FWE 458 Environmental Data Science

Spring 2024

Instructor: Min Chen

Jan 23, 2024

About me

- Assistant Professor of Forest and Wildlife Ecology
- Affiliated with UW Data Science Institute, Geography, Atmospheric and Oceanic Sciences, Nelson Institute Center for Climatic Research, and Energy Analysis and Policy Program
- B.S. Computer Science; M.S. Remote Sensing; Ph.D. Earth and Atmospheric Sciences; Postdoc Global Ecology
- Lab website: https://globalchange.cals.wisc.edu/

Welcome!

	Name	Login ID	SIS ID	Section	Role
	CAMERON ABPLANALP *(He/him)*	CABPLANALP@WISC.EDU	UW861X664	F&W ECOL 458:LEC001	Student
	AMNAH ALJALAL	AALJALAL@WISC.EDU	UW966E698	F&W ECOL 458:LEC001	Student
	DOREEN ANANDE	ANANDE@WISC.EDU	UW970Y195	F&W ECOL 458:LEC001	Student
	DANE BACH	DOBACH@WISC.EDU	UW751B215	F&W ECOL 458:LEC001	Student
	MIN CHEN	MCHEN392@WISC.EDU	UW872T340	F&W ECOL 458:LEC001	Principal Instructor
	RUSAL FERUS	FERUS@WISC.EDU	UW991A581	F&W ECOL 458:LEC001	Student
	Annabelle MAJERUS *(She/her)*	AMAJERUS@WISC.EDU	UW847F494	F&W ECOL 458:LEC001	Student
	NADIA NACKERS *(She/her)*	NNACKERS@WISC.EDU	UW932C140	F&W ECOL 458:LEC001	Student
	TREVOR ROBERTS	THROBERTS@WISC.EDU	UW104Y825	F&W ECOL 458:LEC001	Student
	IPSITA SRINIVAS *(She/her)*	ISRINIVAS@WISC.EDU	UW102G777	F&W ECOL 458:LEC001	Student
	XIAOYU WANG	XWANG2696@WISC.EDU	UW977K848	F&W ECOL 458:LEC001	Student

slido

Join at slido.com
#2350329

ⓘ Click **Present with Slido** or install our Chrome extension to display joining instructions for participants while presenting.

slido

How do you describe your programming skills

ⓘ Click **Present with Slido** or install our Chrome extension to activate this poll while presenting.

slido

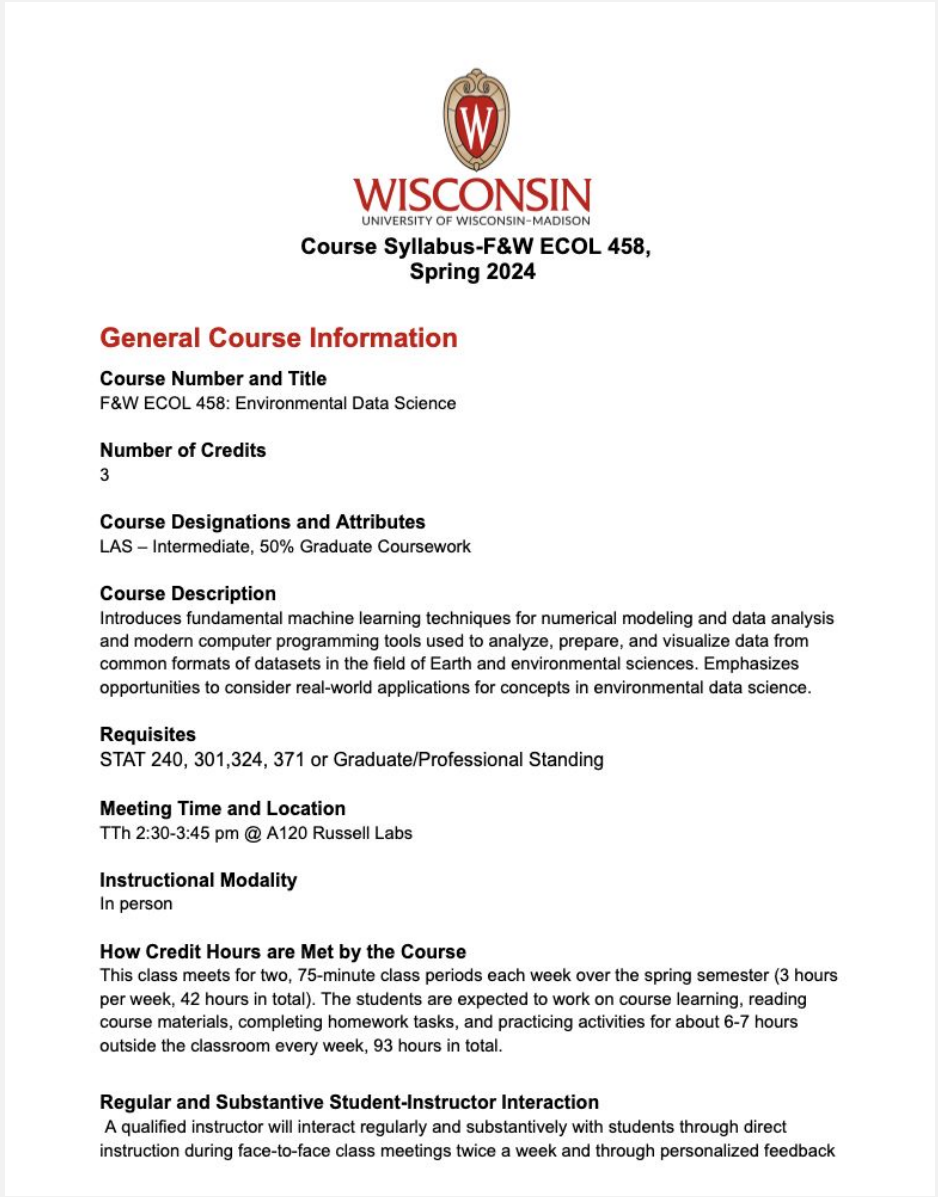
What are your most familiar computer languages?

ⓘ Click **Present with Slido** or install our Chrome extension to activate this poll while presenting.

slido

What are you expecting to learn from this course?

ⓘ Click **Present with Slido** or install our Chrome extension to activate this poll while presenting.



WISCONSIN
UNIVERSITY OF WISCONSIN-MADISON

Course Syllabus-F&W ECOL 458, Spring 2024

General Course Information

Course Number and Title
F&W ECOL 458: Environmental Data Science

Number of Credits
3

Course Designations and Attributes
LAS – Intermediate, 50% Graduate Coursework

Course Description
Introduces fundamental machine learning techniques for numerical modeling and data analysis and modern computer programming tools used to analyze, prepare, and visualize data from common formats of datasets in the field of Earth and environmental sciences. Emphasizes opportunities to consider real-world applications for concepts in environmental data science.

Requisites
STAT 240, 301,324, 371 or Graduate/Professional Standing

Meeting Time and Location
TTh 2:30-3:45 pm @ A120 Russell Labs

Instructional Modality
In person

How Credit Hours are Met by the Course
This class meets for two, 75-minute class periods each week over the spring semester (3 hours per week, 42 hours in total). The students are expected to work on course learning, reading course materials, completing homework tasks, and practicing activities for about 6-7 hours outside the classroom every week, 93 hours in total.

Regular and Substantive Student-Instructor Interaction
A qualified instructor will interact regularly and substantively with students through direct instruction during face-to-face class meetings twice a week and through personalized feedback

Course Syllabus

- Check it out on Canvas
- Highly Practical
- Weekly Homework Assignments
- Due in one week (24*7 hours) since the release of the homework assignment
- For example, if you receive the assignment on the class of a Tuesday, the homework is due before the next Tuesday class
- No exams; final project is required

Course Schedule

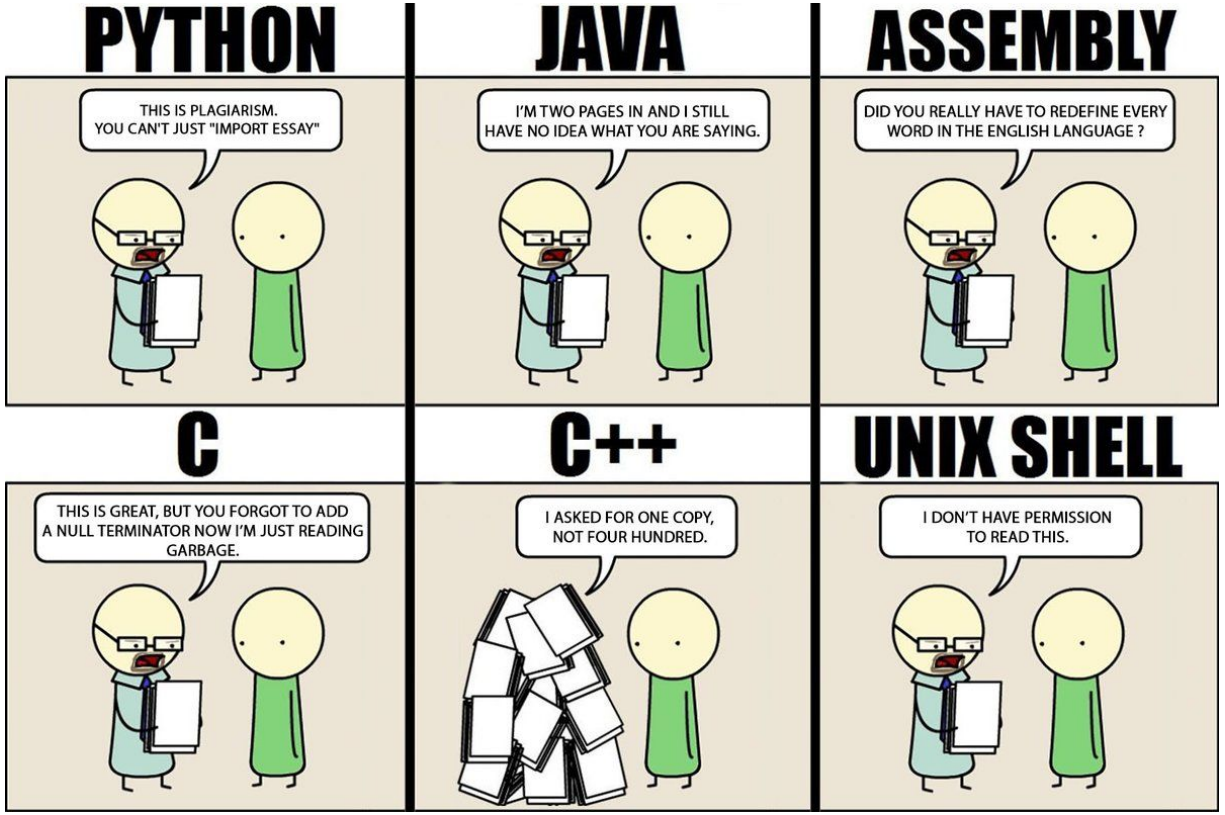
Meeting Dates	Topic
Jan 23, 25	Course overview; Setting up the programming environment
Jan 30, Feb 1	Fundamentals of Python
Feb 6, 8	Fundamentals of Python, cont.
Feb 13, 15	Python Scientific packages
Feb 20, 22	Data visualization
Feb 27, 29	Analyzing spatial data
Mar 5, 7	Analyzing time series data
Mar 12, 14	Fundamentals of Machine learning; Linear regression and logistic regression
Mar 19, 21	Supporting Vector Machine; Naive Bayes Classification
Mar 26, 28	Spring recess, No class
Apr 2, 4	KNN; K-Means
Apr 9, 11	Decision trees and random forests;
Apr 16, 18	Artificial Neural networks and Deep Learning
Apr 23, 25	Other things you need to know; Guest lecture
Apr 30, May 2	AI tools; Questions and Answer session
May 7, 9	Final project due

Setup your programming Environment

- We will use Python + Jupyter Notebook
- Online: Google colab platform
- Offline: VScode

Why Python?

- High level computer language
- Interpretable
- Easy to learn
- Fast growing
- Broad application



Low-level language example

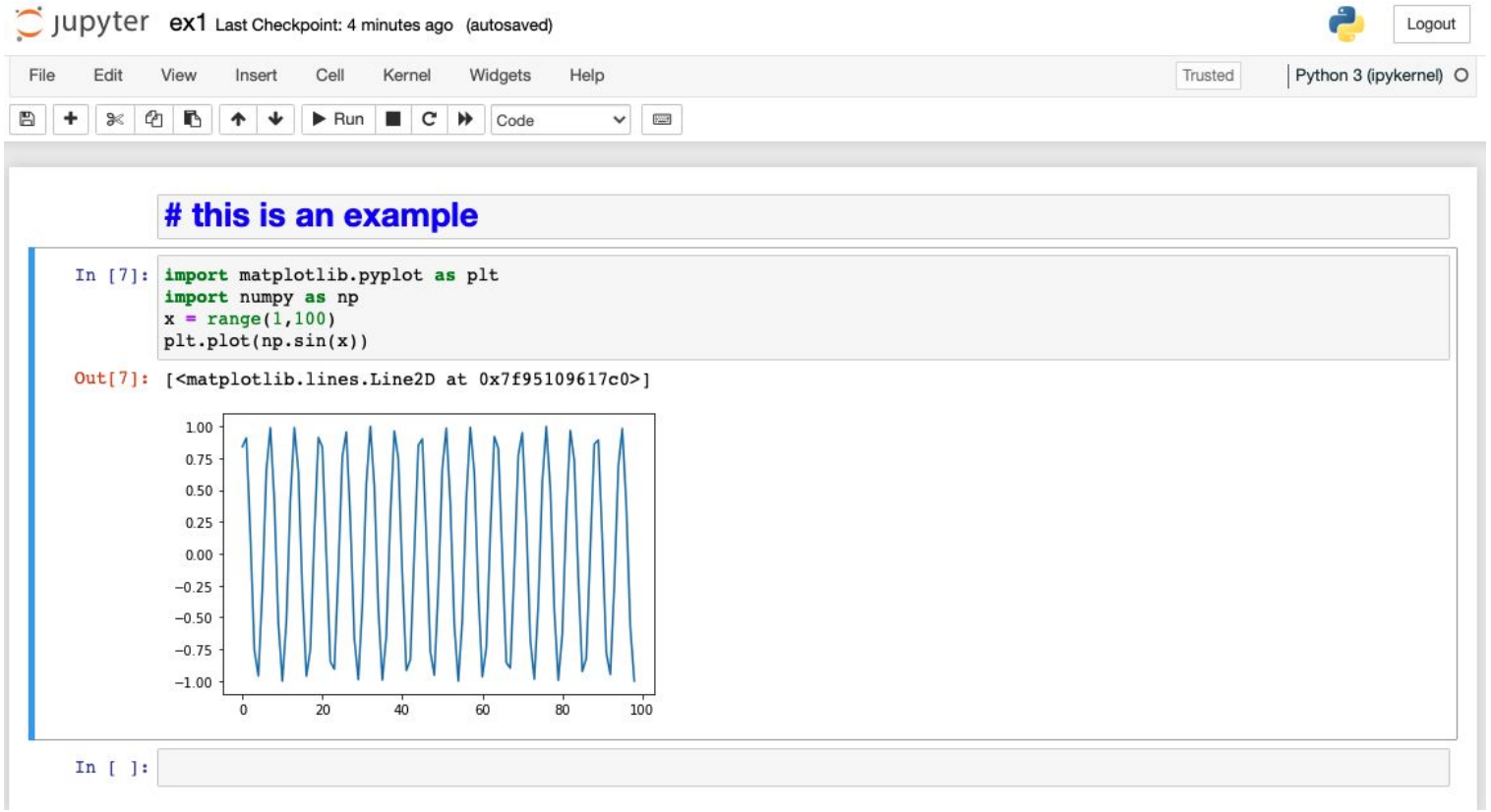
```
MONITOR FOR 6802 1.4          9-14-80  TSC ASSEMBLER  PAGE    2


 C000                 ORG    ROM+$0000 BEGIN MONITOR
 C000 8E 00 70  START  LDS    #STACK

                ***************************************
                * FUNCTION: INITA - Initialize ACIA
                * INPUT: none
                * OUTPUT: none
                * CALLS: none
                * DESTROYS: acc A

 0013           RESETA  EQU    %00010011
 0011           CTLREG  EQU    %00010001

 C003 86 13     INITA   LDA A  #RESETA   RESET ACIA
 C005 B7 80 04          STA A  ACIA
 C008 86 11             LDA A  #CTLREG   SET 8 BITS AND 2 STOP
 C00A B7 80 04          STA A  ACIA

 C00D 7E C0 F1          JMP    SIGNON    GO TO START OF MONITOR

                ***************************************
                * FUNCTION: INCH - Input character
                * INPUT: none
                * OUTPUT: char in acc A
                * DESTROYS: acc A
                * CALLS: none
                * DESCRIPTION: Gets 1 character from terminal

 C010 B6 80 04  INCH    LDA A  ACIA      GET STATUS
 C013 47                ASR A            SHIFT RDRF FLAG INTO CARRY
 C014 24 FA             BCC    INCH      RECIEVE NOT READY
 C016 B6 80 05          LDA A  ACIA+1    GET CHAR
 C019 84 7F             AND A  #$7F      MASK PARITY
 C01B 7E C0 79          JMP    OUTCH     ECHO & RTS

                ***************************************
                * FUNCTION: INHEX - INPUT HEX DIGIT
                * INPUT: none
                * OUTPUT: Digit in acc A
                * CALLS: INCH
                * DESTROYS: acc A
                * Returns to monitor if not HEX input

 C01E 8D F0     INHEX   BSR    INCH      GET A CHAR
 C020 81 30             CMP A  #'0       ZERO
 C022 2B 11             BMI    HEXERR    NOT HEX
 C024 81 39             CMP A  #'9       NINE
 C026 2F 0A             BLE    HEXRTS    GOOD HEX
 C028 81 41             CMP A  #'A
 C02A 2B 09             BMI    HEXERR    NOT HEX
 C02C 81 46             CMP A  #'F
 C02E 2E 05             BGT    HEXERR
 C030 80 07             SUB A  #7        FIX A-F
 C032 84 0F     HEXRTS  AND A  #$0F      CONVERT ASCII TO DIGIT
 C034 39                RTS

 C035 7E C0 AF  HEXERR  JMP    CTRL      RETURN TO CONTROL LOOP
```

hello.asm

```
; ------------------------------------------------------------------------
; Writes "Hello, World" to the console using only system calls. Runs on 64-bit Linux only.
; To assemble and run:
;
;     nasm -felf64 hello.asm && ld hello.o && ./a.out
; ------------------------------------------------------------------------

          global    _start

          section   .text
_start:   mov       rax, 1                  ; system call for write
          mov       rdi, 1                  ; file handle 1 is stdout
          mov       rsi, message            ; address of string to output
          mov       rdx, 13                 ; number of bytes
          syscall                           ; invoke operating system to do the write
          mov       rax, 60                 ; system call for exit
          xor       rdi, rdi                ; exit code 0
          syscall                           ; invoke operating system to exit

          section   .data
message:  db        "Hello, World", 10      ; note the newline at the end
```

Why Jupyter Notebook?

- Jupyter = Julia + Python + R
- Easy to use
- Interactive Development Environment



Install Python on your computer

Navigate to https://www.anaconda.com/download

- **From wikipedia: Anaconda** is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment.
- There are other distributions (CPython etc.), but Anaconda/Miniconda
  - comes with a large collection of pre-installed packages for scientific computing. Miniconda is a smaller, minimal installer for Anaconda
  - Focuses on data science and includes a rich set of libraries for this purpose. It also includes the Conda package manager

Use Google Colab

Navigate to https://colab.research.google.com/

Colab, or "Colaboratory", allows you to write and execute Python in your browser, with

- Zero configuration required
- Access to GPUs free of charge
- Easy sharing

Next Lecture: Git and Github