CMSI 1024

SOFTWARE CARPENTRY*

http://dondi.lmu.build/fall2026/cmsi1024

Fall 2026—Pereira 121
MW 3:40–4:50pm; 2 semester hours
Office Hours MW 5–6pm, TR 3–5pm,
or by appointment *(don't hesitate to ask!)*

John David N. Dionisio, Ph.D.
dondi@lmu.edu
Doolan 102
(310) 338-5782

Objectives and Outcomes

* The phrase "software carpentry" was coined by Greg Wilson and Brent Gorda in 1998:

https://software-carpentry.org/

This is a two-unit practicum course, meaning you will receive training and gain experience building systems according to modern best practices. This particular practicum focuses on the *tools* that you use for software and not the software development itself. Furthermore, we will predominantly use the *command interaction style*—widely considered to be the most efficient way to get work done…but the hardest one to learn. It will be worth it though! Long after the course concludes, my hope is that you will be able to:

1. **Show proficiency with the command line interaction style**
2. **Manage, navigate, and modify files exclusively within an operating system shell**
3. **Perform version control operations exclusively with commands**
4. **Write and manage code collaboratively, including but not limited to resolving conflicts due to concurrent work**

In addition to the course-specific content, you are also expected to:

5. **Follow disciplinary best practices throughout the course.**

Prerequisites/Prior Background

You should have done some computer programming before—and in fact, you'll be asked to revisit some previously-written programs, so have those ready. Most importantly, you must bring a growth mindset and an openness to new and unfamiliar ideas. This course involves activities that are unique to software development, and as such are outside the realm of typical technology interactions. Mindful practice and repetition will be key.

Materials and Texts

Most of the course content will be delivered live in class, immediately followed by hands-on practice and repetitions. The same information can be found online in many forms, but may be intended for a different audience and varies highly in quality and timeliness. As such, the recommendation is to encounter the content in class first, then use online resources as references and/or supplements.

As a practicum, we focus on specific technologies to keep things cohesive and concrete. For this offering, we will be using commands that follow the conventions of Unix and Unix-style operating systems. The concepts do transfer over to other technologies such as PowerShell or specialized systems like databases and analytical tools, and I will aim to point out these underpinnings as we go. For in class activities and assignments, you'll need:

- A terminal/command-prompt program which supports Unix-style commands. Linux and macOS come with this already; on Microsoft Windows, Git Bash and Windows Subsystem for Linux (WSL) are viable (and free) options
- A "supercommand" that we will explore in depth is *git*—the current de facto version control software in industry today. On Linux, installation depends on the distribution you have; on macOS, *git* comes with the *Command Line Tools for Xcode* package; on Windows, Git Bash includes the *git* command already
- The nature of the course work (see below) will involve recording yourself and your screen on video. For this, Zoom will suffice, but other apps will also do as long as they create video files in a standard format and encoder

Course Work and Grading

This course focuses on skills, so deliverables predominantly involve *recording* yourself in action. For additional cohesiveness, the assignments are

framed as episodes of a podcast or video channel—who knows, you might be a highly effective tech communicator just waiting to be discovered! As an incentive, additional credit may be given for videos with an appreciable "fun factor" or creative premise. Do take note that although entertainment value makes informative videos better, it does not redeem uninformative ones. This will be reflected in the way assignments are graded.

Your final grade will be based on the percentage of the points you get for the following episodes and your demonstrated knowledge & understanding of the episode content, with these total points:

Filescape navigator	100
Curator personalis	100
Edits, envs, and execution	100
To [Git]hub and to hold	100
The non-accidental programmer	100
Dynamic duo	200
Welcome to the multiverse	200
Use the force you should not	100
Total	1000 points

Percentages ≥ 90% get an A– or better; ≥ 80% get a B– or better; ≥ 70% get a C– or better. I may nudge grades upward based on qualitative considerations such as degree of difficulty, effort, class participation, time constraints, and overall attitude toward the course.

Inherent Knowledge & Understanding

The degree that you will eventually attain is meant to reflect your *knowledge & understanding* of the discipline represented by that degree. The work artifacts you generate along the way are *means* to this end, and not the ends in themselves. This is the key difference between a class and a job: on the job, you are expected to produce but in class, you are expected to learn. And learning comes first: your future ability to produce is bounded by what you have learned and what you will be able to learn.

As such, although producing the course's deliverables contribute to your grade, these deliverables alone comprise half of the points for each milestone. The other half represents how well you have learned the underlying principles that lead to such deliverables—i.e., your *inherent knowledge & understanding*. Engagement in class and office hours—the questions you ask, the answers you provide, the observations you make—will contribute points to this category over time. As such, this engagement is essential and meaningful, and you will genuinely miss out if we miss you.

To this end, class discussions will work to ensure that everyone has a chance to express themselves in this manner. If you prefer a smaller, more private setting, make sure to drop by office hours so that you can have this opportunity.

Demonstrating your inherent knowledge & understanding includes, but is not limited to:

- Assimilating and using the core concepts and vocabulary of a subject area
- Identifying the boundaries of your current understanding and forming questions to help expand those boundaries
- Formulating novel solutions without a starter template or pattern, and validating their results
- Expressing, questioning, and navigating decisions or tradeoffs
- Reading, understanding, and providing feedback on work products created by you or by others

All this must be doable *without crucial reliance on technology or tools*. The forward-looking thread that connects all of these activities is that they align with how you would demonstrate and apply this knowledge beyond the classroom. Even if not every class you take will carry over to your life trajectory, *during* the course you are expected to approach the subject in that manner.

Disciplinary Best Practices (Objective 5)

Appearance of these in the syllabus constitutes fair warning of the consequences of not heeding them.

Punctuality

An assignment's (episode's) number is its due date in *mmdd* format, due by 11:59:59.999pm of that date. If you primarily work in the Keck Computer Science Lab, plan on finishing at home or a few hours earlier: the lab isn't officially open that late so access is not guaranteed.

Point values are based on the state of your assignments at that moment. If there is an issue that prevents you from submitting on time (e.g., personal or family issues, sickness, conferences, can't-miss raids, job interviews, family travel, or emergencies), let the instructor know ahead of time.

Version Control

Version control is an indispensable part of today's computer science landscape in industry, the academe, and the open source community. Effective use of version control is a focus of this course, with the intent that *you will demonstrate these good habits in future semesters and beyond.*

Such habits include but are not limited to: steady, incremental progress rather than "one and done" commits; descriptive messages showing mindful units of work; and the ability to collaborate with others and resolve conflicts in overlapping work. Make sure to acquire these habits not solely to do well in this course, but also to learn a key best practice within our discipline.

Workload Expectations

In line with the *LMU Credit Hour Policy*, the workload expectation for this two-credit-hour course is a minimum of $2 \times 3 = 6$ hours of work by an average student per week, including the time that we spend together in the classroom.

Attendance

Attendance at all sessions is expected for inherent knowledge & understanding credit. If you must miss class, it is your responsibility to notify me about this and keep up with the course.

The last day to add or drop a class without a grade of W is September 4. The withdrawal or credit/no-credit deadline is November 13.

Academic Integrity

Loyola Marymount University is a community dedicated to academic excellence, student-centered education, and the Jesuit and Marymount traditions. As such, the University expects all members of its community to act with honesty and integrity at all times, especially in their academic work. Academic honesty requires that all members of the LMU community act with integrity, respect their own intellectual and creative work as well as that of others, acknowledge sources consistently and completely, act honestly during exams and on assignments, and report results accurately. As an LMU Lion, by the Lion's Code, you are pledged to join the discourse of the academy with honesty of voice and integrity of scholarship.

Academic dishonesty will be treated as an extremely serious matter, with serious consequences that can range from receiving no credit for assignments/tests to expulsion. It is never permissible to turn in any work that has been copied from another student or copied from any source (including the Internet) without properly acknowledging/citing the source. It is never permissible to work on an assignment, exam, quiz, or any project with another person unless your instructor has indicated so in the written instructions/guidelines. It is your responsibility to make sure that your work meets the standard of academic honesty set forth in the "Academic Honesty Policy" found at:

https://academics.lmu.edu/honesty

Technology Use and Academic Integrity

The use of technology in alignment with academic integrity serves the goal of attaining inherent knowledge and understanding:

- ☑ Technology may be used for *instrumental* help: building skills, learning independently, etc.
- ✗ Technology may *not* be used for *executive* help: generating work without understanding, short-circuiting your agency & accountability, etc.

Examples of executive help include:

- ✗ Generating work artifacts that you do not understand yourself
- ✗ Fixing defects without personally knowing their root causes and the nature of the fix
- ✗ Copy-pasting content from other sources without being able to explain what it does, how it works, nor why it is the best alternative

Examples of instrumental help include:

- ☑ Seeking explanations to gain understanding
- ☑ Navigating work products to think about them critically and produce them on your own
- ☑ Accelerating rote, routine, or mechanical tasks that you could otherwise do yourself

TL;DR *Never use technology (including but not limited to AI) to avoid learning, but do use it to augment, amplify, or accelerate (your) learning.*

Special Accommodations

The Disability Support Services (DSS) Office offers resources to enable students with ADD/ADHD; physical, learning, and psychiatric disabilities; and those on the autism spectrum to achieve maximum independence while pursuing their educational goals. Staff specialists interact with all ar-

eas of the University to eliminate physical and attitudinal barriers. Students must provide documentation for their disability from an appropriate licensed professional. Services are offered to students who have established disabilities under state and federal laws. DSS personnel also advise students, faculty, and staff regarding disability issues. Students who need reasonable modifications, special assistance, academic accommodations or housing accommodations should direct their request to the DSS Office as soon as possible. All discussions will remain confidential. The DSS Office is located on the 2nd floor of Daum Hall and may be reached by email at *dsslmu@lmu.edu* or phone at (310) 338-4216. Please visit *http://www.lmu.edu/dss* for additional information.

Course Evaluations

Student feedback provides valuable information for continued improvement. All students are expected to fairly and thoughtfully complete a course evaluation for this course. This semester, course evaluations will be administered online through the Blue™ evaluation system. You will receive an email notification at your Lion email address when the evaluation form is available. You may also access the evaluation form on Brightspace (*https://brightspace.lmu.edu*) dashboard during the evaluation period. Your responses will be anonymous and will not be linked to you in any way.

Topics and Important Dates

Specifics may change as the course progresses. University dates (italicized) are less likely to change.

September	Command line fundamentals: files, folders/directories, autocomplete, history, common shortcuts
September 4	*Last day to add or drop a class without a grade of W*
September 7	*Martin Luther King, Jr., Day; no class*
October	Manipulating files on the command line; editing within a terminal application; environment variables; introduction to *git*; core *git* flows and subcommands (init, clone, add, remove, mv, commit, push, pull)
November	Introduction to Markdown; monitoring how code changes (log, diff, --staged); collaboration and merge conflict resolution (switch, branch, stash, merge)
November 13	*Last day to withdraw from classes or apply for Credit/No Credit grading*
November 25–27	*Thanksgiving break; no class*
December	With *git* power comes *git* responsibility): restore, revert, reset, rebase, forced pushes, cherry-pick etc.
December 15	*Final podcast "screenings"*

You can view my class calendar and office hour schedule in any iCalendar-savvy client. Its subscription link can be found on the course website (it's too long to provide in writing).

Tentative Nature of the Syllabus

If necessary, this syllabus and its contents are subject to revision. Students are responsible for any changes or modifications announced or distributed in class, emailed to students' LMU Lion accounts, or posted on LMU's course management system, Brightspace. If you are absent from a synchronous class session, it is the student's responsibility to check Brightspace and with the professor to see if you missed any important class announcements. Students should not rely on word-of-mouth from classmates.