CMSE 890-602: Reproducible Computational Workflows

Instructor: Andrew Fullard, fullarda@msu.edu

Course Description

Computational workflows are at the heart of scientific research in fields from high energy physics to textual analysis in the humanities. As the amount of science we do grows, reproducibility becomes ever more important. This course will introduce you to the concepts of computational workflows as applied to research, from the basics of design and planning, through development best practices, and finally the application of cutting edge software packages to research problems.

By the end of this course, you will be able to:

1. Create Data Flow Diagrams to describe a computational research problem
2. Understand version control, git and GitHub as development tools
3. Apply an automated documentation generator and a 'Style Guide' to code
4. Apply unit testing and code reviews
5. Create a rectangular data matrix amenable to quantitative analysis
6. Create and query an SQLite database
7. Analyze an existing computational project for modular organization
8. Apply checkpoints to automated workflows
9. Develop a portable, modular workflow for a computational research problem

We will work toward the goals expressed above throughout this course using a range of activities – primarily by writing software both individually and in small groups, but also through discussion, presentations, and other types of exercises.

Topics covered

The primary topics covered in this course include:

- Introduction to workflows
- Git, GitHub and version control
- Code design and style
- Testing and documentation
- Code review
- Data formats, tables and databases
- Modular workflows, related software and development
- Portable workflows and related software

Expectations for you (prior to the start of the semester)

In order for you to fully participate in this class, you need to:

- Be able to use the Unix command line to do basic operations (creating and changing directories, listing their contents, creating, deleting, and moving files, and so forth).
- Be able to program comfortably in a programming language used for scripting such as Python or R.
- Have a computer that has a programming environment installed suitable for your preferred language. I strongly urge you to install the Anaconda Python distribution if you use Python, available at https://www.anaconda.com/products/distribution .
- Install Microsoft's Visual Studio Code integrated development environment, which is available at https://code.visualstudio.com/.
- Have a laptop that you can bring to class every week with the required development environment. If you do not have a laptop, please let me know as soon as possible.

Required reading materials

There are no required textbooks for this course. Online video and text materials will be used as part of the course.

Other required materials

In-class programming assignments are a critical part of the learning process in this course, as is collaborative learning. To that end, you are expected to have access to a computer and network connection when required.

Course activities

Class participation: Active class participation (led both by the instructor and by students) is critical to the success of this course. As such, you are expected to attend class every week, bring the required materials, and to actively participate in the in-class activities.

Pre-class assignments: I will assign short assignments that are due prior to class. The purpose of these assignments is to introduce new material and give you some practice with it so that we can focus on experimentation and implementation in class. These assignments may consist of one or more short videos or reading assignments and related questions and will be **due at 11:59 p.m. the night before class** via the course's GitHub Classroom. We will discuss these items in class.

In-class assignments: Class sessions will be held once a week at the time and location described below, and will be broken up into presentations, discussions, and activities that will allow you to immediately implement (and get instant feedback on) what you have just learned. In class activities will be turned in at the end of the class session via GitHub. I will

accept assignments in various formats as appropriate for the activity (e.g. a diagram may be in PDF while code may be in a programming language).

Homework: You will have periodic homework assignments (every ~3 weeks) that will provide a more in-depth exploration of the materials covered in class. These will be pursued either individually or in pairs, and will be turned in by the given deadline using GitHub.

Semester Projects: This class will have a semester project that will involve synthesizing some subset of the techniques that you have learned about for a project that relates to your research interests. There will be some programming involved, as well as a brief writeup and a presentation at the end of the semester. More details will be available near the middle of the semester.

Course meeting time and location

This class will meet on Fridays from 12:40-2:30 p.m. in 1455 BPS (inside the ICER office suite). The first day of class is August 29th. There will be no hybrid or online options for this course.

Other important information

Course Website, Calendar, and discussion channel: This course uses a GitHub repository for course organization. Accompanying course information, including the official course calendar, can be found in this repository. Most assignments will be handed out and turned in via a GitHub Classroom – as a result, we will work to create a GitHub account for each student during class. If you already have an account, please send it to me. Please note that the course nominally has a Desire2Learn page (at http://d2l.msu.edu), but it will not be used.

Class attendance: This class is heavily based on material presented and worked on in class, and it is critical that you attend and participate fully every week! Lack of attendance or habitual lateness will be dealt with first by sarcasm on the instructor's part, and then by more formal means.

Classroom behavior: Respectful and responsible behavior is expected at all times, which includes not interrupting other students, turning your cell phone off, refraining from non course-related use of electronic devices, and not using offensive or demeaning language in our discussions. Flagrant or repeated violations of this expectation may result in ejection from the classroom, grade-related penalties, and/or involvement of the university Ombudsperson. In particular, behaviors that could be considered discriminatory or harassing, or unwanted sexual attention, will not be tolerated and will be immediately reported to the appropriate MSU office.

Academic Honesty: Intellectual integrity is the foundation of the scientific enterprise. In

all instances, you must do your own work and give proper credit to all sources that you use in your papers and oral presentations – any instance of submitting another person's work, ideas, or wording as your own counts as plagiarism. This includes failing to cite any direct quotations in your essays, research paper, class debate, or written presentation. The MSU College of Natural Science adheres to the policies of academic honesty as specified in the General Student Regulations 1.0, Protection of Scholarship and Grades, and in the all University statement on Integrity of Scholarship and Grades, which are included in Spartan Life: Student Handbook and Resource Guide. Students who plagiarize will receive a 0.0 in the course. In addition, University policy requires that any cheating offense, regardless of the magnitude of the infraction or punishment decided upon by the professor, be reported immediately through an Academic Dishonesty Report, which is sent to the appropriate dean of the student's college as well as to the Dean of Students (for undergraduates) or Dean of the Graduate School (for graduate students).

It is important to note that **plagiarism in the context of this course includes, but is not limited to,** directly copying another student's solutions to pre-class, in-class, or homework problems; copying materials from online sources, textbooks, or other reference materials *without citing those references in your source code or documentation*, or having somebody else do your pre-class work, in-class work, or homework on your behalf. Any work that is done in collaboration with other students should state this explicitly, and have their names as well as yours listed clearly.

More broadly, we ask that students adhere to the Spartan Code of Honor academic pledge, as written by the Associated Students of Michigan State University (ASMSU): "As a Spartan, I will strive to uphold values of the highest ethical standard. I will practice honesty in my work, foster honesty in my peers, and take pride in knowing that honor is worth more than grades. I will carry these values beyond my time as a student at Michigan State University, continuing the endeavor to build personal integrity in all that I do."

Generative AI: Generative AI tools such as ChatGPT can be used to assist your work in this class. You must state when you have used them as part of graded assignments and provide the prompts you used to generate the output.

You are responsible for the information you submit based on an AI query (for instance, that it does not violate intellectual property laws, or contain misinformation or unethical content). Your use of AI tools must be properly documented and cited in order to stay within university policies on academic integrity and the Spartan Code of Honor Academic Pledge.

Instructor information

Contact information:

Andrew Fullard
Research Consultant, ICER
BPS office: 1450 BPS (inside ICER offices)
Email: fullarda@msu.edu

Office hours: By appointment – please contact me by email to schedule a time for us to speak either in person or on Zoom. I'm typically free at most times except Wednesday mornings.

Scheduled travel: None

Grading information

Activity Grade percentage
Pre-class assignments: 5
Participation/attendance: 20
In-class assignments: 30
Homework assignments: 20
Semester project 25

Total: 100% **Grading scale**

4.0 $\geq$ 90%
3.5 $\geq$ 85%
3.0 $\geq$ 80%
2.5 $\geq$ 75%
2.0 $\geq$ 70%
1.5 $\geq$ 65%
1.0 $\geq$ 60%
0.0 < 60%

Note: I am using the standard CMSE department expectations regarding the grading scale for elective 800-level courses. Students that actively participate and do their best on all assignments will receive a reasonable grade.