installing the compiler

run "`gcc -v`" in a terminal to check whether you have the compiler

Linux

- you probably have everything you need installed by default
- if not, try this (Debian or Ubuntu): `sudo apt install build-essential`

Mac

- open a Terminal then run this command: `xcode-select --install`
- if it asks to install "command line tools", click the "Install" button

Windows

- use Windows Subsystem for Linux (WSL2) ← *highly recommended!!*
    - see `https://learn.microsoft.com/en-us/windows/wsl/install`
    - reboot (if needed) and after choosing a Linux username and password, run:
      `sudo apt install build-essential`
- **or** use the Chocolatey package manager to install a Windows C compiler
    - see next page →

installing a compiler on Windows using Chocolatey

install Chocolatey package manager

1. in search box, type "cmd" to search for 'command prompt'
2. right-click and choose "Run as administrator" (and confirm it in the pop-up)
3. open a web browser here:
   `https://community.chocolatey.org/courses/installation/installing`
4. set "Choose installation method" to "Basic Chocolatey Install"
5. scroll down to "Install with cmd.exe", copy the code, paste it into your `cmd` window, press return

close the `cmd` window

run PowerShell as administrator

- `choco install unzip mingw make -y`

optional: install a better editor than Notepad or Notepad++

- `choco install nano-win -y`   (as used in Information Literacy)

(to check which packages are installed, you can run: `choco list --local`)

16

installing a shell and terminal on Windows using Chocolatey

optional: install a standard shell, command-line utilities, and terminal

- run: `choco install git.install microsoft-windows-terminal -y`

- in the search box, type "terminal" and run the Terminal app

- under Settings, select "+ Add a new profile" and fill in:
  - Name: `bash`
  - Command line: `C:\Program Files\Git\bin\bash.exe`
  - Starting directory: `C:\Users\`*your-username*

- click "Save"

- under Startup, set:
  - Default profile: `bash`

- click "Save"

close and re-open the Terminal app and you should now be using the bash shell

('serious' text editors such as Emacs, remote access with `ssh`, etc., will all work perfectly in this terminal and shell)