

# Course Syllabus for SIADS 515: Efficient Data Processing

## Course Overview and Prerequisites

This course will introduce students to the basics of the linux command-line interface, debugging concepts, basic algorithmic principles such as memoization, recursion, caching, and generators, as well as efficiency and code profiling.

There are no prerequisites.

## Instructor and Course Assistants

Instructor: Chris Teplovs, Ph.D. Lecturer III in Information and Research Investigator, School of Information

Course Assistant: Kris Steinhoff; Sean Wu

A specific thank you to McCoy Doherty who assisted in the development and design of this course.

## Communication Expectations

Slack is the preferred communication tool for this course. If you have questions about course content (e.g. lecture videos, quizzes, or assignments), please make sure to use Slack. Instructor and course assistant response time to Slack messages will be within 24 hours.

Personal communication that may involve sensitive information may be emailed directly to the instructor or course assistant. If you email the instructor or course assistant, please include SIADS515 in the email subject. Instructor and course assistant response time to email messages will be within 24 hours.

Instructor grading response time will be 4 days after assignments are submitted.

## Required Weekly Textbooks and Readings

Week 1: Linux Command-Line Interface

[\*The Linux Command Line\*](#) by William Shotts

Week 2: Iterators, Generators, Decorators

Iterators and Generators

[\*Effective Python: 90 Specific Ways to Write Better Python\*](#) by Brett Slatkin

Decorators

[\*Head First Python\*](#) Second Edition by Paul Barry (Chapter 10)

Week 3: Debugging

[\*The Visual Python Debugger for Jupyter Notebooks You've Always Wanted\*](#) by David Taieb

Week 4: Complexity and Optimization

[\*A Common-Sense Guide to Data Structures and Algorithms\*](#) by Jay Wengrow

[\*Python Data Science Handbook\*](#) by Jake VanderPlas

To access the required textbooks, simply click on the links above, which will direct you to the U-M Library website. Scroll down the page and click on *Available Online* (some texts will have multiple online options, but it is recommended you choose **Safari books online**).

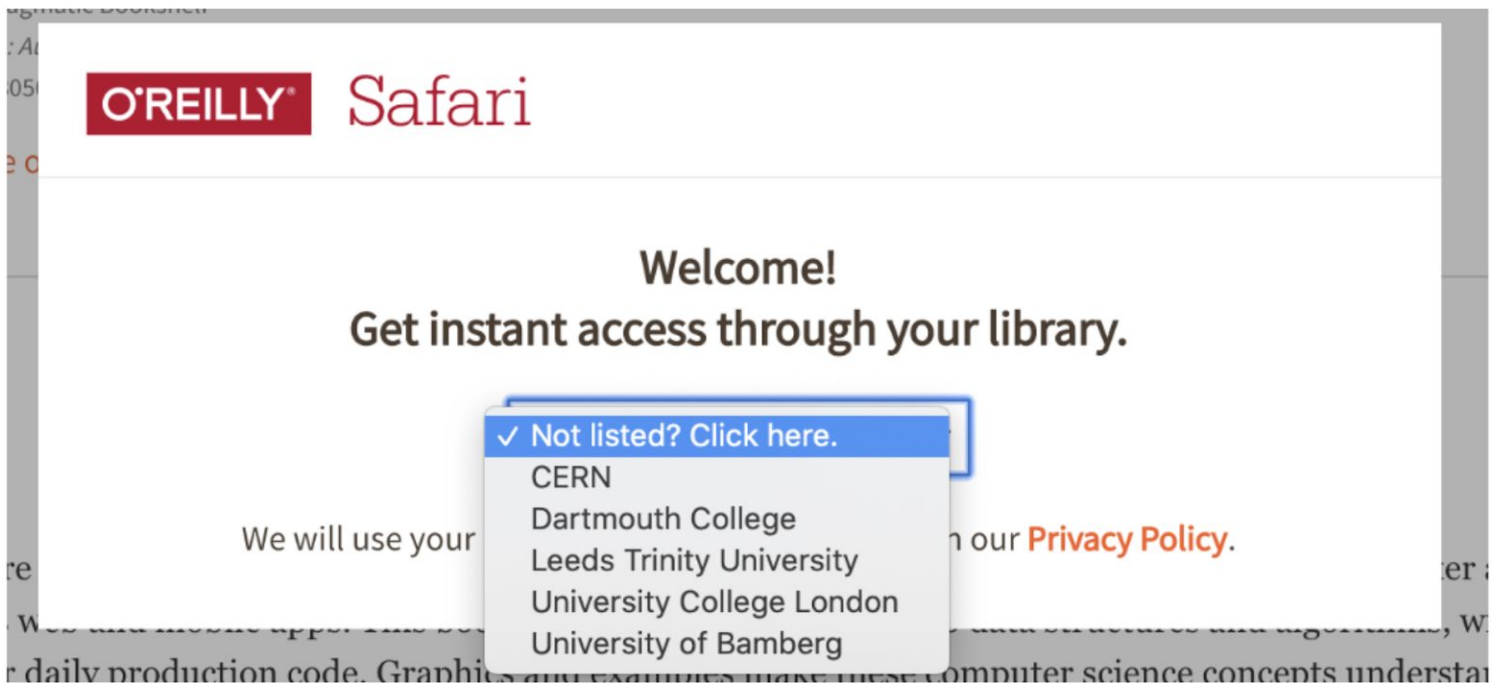
[Available online](#)

[Get This](#)

N/A

Access to the Safari books online version restricted; authentication may be required:

After you are directed to the textbook, you will see an O'Reilly pop-up window asking for you to select your institution. U-M is not an available option, so you will need to select the option "Not Listed? Click Here".



You will be prompted to input your U-M email address (no password required).

Welcome!  
Get instant access through your library.

Just enter your academic institution email below:

Academic email

Already a user? [Click here.](#)

We will use your personal data in accordance with our [Privacy Policy](#).

Let's Go

## Learning Outcomes

Familiarity with the linux command-line interface

Understanding generators, memoizing and cacheing

Debugging

Efficient data structures

Theory and measurement of complexity

## Course Schedule

- This course begins on **Wednesday, February 3, 2021** and ends on **Tuesday, March 2, 2021**.
- Weekly assignments will be **due on Tuesdays at 11:59 pm** (Ann Arbor, Michigan time-Eastern Daylight Time - EDT, UTC -4).

In **Week 1**, you will be introduced to the linux command line interface (CLI) and you will learn basic data manipulation techniques without having to use Python.

In **Week 2**, you will learn about iterators, generator, decorators, caching, and memoization -- all useful techniques to improve the writing and execution of your code.

In **Week 3**, we will review some basic data structures, learn to use advanced ones to help with data manipulation, and debugging techniques for data science.

In **Week 4**, we will discuss algorithmic complexity and look at how we can use profiling to improve the efficiency of our code.

### Weekly Office Hours via Zoom (Ann Arbor, Michigan time):

Your instructor will hold weekly, synchronous office hours using the video-conferencing tool, Zoom. The schedule of office hours can be found by clicking on the **Live Events** link in the lefthand navigation menu. Additionally, all office hours will be recorded and archived so that you can retrieve at a later date. Archived office hours can be found in respective modules of the course.

### Grading

| Course Item       | Percentage of Final Grade | Due                        |
|-------------------|---------------------------|----------------------------|
| Week 1 Assignment | 25%                       | Tuesday, February 9, 2021  |
| Week 2 Assignment | 25%                       | Tuesday, February 16, 2021 |
| Week 3 Assignment | 25%                       | Tuesday, February 23, 2021 |
| Week 4 Assignment | 25%                       | Tuesday, March 2, 2021     |
| <b>Total</b>      | <b>100%</b>               |                            |

Note: All assignments are required to earn credit for this course.

### Letter Grades, Course Grades, and Late Submission Policy

We realize that the occasional crisis might mess up your schedule enough to require a bit of extra time in completing a course assignment. Thus, we have instituted the following late policy that gives you a limited number of flexible “late day” credits.

You have three (3) free late days to use during SIADS 515. One late day equals exactly one 24-hour period after the due date of the assignment (including weekends). No fractional late days: they are all or nothing. Once you have used up your late days, 25% penalty for each subsequent 24-hour period after the deadline that an assignment is late For example, if the due date is 5:00pm Tuesday, with no late days left, penalties would be:

Before 5pm Friday: 25% deduction

Before 5pm Saturday: 50% deduction

Before 5pm Sunday: 75% deduction

After 5pm Sunday: 100% deduction

You don't need to explain or get permission to use late days, and we will track them for you. Note that resubmissions after the deadline will be counted as late submissions. Also, late days may not be applied to the Week 4 Homework.

The grading scale for this course is as follows:

|    |     |
|----|-----|
| A+ | 97% |
| A  | 93% |
| A- | 90% |
| B+ | 87% |
| B  | 83% |
| B- | 80% |
| C+ | 77% |
| C  | 73% |
| C- | 70% |
| D+ | 67% |
| D  | 63% |
| D- | 60% |
| F  | 0%  |

## Academic Integrity/Code of Conduct

Refer to the [Academic and Professional Integrity](#) section of the UMSI Student Handbook. (access to Student Orientation course required).

## Accommodations

Refer to the [Accommodations for Students with Disabilities](#) section of the UMSI Student Handbook (access to the Student Orientation course required).

Use the [Student Intake Form](#) to begin the process of working with the University's Office of Services for Students with Disabilities.

## Accessibility

Refer to the [Screen reader configuration for Jupyter Notebook Content](#) document to learn accessibility tips for Jupyter Notebooks.

## Help Desk(s): How to get help

- Degree program questions or general help - [umsimadshelp@umich.edu](mailto:umsimadshelp@umich.edu)
- Coursera's Technical Support (24/7) - <https://learner.coursera.help/>

## Library Access

Refer to the [U-M Library's information sheet](#) on accessing library resources from off-campus. For more information regarding library support services, please refer to the [U-M Library Resources](#) section of the UMSI Student Handbook (access to the Student Orientation course required).

## Student Mental Health

Refer to the University's [Resources for Stress and Mental Health website](#) for a listing of resources for students.

## Student Services

Refer to the [Introduction to UMSI Student Life](#) section of the UMSI Student Handbook (access to the Student Orientation course required).

## Technology Tips

- Recommended Technology
- This program requires Jupyter Notebook for completion of problem sets and Adobe or other PDF viewer for reading articles.
- Working Offline
- While the Coursera platform has an integrated Jupyter Notebook system, you can work offline on your own computer by installing Python 3.5+ and the Jupyter software packages. For more details, consult the [Jupyter Notebook FAQ](#).