

COURSE OVERVIEW

CSC 100 - Introduction to programming - Spring 2024

Marcus Birkenkrahe

January 14, 2024

Contents

1	What was my first programming language?	2
2	Which careers do our graduates have?	5
3	What will you do in this course?	6
4	GCC installation on MacOS (ChatGPT)	6
5	GCC installation on Windows	10
6	How will you be evaluated?	11
7	Which tools are you going to use?	12
8	Learning Management System: Canvas	12
9	Software development platform: GitHub	16
10	The (feared) Emacs editor & Literate Programming	16
11	What are we going to do besides programming?	17
12	Practice: first "literate" C program! (30 min)	17
13	What did you just learn?	21
14	When is the first assignment due?	22
15	Glossary	23

1 What was my first programming language?

- My first programming languages:
 - BASIC - Beginner's All-purpose Symbolic Instruction Code (1979)
 - FORTTRAN - FORMula TRANslator (1985)
 - C++ (1989)
- My last programming languages:
 - R (2019)
 - Python (2023)
 - Lisp (2024)
- Languages in between:
 - Shell (OS)
 - SQL/SQLite (databases)
 - Snap! (visual)
 - Perl (text)
- Languages that I still want to learn:
 - Assembly (machine)
 - Haskell (functional)
 - Verilog (parallel)
 - Prolog (logical)



Figure 1: Poster for this course created by DALL-E 3



Figure 2: Texas Instruments TI-99/4 16-bit microcomputer, 1979

2 Which careers do our graduates have?



Where our CS and DS graduates went to work in the last 2 years:

- Full stack developer @OneBrief (military planning software)
- Computer Aided Design at Bad Boy Mowers (comp sci/math/engineering)
- IT project specialist at Bad Boy Mowers (data science/math/comp sci)
- Data analyst at Future Fuel Chemical (data science/comp sci)
- Database specialist at the Department of Homeland Security
- Software engineer at Riot Games (League of Legends/game company)
- Cybersecurity engineer at Sierra Nevada (aircraft engineering)
- Head of data analysis at Enterprise Inc. (rental cars)
- Programmer at Outlaw Snax Chips (snack foods)
- Manager Enterprise Applications at Lyon College IT Services

3 What will you do in this course?

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

4 GCC installation on MacOS (ChatGPT)

How can I install the GNU gcc compiler on MacOS?

To install the GNU `gcc` compiler on MacOS, you can use the `Homebrew` package manager. Follow these steps:

1. Open a Terminal window.
2. Install Homebrew by entering the following command and following the on-screen instructions:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/i..."
```

1. Once Homebrew is installed, you can install `gcc` by entering the following command:

```
brew install gcc
```

This will install the latest version of `gcc` available through Homebrew.

1. After the installation is complete, you can verify that `gcc` is installed by running the following command:

```
gcc --version
```

This command should display the version of `gcc` that you have installed.

That's it! You should now have the GNU `gcc` compiler installed on your MacOS system.

WK	ASSIGNMENT	TEXTBOOK CHAPTER	TEST
1	Emacs Tutorial	1 Introducing C	1
2	Program 1	2 C Fundamentals	2
3	Program 2	3 Input/Output	3
4	Program 3	4 Expressions	4
5	Program 4	5 Selection	5
6	Program 5	6 Loops	6
7	Program 6	7 Basic types	7
8	Program 7	8 Arrays	8
9	Program 8	9 Functions	9
10	Program 9	11 Pointers	10
11	Program 10	12 Pointers & Arrays	11
12	Program 11	13 Strings	12
13	Program 12	14 The Preprocessor	13
14	Program 13	15 Structs, Unions & Enum	14
15	Program 14	16 Advanced Pointers	15
16	Program 15	From C to C++	Final

Figure 3: Syllabus, Canvas (lyon.instructure.com) or GitHub (github.com/birkenkrahe/cpp)



Figure 4: GCC installation on MacOS (DALL-E3)



Figure 5: GCC installation on Windows OS (DALL-E 3)

5 GCC installation on Windows

Step-by-step instructions for installing MinGW via MSYS2:

Step 1: Download MSYS2 installer

- Go to the MSYS2 website: <https://www.msys2.org/>
- Download the installer

Step 2: Run the installer

- Double-click on the downloaded installer to run it.
- Choose the installation directory (the default is usually fine).
- Click "Next" and then "Install" to start the installation.
- The installer will download and install the necessary files. This may take some time, so be patient.

Step 3: Update MSYS2

- After the installation is complete, you'll see an MSYS2 terminal. Type the following command after the '\$' prompt, and press Enter:

```
pacman -Syy
```

- This will update the MSYS2 installation with the latest packages. Follow the prompts and press Enter when asked.
- For updates (using the same command), add the MSYS2 terminal to your taskbar and run the update regularly.

Step 4: Install MinGW

- Once the update is finished, you can install MinGW by typing the following command and pressing Enter:

```
pacman -S gcc
```

- Follow the prompts and press Enter when asked.
- This step can take a few minutes, be patient.

Step 5: Test the installation

- To test if MinGW is installed correctly, open a new MSYS2 terminal.
- Type the following command and press Enter:

```
gcc --version
```

- You should see the version of GCC (the GNU Compiler Collection) installed with MinGW.
- If you see the version number, congratulations! MinGW is installed and ready to use.

Step 6: Put MSYS2 in the PATH of your PC

- Type PATH in your Windows search field and open "Edit the system environment variables".
- Click on the "Environment Variables..." button.
- In the "System variables" menu, find the line for "Path" with your cursor, and click "Edit".
- In the menu, click "New" and enter C:\msys64\mingw64\bin.
- Click "OK" for all three menus.
- You can now execute `gcc` from anywhere on your computer (e.g. from inside Emacs, or from the Windows CMD line).

That's it! You have successfully installed MinGW via MSYS2. You can now use MinGW to compile and run C/C++ programs on your system.

6 How will you be evaluated?

- All course requirements have firm deadlines. However:
- If you have to miss a deadline, come talk to me first.
- Unexcused late submissions will be penalized (50%).
- Final exam is optional and will be sourced by term test questions.
- Programming assignments are home assignments.
- In-class practices are "literate programming" exercises.
- All assignments require submission of Emacs Org-mode files.

WHEN	DESCRIPTION	IMPACT
Weekly	Programming assignments	25%
Weekly	Multiple choice tests	25%
Weekly	Participation (class)	25%
TBD	Final exam (optional)	25%

Figure 6: Source: syllabus, Canvas (lyon.instructure.com) or GitHub (github.com/birkenkrahe/ml)

7 Which tools are you going to use?

- Canvas (learning management system) - get the mobile app.
- GitHub repository (all course materials except tests/announcements).
- GNU Emacs (literate programming environment) - see also: FAQ.
- Command line interface (shell / REPL) see also: replit.com.
- Google Drive for whiteboard screenshots.

8 Learning Management System: Canvas

- Announcements (you'll get them also via Email)
- Assignments (Programming assignments and in-class practice)
- Zoom links (cloud recording - will be deleted after a few weeks)
- Course links (GitHub, Whiteboard, Emacs/GCC download)
- Gradebook (up to date grades) & Attendance record



Figure 7: How the AI imagines our toolshop (DALL-E 3)

The screenshot shows the Canvas student view for the course CSC 100 01. The left sidebar has a dark blue background with white icons and text. The main content area has a light gray background with a header bar.

Header:

- Course icon: A yellow shield with a building and a tree.
- Course name: CSC 100 01 > Syllabus
- Immersive Reader button

Header Bar:

2023-2024 - Spring Term - F...

Syndication:

Course Syllabus Jump to Today

Left Sidebar:

- Account
- Dashboard
- Courses
- Calendar
- Inbox
- History
- Help (with a green notification badge)

Main Content Area:

Course Syllabus

1. General Course Information

- Meeting Times: Monday/Wednesday/Friday, 9:00-9:50 AM
- Meeting place: Lyon Building Room 104 (computer lab)
- Professor: Marcus Birkenkrahe
- Office: Derby Science Building 210
- Phone: (870) 307-7254 (office) / (501) 422-4725 (private)
- Office hours: MonWedFri 10-10:50 am & 3-3.50 pm, Tue-Thu 4.15-4.45 pm & by appointment
- Textbook: King (2008). C Programming - A Modern Approach. New York: Norton. [Online: knking.com](#) and see also [this GitHub repo](#).
- See also other books used for this course:
 - Malik DS, C++ Programming: Program Design including Data Structures (8e), [Cengage 2018](#) ([Amazon](#)): New from \$83.87 - Dec 2023). This is also the textbook for [CSC 240 Data Structures with C++](#) (Fall 2024).
 - Oualine S, Practical C++ (2e), [O'Reilly 2002](#),
 - Stroustrup B, Programming - Principles and Practice Using C++, Addison Wesley 2014. See also [Author's website](#),
 - Haephrasti M and R, Learning C++, [Manning 2024](#).

Figure 8: Canvas (Student view - Syllabus page)

The screenshot shows my GitHub profile page with a dark theme. At the top, there's a navigation bar with links for Overview, Repositories (38), Projects, Packages, Stars (43), and Sponsoring. A search bar and various icons are also present.

Pinned:

- FAQ, diary, organizational stuff for all my courses @Lyon** (org, Public)
Jupyter Notebook, 2 stars, 2 forks
- Introduction to programming in C++ (Spring 2024)** (cpp, Public)
C++
- Introduction to advanced data science with R, Python and SQL, Lyon College, spring 2024** (ds205, Public)
Introduction to advanced data science with R, Python and SQL, Lyon College, spring 2024
- Database Theory and Applications, Lyon College, Spring 2024** (db24, Public)
Database Theory and Applications, Lyon College, Spring 2024
- Operating systems using Raspberry Pi computers and Linux, Lyon College, Batesville AR, spring 2024** (os24, Public)
Operating systems using Raspberry Pi computers and Linux, Lyon College, Batesville AR, spring 2024
- Copilot demo and test** (copilot, Public)
Copilot demo and test, Python, 1 star

Marcus Birkenkrahe
birkenkrahe

Associate professor of computer and data science at Lyon College.
Former professor of business informatics at Berlin School of Economics and Law.

Edit profile

36 followers · 37 following

Lyon College
Batesville, AR
birkenkrahe@lyon.edu
<https://lyon.edu/marcus-birkenkrahe>
[@birkenkrahe](#)
[in/birkenkrahe](#)

Achievements

Contribution settings ▾

2,580 contributions in the last year

Learn how we count contributions

Less More

Activity overview

Contributed to [birkenkrahe/admin](#), [birkenkrahe/py](#), [birkenkrahe/org](#) and 29 other repositories

Code review

Figure 9: My GitHub dashboard - close to 4,000 commits in 2022 (100,000 lines of code?)

9 Software development platform: GitHub

- GitHub is the most popular software development platform but not just for programmers (cp. Braga et al., 2023)
- It's now owned by Microsoft (which is becoming a problem)
- Here for you: all course materials, and the course FAQ
- To get some bonus points, watch "How GitHub works", complete the "Hello world exercise" on the GitHub platform and submit the result.

10 The (feared) Emacs editor & Literate Programming

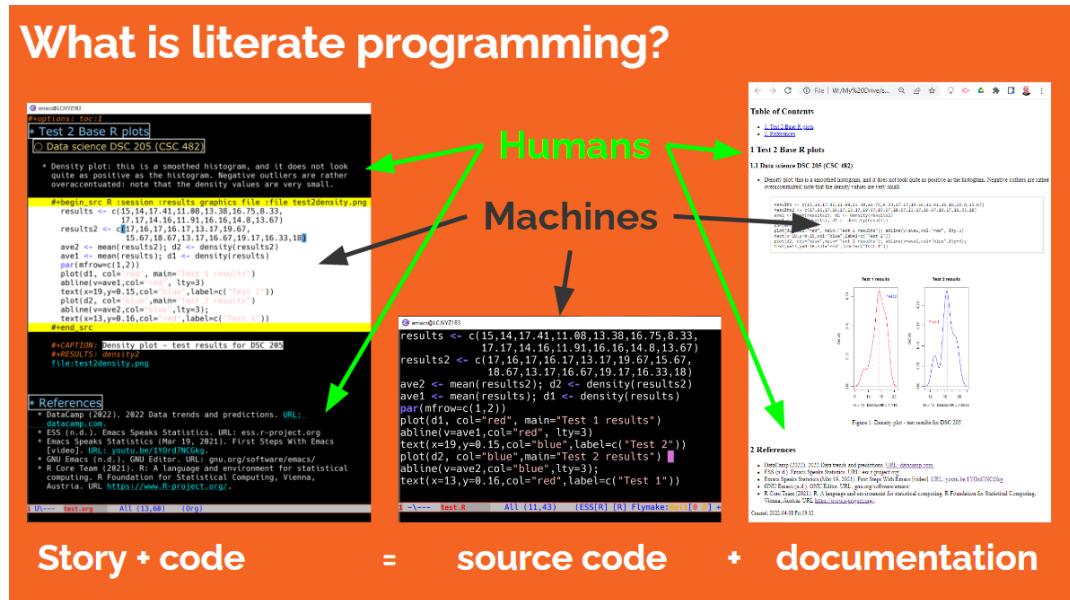


Figure 10: My GitHub dashboard

"Story + code = source code + documentation"

- Humans can't read machine code - they require stories.
- Literate programming delivers human-readable programs.
- Emacs + Org-mode is our platform to do this.

11 What are we going to do besides programming?

- There are many layers between your input and the CPU.
- Most of the software enabling you to do stuff is written in C.
- The primary software to connect to the OS is the shell program.

Does anybody know what a "shebang" in computing is?¹

12 Practice: first "literate" C program! (30 min)

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

Let's set emacs up, write and run a first "literate" c program! it is very important that you enter everything **exactly as shown**. if you get something wrong just go back one step. Contact me if you need me after checking with your neighbor if he or she can help.

1. Open the command line terminal with `cmd` in the search field
2. At the prompt, type `gcc --version`
3. Open <https://github.com/birkenkrahe/org/> in a browser
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 to open it quickly next time!
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, enter `CTRL + x CTRL + f` to enter a new file.

¹A shebang is the command sequence `#!` at the beginning of a shell script. It is followed by the PATH to an executable program, e.g. `bash`.



Figure 11: Layers between you and a document on your computer (DALL-E3)



Figure 12: Books aren't the only way to be "literate" in programming!

14. At the prompt at the bottom of the screen, enter `first.org`
15. Enter the following text (replace `yourname` with your own name):

```
#+title: First C program
#+author: Marcus Birkenkrahe (pledged)

* My first C program

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

#+begin_src C :results output :tangle first.c

#include <stdio.h>

int main() {

    printf("Hello, world!\n");

    return 0;
}

#+end_src
```

16. 'Run' the program by putting the cursor anywhere on the code block

and typing `CTRL-c` `CTRL-c`. You should see the result on the screen.

17. Save the file with `CTRL + x` `CTRL + s`
18. 'Tangle' the code with `CTRL + c` `CTRL + v` `t` (or `M-x org-babel-tangle`):
Emacs reports "Tangled 1 code block from `first.org`".

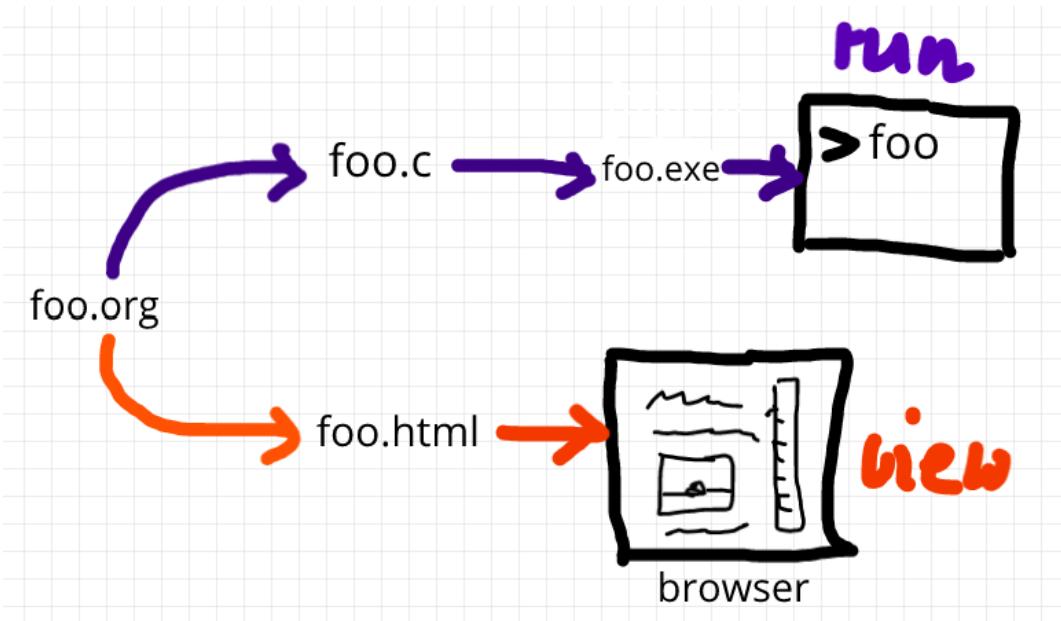


Figure 13: What happens when you tangle or weave a literate program

19. Open a shell inside Emacs by entering `ALT-x eshell`
20. At the `$` prompt, enter `ls -l first*` - you should see `first.c` listed
21. Display `first.c` by entering `cat first.c`
22. Enter `gcc first.c -o hello` to compile the C program into an executable
23. Enter `hello` to run the executable. You should see the output.
24. Exit and close Emacs with `CTRL-x` `CTRL-c`
25. Exit and close the shell by entering `exit` after the prompt

26. Save your file to a directory on your GDrive (you can do this with File Explorer, or directly in Emacs with the following commands (you don't have to worry about spaces etc. because you can auto-complete using the <TAB> key):

```
C-x C-w          ;; write file
w:/My Drive/    ;; target directory
C-x d w:/My Drive/  ;; open target directory
s                ;; sort to see recent files at top
```

You can also do it in the Emacs eshell that you used earlier to compile and run the file on the shell (auto-complete with <TAB>):

```
cp first.org w:/My\ Drive/      # copy file to target directory
cat w:/My\ Drive/first.org     # view copy of file at target location
```

27. Upload `first.org` as your first in-class assignment:

- (a) Open a browser to GDrive and upload the file
- (b) Open the assignment in Canvas at lyon.instructure.com
- (c) Upload the file from GDrive (click on "More")
- (d) When you see it attached, click on `Submit Assignment`.

13 What did you just learn?

You learnt:

1. How to open and close a command line (aka shell) on Windows.
2. How to check the version of the C compiler `gcc`.
3. How to find a file on the GitHub platform and download it.
4. How to list, display, and move files from the command line.
5. How to open and close the GNU Emacs editor.
6. How to create, save, and write an Emacs Org-mode file.
7. How to create, compile, and run a C program.
8. How to tangle a literate program into source code.

9. How to save a file on your GDrive in three ways.
10. How to submit a completed assignment to Canvas.

It would be worth repeating these steps on your own without peeking in your notes to make sure that you understood what you did and that you can do it again - we'll do this hundreds of times in class!

You can watch me complete this exercise in this video (30').

14 When is the first assignment due?



Figure 14: Four monks by Claudio Rinaldi (1852-1909)

- The first in-class assignment (`first.org`, see Canvas) is due no later than Friday, January 26. For late submissions, you lose 50% of your points. You just did this, all you have to do is submit it!

- The second home assignment ([Emacs tutorial](#), see Canvas) is due on Friday, January 26, too. For late submissions, you lose 50% of your points. We'll start this assignment together in the next session.
- The first multiple-choice test is due on Monday, 22 January, including all of the material that was covered until then.

For grading details, see syllabus.

15 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

16 References

- Braga, P. H. P., Hébert, K., Hudgins, E. J., Scott, E. R., Edwards, B. P. M., Sánchez Reyes, L. L., Grainger, M. J., Foroughirad, V.,

Hillemann, F., Binley, A. D., Brookson, C. B., Gaynor, K. M., Shafiei Sabet, S., Güncan, A., Weierbach, H., Gomes, D. G. E., & Crystal-Ornelas, R. (2023). Not just for programmers: How GitHub can accelerate collaborative and reproducible research in ecology and evolution. *Methods in Ecology and Evolution*, 14, 1364–1380. <https://doi.org/10.1111/2041-210X.14108>

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