

Installation

CSC100 Introduction to programming in C/C++

Table of Contents

- [1. What are you going to learn?](#)
- [2. Why we don't just get coding](#)
 - [2.1. Philosophy](#)
 - [2.2. Technology](#)
- [3. Infrastructure Setup](#)
- [4. GitHub](#)
 - [4.1. What is it?](#)
 - [4.2. Why are we using it?](#)
 - [4.3. What will you have to do?](#)
 - [4.4. What else can you do?](#)
- [5. Install GNU CC](#)
 - [5.1. Windows](#)
 - [5.2. MacOS](#)
 - [5.3. Linux](#)
- [6. Emacs for C programming](#)
 - [6.1. What is Emacs ?](#)
 - [6.2. How do you use Emacs?](#)
 - [6.3. How will we use Emacs?](#)
 - [6.4. Does it really have to be Emacs?](#)
 - [6.5. What about Emacs' famously "steep learning curve" ?](#)
- [7. Install GNU Emacs](#)
 - [7.1. Download and Installation for Windows](#)
 - [7.2. Download and Installation for MacOS](#)
- [8. Customize Emacs](#)
 - [8.1. Create configuration file](#)
 - [8.2. Create sample notebook](#)
 - [8.3. Layout changes](#)
 - [8.3.1. Customize theme and font](#)
 - [8.4. Installing additional packages](#)
 - [8.5. Presenting in Emacs](#)
 - [8.6. Definitions and functions](#)
 - [8.7. Adding images and links to Org-mode files](#)
 - [8.8. Tables](#)
 - [8.9. Export](#)
- [9. Summary](#)
- [10. Glossary](#)
- [11. References](#)

1 What are you going to learn?

- Why we don't just get coding
- Setting up infrastructure
- Install a C compiler
- Install the Emacs editor and IDE
- Customize the IDE
- Understand and learn to use GitHub
- Summary of concepts/code
- What's next

2 Why we don't just get coding

There are two BIG reasons - one is philosophical, the other is technical.

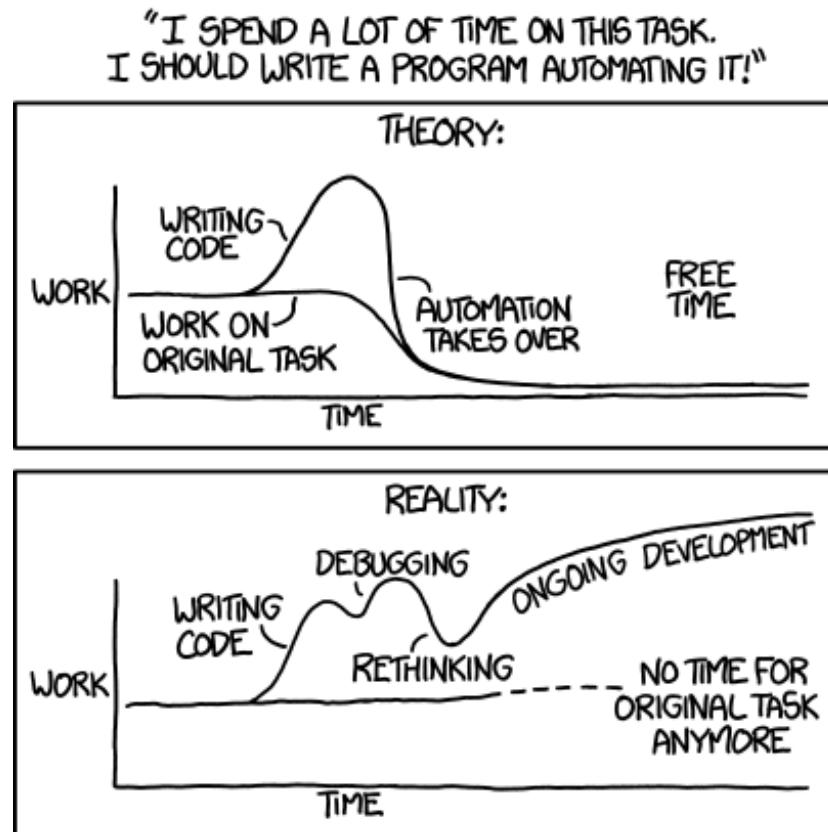


Figure 1: Automation by xkcd.

2.1 Philosophy

"Understanding underlying technologies helps you develop a sense of what can go wrong. Knowing just high-level tools makes it easy to ask the wrong questions. It's worth learning to use a hammer before graduating to a nail gun. Learning underlying systems and tools gives you the power to build new tools, which is important because there will always be a need for tool builders, even if tool users are more common. Learning about computers so that the behavior of programs isn't a mystery enables you to craft better code." (Steinhart, 2019)

- Understanding technologies makes them safer
- Knowing low-level tools helps asking the right questions
- Learning low-level tools helps you build new tools
- Learning about computers enables you to craft better code

2.2 Technology

The other reason has to do with C itself:

TASK	SOFTWARE	EXAMPLE
C source code needs to be written and edited	editor	GNU Emacs
C source code needs to be compiled, linked and debugged	compiler	GNU CC
C object code needs to be run	shell	GNU bash

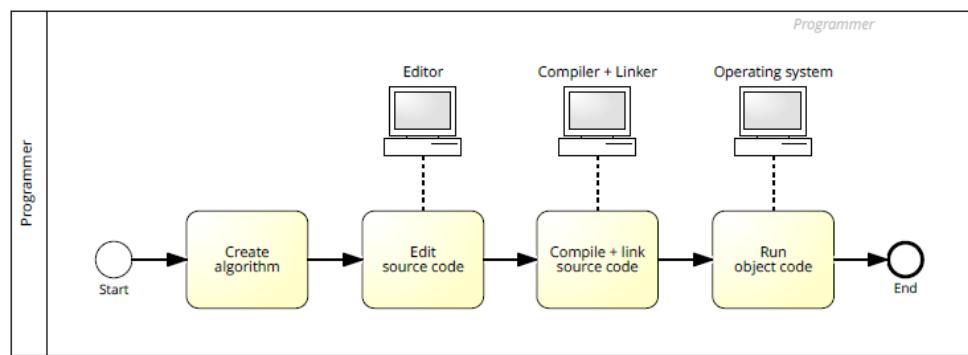


Figure 2: C programming workflow (BPMN model)

There are different routes to get C running on your computer, depending on your operating system (Linux, MacOS or Windows)¹.

The problem with most IDEs is that they're clunky. They take some time getting used to, have many functions that you'll never need, and any time spent on learning the IDE is wasted unless you keep working with it. IDEs do not teach you transferable skills.

3 Infrastructure Setup

The sections below cover most of the infrastructure used in my courses. They are also available as [FAQs on GitHub](#).

If you look at this at GitHub, look at *.org files for syntax highlighting. The Markdown (*.md) files do not have it.

PLATFORM	SOFTWARE	PURPOSE
GitHub	Git	Hosting
MinGW	GCC	Compiler
GNU	Emacs	Editor

4 GitHub

4.1 What is it?

- Software development platform
- Built around Git by Linus Torvalds
- Bought by Microsoft in 2018
- AI support (e.g. [GitHub Copilot](#))

Watch: "[What is GitHub?](#)" (GitHub, 2016)



Gif: "So long binder of requirements" Source: GitHub

4.2 Why are we using it?

Image: Org-mode file in GitHub

main · cc100 / 2_installation / babel_c.org

birkenrahe tweaks · Latest commit f47abc2 1 hour ago · History

All 1 contributor

54 lines (39 sloc) | 1.5 KB

<<babel.org>>

Babel test

This file demonstrates working with source code in Emacs for a number of different languages.

1. To run a code chunk as a whole, type `C-c C-c`. The result will appear immediately below the chunk.[fn:1]
2. look at the code in a separate buffer and run them in parts. To open a buffer with the code, type `~C-c `~`.
3. To print an org-mode file, type `C-c C-e` and choose a print format from the list.

Running chunks will only work if Emacs can find the respective programs[fn:2], and if a compiler (for C), or an interpreter (for R and SQLite) were installed.

The code block needs to be named as shown. If you want the result and the code shown in the printout, you need to specify `:exports both`.

```
#include <stdio.h>

int main(void) {
    puts("hello world");
    return 0;
}
```

hello world

In the second version, both the header and the function definition are present so that you can see the inside of the function only.

```
puts("hello world");
```

hello world

Footnotes

[fn:2]This is why we changed the Windows PATH variable during the installation of the programs R and GNU gcc. ([here](#)).

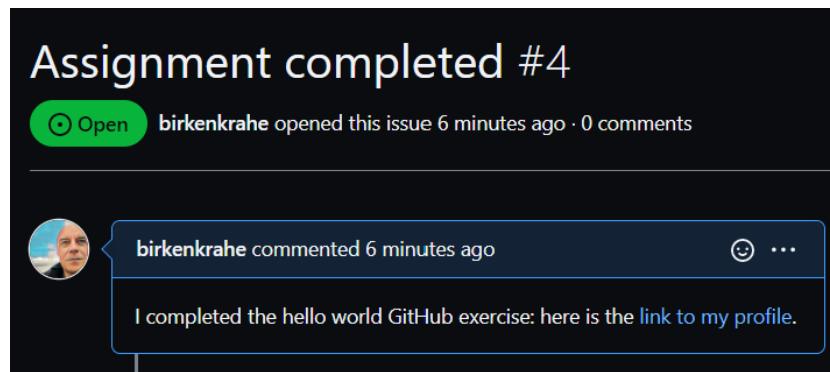
[fn:1]Provided the block has been formatted correctly.

- It's free
- To host course materials
- Upload assignments (esp. Org-files)
- Discussion
- Wiki for collaboration
- Complements Schoology²

4.3 What will you have to do?

- [Sign up with GitHub](#) - use Lyon Email
- Pick an available username **using your own first and last name**, e.g. MarcusBirkenkrahe, or DonaldTrump
- [Complete the "Hello World" exercise \(FAQ\)](#)
- Give me your GitHub username so that I can add you as a collaborator to my private cc100 repository
- [Create an issue](#) from the [cc100 repository](#) like in the example below (except from your account instead of mine).

Image: Issue "Assignment completed"



If you do have a GitHub account already, do the exercise anyway using your existing account (it takes 10 min)! Make sure you let me know what your user name is so that I can add you to my repo.

4.4 What else can you do?

- You can [fork](#) the [cc100](#) repository
- You can [watch](#) the [cc100](#) repository - and set [Notifications](#) to Participating and @mentions so that you see my comments (see image below).

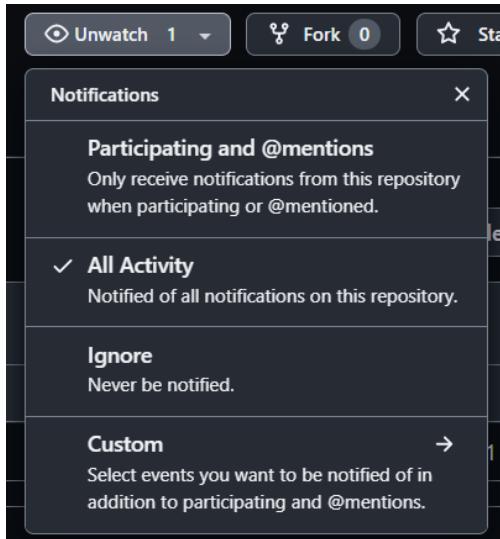
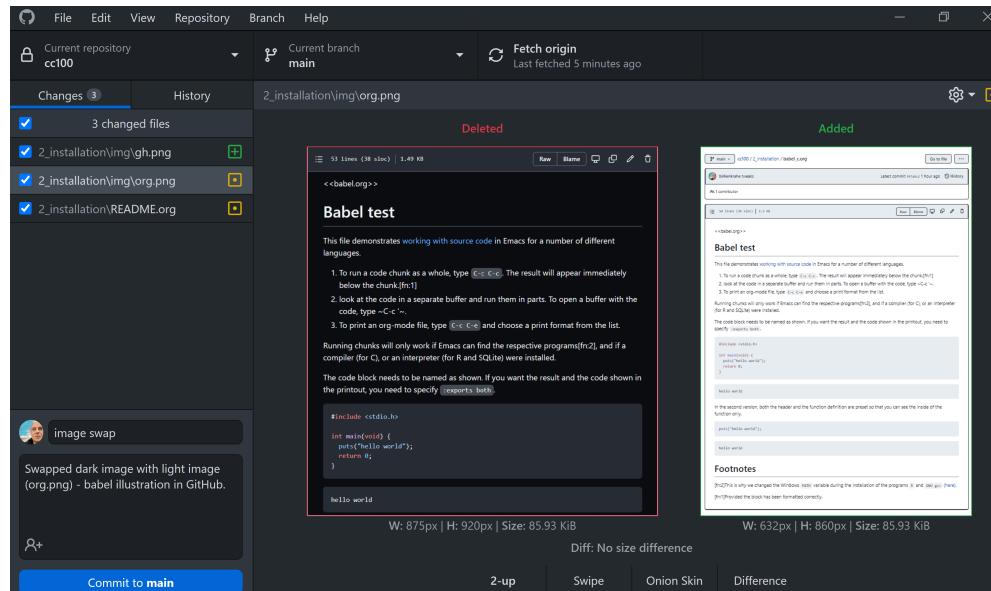


Image: Notifications settings when watching a repository

- You can [submit issues](#) from the repository (e.g. if you notice mistakes or if you want extra information, or to share a link)
- You can participate in [discussions](#) (sometimes I will make you)
- You can add to the [wiki](#) (e.g. comments and links to interesting resources)
- You can install the [mobile app](#) on your smartphone³
- You can use it as a platform for [projects](#) or [coding](#)
- You can download the [desktop client](#) to manage repos on your PC (see image below).

Image: GitHub desktop client commit



5 Install GNU CC

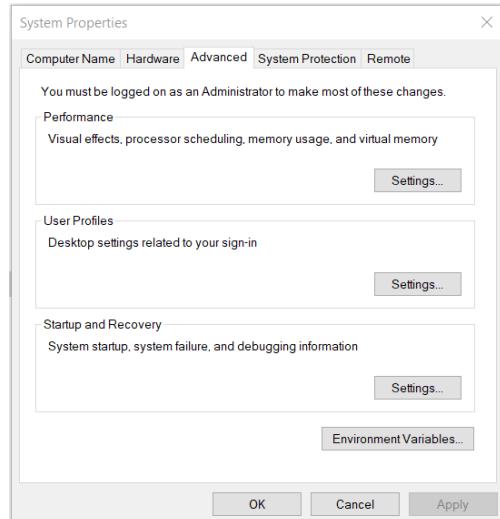
- Compiler & linker are usually bundled
- [GCC](#) ("GNU cc") is a popular C/C++ compiler
- [MinGW](#) ("Minimalist GNU for Windows") is a port of GNU gcc to Windows

Image: Shameless plug for the Free Software Foundation - Source: fsf.org.

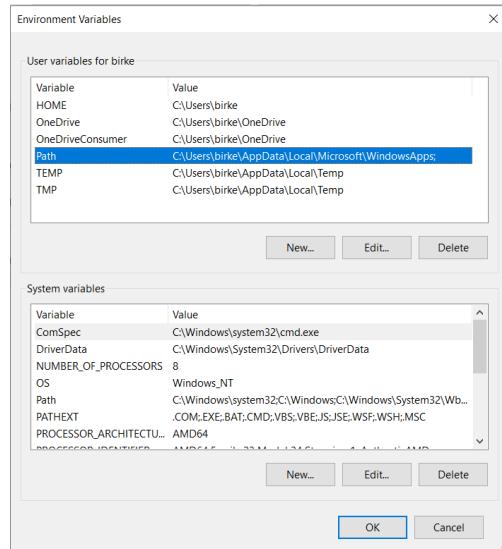


5.1 Windows

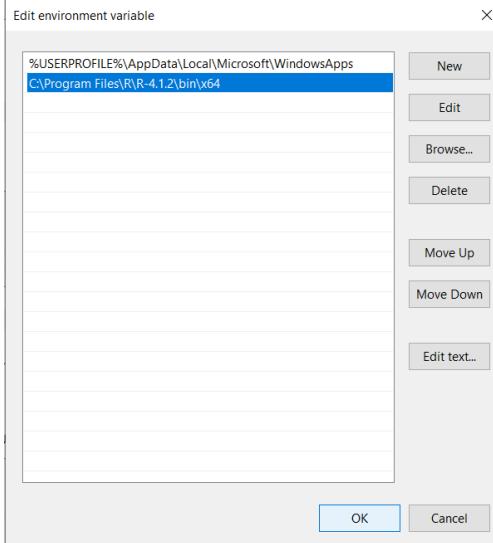
- Download the installer [from sourceforge](#).
- Run the installer - accept all presets.
- Add the location of the executable file gcc to the Windows PATH (you can find it in C:\Program Files(x86)\mingw-w64\):
 - Search for PATH and open the menu System Properties



- Open the menu Environment Variables, click on the PATH variable and choose Edit



- Click on New and paste the path into the empty line. Confirm three times with OK to close all menus.



- To test, search for CMD, open a terminal and enter gcc --version - you should get the output shown below (or similar). Close the window.

```
Command Prompt
Microsoft Windows [Version 10.0.19043.1415]
(c) Microsoft Corporation. All rights reserved.

C:\Users\birke>gcc --version
gcc (i686-posix-dwarf-rev0 Built by MinGW-W64 project) 8.1.0
Copyright (C) 2018 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

C:\Users\birke>
```

5.2 MacOS

Apple no longer allows GNU tools - instead, you can install the free [Clang](#) compiler as part of the [Xcode](#) development suite.

- Check if you already have a C compiler: open a terminal (search for terminal app) and enter `cc -v`.
- If no compiler is found, download it by entering `xcode-select --install`
- Test it by typing `cc -v`.

```
$ cc -v
Apple clang version 12.0.0 (clang-1200.0.32.29)
Target: x86_64-apple-darwin19.6.0
Thread model: posix
InstalledDir: /Library/Developer/CommandLineTools/usr/bin
$
```

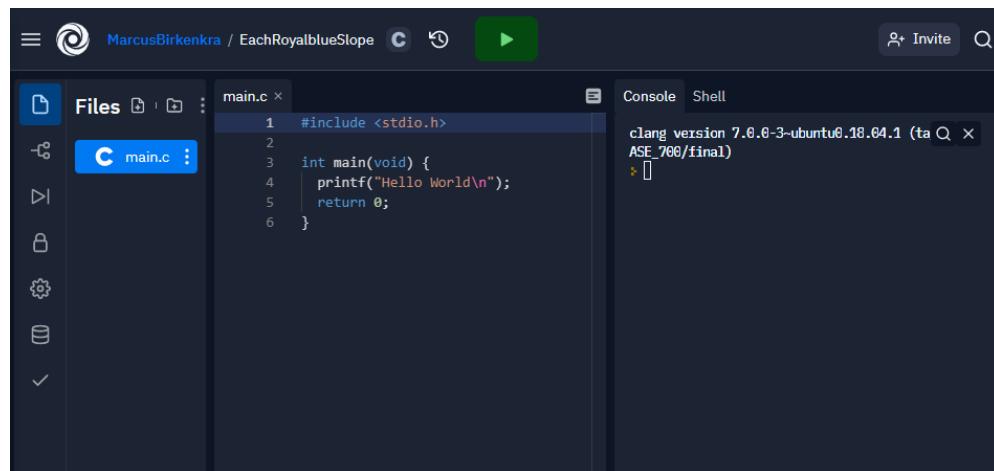
5.3 Linux

Under Linux, the installation of a binary (executable) for that distribution and computer architecture, is done with a simple command from the command line, usually using the package manager, e.g. `apt` for Ubuntu 20.04: `sudo apt install build-essential`. This will install `gcc`, `g++`, and the powerful [GNU make program](#).

6 Emacs for C programming

With the Emacs editor + Org-mode, you can almost program interactively (live code) with C - akin to Python or R. Org-mode inside Emacs works like a REPL (Read-Evaluate-Print-Loop).

A resource to look at, and use (for free, at first) that uses the REPL concept, is [repl.it](#). See image below for the "hello world" program in C.



6.1 What is Emacs ?

PROPERTY	WHAT THIS MEANS
Extensible editor	You can adapt it to your needs ⁴
Written in C with Emacs Lisp	It's fast and smart (via Lisp ⁵)
Ancient software	Written 1976, released in 1985 ⁶
Ca. 1.5M lines of code	By comparison: Windows ca. 50M; Linux kernel ca. 30M

```

File Edit Options Buffers Tools Table Org Text Help
D | X | S | D | X | Q

156 *** Teach a man to fish...
157 #begin_quote
158 Give a man a fish, and you feed him for a day. Teach a man to fish, and you feed
159 him for a lifetime. --- Anne Isabella
160 #end_quote
161
162 Most popular editors have a simple and pretty settings interface, filled with
163 check-boxes, selects, and the occasional text-box. This makes it easy for the
164 user to pick between common desirable behaviours. To me this is now like /giving
165 a man a fish.
166
167 What if you want one of those "check-box" settings to be only on in certain
168 conditions? Some editors have workspace settings, but that requires you to
169 manually set the value for every single instance. Urgh, what a pain.
170
171 What if you could set the value of that "check-box" setting to be the result of
172 an arbitrary expression evaluated for each file? This is where an editor like
173 Emacs comes in.
174 Configuration for Emacs isn't a list of settings in JSON etc. It's an executable
175 program which modifies the behaviour of the editor to suit your likings.
176 This is "Teaching a man to fish".
177
178 Emacs is built in the same language you configure it in (Emacs Lisp, or elisp).
179 It comes with a broad array of useful functions for text-editing, and Doom adds
180 a few handy little convenience functions.
181
182 Want to add a keybinding to delete the previous line? It's as easy as
183 #+name: Keybinding to delete the previous line
184 #+begin_src emacs-lisp :tangle no
185 (map! "C-d"
186   (cmd! (previous-line)
187     (kill-line)
188     (forward-line)))
189 +end_src
U:*** config.org 3% L173 Git:master (Org ws SP wb ~ dtrt-indent GCMH WK better- U:*** emacs.d 3% L173 Git:master (Org ws SP wb ~ dtrt-indent GCMH WK better- /image: "Emacs 27.1 showing Org, Magit and Dired buffers with the modus-operandi theme, without window titlebar or borders." Source: Wikipedia/
```

Challenge: which Emacs properties can you deduce from this image alone?⁷

6.2 How do you use Emacs?

See [FAQ](#). I use Emacs for most of my computing needs:

- Writing (teaching, research)
- Planning (Calendar, ToDo)
- Organizing (Files)

See also the article "[Getting started with Emacs](#)" (Kenlon, 2020), and the video "[The Absolute Beginner's Guide to Emacs](#)" (System Crafters, 2020) with [my notes](#).



Image: DESY APE research group (1994). Can you find me?

Other uses:

- As [window manager](#) (only under Linux)
- As [email client](#)
- Remote access (with [GNU Tramp](#))

6.3 How will we use Emacs?



We'll use it as:

- EDITOR to write source code,
- NOTEBOOK to write literate programs, and
- SHELL to build and run code.

"Emacs outshines all other editing software in approximately the same way that the noonday sun does the stars. It is not just bigger and brighter; it simply makes everything else vanish." – Neal Stephenson, In the Beginning was the Command Line (1998)⁸

We will not use Emacs as a substitute for religion even though there is a "[Church of Emacs](#)" (EmacsWiki)! Huh?! What?!

Image: Notre Dame de Paris. Source: Wikipedia.



6.4 Does it really have to be Emacs?

You'll handle it. Keep calm and carry on coding.

If you look around, you'll see a lot of discussion on different source code editors and IDEs. Currently [Microsoft's Visual Studio \(VS\) Code](#) seems to be the most popular contender. However, as one developer said:

"One thing that cannot be replaced by any extension in VS code, VIM or any other editor: Emacs' Org mode. Org mode is for sure one of the most amazing pieces of software I have ever seen or worked with. It does things that no other text-based word processor can do, even if you are writing complex scientific reports. VS code has an extension which brings less than 5% of Org mode functionality, tops and that is mostly the code highlighting." ([Timachi, 2019](#))

6.5 What about Emacs' famously "steep learning curve" ?

"Emacs can be a challenge if you are used to using mouse pointer. One should be willing to leave the mouse and stick with the keyboard." ([Timachi, 2019](#))

Using the keyboard for everything is much faster (than mouse-only, or mouse + keyboard) but takes getting used to. During the writing of this paragraph, I used the following keystrokes (with the command behind the keys, which your fingers will learn):

KEY	COMMAND
<q RET	org-self-insert-command
C-M-\	indent-region
M-q	org-fill-paragraph
C-a	org-beginning-of-line
C-e	org-end-of-line
C-x C-s	save-buffer

Computer science, and IT, are largely about mastering, and creating new tools. Therefore, almost any effort is justified that goes into improving your **meta skills**⁹ in this area.

7 Install GNU Emacs



Image: GNU Emacs logo. Source: [GNU Project](#)

7.1 Download and Installation for Windows

- Download GNU Emacs + ESS as a modified version for [Windows](#).
- Run the installer - accept all presets.
- Check out the [guided tour](#).
- Open Emacs, type CTRL-h t (C-h t) and complete the tutorial.

Note: the next quiz will include material from the Emacs tutorial!

7.2 Download and Installation for MacOS

- Download GNU Emacs + ESS as a modified version for [MacOS](#).
- Run the installer - accept all presets.
- Check out the [guided tour](#).

- Open Emacs, type **CTRL-h t** (**C-h t**) and complete the tutorial.

Note: the next quiz will include material from the Emacs tutorial!

8 Customize Emacs

GNU Emacs is much more than a text editor and an IDE. It's more like an operating system inside your operating system. Among the many things that Emacs is capable of, we only need one for this class: the ability to create and run interactive notebooks.

This will give you the power of [Jupyter notebooks](#) or [Colaboratory](#) on your computer, without language limitations, and you can share notebooks with anyone, who has Emacs (or Markdown, for reading only).

The central package for many day to day tasks is `Org-mode`. Here is a set of [Org-mode tutorials](#) (with videos) covering many interesting applications. Org-mode is especially popular among scientists, and among these, physicists (my original tribe), who developed it.

And here is an excellent video tutorial by someone who is also getting started with Emacs for the first time like you:

- [The Absolute Beginner's Guide to Emacs](#) (System Crafters, 2021) - 1hr11min long - time well invested ([I made some notes](#)).

8.1 Create configuration file

To create interactive computing notebooks in Emacs, we use the [Org-mode](#) and [Babel](#) packages. Both are already installed in your version of Emacs, but you have to tell Babel, which languages you want to work with.

Customization like this is done with a configuration file `.emacs`, which is placed in your home directory (`$HOME`). Where this folder is actually located on your computer depends on your operating system^{[10](#)}.

Download the configuration file [from GitHub](#). Here is the code: you can also copy and paste it into a `.emacs` file.

```
;; bury a directory buffer with 'a'
(put 'dired-find-alternate-file 'disabled nil)

;; require ob-sqlite and ob-sql (for compilation in org src blocks) & tangle
(require 'ob-sqlite)
(require 'ob-sql)
(require 'ob-emacs-lisp)
(require 'ob-R)
(require 'ob-C)
(require 'ob-shell)

;; active Babel languages
(org-babel-do-load-languages
 'org-babel-load-languages
 '((R . t)
   (sql . t)
   (shell . t)
   (emacs-lisp . t)
   (C . t)))

;; Syntax highlight code in your SRC blocks The last variable removes
;; the annoying "Do you want to execute" your code when you type:
;; C-c C-c
(setq org-confirm-babel-evaluate nil
      org-src-fontify-natively t
      org-src-tab-acts-natively t)

;; get packages from MELPA package manager
(require 'package)
(add-to-list 'package-archives
             ('("melpa-stable" . "https://stable.melpa.org/packages/"))

;; require org-tempo so that <s <TAB> self-insert works
(require 'org-tempo)

;; remove GUI tool bar and menu bar
(menu-bar-mode -1)
(tool-bar-mode -1)
(toggle-scroll-bar -1)

;; enable recentf mode and bind it to
(recentf-mode 1)
```

```
(global-set-key (kbd "C-x rf") 'recentf-open-files)
(setq-default org-hide-emphasis-markers t)
```

Once you've created the `.emacs` file, you can start Emacs. Everything that follows is optional. I will bring one interesting Emacs package per week to class for you to try if you like. But even just by using Emacs as your editor for assignments, you'll become quite an expert, [almost a "hacker"](#) (Wulff, 2021).

8.2 Create sample notebook

To create a notebook using Org-mode, create an `.org` file. Then type `C-c C-`, and select your chunk from the list. You can also abbreviate this by entering `<s` on any line.

Check out [./babel_c.html](#) for examples with C code blocks¹¹.

8.3 Layout changes

You can completely change anything about the way Emacs looks, feels and behaves. Here are a few suggestions with code snippets based on my own customizations.

If you change your `~/.emacs` file, you need to evaluate the file (`M-x h evaluate-region`) or restart Emacs to see the changes.

Emacs Lisp is a fun language to learn, because through Emacs you can play around with it and see what it does much more easily than with other languages. Here is a [complete tutorial for non-programmers](#). Lisp (and Emacs Lisp) is a functional programming language (like R).

8.3.1 Customize theme and font

To change the theme, enter `M-x custom-themes`. Activate `Save theme settings` if you want the settings to become permanent. This will modify your `.emacs` configuration file.

You can also upload fonts and change fonts. You can do this easiest by opening the `Options` menu at the top of the Emacs screen and selecting `Set default font` from the list.

If you don't have the menu bar, enter `M-x menu-bar-mode` - this will toggle the menu bar, i.e. you can make it appear or disappear with this command. If you don't have a mouse, you can open the menus with `<F10>`. I don't tend to use it at all, since one of the advantages of Emacs is that everything can be done with the keyboard (which is way faster than the mouse).

If you want to get into this for whatever reason, [check this out](#) (Zamboni, 2018).

8.4 Installing additional packages

There are hundreds of useful packages available for instant installation. To see them, enter `M-x package-list-packages`.

The screenshot shows part of the listing, with `available`, `installed` (by me), and `built-in` (by GNU Emacs) files.

Package	Version	Status	Archive	Description
zetteldeft	0.3	available	melpa-s...	Turn deft into a zettelkasten
zettelkasten	0.4.0	available	melpa-s...	Helper functions to organise
zetz-mode	0.1.0	available	melpa-s...	A major mode for the ZetZ pr
zmq	0.10.10	available	melpa-s...	ZMQ bindings in elisp
znc	0.4	available	melpa-s...	ZNC + ERC
zombie-trellys...	0.2.1	available	melpa-s...	A minor mode for interaction
zone-nyan	0.2.2	available	melpa-s...	Zone out with nyan cat
zones	2019.7.13	available	gnu	Zones of text - like multipl
zones	2019.7.13	available	gnu	Zones of text - like multipl
zoom	0.2.3	available	melpa-s...	Fixed and automatic balanced
zoom-window	0.6	available	melpa-s...	Zoom window like tmux
zap-to-char	1.1	available	melpa-s...	A replacement of zap-to-char
zotelo	1.3	available	melpa-s...	Manage Zotero collections fr
zotxt	5.0.5	available	melpa-s...	Tools to integrate emacs wit
zoutline	0.2.0	available	melpa-s...	Simple outline library.
zoxide	1.1.0	available	melpa-s...	Find file by zoxide
ztree	1.0.6	available	gnu	Text mode directory tree
ztree	1.0.6	available	gnu	Text mode directory tree
zygospore	0.0.3	available	melpa-s...	reversible C-x 1 (delete-othe
zzz-to-char	0.1.3	available	melpa-s...	Fancy version of zap-to-char
dracula-theme	1.7.0	installed		Dracula Theme
org-beautify-theme	0.3.1	installed		A sub-theme to make org-mode
org-bullets	0.2.4	installed		Show bullets in org-mode as
pdf-tools	0.90	installed		Support library for PDF docu
yasnippet	0.14.0	installed		Yet another snippet extensio
tablist	1.0	dependency		Extended tabulated-list-mode
allout	2.3	built-in		extensive outline mode for u
allout-widgets	1.0	built-in		Visually highlight allout ou
ansi-color	3.4.2	built-in		translate ANSI escape sequen
antlr-mode	2.2.3	built-in		major mode for ANTLR grammar
artist	1.2.6	built-in		draw ascii graphics with you
auth-source-pass	5.0.0	built-in		Integrate auth-source with p
backtrace	1.0	built-in		generic major mode for Elisp
bs	1.17	built-in		menu for selecting and displ
cc-mode	5.33.1	built-in		user customization variables
cedet	2.0	built-in		Data structure debugger
cfengine	1.4	built-in		mode for editing Cfengine fi
chart	0.2	built-in		Draw charts (bar charts, etc
checkdoc	0.6.2	built-in		check documentation strings
cl-generic	1.0	built-in		CLOS-style generic functions
cl-lib	1.0	built-in		Common Lisp extensions for E
cl-print	1.0	built-in		CL-style generic printing
cwarn	1.3.1	built-in		highlight suspicious C and C

To install a package

- search and find it (forward search with **C-s** or backward search with **C-r**)
- enter **i** to mark the package for installation
- enter **x** to install it.

8.5 Presenting in Emacs

I often present in Emacs, especially when I use interactive notebooks. I use `org-slide-tree-mode` for that ([see documentation](#)). You need to install the package `org-tree-slide` and put the code below into your `/.emacs` file.

U:%%- *Packages*	65% of 331k L2216 (Package Menu +2)
Packages to install: 1 (<code>org-tree-slide-2.8.4</code>). Proceed? (y or n) [Please answer y or n]	

```
;; org-tree-slide: https://github.com/takaxp/org-tree-slide
;; to activate: M-x org-tree-slide-mode or <f9> - stop S-<f9>
(require 'org-tree-slide)
(with-eval-after-load "org-tree-slide"
  (global-set-key (kbd "<f9>") 'org-tree-slide-mode)
  (global-set-key (kbd "S-<f9>") 'org-tree-slide-skip-done-toggle)
  (define-key org-tree-slide-mode-map (kbd "<f8>") 'org-tree-slide-move-previous-tree) ;; move forwards
  (define-key org-tree-slide-mode-map (kbd "S-<f8>") 'org-tree-slide-move-next-tree) ;; move backwards
  )
  (setq org-image-actual-width nil)
  (setq org-tree-slide-skip-outline-level 0)
  (setq org-tree-slide-slide-effect t)
  (org-tree-slide-simple-profile) ;; no headers
```

In the code, `<f9>` is used to switch the mode on or off (`SHIFT + <f9>`), and `<f8>` to move one slide forward or backward (`SHIFT + <f8>`). Slide headers have been removed. If you want slide headers, comment the last line by putting `;` in front of it like this:

```
;; (org-tree-slide-simple-profile) ;; no headers
```

8.6 Definitions and functions

You can use `M-Q` to fill a region (wrap the text and cut it off after 70 characters, a value set in `fill-column`). Sometimes it is useful to unfill a region (put it on one line, for example to copy it into an email). If you put the following definition into your `~/.emacs` file, you can use `M-x unfill-region` to achieve that.

```
;; unfill region
(defun unfill-region (beg end)
  "Unfill the region, joining text paragraphs into a single
   logical line. This is useful, e.g., for use with
   `visual-line-mode'.
(interactive "*r")
(let ((fill-column (point-max)))
  (fill-region beg end)))
```

If you like to bind the function to a key sequence, you can use this code - now `C-M-Q` will invoke the function:

```
;; bind unfill-region to C-M-Q
(define-key global-map "\C-\M-Q" 'unfill-region)
```

8.7 Adding images and links to Org-mode files

My lecture scripts and notebooks often contain images and links. It is easy to add image and links (internal to Emacs or Internet URLs) to an Org-mode file.

Images can be named and given captions. Here is an example with figure 17 below. To show/hide images, use `C-c C-x C-v` (`org-toggle-inline-images`). The `#+ATTR_HTML: :width 400px` line sets the display size of the image (both in Emacs and in the HTML export).

```
#+CAPTION: Google search trends for popular editors
#+NAME: fig:trend
#+ATTR_HTML: :width 400px
[./img/trend.png]
```

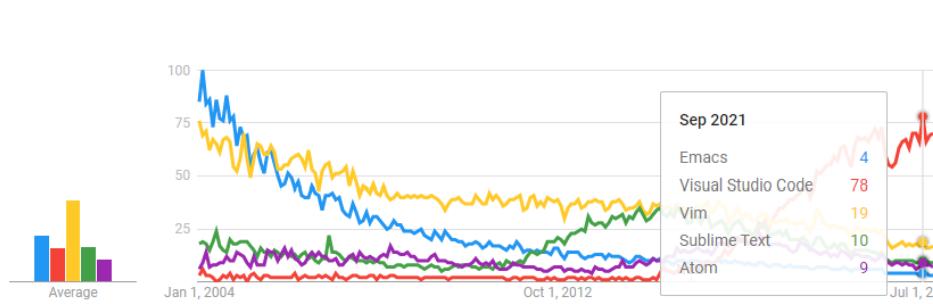
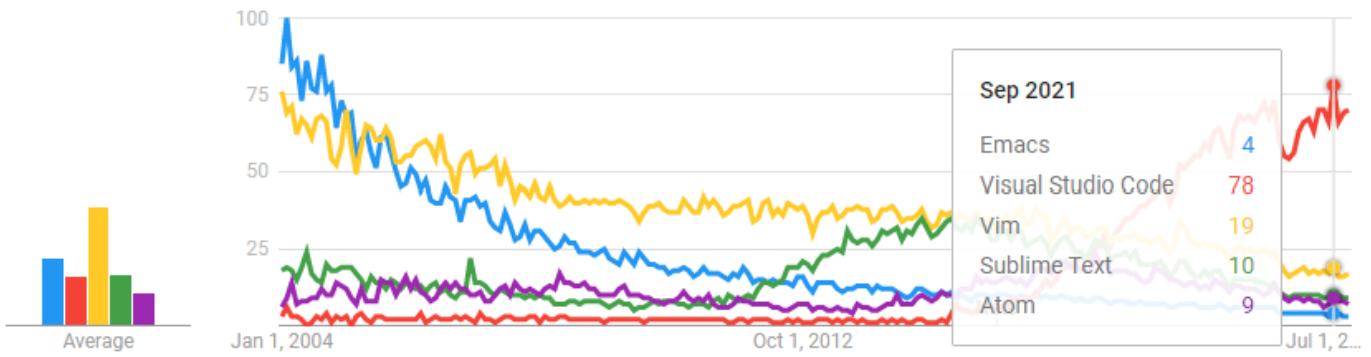


Figure 17: Google search trends for popular editors

And here is the link to the image - when viewing `setup.org` in Emacs, you can open links with `C-c C-o` (`org-open-at-point`).



8.8 Tables

Org-mode has powerful table manipulation capabilities. I don't use Excel, I use active tables in Org-mode for my spreadsheet needs (e.g. computation of grades). There is too much to learn here - I suggest working through this short [tutorial](#). For using tables as spreadsheets, see this short [tutorial](#).

8.9 Export

You can see the different export options for any Emacs buffer with `c-c c-e` (`org-export-dispatch`). This command requires you to pick an option and enter the corresponding code in the mini buffer - see image.

```

403      You can see the different export options for any Emacs buffer with
404      ~C-c C-e~ (~org-export-dispatch~). This command requires you to
405      pick an option and enter the corresponding code in the mini buffer.]
406
407 [● Summary
U\--- README.org<2_installation> 81% of 18k L406 Git:main (Org ARev Fill)
Use SPC, DEL, C-n or C-p to navigate.
[C-b] Body only: Off [C-v] Visible only: Off
[C-s] Export scope: Buffer [C-f] Force publishing: Off
[C-a] Async export: Off

[c] Export to iCalendar
  [f] Current file [a] All agenda files
  [c] Combine all agenda files

[h] Export to HTML
  [H] As HTML buffer [h] As HTML file
  [o] As HTML file and open

[l] Export to LaTeX
  [L] As LaTeX buffer [l] As LaTeX file
  [p] As PDF file [o] As PDF file and open

[m] Export to Markdown
  [M] To temporary buffer [m] To file
  [o] To file and open

[o] Export to ODT
  [o] As ODT file [o] As ODT file and open

[t] Export to Plain Text
  [A] As ASCII buffer [a] As ASCII file
  [L] As Latin1 buffer [l] As Latin1 file
  [U] As UTF-8 buffer [u] As UTF-8 file

[p] Publish
  [f] Current file [p] Current project
  [x] Choose project [a] All projects

[&] Export stack [#] Insert template
[q] Exit

1\**- *Org Export Dispatcher* All of 1.1k L1 (Fundamental)

```

However, if an export is successful depends on the availability of programs in the background. For example, you need some extras to generate a PDF file straight from a LaTeX file. .odt files are Apache [OpenOffice](#) files (XML formatted) that can also be opened in Google Docs¹².

What always works is HTML (.html) export, and Markdown (.md) export. Markdown is the standard format for GitHub text files. However, to get the markdown export option with C-c C-e you need to export once per Emacs session manually by entering M-x org-export-to-markdown.

The HTML export is displayed using your default browser and looks as shown below for this file. You can print it from the browser if you need a paper print version.

Installation

CSC100 Introduction to programming in C/C++

Table of Contents

- [1. What are you going to learn?](#)
- [2. Why we don't just get coding](#)
- [3. Emacs for C programming](#)
- [4. Infrastructure Setup](#)
- [5. Install a C compiler \(Windows + MacOS\)](#)
- [6. Install Emacs](#)
- [7. Customize Emacs](#)
- [8. Summary](#)
- [9. Jargon](#)
- [10. What's next](#)
- [11. References](#)

1 What are you going to learn?

- Why we don't just get coding
- Setting up infrastructure
- Install a C compiler
- Install the Emacs editor and IDE
- Customize the IDE
- Summary of concepts/code
- What's next

2 Why we don't just get coding

There are two reasons - one is philosophical, the other is technical.

"Understanding underlying technologies helps you develop a sense of what can go wrong. Knowing just high-level tools makes it easy to ask the wrong questions. It's worth learning to use a hammer before graduating to a nail gun. Learning underlying systems and tools gives you the power to build new tools, which is important because there will always be a need for tool builders, even if tool users are more common. Learning about computers so that the behavior of programs isn't a mystery enables you to craft better code." (Steinhart, 2019)

The other reason has to do with C itself:

TASK	SOFTWARE
C source code needs to be written and edited	editor
C source code needs to be compiled, linked and debugged	compiler
C object code needs to be run	shell

What works really well in HTML are mathematical formulae. This LaTeX equation for example only renders well in HTML (see image):

$$Q^\pi = E\left[\sum_{\tau=1}^{\infty} \gamma^{\tau-1} r_\tau | s_t = s, a_t = 1\right] \quad (1)$$

What works really well in HTML are mathematical formulae. This LaTeX equation for example only renders well in HTML:

$$Q^\pi = E\left[\sum_{\tau=1}^{\infty} \gamma^{\tau-1} r_\tau | s_t = s, a_t = 1\right] \quad (1)$$

9 Summary

- To program in C, we need a computer, a compiler, and an editor
- You'll have to download the compiler for Windows or MacOS
- You can download and install Emacs (ready for data science)
- Emacs is a highly customizable editor (using Emacs Lisp)
- Org-mode is a literate programming environment

10 Glossary

CONCEPT	EXPLANATION
Source code	Human-readable program
Compiling	Translating source
Linking	Linking compiled program to libraries
Library	Bundle of reusable macros or functions
Object code	Code ready for execution by a machine
Execution	Running object code on a machine
Interpreter	Machine that interprets and executes source code
Script	Source code for an interpreter
Emacs	Extensible text editor (via Emacs Lisp)
Literate Program	Readable code - expands into doc + executable
GNU	"GNU's not UNIX"
GNU/Linux	Free, open source operating system
Richard Stallman	Creator of the GNU project and Emacs
Org-mode	Emacs package for literate programming (and more)

11 References

- Biggs/Donovan (November 9, 2020). Modern IDEs are magic. Why are so many coders still using Vim and Emacs? [Blog]. URL: stackoverflow.org.
- DistroTube (October 4, 2019). Switching to GNU Emacs [video]. [URL:youtu.be/Y8koAgkBEnM](https://youtu.be/Y8koAgkBEnM).
- Galov (August 9, 2021). 111+ Linux Statistics and Facts - Linux Rocks! [blog]. [URL: hostingtribunal.com](https://hostingtribunal.com).
- GCC, the GNU Compiler Collection. [URL: gcc.gnu.org](https://gcc.gnu.org).
- GitHub (Dec 19, 2016). What is GitHub? [video]. [URL:youtu.be/w3jLJU7DT5E](https://youtu.be/w3jLJU7DT5E).
- GNU Emacs, an extensible, customizable, free/libre text editor. [URL: gnu.org/software/emacs](https://gnu.org/software/emacs).
- Kenlon (March 10, 2020). Getting started with Emacs [blog]. [URL: opensource.com](https://opensource.com).
- MinGW-w64 - Minimal GCC for Windows. A complete runtime environment for GCC & LLVM for 32 and 64 bit Windows. [URL: mingw-w64.org](https://mingw-w64.org).
- Steinhart (2019). The Secret Life of Programs. NoStarch Press. [URL: nostarch.com](https://nostarch.com).
- System Crafters (March 8, 2021). The Absolute Beginner's Guide to Emacs [video]. [URL:youtu.be/48JlgIBpwI](https://youtu.be/48JlgIBpwI).
- System Crafters (November 28, 2021). M-x Forever: Why Emacs will outlast text editor trends. Emacs conference 2021 [video]. [URL:youtu.be/9ahR5K_wkNQ](https://youtu.be/9ahR5K_wkNQ).
- Timachi (Dec 7, 2019). Why I switched from VScode to Emacs | Why I switched from VScode to Emacs [blog]. [URL: hadi.timachi.com](https://hadi.timachi.com).
- Wulff (Jul 27, 2021). 8 Reasons Why Emacs is the Best Text Editor for Programming [blog]. [URL: hackernoon.com](https://hackernoon.com).
- xkcd (n.d.). A webcomic of romance, sarcasm, math, and language [website]. [URL: xkcd.com](https://xkcd.com).
- Zamboni (March 21, 2018). Beautifying Org Mode in Emacs [blog]. [URL: zzamboni.org](https://zzamboni.org).

Footnotes:

¹ Code::Blocks, CodeLite, Netbeans, Microsoft Visual Studio (VS), are all free IDEs for C/C++, with VS being the most popular one right now.

² Last term when I began to use GitHub, some students were complaining about the extra platform - however, it should be clear by now, that Schoology cannot substitute for GitHub. The latter is a software engineering platform, well suited for computer science teaching and code development, the former is a learning management system focused on distributing material, creating tests and computing grades.

³ Only Markdown (.md) files are rendering in the mobile app. Org-mode files (.org) do not. Since you have Emacs, feel free to add a Markdown version of an Org file if you want one because you use the mobile version a lot.

⁴ Here is an example from my /.emacs file: I defined the function iwb to indent a whole buffer according to the buffer's mode - something that can also be done with the key sequence C-x h C-M-\ (mark-whole-buffer + indent-region).

```
;; re-indenting of whole buffer according to mode
(defun iwb ()
  "indent whole buffer"
  (interactive)
  (delete-trailing-whitespace)
  (indent-region (point-min) (point-max) nil)
  (untabify (point-min) (point-max))
)
```

⁵ Emacs Lisp is a Lisp dialect. Lisp was one of the first languages used for Artificial Intelligence research (cp. [SHRDLU](#), an early natural language processing system).

⁶ Written in 1976 by Richard Stallman, who then tinkered with it for ten years before releasing it. Emacs is also one of the two contenders, along with vi, of the famous editor wars of the UNIX culture. UNIX is the "mother" of all operating systems, the systems that make computer run and do stuff.

⁷ (1) Emacs has versions (at the time the screenshot was taken: 27.1); (2) Emacs has named "buffers", and you can open several simultaneously [the names correspond to Emacs plugins or packages for organization (org), Git (magit) and file management (dired); (3) Emacs has layout themes with title and borders. (4) Each buffer is accompanied by a status line at the bottom [modeline].

⁸ Neal Stephenson is a sci-fi author who also coined the term "cyberspace", and developed a spacecraft and launch system for Bezos' Blue Origin.

⁹ "Meta skills" are transferable skills that you learn, or improve, while you learn something specific (like Emacs or Org-mode). While the special skills might become obsolete or less important to you over time (because of a change of job, interest, or the market), meta skills stay important and fresh forever, because you can use them for every new special skill learning project.

¹⁰ On my Windows machine, \$HOME is C:\Users\birkenkrahe\. On my Linux box, it is /home/marcus/.

¹¹ This link also shows you how to link notebooks. You can set a link anywhere (inside Emacs or Internet) with `c-c c-l`. If the target is another file, that file needs to be found (the path must be correct), and an anchor with the link name must be put into the file, in this case, the link is `~/babel.org`, and the anchor is `<<babel.org>>`

¹² However, on my Windows 10 PC, WORD refuses to open OpenOffice files (perhaps because the package is only available as a 32-bit version from [Apache OpenOffice](#)?).

Author: Marcus Birkenkrahe

Created: 2022-05-19 Thu 12:26