Syllabus: Introductory computer programming for geoscientists, EART 119

Instructor: Mathis Hain, EMS 308A, mhain@ucsc.edu
TA: Megan Kelley, EMS C317, megankelley@ucsc.edu
Lectures: MWF 9:20-10:25am, EMS D258
Discussion: MW 6:00-7:00pm, Soc Sci 1 135
Canvas course website: https://canvas.ucsc.edu/courses/27502

Problem sets: 8 weekly problem sets, not in exam weeks
Mastery challenges: 4 assignments for you to demonstrate mastery
Exams: Midterm (date TBD), Final (12/9, 12pm)
Requirements: You will need a computer for this class.
Textbook: 'Think Python' is freely available as a PDF file.

Short description:
This course provides an introduction to programing for problem solving in the Earth and Environmental Sciences, assuming that the students have no prior experience with programing. The goal of this course is to develop useful programing skills for prospective scientists, rather than to cover every last aspect of programing with Python.

Learning goals:
(A) Learn how to communicate with your computer.
(B) Develop skills how to load, process and plot scientific data.
(C) Leveraging computational tools to solve scientific problems.

Topics covered:
# BASICS: You will learn how to set up and use Python on your computer. This includes terminology, variables, expressions, functions, branches, loops, etc.

# PLOTING: You will learn how to handle and analyze common types of Earth data and how to construct advanced visualizations based on that data. This includes histograms, scatter plots, fitting models to data, time series analysis, maps and videos.

# COMPUTE: Oftentimes scientific problems can be expressed as mathematical equations. You will learn how to use your computer to carry out the computations required to solve these problems, including algebra and calculus.

Group work: You are allowed and even encouraged to work in groups. If you have an interesting discussion please bring it to class. You are trusted to be honest and that each of you hands in your own work.

Inclusivity: Everyone is welcome to this class, and we can all expect to be treated with respect. This course is designed to be useful regardless of you prior experience with programing.

Week-by-week topics:

(1) Introduction to Computer programing with Python

- Computers, file system, programs, command line
- Setting up ‘Python’ programing environment, including SciPy; Hello World
- Debugging: Syntax, semantics & runtime errors

(2) Variables, expressions & statements

- Strings, numbers, lists; arrays, vectors and matrices
- Operators, functions, & modules
- Loops & branches

(3) Input & output

- User-input, file-input, caller-input
- Screen-output, file-output, caller-output
- Plotting, saving plots

(4) Univariate, bivariate & multivariate data

- Types of plots: histogram, boxplot, 2D-plot, 3D-plot, subplots
- Regression: fitting linear and non-linear models to data
- Classification, clustering and principle component analysis

(5) Time-series data

- Resampling and interpolation
- Basic time-series analysis and visualization (trends, autocorrelation, covariance)
- Advanced time-series analysis and visualization (filters,

(6) Spatial data and maps

- Types of plots: contours, shapes, color maps, maps
- Maps and map projections
- Processing spatial data: interpolation, gradients

(7) Data in space and time

- Slicing multidimensional data
- Image processing
- Animations and videos

(8) Calculus

- Numerical differentiation and root-finding
- Solving one ordinary differential equation
- Solving systems of ordinary differential equation

(9) Algebra

- Evaluating algebraic expressions
- Solving algebraic expressions
- Optimization and inverse problems

(10) Advanced topics in scientific computing

- Performance
- Parallel computing
- Code sharing

Grading: There will be a final exam (40%) and a mid-term exam (20%) as well as eight problem sets (5% each) and four challenges (must pass). With a total of 100% possible points the following scale determines the final grade: >85% As, 85-70% Bs, 70-60% Cs. If this scale results in too few As and Bs, then a curve will be used to assign at least 10% As and 20% Bs.

Expectation: According to Senate Regulation 760 a 5-credit course is to average 15 hours of work per week. This includes class attendance (3h), discussion sections (2h), class preparation (2h), independent programing (3h), after-class review (2h), problem sets/quizzes (3h).