Teaching computer and data science with literate programming tools:
How I made Emacs + Org-mode mandatory in all my courses

Marcus Birkenkrahe - Lyon College
EmacsConf 2023
teaching programming with emacs
Teaching Data Science with Literate Programming Tools

by Marcus Birkenkrahe

Department of Math and Science, Lyon College, Batesville, AR 72501, USA

Digital 2023, 3(3), 232-250; https://doi.org/10.3390/digital3030015

Received: 25 July 2023 / Revised: 4 September 2023 / Accepted: 6 September 2023 / Published: 8 September 2023

(This article belongs to the Collection Multimedia-Based Digital Learning)
What is data science?
Computing + Math/Stats + Your Stuff
What is data science?

data + code + stats = story
Computer science is a craft

- Take it apart
- Learn tools
- Fix many cars
- Mechanical literacy
- Inferential thinking
The problem

- Computers are seen as bricks with buttons
- **Students cannot find downloaded files**
- Cannot distinguish between browser, PC, network, cloud, client, server ...
- **Convenience, not customization, rules**
- Machines have all the power
The solution: Emacs + Org-mode
GNU Emacs

- Programmable platform
- Self-documenting
- Fully extensible & transparent
- Text editor + operating system
- Keyboard-heavy
- Lisp machine
- Free software
- UNIX / Linux methodology
- Created 1975, launched 1985
- Used by me since 1991
- Hard to learn, easy to use
- Emacs configuration file

- Run C/C++, R, SQL, SQLite, Python and bash
- Update Emacs from the melpa repository
- Create code blocks with skeleton commands
- Autoload ESS
- Disable toolbar and graphical menu bars
-Story + code =

source + documentation

What is literate programming?
What is literate programming?

Story + code = source code + documentation

Humans

Machines
Emacs as a literate programming tool

- Execute code blocks
- Code in 43 programming languages
- Display results
- Interact with shell
- Extract source code
- Render documentation
- Manage tasks & projects
- 5,000 add-on packages
Case study: basic setup

- Introductory to advanced
- Different computing applications
- Taught over 3 terms
- 6-28 participants
- Used GitHub, Canvas, DataCamp

The material of all of my courses is available online at github.com/birkenkrahe
Emacs + Org-mode notebooks used for:

- Code along lectures
- Home assignments
- Practice in class
- Student projects
- GitHub repository
Onboarding: simplified Emacs tutorial
Instruction + Interaction

- Emacs + Org pre-installed
- All lectures code-along
Assignments + Projects

- Submit literate Org-mode files
- Communicate throughout

1. Write a program that prompts the user to enter a telephone number in the form (xxx) xxx-xxxx, and then displays the number in the form xxx.xxx.xxxx.

2. Example input/output of the first program, `phone1.c`:

   Enter phone number [(xxx) xxx-xxxx]: (870) 456-7890
   You entered: 870.456.7890

3. Write another program that asks for the input format in the form \texttt{\textbackslash x\textbackslash x\textbackslash x\textbackslash x\textbackslash x\textbackslash x\textbackslash x\textbackslash x}, and then displays the number in the form \texttt{\textbackslash x\textbackslash x\textbackslash x\textbackslash x\textbackslash x\textbackslash x\textbackslash x\textbackslash x}.

4. Example input/output of the second program, `phone2.c`:

   Enter phone number [\texttt{\textbackslash x\textbackslash x\textbackslash x\textbackslash x\textbackslash x\textbackslash x\textbackslash x\textbackslash x\textbackslash x}]: 870.456.7890
   You entered: (870) 456-7890

5. Submit one Emacs Org-mode file `phone.org` with both programs in it as code blocks that can be tangled as `phone1.c` and `phone2.c`, respectively.

6. The header information of your Org-mode file should look like this:

   ```org
   #+TITLE: Phone number conversion
   #+AUTHOR: [your name]
   #+HOWTO: pledged
   #+PROPERTY: header-args:C :main yes :includes (stdio.h) :results output :tangle yes
   ```

7. Tip: some characters, like \textbackslash, are protected because they are part of the file \texttt{\textbackslash PATH}. If you want to use them, you have to "escape" them with an extra \textbackslash, like the newline character \textbackslash\textbackslash. So to print (or to scan) the character \textbackslash\textbackslash, you use \texttt{\textbackslash\textbackslash}.

8. Here is a short video \(\rightarrow\) (9 min) that explains in detail how to get started with this exercise in Emacs + Org-mode + C.
Before | After introducing literate programming

Test Results CSC 100 Spring 2022

Test 1
Test 2

Density

N = 15  Bandwidth = 0.9945

Test Results CSC 482/DSC 205 Spring 2022

Test 1
Test 2

Density

N = 13  Bandwidth = 0.8042
Overall results positive:

- Emacs hard for all but all succeeded across all courses
- Documentation results uneven but higher quality than ever
- Interactivity praised by all students
- Computing and infrastructure competences much improved
Conclusion & outlook

- Immersion and interaction is everything
- Emacs + Org-mode perform well as central literate programming platform
- Pre-configuring and onboarding are important to train students quickly