CSCI 103: Introduction to Programming - Lab 14
Part 1: Survey

For the first 20-25 minutes of lab (or until all students have finished) we will fill out this form:

- https://forms.gle/Mi2wqtDQy69j92rK6

To receive credit for this lab: TAs and staff will verify you submitted the form and completion of the second part of today’s lab.
Part 2: Developing Outside of Codio

1. While we ALL love Codio, it will be important to be able to develop, compile, and test code on your laptop, natively. Thus, we want to help you install a working C++ compiler, editor, and other tools on your laptop and have you write and run a simple "Hello world" application.
Part 3: Streaming Project Questions

1. Stick around to get help on the Streaming project if you finish the other tasks early.
Tutorial of MS VS Code
MS Visual Studio Code

• Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.
• It is ONLY an editor, but can be linked to a compiler and a debugger if they are installed on your machine
• We will have you install those as well
Note About AI-Related Extensions

- We know there are extensions for VSCode that can connect to AI-bots (like chatGPT) in the background to generate a line of code for you OR entire functions at a time.
- For your CS course (103, 104, 356, etc.) these are **NOT allowed** and represent a violation of academic integrity, since you are not the author. Use them **ONLY** if the instructor has specifically allowed you to.
Instructions Based on OS

- Windows (slide 9)
- Mac (slide 15)

If you already have your own toolchain (IDE or separate editor/compiler) installed, great! Demonstrate that you can compile and run a simple C++ program to get credit for this lab.
Windows OS Instructions
Using GCC with MinGW on **Windows**

(MacOS?? Go to slide 15)

- In this tutorial, you’ll configure Visual Studio Code to use the GCC C++ compiler (g++) and GDB debugger from mingw-w64 to create programs that run on Windows.
- After configuring VS Code, you will compile and debug a simple Hello World program in VS Code.
- If you have any problems, feel free to check the VS Code documentation repository.
Prerequisites

• Install **Visual Studio Code**
• Install the **C/C++ extension for VS Code**. You can install the C/C++ extension by searching for 'c++' in the Extensions view.

![Visual Studio Code Extensions](image)

• Get the latest version of Mingw-w64 via **MSYS2**, which provides up-to-date native builds of GCC, Mingw-w64, and other helpful C++ tools and libraries. You can download the latest installer from the MSYS2 page or use [this link to the installer](link).
• Follow the Installation instructions on the [MSYS2 website](link) to install Mingw-w64. Take care to run each required Start menu and pacman command.
Prerequisites

• Install the Mingw-w64 toolchain (\texttt{pacman} -S --needed base-devel mingw-w64-x86_64-toolchain). Run the \texttt{pacman} command in a MSYS2 terminal. Accept the default to install all the members in the \texttt{toolchain} group.

• Add the path to your Mingw-w64 \texttt{bin} folder to the Windows \texttt{PATH} environment variable by using the following steps:
  – In the Windows search bar, type 'settings' to open your Windows Settings.
  – Search for \texttt{Edit environment variables for your account}.
  – Choose the \texttt{Path} variable in your \texttt{User variables} and then select \texttt{Edit}.
  – Select \texttt{New} and add the Mingw-w64 destination folder path to the system path. The exact path depends on which version of Mingw-w64 you have installed and where you installed it. If you used the settings above to install Mingw-w64, then add this to the path: \texttt{C:\\msys64\\mingw64\\bin}.
  – Select \texttt{OK} to save the updated PATH. You will need to reopen any console windows for the new PATH location to be available.
Check your MinGW

• To check that your Mingw-w64 tools are correctly installed and available, open a new Command Prompt (do NOT use the previous one you had open) and type:
  – gcc --version
  – g++ --version
  – gdb --version

• If you don't see the expected output or g++ or gdb is not a recognized command, make sure your PATH entry matches the Mingw-w64 binary location where the compilers are located. If the compilers do not exist at that PATH entry, make sure you followed the instructions on the MSYS2 website to install Mingw-w64.

• If gcc has the correct output but not gdb, then you need to install the packages you are missing from the Mingw-w64 toolset.
  – Missing the mingw-w64-gdb package is one cause of the "The value of miDebuggerPath is invalid." message upon attempted compilation if your PATH is correct.
Create Hello World

• Follow Instructions on [this link](#)
Mac OS Instructions
Using GCC with MinGW on MacOS

- In this tutorial, you configure Visual Studio Code on macOS to use the Clang/LLVM compiler and debugger.
- After configuring VS Code, you will compile and debug a simple C++ program in VS Code.
- If you have any problems, feel free to check the VS Code documentation repository.
Prerequisites

• Install Visual Studio Code on macOS.
• Install the C++ extension for VS Code. You can install the C/C++ extension by searching for 'c++' in the Extensions view.

Ensure Clang is installed

• Clang may already be installed on your Mac. To verify that it is, open a macOS Terminal window and enter the following command:
  – clang --version

• If Clang isn't installed, enter the following command to install the command line developer tools:
  – xcode-select --install
Create Hello World

• Follow Instructions on this link
Debugging

- If you get the error message
  - *expected ; at the end of declaration vector<string> msg*
- Visual Studio Code is using C++03 by default for g++
- To fix this, go to .vscode folder → tasks.json and...
  - add “-std=c++11” under args
- Reopen the workspace/file in VSCode
Checkoff

- Show your course staff that you were able to compile and run the test, "Hello world" program and that you completed the form.