

# 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: Kris Steinhoff, Intermittent Lecturer in Information, School of Information and Staff Data Engineer, Toyota Research Institute

Course Assistant: Sean Wu, Intermittent Lecturer in Information, School of Information

Chris Teplovs designed and developed this course with the assistance of McCoy Doherty and Kris Steinhoff.

## Communication Expectations

If you have questions about course content (e.g. lecture videos, quizzes, or assignments), please use the class Slack channel to discuss with classmates and the instructional team. Instructor and course assistant response time to Slack messages will be within 24 hours.

Personal communication that may involve sensitive information can be emailed directly to the instructional team at [siads-515-summer-2021-instructors@umich.edu](mailto:siads-515-summer-2021-instructors@umich.edu). Instructor and course assistant response time to email messages will be within 24 hours.

## Required Weekly Textbooks and Readings

- See the [Library's access instructions](#) for the readings hosted by O'Reilly Online.

Week 1: Linux Command-Line Interface

[The Linux Command Line](#) by William Shotts (Chapters 1-4 only)

Week 2: Iterators, Generators, Decorators

Iterators and Generators

[Effective Python: 90 Specific Ways to Write Better Python](#) by Brett Slatkin (Chapter 4. Comprehensions and Generators)

Decorators

[Head First Python, 2nd Edition](#) Second Edition by Paul Barry (Chapter 10. Function Decorators: Wrapping Functions)

Week 3: Debugging

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

[Python Scope & the LEGB Rule: Resolving Names in Your Code](#) by Leodanis Pozo Ramos

## Week 4: Complexity and Optimization

[A Common-Sense Guide to Data Structures and Algorithms](#) by Jay Wengrow (Chapter 3)

[Python Data Science Handbook](#) by Jake VanderPlasw (Chapter 1, section 7: Profiling and Timing Code)

## Learning Outcomes

Familiarity with the linux command-line interface

Understanding generators, memoization and caching

Debugging

Efficient data structures

Theory and measurement of complexity

## Course Schedule

- This course begins on **Tuesday, June 1, 2021** and ends on **Monday, June 28, 2021**.
- Weekly assignments will be due on **Monday 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, generators, 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 left-hand navigation menu. Additionally, all office hours will be recorded and archived so that you can retrieve them 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%	Monday, June 7, 2021

Week 2 Assignment	25%	Monday, June 14, 2021
Week 3 Assignment	25%	Monday, June 21, 2021
Week 4 Assignment	25%	Monday, June 28, 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

Late assignments will be penalized by 20% times the number of days late. Homework that is late 5 days or more will receive zero credit.

The grading scale for this course is as follows:

A	95%
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](#).