

# EECE.2160: ECE Application Programming

Spring 2018

## Programming Assignment #1: A Simple C Program

Due **Monday, 1/29/18**, 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](mailto: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

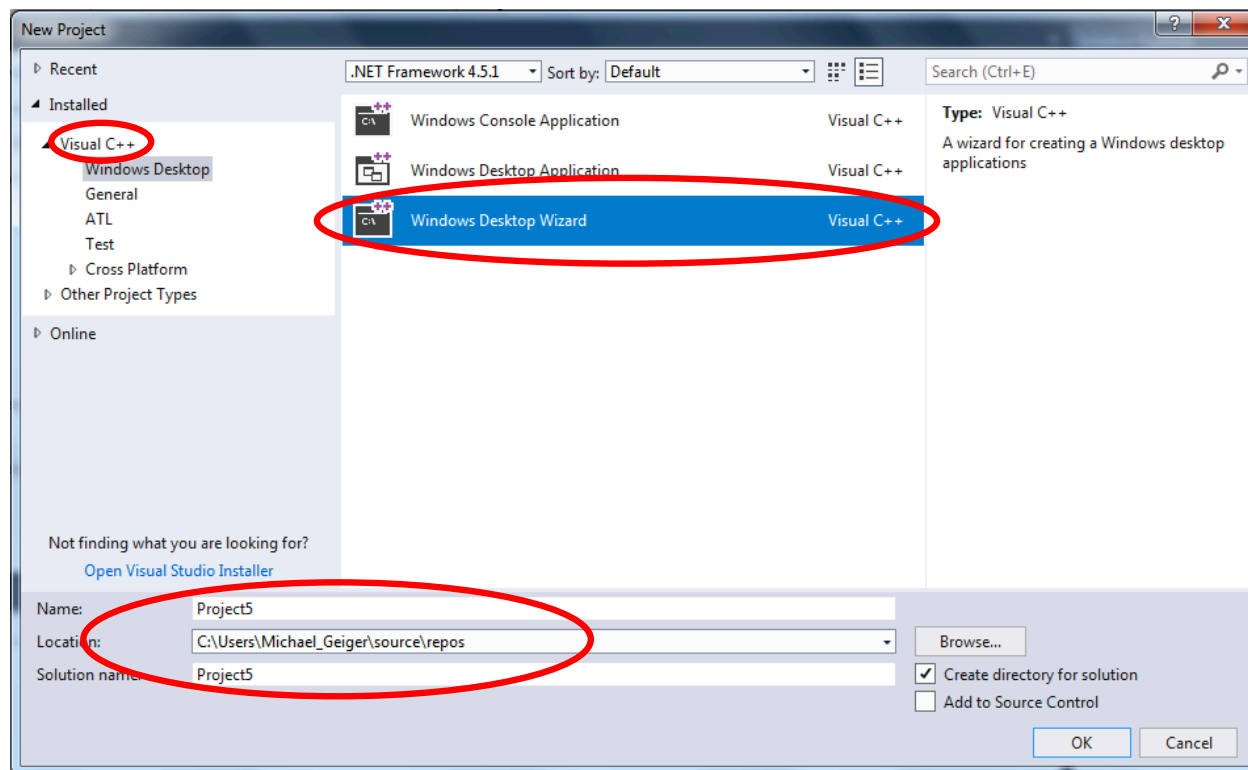
*Mac users: Please skip to Section 5, “Using Xcode,” which describes how to set up a project using that IDE.*

We will create a sample project to help illustrate the use of Visual Studio. Note that this tutorial assumes the use of Visual Studio Community 2017. (Figures 1 and 2 actually show screens from Visual Studio 2012, which works similarly.) **Students using V Labs must follow slightly different steps, which are detailed in a separate document posted on the**

1. After starting Visual Studio, select **File→New Project** from the main menu, or simply click the “**Create New Project ...**” link on the welcome screen.

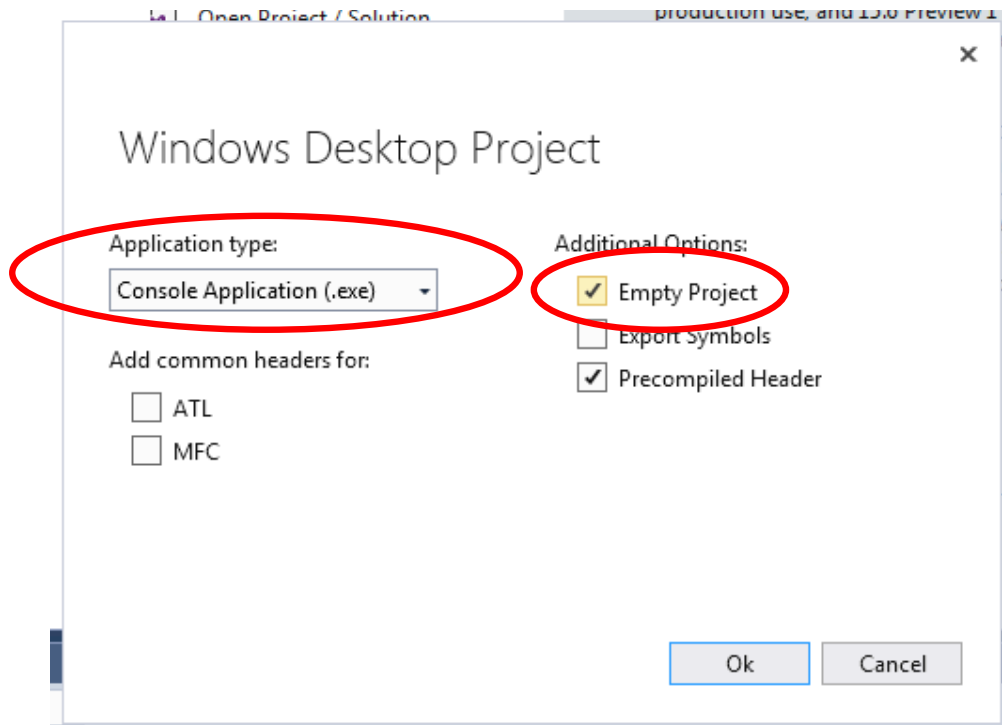
The dialog window that appears allows you to choose the type and name of your project. After selecting **Visual C++** in the list of templates on the left, choose **Windows Desktop Wizard** from the list of project types in the middle, as shown in Figure 1. Use the boxes at the bottom of this window 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.



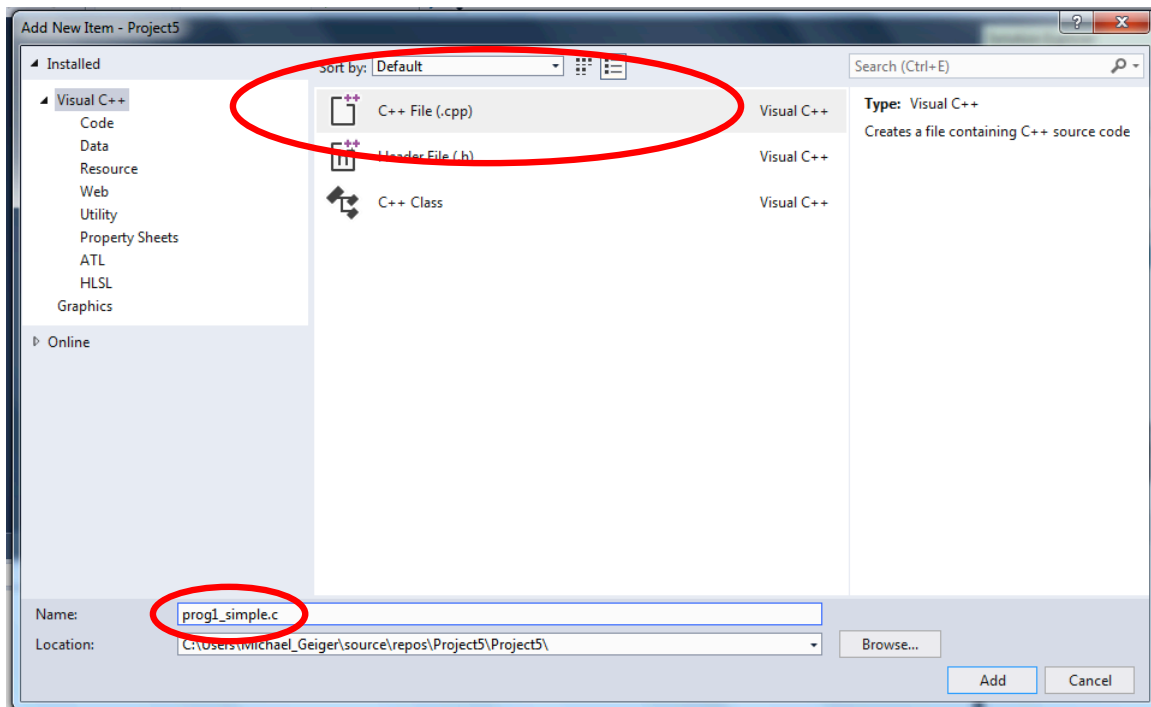
**Figure 1:** Creating a new Visual Studio project by selecting “Visual C++” from the list of templates, “Windows Desktop Wizard” from the list of project types, and specifying a name and location for your project. In this example, the project—not the source code file—is called “Project5” and stored in “C:\Users\Michael\_Geiger\source\repos”.

2. In the dialog box that appears, set the “Application type” to **Console Application (.exe)** and click the **Empty Project** box under “Additional Options,” as shown in Figure 2 below. After selecting these options, click Ok, and Visual Studio will set up an empty project for you.



**Figure 2:** Completing project setup by choosing “Console Application (.exe)” as the Application Type and specifying the Empty Project option

3. To create your first file, simply right click the “Source Files” folder in the Solution Explorer window, choose **Add → New Item**, choose **C++ file (.cpp)** as the file type, then type the full name (including the .c extension) for the file you’d like to create (for example, prog1\_simple.c). The file should appear under the “Source Files” folder, which is an indication the file is part of your project and it will be compiled each time you rebuild your project.



**Figure 3:** Specifying the type and name of your new file after right-clicking the Source Files folder in the Solution Explorer pane. The file type should always be “C++ File (.cpp)”, and the file name you specify should exactly match the program specification. For example, in this assignment, your file must be named *prog1\_simple.c*.

## 5. Using Xcode

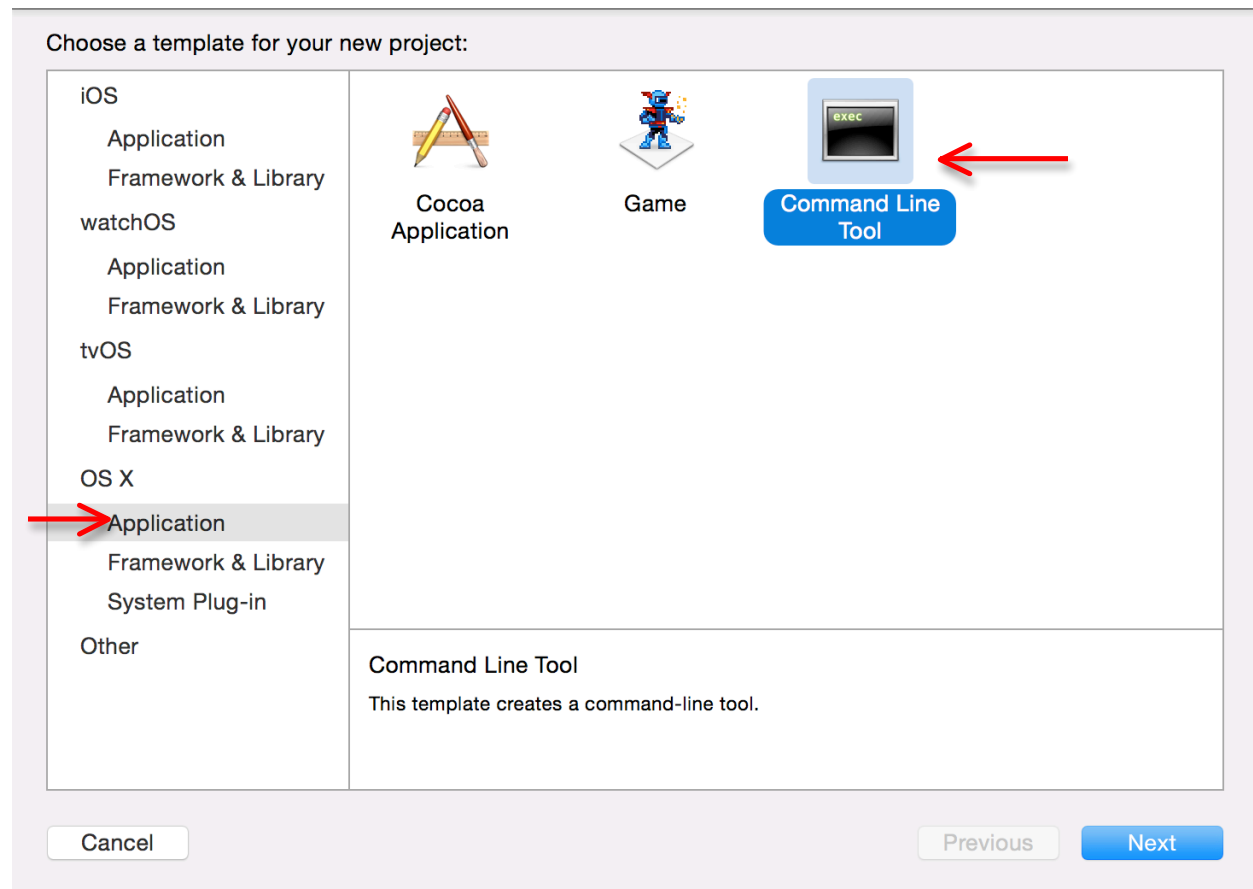
This section describes the creation of a sample project to illustrate the use of Xcode:

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 4:** Xcode welcome screen. Choose “Create a new Xcode project” when this screen appears.

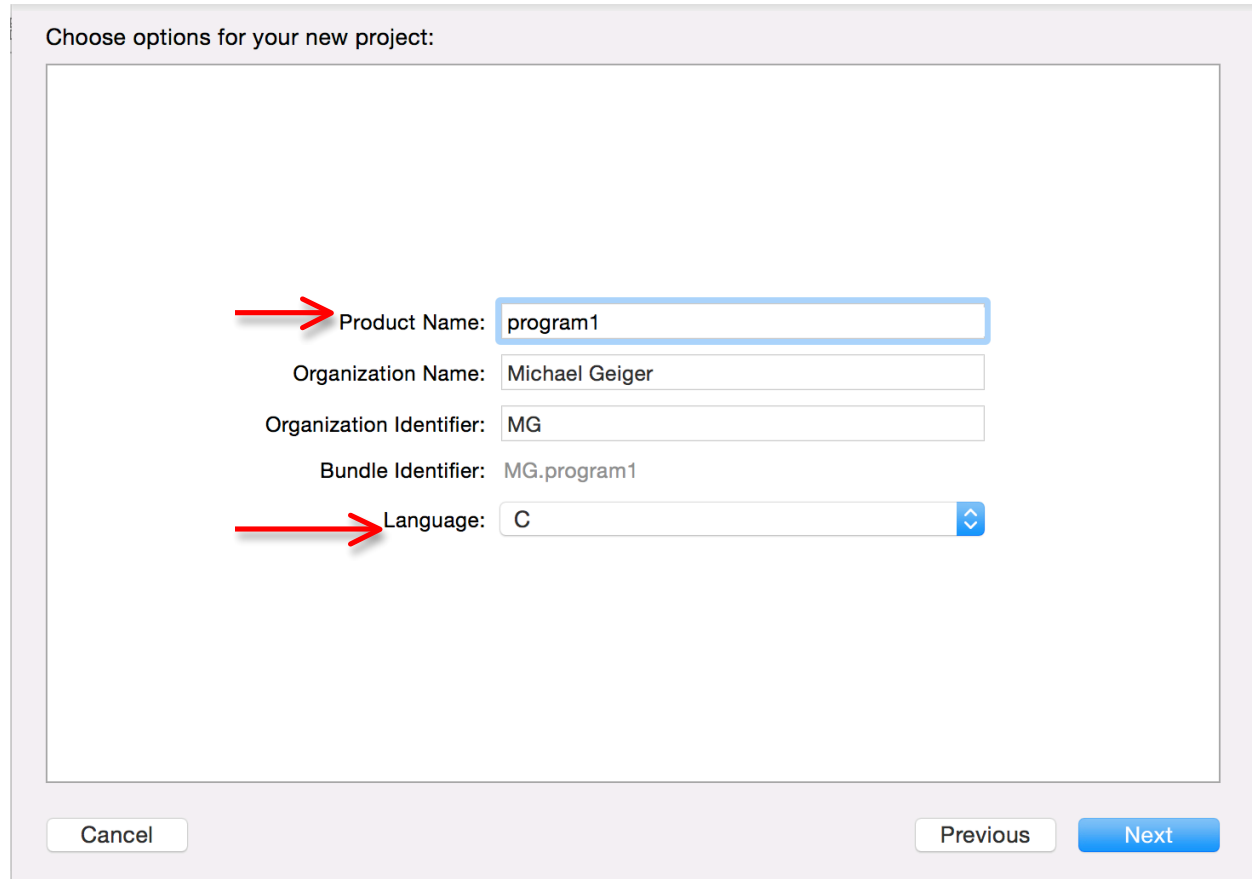
2. When prompted to choose a template for your new project, select **Application** from the list of **OS X** choices on the left, then choose **Command Line Tool** from the options that appear (Figure 9).



**Figure 5:** 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.



Choose options for your new project:

Product Name: program1

Organization Name: Michael Geiger

Organization Identifier: MG

Bundle Identifier: MG.program1

Language: C

Cancel Previous Next

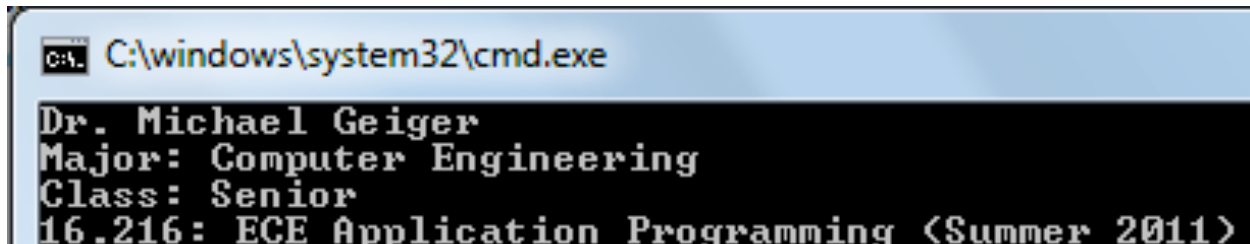
**Figure 6:** Naming your project. Make sure that the language for the project is set to C.

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:

A screenshot of a Windows command prompt window. The title bar at the top reads "C:\windows\system32\cmd.exe". The command prompt shows the following text:

```
Dr. Michael Geiger  
Major: Computer Engineering  
Class: Senior  
16.216: ECE Application Programming <Summer 2011>
```

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.

**VISUAL STUDIO COMMUNITY 2017 USERS:** If you start with an empty project, as described earlier, your program will not pause by default at the end of execution when using "Start Without Debugging." So, you have to modify the project settings to get this behavior.

Right click on your project, then choose **Properties → Configuration Properties → Linker → System**. Under the "SubSystem" option, select "Console" (/SUBSYSTEM:CONSOLE). Your project should now work correctly when using "Start Without Debugging."