16.216: ECE Application Programming

Summer 2015

Programming Assignment #1: A Simple C Program

Due **Friday, 5/22/15**, 11:59:59 PM

1. Introduction

This program simply tests your ability to write, compile, execute, and submit programs using the tools available for this course. Your program will print some basic information, to build upon the basic example shown in class. Please note that, unlike most assignments, this program is worth **50 points** due to its simplicity. A typical assignment will be worth 100 points.

2. Deliverables

40 points: Submit your source file to Dr. Geiger (Michael_Geiger@uml.edu) as an e-mail attachment. Ensure your source file name (project name does not matter—see Section 4) is ***prog1_simple.c***.

You should submit only the .c file. Failure to meet this specification will reduce your grade, as described in the program grading guidelines, which you are strongly encouraged to read before starting the assignment.

The .c file will typically be stored in a project subdirectory. For Visual Studio users, the file will be located in the project subdirectory that has the same name as the project itself. In the example shown in Section 4, since the project is stored in "C:\Users\Michael_Geiger\Documents\Visual Studio 2012\Projects\" and its name is "test_project", you can find the source file for this project in:

C:\Users\Michael_Geiger\Documents\Visual Studio 2012\Projects\test_project\test_project\

10 points: Introduce yourself to Dr. Geiger, either before/after class or at his office (Perry 118A). Dr. Geiger is available before class or during office hours (TTh 12-1). Make an appointment if these times do not work for you. Students who have met Dr. Geiger previously must still meet with him this semester to earn these points.

3. Specifications

For this assignment, write a simple C program that prints the following information. Each bullet point below corresponds to a single line of output:

- Your name
- Your major
- Your class (i.e. freshman, sophomore, etc.)
- The name and semester of this course

Ensure that your code contains appropriate comments, as discussed in the grading guidelines and in class. For this program, you can just write a header comment giving your name, the date, and a brief description of the assignment.

4. Using Visual Studio

Mac users: Please skip to Section 5, using Xcode, which describes how to set up a project using that IDE.

We will create a sample project to help illustrate the use of Visual Studio. Note that this tutorial assumes the use of Visual Studio 2012; Visual Studio 2013 works similarly.

After starting Visual Studio, select **File→New→Project** from the main menu, or simply click the "**New Project …** " link on the welcome screen.

The dialog window that appears allows you to choose the type and name of your project. After selecting **Visual C++** in the list of templates on the left, choose **Win32 Console Application** from the list of project types in the middle. (Note that your choices may not exactly match those shown in Figure 1.) Use the boxes at the bottom of this window to specify a name and location for your project.

Important note: Your project name **will not need to match the assignment name**. To avoid confusion, I suggest choosing a generic name for the project (for example, "program1") and then naming your source file appropriately later.

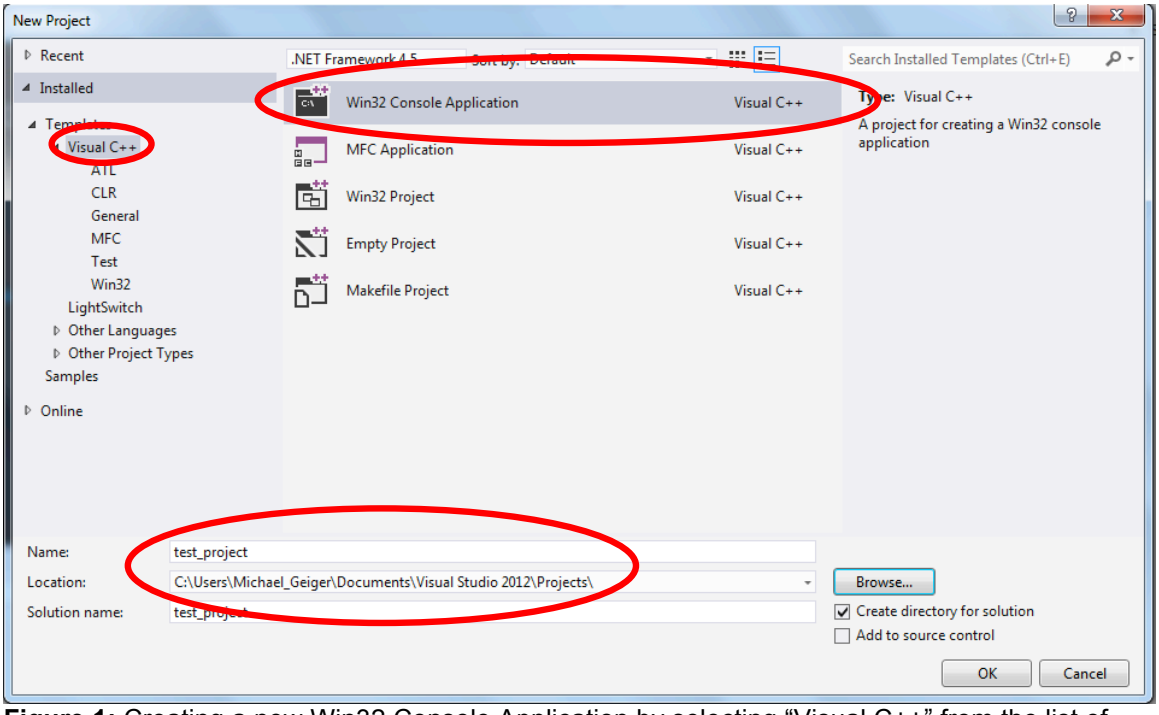


Figure 1: Creating a new Win32 Console Application by selecting "Visual C++" from the list of templates, "Win32 Console Application" from the list of project types, and specifying a name and location for your project. In this example, the project—not the source code file—is called "test_project" and stored in "C:\Users\Michael_Geiger\Documents\Visual Studio 2012\Projects\".

After accepting these settings, a window appears that you can use to set application settings. Click **Next**, then select the check box next to **Empty project**, which is under **Additional options**, in the following window. Click **Finish** to create your project.

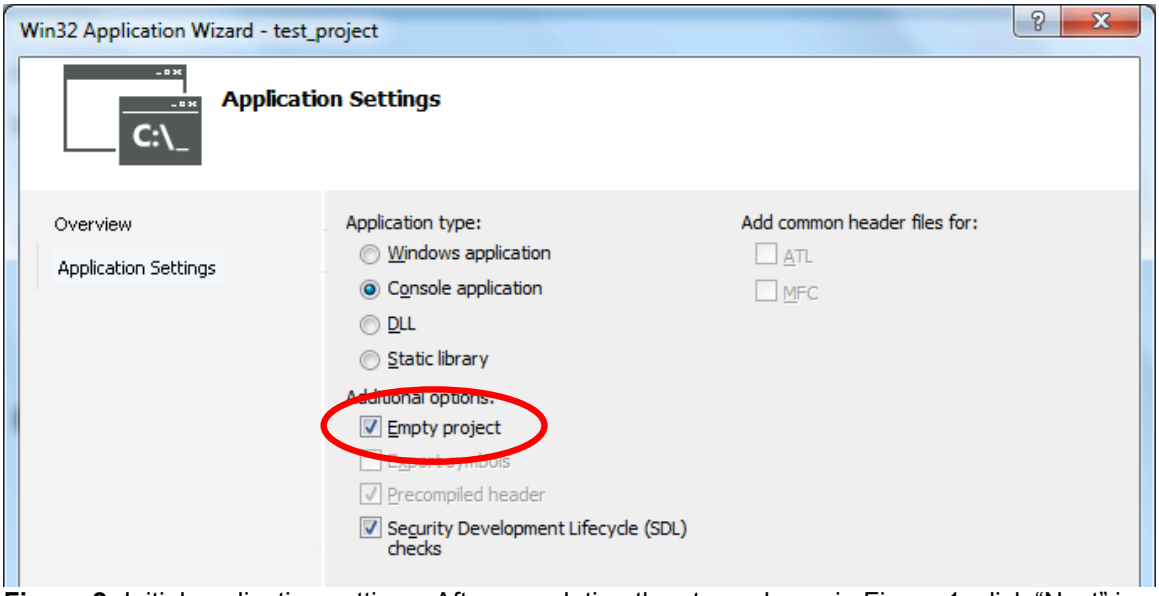


Figure 2: Initial application settings. After completing the steps shown in Figure 1, click "Next" in the first window that appears to reach this window—do not choose "Finish." Once you reach this window, be sure to select the "Empty project" check box before clicking "Finish".

To create your first C file, right click on **Source Files**, in the **Solution Explorer** window. Choose **Add → New Item**. In the list that appears, choose **C++ file (.cpp)**, then name your file. **Remember that, in your programming assignments, source file names are specified for you. You must choose a name with a .c extension so that the program will be saved as C source code, not C++ source code.** See Figures 3 and 4 below.

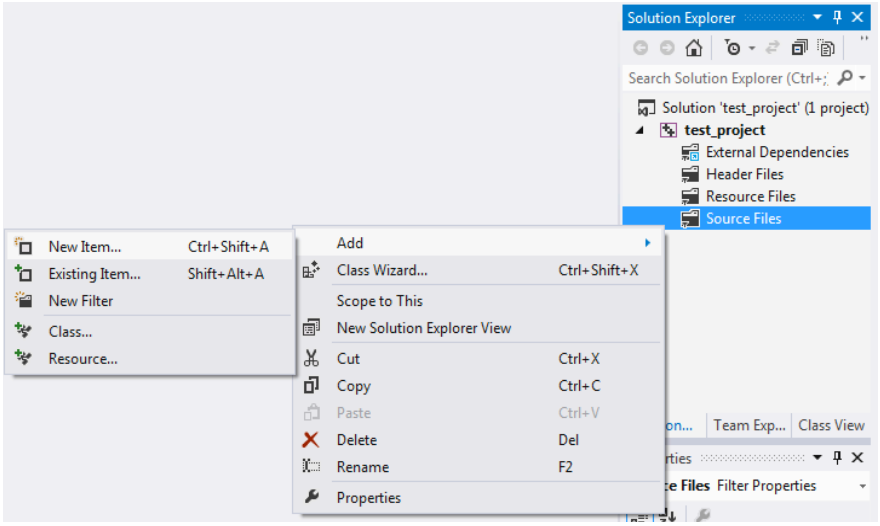


Figure 3: Adding a new source file to your project

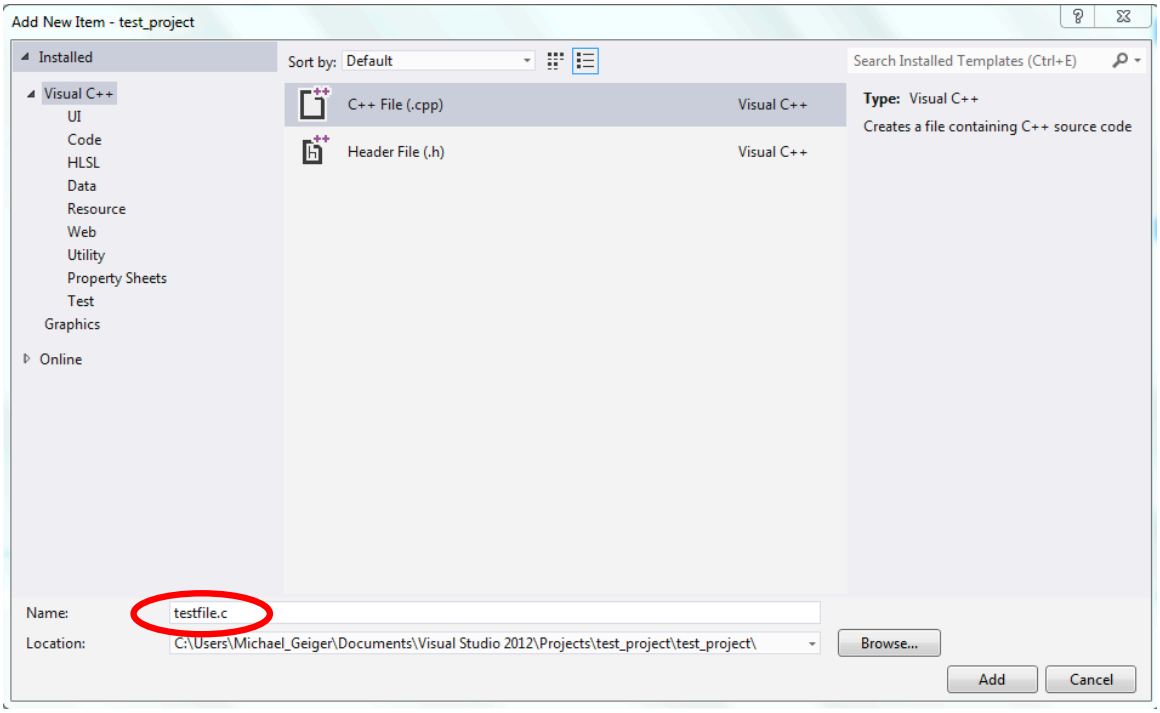


Figure 4: Naming your source file. Note that the file name must end with ".c".

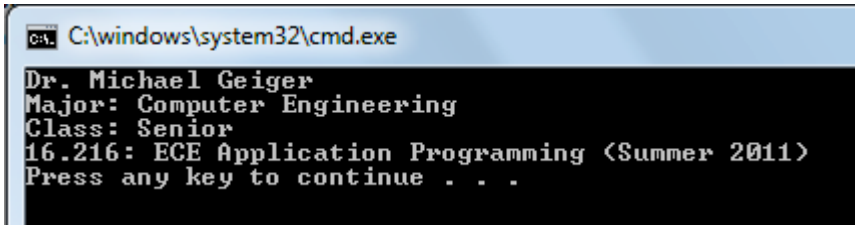
5. Using XCode

I unfortunately don't have figures to describe the steps required to set up a new project in Xcode. However, the following is a list of the required steps:

- Choose **Start New Project** when Xcode opens
- Setting the project type
  - Under the list of **OS X** choices on the left, choose **Application**
  - Choose **Command Line Tool** from the options that appear
- Name your project (*project name doesn't matter*) and choose a directory. Also, ensure that the **type of project is set to "C"** using the appropriate drop-down menu
- By default, the project will include a simple C file named "main.c"
  - You can edit this file to include your own code
  - **Rename this file to match the program specification** (in other words, for this assignment, change "main.c" to "prog1_simple.c")

6. Test Cases

Given the simplicity of this assignment, the "test cases" simply show the appropriate formatting for your output:



Note: to get your program to terminate with a message saying, “Press any key to continue …”, use the **Start Without Debugging** command (press Ctrl + F5) to run your code. Do not use additional code—such as the `system("pause")` function or an infinite loop—in your code to achieve the same result. Doing so will render our grading program useless.