>th</sup>

Due: Wednesday, January 27<sup>th</sup>, *before midnight*

Value: 20 points (for successfully submitting a correct program to the assignment manager for your section before the deadline). Late submissions will receive a 0.

## Executive Summary

You will write a program that on execution displays a greeting to your name.

## Purpose

- Gain familiarity with Visual Studio: writing, compiling and running a program
- Gain familiarity with version control

## Preliminaries

- In case you do not have access to a computer and need to use Cockrell School of Engineering computers (physically or remotely) you need an Austin domain account. You can enable your account here: <https://apps.engr.utexas.edu/dnapps/EnableAccount/default.aspx>
- Download and install Microsoft Visual Studio Ultimate 2013 (Described in the FAQ)
- Review the assignment submission process via Canvas (Described in the FAQ)

## Deliverables

The `HelloToYou.c` program source file.

## Steps

## Setup

## Subversion for Version Control

We will use a hosted SVN service, specifically, [Assembla.com](http://Assembla.com), for [Version Control](#)

1. Create an [Assembla](#) account. Use this convention for your log in ID: `ee312_<lower case uteid>` (e.g. `ee312_aa123`). After registration, create a free **private subversion repo** with space name `ee312_<lower case uteid>`, this is where all your labs will reside. You can ignore the three folders already present.

When submitting your assignment make sure that the files for each assignment are contained in a separate folder called `assignment<number_of_assignment>` (for example, for assignment 1 your svn repositories should contain a folder called "assignment1" all lowercase and no spaces. The folder "assignment1" should contain all the files for the first assignment)

2. Install Subversion.

You can use any SVN client you want, here are some suggestions:

- Windows: [TortoiseSVN](#) may be a good choice for you.

- Other: you can use [Apache Subversion](#). [Download the binary package](#) and install SVN.

3. For Tortoise. Create or choose a local directory for your code base. Right click on the directory, select TortoiseSVN->create repo here. Then right click the directory again, choose SVN checkout. Paste your url from assembla to the “Url of Repository” line.

4. A folder will be created within the directory. You can save your Visual Studio projects to this folder. Remember to commit your changes to your assembla repository before you log off.

5. Learn [basic SVN operation](#)

- [Checkout](#): svn checkout (co) — Check out a working copy from a repository.
- [Add](#): svn add — Add files/directories to the project
- [Commit](#): svn commit (ci) — Send changes from your local copy to the repository.
- [Update](#): svn update (up) — Update your working copy.

[Difference between import, export, commit and checkout](#)

6. Share your assembla project with your TA, whose assembla IDs are **elieantoun**, **kespinoza** and **mtumbokon**

- Go to the “Team” tab
- On the right, enter the TA’s ID and choose ‘Member’ as the role
- Invite your TA.

**You have to finish all the above steps for your labs to be graded otherwise you will receive a ‘0’**

## Creating The C Program

### Create the Visual Studio Project

1. Open Microsoft Visual Studio 2013 (refer to the syllabus for installation instructions). If the “Choose Default Environment Settings” window appears, choose Visual C++ Development Settings and press Start Visual Studio.
2. Select the *File menu > New > Project...* In the Visual C++ Projects section, select Win32 Console Application. Name the project *assignment1*, make sure the project directory corresponds to your svn repository (created above), and press OK.
3. On the wizard screen that appears, select the *Application Settings* tab on the left. Check the box for *Empty Project* and click Finish.
4. If the Solution Explorer pane is not visible, select the View menu > Solution Explorer to show it. Right click the *assignment1* project (it’s the second item, not the solution) and select Add > Add New Item... From the Visual C++ category, select C++ File (.cpp) in Templates. Name the file **HelloToYou.c** and press Add.

### Writing the Program Code

5. Write the header comment for your file (see the FAQ for a description of what goes in the header comment)
6. Below your header comment, type in C program shown below, filling in your name in the portion denoted by < >. (The meaning of each of the constructs in this program will be explained in class.)

```
#include <stdio.h>
int main( void )
{
    printf("Hello to: <insert your own name in here>\n");
    /* causes the display to pause until you strike a ENTER */
    getchar();
    return 0;
}
```

## Build Your Program

7. Now you'll want to *build* (compile and link) your program. Save your file<sup>1</sup>, then select the *Build menu > Build Solution*. If an error occurs, make sure you typed the program correctly and try compiling again. (Inspect the error messages carefully, they should give you a good idea as to what the problems are.)

## Running Your Program for the First Time

8. Now it's time to run (execute) your program. Select the *Debug menu > Start Debugging*. Your program will display a window with your greeting message. Press ENTER to close the window and return to Visual Studio.
9. Congratulations, you have now written a program and ran it to see the output.

## Finishing Up and Submitting Your Work

10. When you are done running your program, you can close Visual Studio. You can find the `HelloToYou.c` file under the directory created for your project at the beginning of this lab (for example, `H:\EE312\Assignment1\HelloToYou.c`). Make sure it is the file that contains the latest version of your C source code.
11. Commit your `HelloToYou` project to assembla
12. Submit your finished `HelloToYou.c` source code file via the Canvas web page for this course
13. Save a backup copy of your program to your flash drive. You may now logout.

---

<sup>1</sup> Visual Studio automatically saves your work before compiling, so all changes you have made are included in the compiled program. However, not all text editors will do this, so it is a good habit to always save your work before you try to compile your program.