

COURSE OVERVIEW

CSC 100 - Introduction to programming - Spring 2023

Marcus Birkenkrahe

January 16, 2023



Figure 1: Dashboard of a sportscar of the 1970s

Who is your lecturer?

- Not cats OR dogs, but cats AND dogs



Figure 2: Photo of Poppy as a puppy

- Physicist by training (making sense of this)
- Teacher by profession (making sense of this)
- Data scientist by choice (making sense of this)
- Lyon College since 2021 (from Berlin, Germany)

My first programming languages



Figure 3: Texas Instruments TI-99/4 16-bit microcomputer, 1979

- My first programming languages:
 1. BASIC - Beginner's All-purpose Symbolic Instruction Code (1979)
 2. FORTTRAN - FORMula TRANslator (1985)
 3. C++ (1989)
- How many programming languages are there today?

The importance of infrastructure

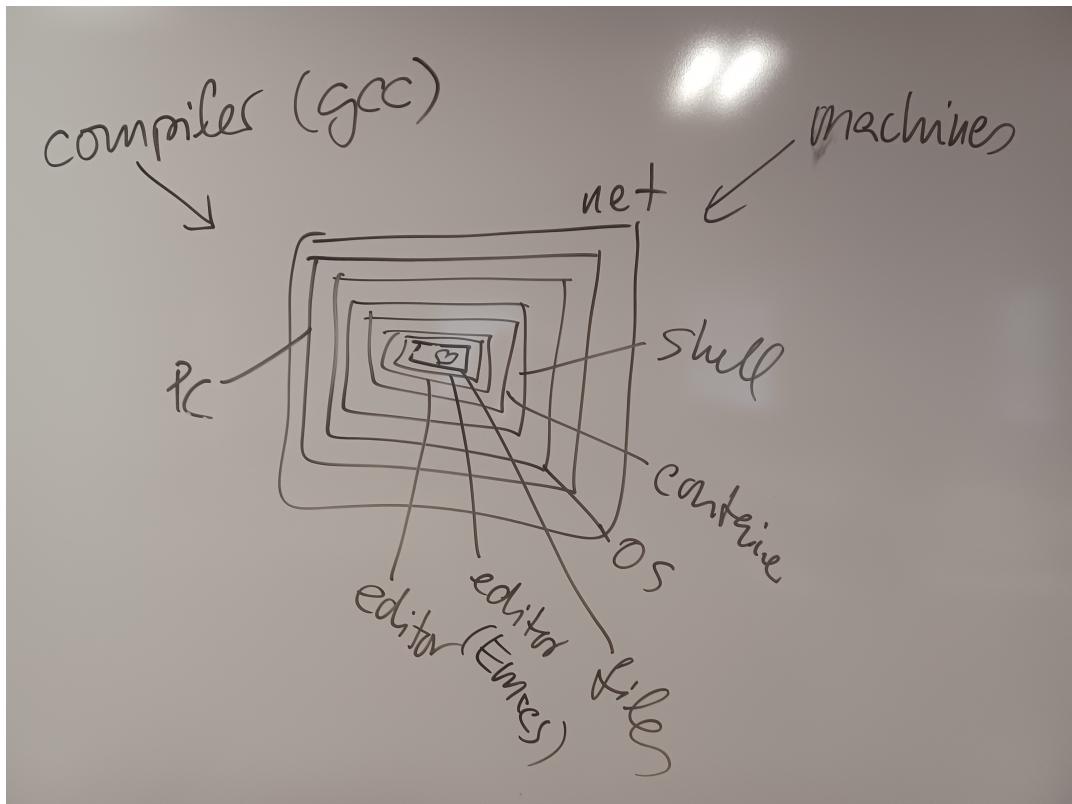


Figure 4: Layers between you and a document on your computer

The image shows eight layers that your computing infrastructure has to manage (there are more in fact) - most of the software enabling you to do this (without you noticing) is written in C.

What do you expect from this course?

- I would like to make a website
- I want to learn as many languages as I can
- I may need it (formally) for my Math major
- I want to make video games

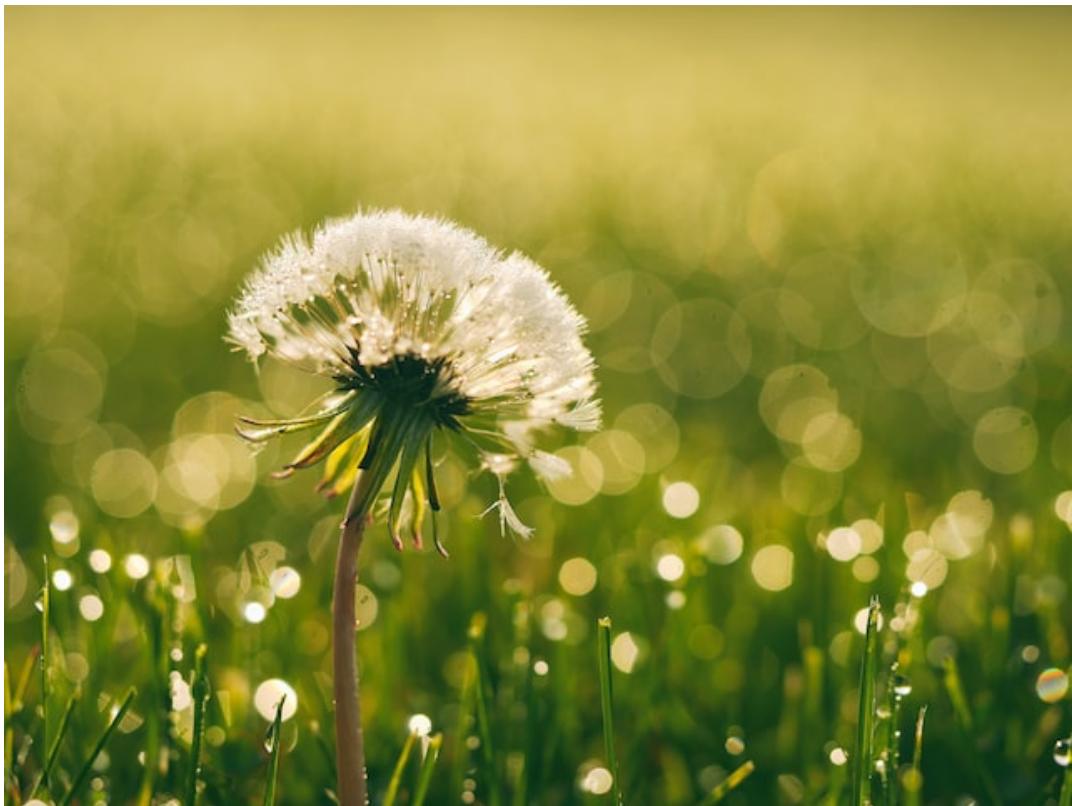


Figure 5: Dandelion in the grass, by Aaron Burden, @Unsplash

- I want to get better at computing because of math/eng
- I want to become a software developer
- I want to see if computing may be in my future

What careers do our students have?



What careers did our students have?

The slide features a photograph of a person from the waist down, wearing a grey blazer over a white shirt and dark trousers, carrying a brown leather briefcase and a folder under their left arm. The background is a blurred outdoor scene.

- Full stack developer @ OneBrief (military planning software)
- Cybersecurity @Sierra Nevada (aircraft engineering)
- Software engineer @Riot Games (League of Legends game company)
- Database specialist @DHS (Department of Homeland Sec)
- IT director @Bad Boy Mowers (High performance lawn mowers)

Figure 6: Careers of recent computer and data science graduates from Lyon College

What will you do in this course?

- We have a **textbook** - King, C Programming - A modern approach (2008)
- We have an **operating system** - Linux (maybe) or Windows/MacOS
- We have (more than) an **editor** - Emacs + Org-mode
- We have a **compiler** - GCC (also on a Mac)

WK	DATE	ASSIGNMENT	TEXTBOOK CHAPTER	TEST
1	Jan 11,13	Emacs Tutorial		
2	Jan 18,20	Program 1	1 Introducing C	1
3	Jan 23,25,27	Program 2	2 C Fundamentals	2
4	Jan 30, Feb 1,3	Program 3	3 Input/Output	3
5	Feb 6,8,10	Sprint Rev 1	4 Expressions	
6	Feb 13,15,17	Program 4	5 Selection	4
7	Feb 20,22,24	Program 5	6 Loops	5
8	Mar 1,3	Program 6	7 Basic types	6
9	Mar 6,8,10	Sprint Rev 2	8 Arrays	
10	Mar 13,15,17	Program 7	9 Functions	7
11	Mar 27,19,31	Program 8	11 Pointers	8
12	Apr 3,5	Program 9	12 Pointers & Arrays	9
13	Apr 10,12,14	Sprint Rev 3	13 Strings	10
14	Apr 17,19,21	Program 10	14 The Preprocessor	
15	Apr 24,26,28	Sprint Rev 4	C vs. C++	
16	May 1, 3			

Figure 7: Syllabus, Canvas (lyon.instructure.com) or GitHub (github.com/birkenkrahe/ds2)

GCC installation on MacOS

The following two answers came from ChatGPT, a conversational machine learning model created by OpenAI.

How will you be evaluated?

- All course requirements have deadlines
- Late submissions will be penalized (loss of points)
- Final exam will be sourced by term test questions
- Programming assignments are home assignments
- In-class practices are "literate programming" exercises

Which tools are you going to use?

- Canvas (learning management system)
- GitHub repository (all course materials except tests)
- GNU Emacs + ESS + R (literate programming environment)
- Command line interface / interpreter - (shell / repl.it.com)

The (new) Canvas LMS

- Announcements
- Assignments (Programming assignments and in-class practice)
- Zoom links (participate/cloud recording)
- Course links (GitHub, Whiteboard)
- Gradebook

A screenshot of a Stack Overflow post titled "How can I install GCC on a Mac". The post includes a profile picture of a man, a snippet of text explaining how to install GCC using Homebrew, a code block with the command "brew install gcc", and a note about XCode command line tools precedence.

How can I install GCC on a Mac

You can install GCC on a Mac by using the package manager Homebrew. First, you will need to install Homebrew by following the instructions on their website: <https://brew.sh/>. Once Homebrew is installed, you can install GCC by running the following command in your terminal:

```
brew install gcc
```

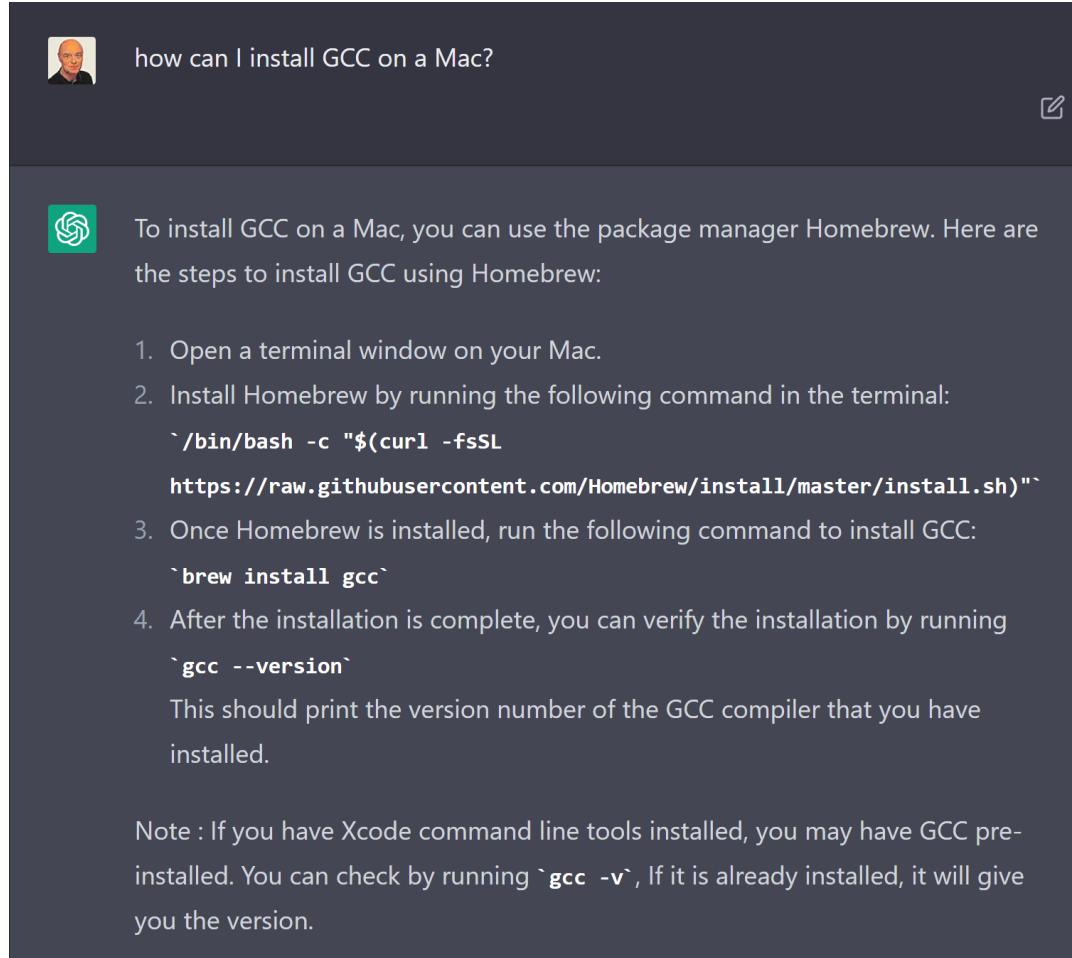
This will install the latest version of GCC on your Mac. To check if GCC is properly installed and to see the version number, you can run the following command:

```
gcc --version
```

Please note that if you have the native XCode command line tools package installed, the version of GCC installed with this package takes precedence over the one installed with Homebrew package manager.

You can also use pre-built binary package from other providers like MacPorts, or use pre-built binary installer for GCC package.

Figure 8: First answer on how to install GCC on MacOS



how can I install GCC on a Mac?

To install GCC on a Mac, you can use the package manager Homebrew. Here are the steps to install GCC using Homebrew:

1. Open a terminal window on your Mac.
2. Install Homebrew by running the following command in the terminal:
`/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"`
3. Once Homebrew is installed, run the following command to install GCC:
`brew install gcc`
4. After the installation is complete, you can verify the installation by running
`gcc --version`
This should print the version number of the GCC compiler that you have installed.

Note : If you have Xcode command line tools installed, you may have GCC pre-installed. You can check by running `gcc -v`, If it is already installed, it will give you the version.

Figure 9: Second answer on how to install GCC on MacOS

REQUIREMENT	UNITS	PPU	TOTAL	% of TOTAL
Final exam	1	100	100	25.
Programming assignments	10	10	100	25.
Class practice	10	10	100	25.
Multiple-choice tests	10	10	100	25.
Project sprint reviews	0	0	0	0
TOTAL			400	100.

Figure 10: Source: syllabus, Canvas (lyon.instructure.com) or GitHub (github.com/birkenkrahe/ml)

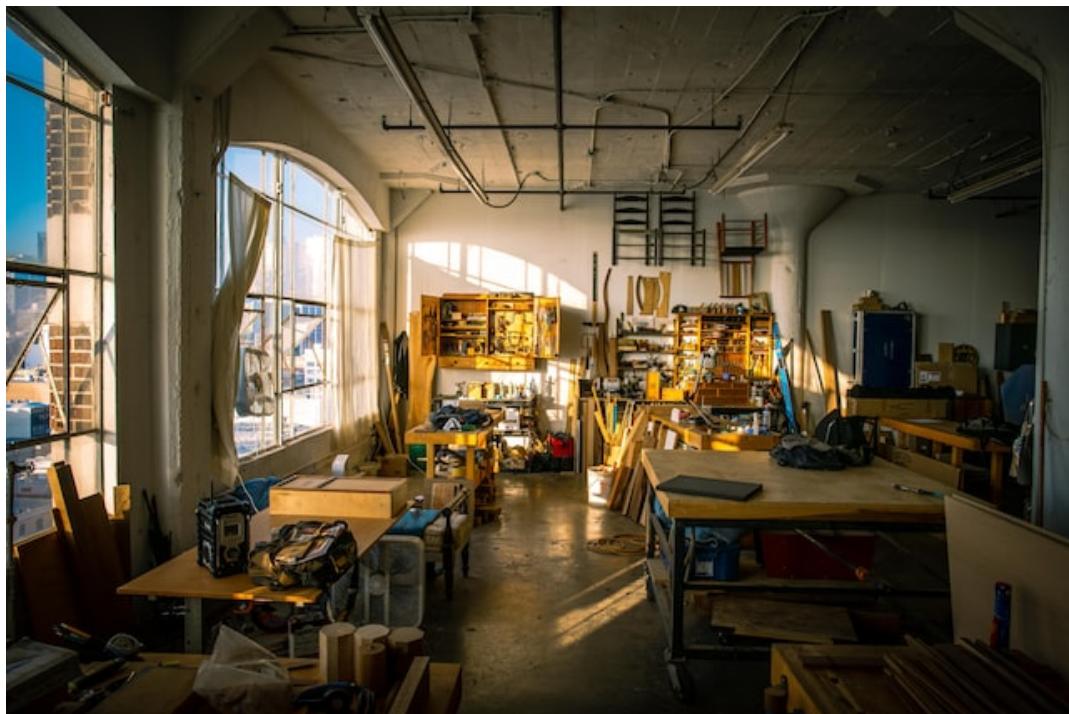


Figure 11: Unsplash, workshop

Lyon

CSC 100 > Syllabus

Immersive Reader

Course Syllabus

1. General Course Information

- Meeting Times: Monday/Wednesday/Friday, 15:00-15:50 hrs
- Meeting place: Lyon Building Room 104 (computer lab)
- Professor: Marcus Birkenrahe
- Office: Derby Science Building 210
- Phone: (870) 307-7254 (office) / (501) 422-4725 (private)
- Office hours: Mon/Wed/Fri 16:15-16:45, Tue/Thu 16:00-16:30
- Textbook: King (2008). C Programming - A Modern Approach. New York: Norton. [Online](#): [knking.com](#)

2. Standard and course policies

- Standard Lyon College Policies** are incorporated into this syllabus and can be found at: [lyon.edu/standard-course-policies](#).
- The **Assignments and Honor Code** and the **Attendance Policy** are incorporated into this syllabus also and can be found at: [tinyurl.com/LyonPolicy](#).

Figure 12: Canvas

The (popular) GitHub platform

- GitHub is the most popular software development platform.
- It's now owned by Microsoft (which is becoming a problem)
- Here for you: all course materials, and the course FAQ

The (feared) Emacs editor

IN PROGRESS Practice: first "literate" C program!

(You can find a PDF of this exercise on GitHub)

Let's set Emacs up, write and run a first "literate" C program!

- Open the command line terminal with CMD in the search field
- At the prompt, type `gcc --version`
- Open <https://github.com/birkenkrahe/org/> in a browser

The screenshot shows the GitHub dashboard of user [birkenkrahe](#). At the top, there is a search bar and navigation links for Pulls, Issues, Codespaces, Marketplace, and Explore. Below the header, there is a pinned profile picture of Marcus Birkenkrahe. The main content area is divided into several sections:

- Pinned:** A grid of five pinned repositories:
 - [org](#) (Public): FAQ, diary, organizational stuff for all my courses @Lyon. Includes a link to Emacs Lisp.
 - [ds2](#) (Public): Advanced introduction to data science (DSC 205) at Lyon College, Spring 2023.
 - [ml](#) (Public): Machine learning course in R at Lyon College, Batesville AR, spring 2023.
 - [tm](#) (Public): Digital Humanities - text mining with R, Lyon College, CSC 105, spring 2023.
 - [cc](#) (Public): Introduction to programming in C and C++ at Lyon College, 2023. Includes a link to grades.
- Customize your pins:** A section for pinning additional repositories.
- Contributions:** A chart showing contributions over the last year, with a legend for 'Less' (light green), 'More' (dark green), and 'Learn how we count contributions'.
- Activity overview:** A summary of recent activity, including contributions to repositories like [birkenkrahe/admin](#), [birkenkrahe/org](#), and [birkenkrahe/os420](#).
- Sponsoring:** A section showing sponsored users.
- Profile details:** Information about the user, including 26 followers, 25 following, and contact information (Lyon College, Batesville, AR, email, website, Twitter handle).
- Edit profile:** A button to edit the user's profile.

Figure 13: My GitHub dashboard - close to 4,000 commits in 2022 (100,000 lines of code?)

What is literate programming?

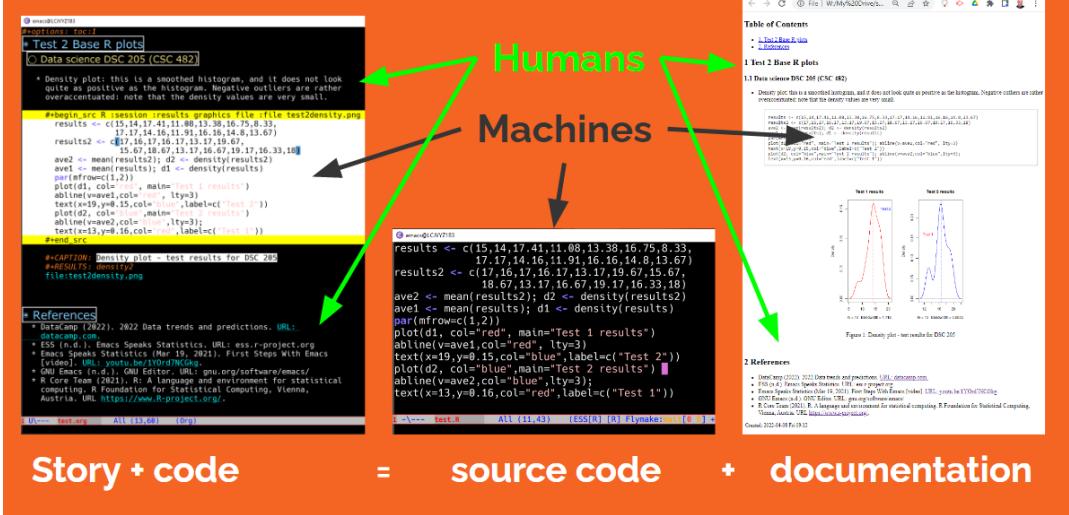
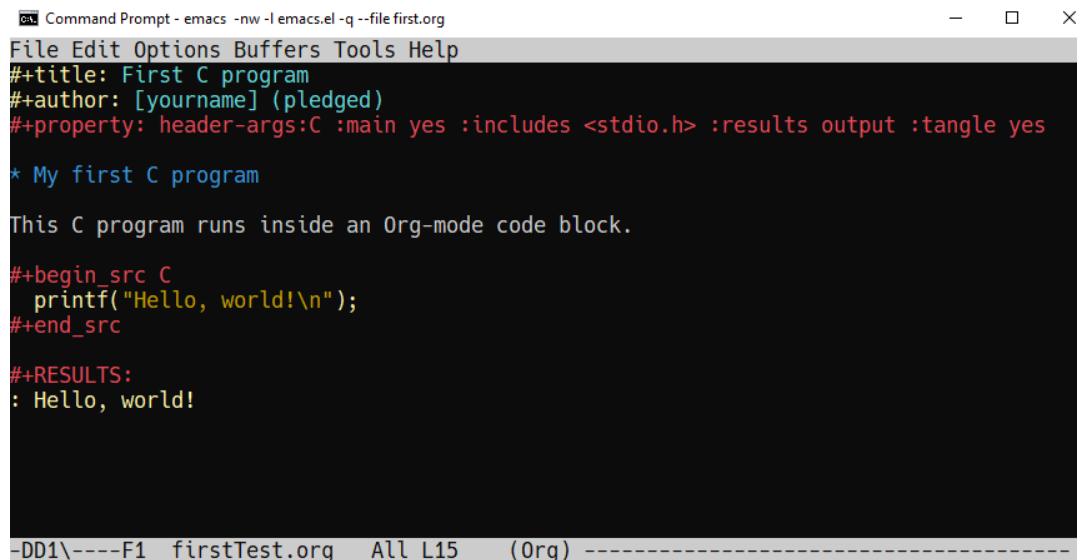


Figure 14: My GitHub dashboard



Figure 15: Books aren't the only way to be "literate" in programming!

4. Navigate to the repository `emacs` in GitHub
5. Click on the file name `.emacs` to open it
6. Open the `Raw` version of this file (there's a button)
7. Right click to `Save as` and save file as `emacs.txt` in `/Downloads`
8. Open a terminal by entering `CMD` in the Windows search bar
9. Pin the terminal to your taskbar
10. At the prompt, enter `DIR emacs.txt` - you should see the file
11. At the prompt, enter `emacs --version`
12. Enter `emacs -nw -l emacs.txt`
13. Inside Emacs, click `<F10>` and `ENTER` to open a new file
14. At the prompt at the bottom of the screen, enter `first.org`
15. Enter the following text (replace `yourname` with your own name):



```
Command Prompt - emacs -nw -l emacs.el -q --file first.org
File Edit Options Buffers Tools Help
#+title: First C program
#+author: [yourname] (pledged)
#+property: header-args:C :main yes :includes <stdio.h> :results output :tangle yes
* My first C program

This C program runs inside an Org-mode code block.

#+begin_src C
  printf("Hello, world!\n");
#+end_src

#+RESULTS:
: Hello, world!
```

-DD1\----F1 firstTest.org All L15 (Org) -----

16. Run the program by putting the cursor anywhere on the code block and typing `CTRL-c CTRL-c` (or `C-c C-c`)

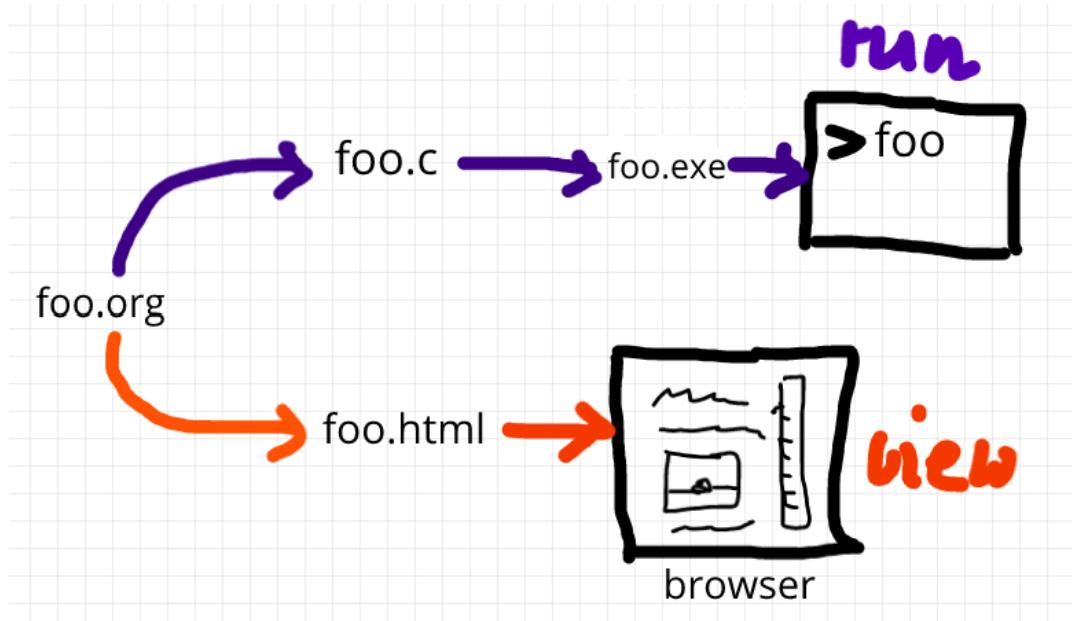


Figure 16: What happens when you tangle or weave a literate program

17. Tangle the code with `C-c C-v t` (or `M-x org-babel-tangle`)
18. Open a shell (terminal program) with `ALT-x eshell`
19. At the prompt, enter `ls -l first*` - you should see `first.C`
20. Enter `gcc first.C -o hello`
21. Enter `hello` to run the program.
22. Upload `first.org` as your first in-class assignment to Canvas!

When is the first assignment due?

- The first in-class assignment (`first.org`, see Canvas) is due on Friday, January 20. For late submissions, you lose 1 point per day (out of 10 possible points)
- The first home assignment (`Emacs tutorial`, see Canvas) is due on Monday, January 23. For late submissions, you lose 1 point per day (out of 10 possible points)



Figure 17: Four monks by Claudio Rinaldi (1852-1909)

- We'll write the first weekly multiple-choice test on January 20, covering all of the material that was covered until then.

What are you looking forward to?





Figure 18: Introduction to C programming



Figure 19: Introduction to C programming

Next

Glossary

TERM	MEANING
Meta data	Data about data (e.g. layout instructions)
Infrastructure	Computing roadworks: hardware and software
Editor	Program to write programs in
Compiler	Program to turn source into machine code
<code>gcc</code>	The GNU C compiler
Source code	Code for humans to read and edit (<code>.c</code>)
Machine code	Code for machines to execute (<code>.exe</code>)
Git	Software version control system (2005)
Emacs	Extensible editor written in Lisp (1985)
FOSS	Free and Open Source Software
Linux	FOSS operating system (1991)
Windows, MacOS	Commercial OS (Microsoft, Apple)
Android	Linux for smartphones (Google)
iOS	MacOS for Apple smartphones
Command line	Terminal, shell program to talk to the OS
Prompt	Location on your computer, e.g. <code>C:\User\</code>
Raw file	No control characters for syntax highlighting
Syntax highlighting	Making programming language visible
<code>DIR</code>	Windows command to list files
<code>cd</code>	Command to change directory
Literate pgm	Doc + code + output for humans and machines
Org-mode	Plugin for Emacs to edit Org files (<code>.org</code>)
Dunning-Kruger effect	Illustrating ignorance of your own ignorance

References

- King K N (2008). C Programming - A Modern Approach. Norton.