EECE.2160: ECE Application Programming

Summer 2017

Programming Assignment #1: A Simple C Program

Due **Thursday, 5/18/17**, 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, thus building 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

10 points: Introduce yourself to your instructor either before/after class or during office hours. Make an appointment if these times do not work for you. Students who have met the instructor previously must still meet him this semester to earn these points.

10 points: E-mail Dr. Geiger at Michael_Geiger@uml.edu with your preferred e-mail address for your Dropbox account so he can provide you with a link to a shared folder you will use to submit all assignments, as described in class. If you do not have a Dropbox account, set one up prior to contacting Dr. Geiger. You should contact Dr. Geiger well in advance of the Program 1 due date to ensure your Dropbox folder is set up with plenty of time to spare.

30 points: Submit your source file by uploading your source file directly to your Dropbox folder. 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\

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.

(Section 4 begins on the next page of this document.)

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 2015. (Figures 1 and 2 actually show screens from Visual Studio 2012, which works similarly.)

1. 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.

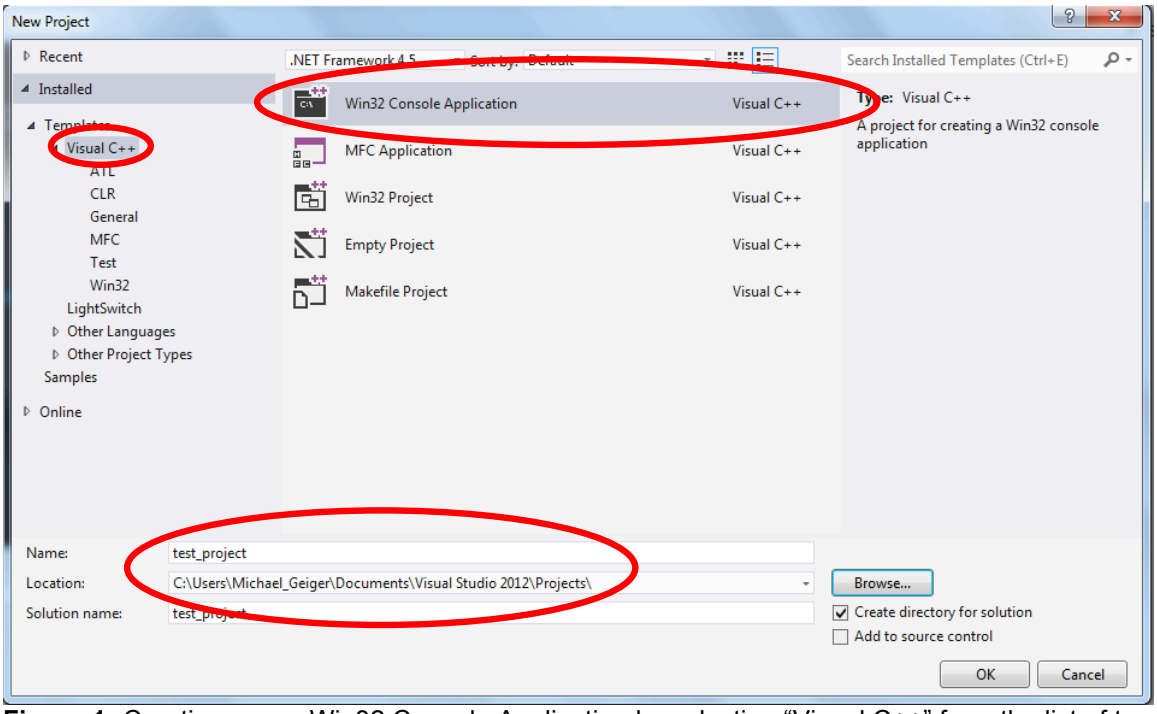


Figure 1: Creating a new Win32 Console Application by selecting "Visual C++" from the list of templates, "Win32 Console Application" from the list of project types, and specifying a name and location for your project. In this example, the project—not the source code file—is called "test_project" and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

2. 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.

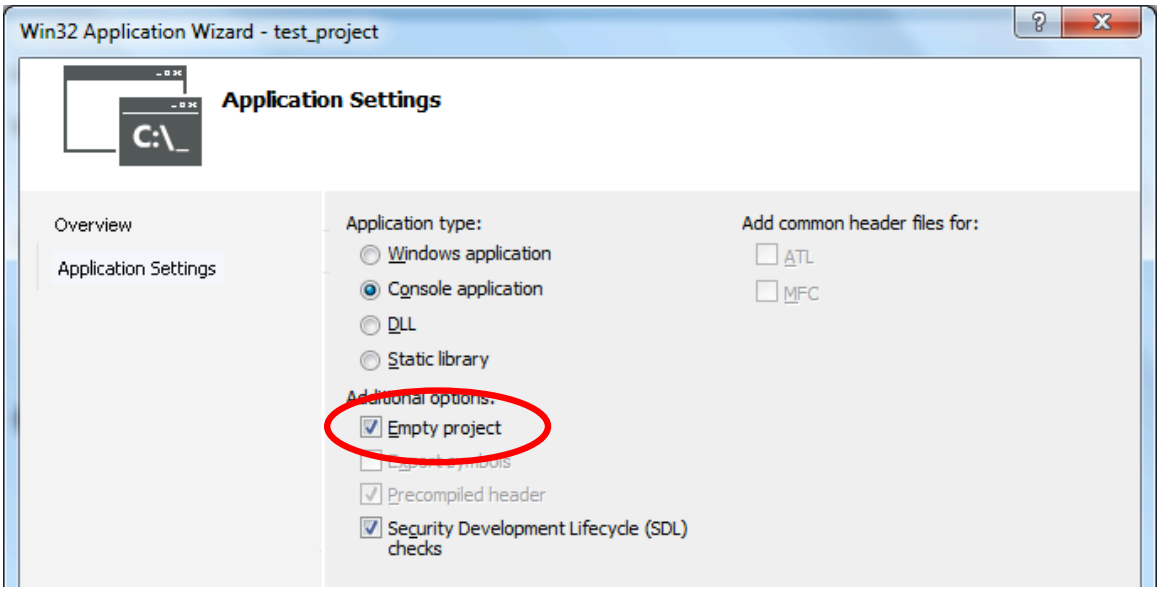


Figure 2: Initial application settings. After completing the steps shown in Figure 1, click "Next" in the first window that appears to reach this window—do not choose "Finish." Once you reach this window, be sure to select the "Empty project" check box before clicking "Finish".

3. To create your first C file, select **File → New File** from the main menu (Figure 3). In the window that appears, choose **Visual C++** from the list of **Installed** templates, then select **C++ file (.cpp)** and select **Open** (or press return) (Figure 4).

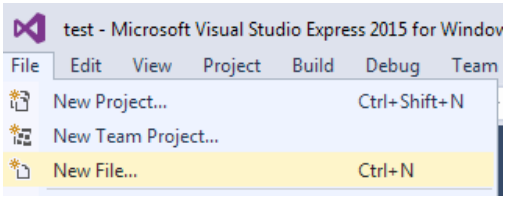


Figure 3: Creating a new file

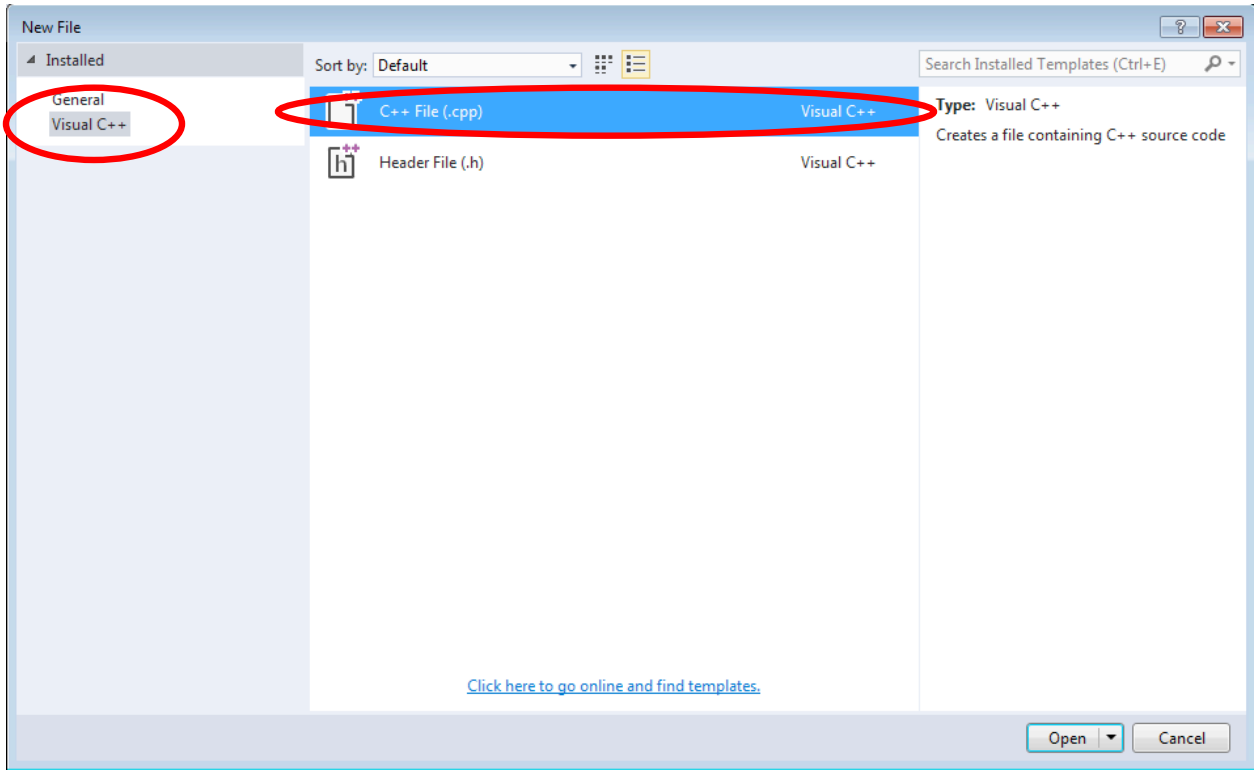


Figure 4: Choosing the appropriate file type when creating a new file

4. To add the new file to the currently open project, it must first be saved to the project directory and then added as an existing item. To save the file, first select the option **Move <fname> into → <projname>** from the main menu, where **<fname>** is the default name for the unsaved file (typically "Source1.cpp") and **<projname>** is the name of your currently open project (in Figure 5 below, "test").

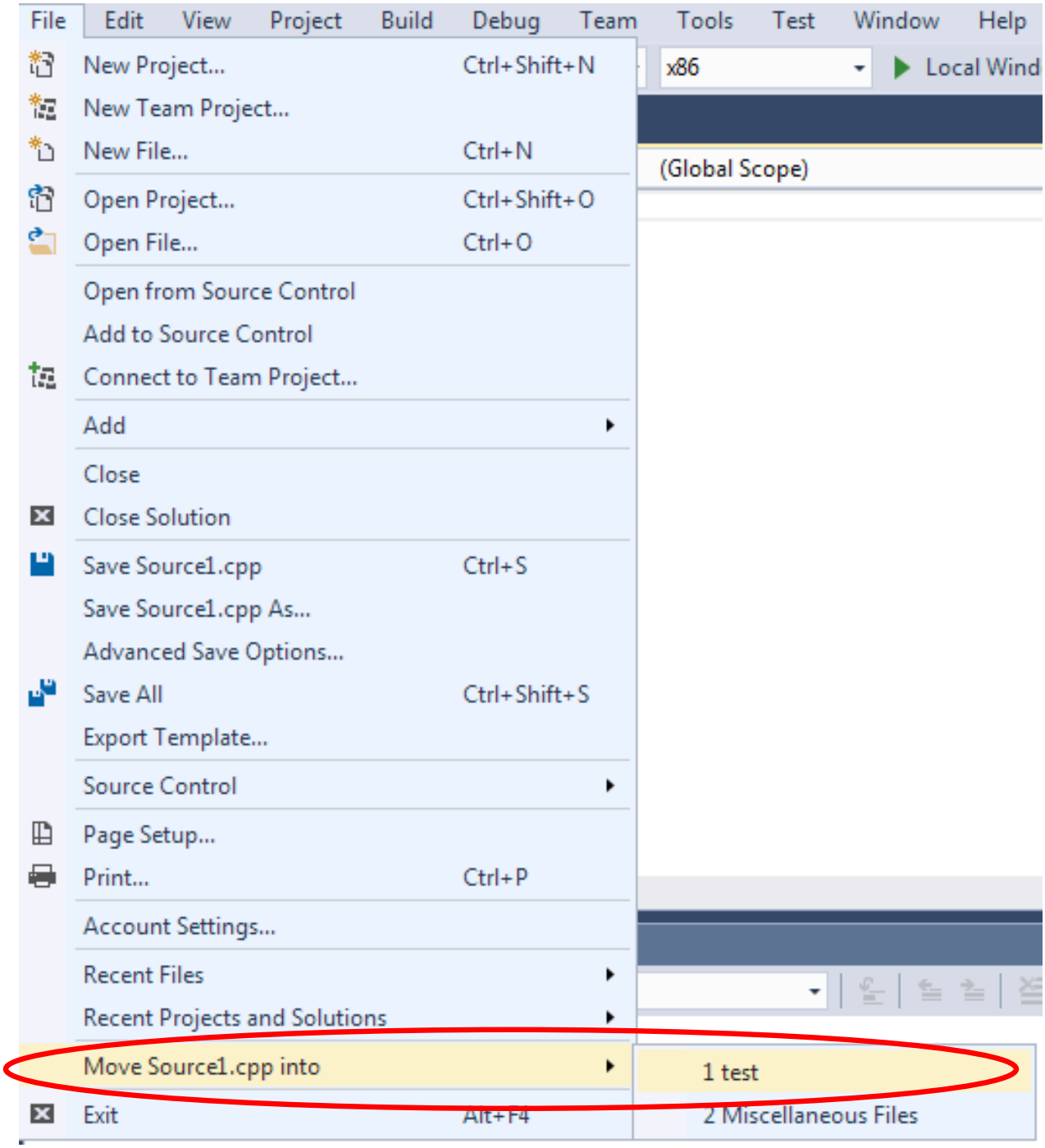


Figure 5: Moving the unsaved file to the project directory

5. In the window that appears, change the file type to a C source file by selecting **C Source File (*.c)** from the **Save as type:** drop-down menu. Type in the desired file name, as shown in Figure 6 below. If you have selected C Source File as your file type, you do not need to add the ".c" extension to the end of the file name. **Remember that, in your programming assignments, source file names are specified for you.**

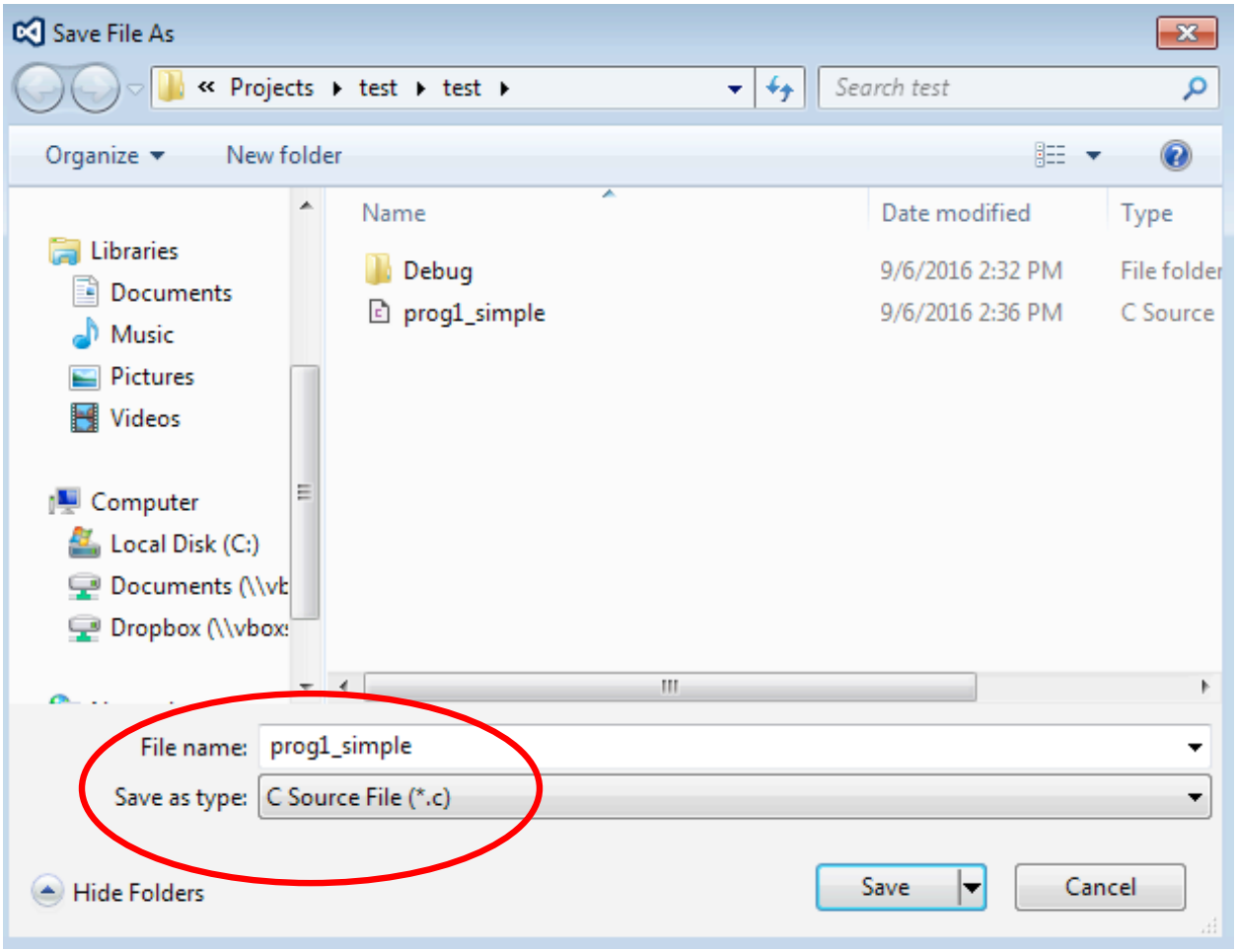


Figure 6: Saving the new source file to the project directory. Remember to select **C Source File (*.c)** as the file type, and ensure the file name matches the name provided in your programming assignment.

6. You can now add the file to the list of source files in your currently open project. If you don't add the new file to the source file list, that file won't be compiled when you build the project.

Note: If you are using a version of Visual Studio earlier than 2015, you should not need to follow this step—step 5 will add the file to the list of source files in your project.

To add the file, right click on **Source Files** in the **Solution Explorer** window. Choose **Add → Existing Item** (Figure 7). In the window that appears, select the file that you saved in the previous step and choose **Add** or press return. Your file will appear under the Source Files folder and will be compiled when you build your project.

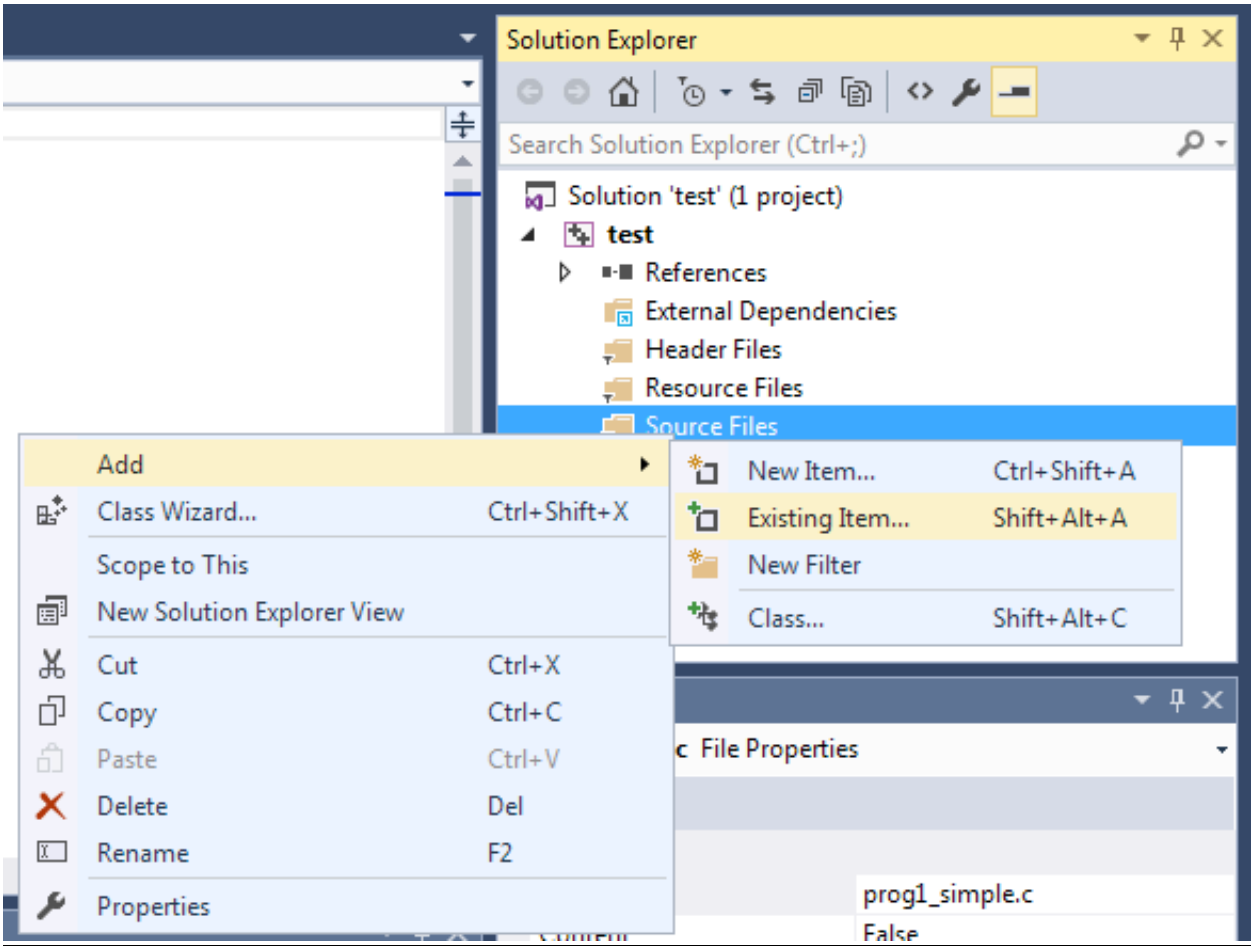


Figure 7: Adding an existing file to your project

5. Using Xcode

This section describes the creation of a sample project to illustrate the use of Xcode:

1. After starting Xcode, select **File→New→Project** from the main menu, or simply click the "**Create a new Xcode project**" link on the welcome screen (Figure 8).

Figure 8: Xcode welcome screen. Choose “Create a new Xcode project” when this screen appears.

2. When prompted to choose a template for your new project, select **Application** from the list of **OS X** choices on the left, then choose **Command Line Tool** from the options that appear (Figure 9).

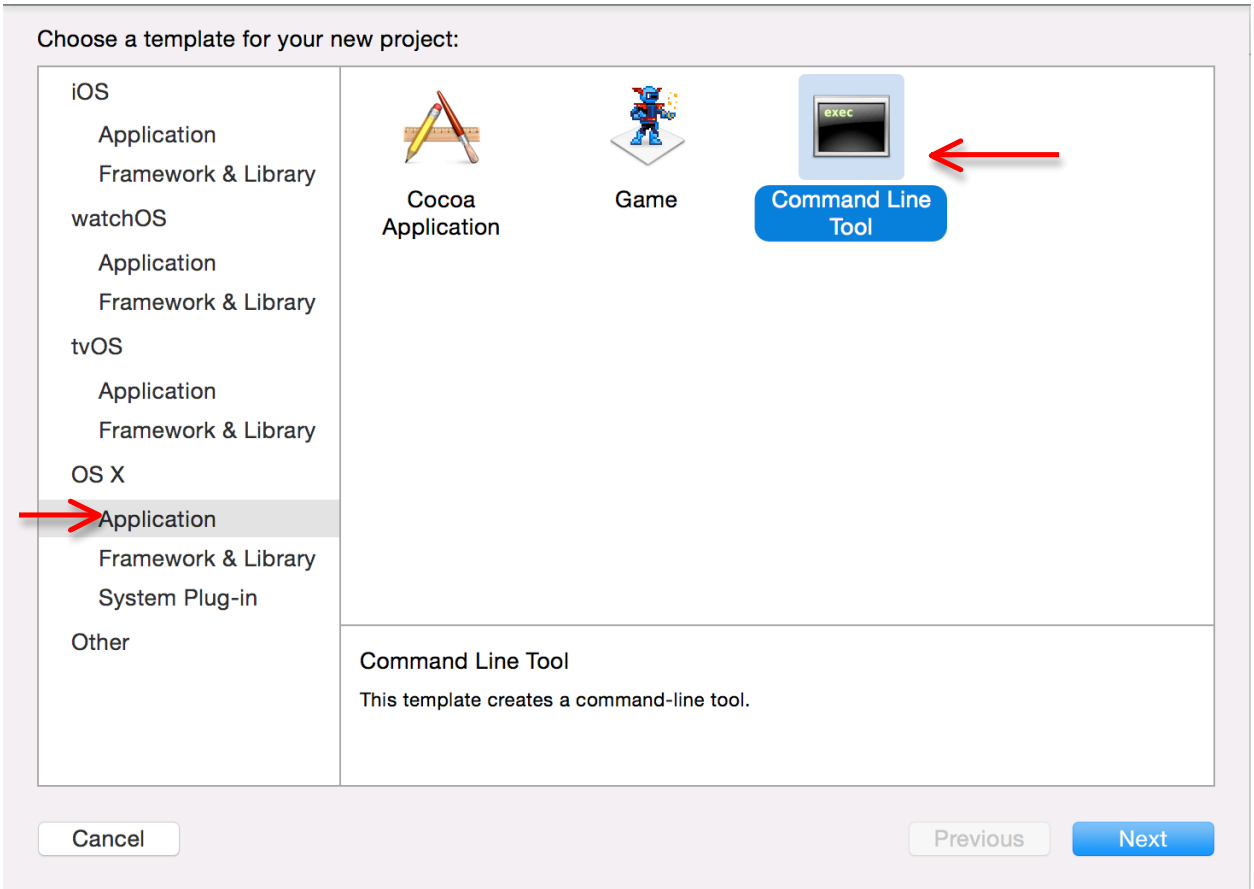


Figure 9: Choosing project type: an OS X Application that runs as a Command Line Tool

3. Name your project (not the same as your source file name). Also, ensure that the **language for your project is set to "C"** using the appropriate drop-down menu (Figure 10). Clicking "Next" at this point will allow you to choose the directory in which the project is stored.

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.

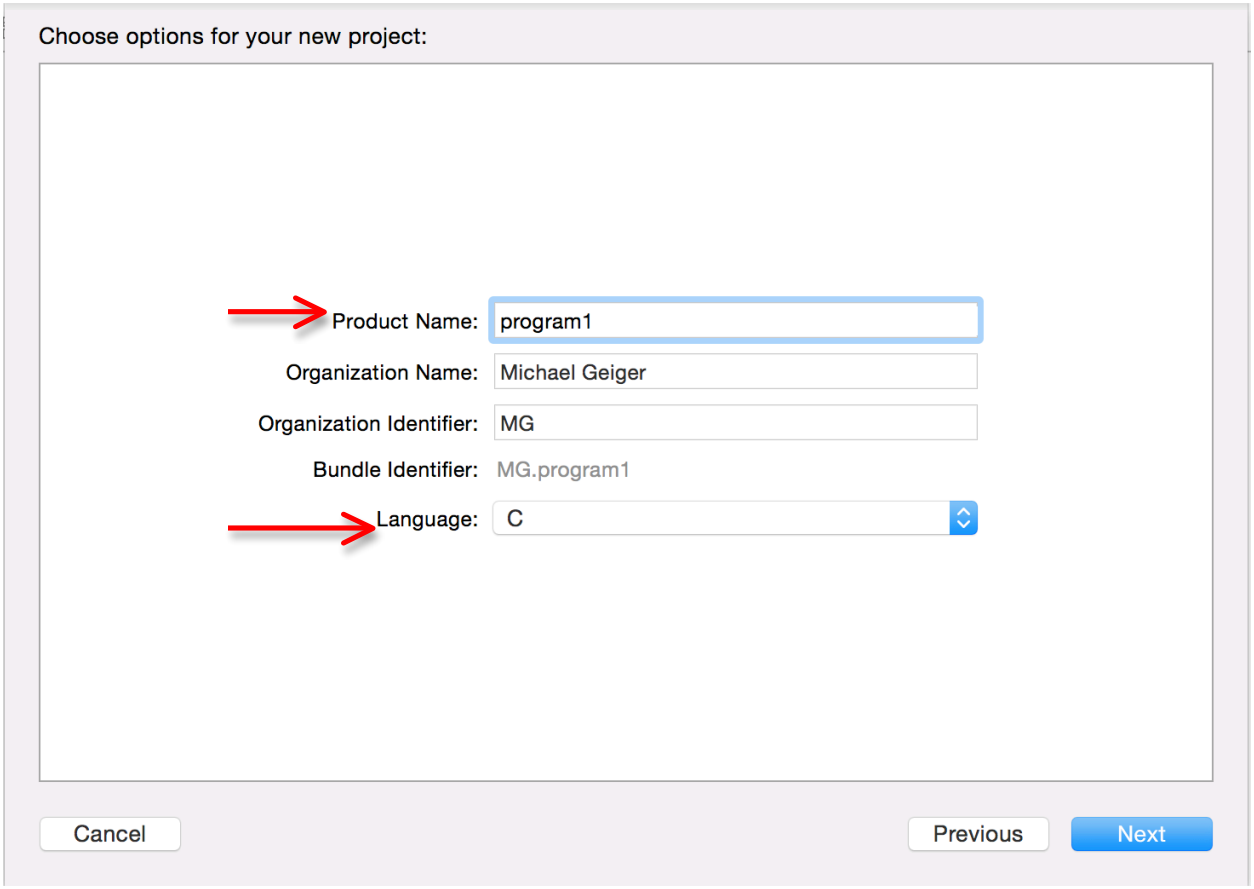
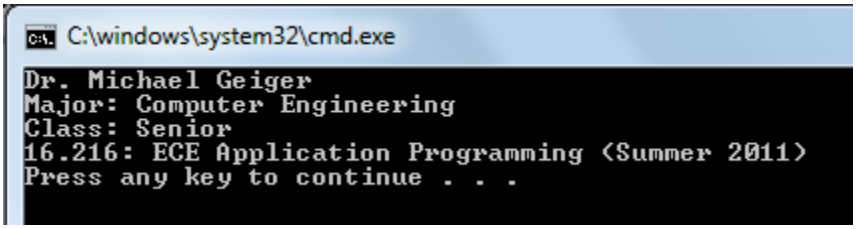


Figure 10: Naming your project. Make sure that the language for the project is set to C.

4. By default, the project will include a simple C file named "main.c," which contains a version of the "Hello, World!" program. You can edit this file to include your own code, but make sure you **rename the 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 …” in Visual Studio only (not Xcode), 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.