C How to Program, Before You Begin

Software

Download and install a compiler for your platform.

- If you're on Linux, GCC is most likely already installed.
- If you're on a Mac, you can use Apple's free Xcode IDE, which you can get from the Mac App Store.
- If you're on Windows, you can use Microsoft's Visual Studio Community edition from http://microsoft.com/express.

Creating a Project in Visual Studio

As for the Before You Begin, on Windows all you need to do is install Visual Studio Community. The C++ compiler is capable of compiling almost every program in C How to Program—there are a couple of recent C features that are not supported and we note those in the book.

To compile a C program:
1. Open Visual Studio.
2. Select **File > New > Project…**
3. In the left side of the **New Project** dialog, expand **Templates**, then select **Visual C++.**
4. In the center of the dialog, select **Win32 Console Application**.
5. At the bottom of the dialog, specify a name and location for your project, then click **OK**

In the Win32 Application Wizard dialog:
1. Click **Next>**
2. Uncheck both **Precompiled header** and **Security Development Lifecycle (SDL) checks**
3. Check **Empty Project**
4. Click **Finish**.

Next, add a source code file to the **Source Files** folder in the **Solution Explorer** at the right side of Visual Studio—if this window is not showing, select **View > Solution Explorer**.

- If you want to run an existing program from our examples, you can simply drag its files from Windows Explorer (Windows 7) or File Explorer (Windows 8 and 10) onto the **Source Files** folder in the **Solution Explorer**. Once you've done this, you can type **Ctrl + F5** to compile and run the program.
- If you want to create a program from scratch—which I recommend even for our programs when learning—right click the **Source Files** folder in the **Solution Explorer** then select **Add > New Item…**. In the dialog, select **C++ file**, BUT name the file with the extension ".c" not ".cpp"—this is how Visual Studio knows to compile the program as C rather than C++.

Compiling/Running Using GNU C++ on Linux

The prompt in the shell on our system uses the tilde (~) character to represent the home directory, and each prompt ends with the dollar sign ($) character. The prompt will vary among Linux systems. We assume here that you've already created or already have a .c file (such as one of our examples) that's ready to be compiled.

Step 1: Locating the Application

From a Linux shell, use the command **cd** to change to the directory containing the C file you want to compile. For this discussion we'll call the file **GuessNumber.c** and assume the **examples** folder is in your user account's folder. We'll also assume that **GuessNumber.c** is in the **examples** subfolder **ch01**.

In the shell window, type

cd examples/ch01

then press *Enter*. You can see the contents of the directory by typing **ls** and pressing *Enter*.

Step 2: Compiling the Application

Before running the application, you must first compile it by typing

gcc -std=c11 GuessNumber.c -o GuessNumber

This command compiles the application for C11 and produces an executable file called **GuessNumber**. For other programs you'd replace **GuessNumber** with the appropriate name. If the program contains multiple .c files and only that program is in the directory, you can use *.c to indicate that all .c files in the directory should be compiled.

Step 3: Running the Application

To run the executable file GuessNumber, type

./GuessNumber

at the next prompt, then press Enter. The **./** tells Linux to run from the current directory and is required to indicate that GuessNumber is an executable file.

Creating a Project in Xcode on MacOS

Step 1: Launching Xcode

Open a Finder window, select Applications and double click the Xcode icon. If this is your first time running Xcode, the **Welcome to Xcode** window will appear. Close this window for now—you can access it any time by selecting **Window > Welcome to Xcode**.

Step 2: Creating a Project

1. **Select File > New > Project….**
2. In the **OS X** (or **MacOS**) subcategory **Application**, select **Command Line Tool** and click **Next**.
3. Provide a name for your project in the **Product Name** field.
4. Ensure that the selected Language is **C** and click **Next**.
6. Specify where you want to store your project, then click **Create.**

By default, Xcode creates a **main.cpp** source-code file containing a simple program that displays "Hello, World!". The window is divided into four main areas below the toolbar: the Navigator area (left), Editor area (center) and Utilities area (right) are displayed initially. We'll explain momentarily how to display the Debug area (bottom) in which you'll run and interact with the program.

Step 3: The main.c File

You can use main.c as the file in which to type a new program, or you can delete that file and add an existing .c file to the project. To type a new program, double click main.c to open it.

To add a different file, select main.c and delete it, then right click the folder that contained main.c and select **Add Files to "folder name"….** You can then navigate to where the existing file(s) is(are) stored and select them to add them to the project. You can also simply drag files from the Finder into the folder.

Step 4: Compiling and Running the Project

To compile and run the project so you can test-drive the application, simply click the run button (looks like a right facing triangular Play button) at the left side of Xcode's toolbar. If the program compiles correctly, Xcode opens the Debug area (at the bottom of the Editor area) and executes the program in the right half of the Debug area. If the program is interactive, you can click in that window and type any required data.