Math 242: Mathematical Programming
Fall 2017

Instructor: Chris Curtis
Office: GMCS 591
Email: ccurtis@mail.sdsu.edu

Meeting Time: MW 15:30-16:45

Prerequisites: Mathematics 150,151.

Official Course Description: Introduction to Python programming. Modeling, problem solving, and visualization.

Textbook: Scientific Computation: Python Hacking for Math Junkies, by Bruce E. Shapiro. ISBN: 9780692366936

Secondary Text: Python for Informatics, by Charles Severance.

Learning Outcomes: The overaching outcomes in this course will be for students to, using the Python programming language,

1. Learn the 'science' of computer programming.
2. Learn the 'art' of comptuer programming.
3. Represent abstract mathematics as computer code, and translate computer code into mathematics.
4. Visualize and describe data.
5. Learn in what ways modern computing is done.

This will be accomplished by achieving the following learning outcomes.

1. **Procedural Programming in Python**: Students will define and use
    - data types

- conditional statements
- while and for loops
- functions

Since no one of these topics is useful without the other, synthesis is critical. Problems will build in complexity from week to week as new skills are learned so as to promote an integrated view of programming.

2. **Basic Data Structures**: For storing and manipulating data, students will use

   - NumPy arrays
   - lists
   - tuples
   - dictionaries

   The students will use "Pythonic" data manipulation techniques such as list comprehension and lambda functions. Basic sorting techniques and their analysis will be covered and implemented. Students will be required to synthesize these techniques with other learning outcomes so that, for example, students master the use of appropriate control structures for different data types and problems involving data. Taken together with the Procedural Programming outcome, this ensures students learn the 'science' of computer programming.

3. **Doing Mathematics on Computers**: Throughout the course, students will practice translating problems and notation in mathematics into algorithms via the Python language. The reverse skill of translating code into mathematical notation will be studied and practiced as well. While this will be a focus throughout the course, the last third of the course is devoted to realizing abstract mathematical frameworks in the form of Python.

4. **Data Visualization and Manipulation**: Students will learn how to read and write to files. They will make several types of plots and use the plotting capabilities available in Matplotlib. Students will be required to synthesize various techniques in order to produce different plots of data, and written interpretations of said plots will be a part of assignments. Taken together with the Doing Mathematics outcome, students will have ample opportunity to practice the 'art' of computer programming.

5. **Working in Modern Computing Environments**: Students will use the features of various programming environments to solve problems and present their work. They will use stand-alone interpereters

such as Canopy. They will use web based interpreters such as Jupyter. They will use Jupyter to present results in mathematics by taking advantage of the LaTeX markup language. Thus students will also be exposed to the breadth of ways in which programming is done.

Grading Policy: Your final score will consist of homework (30%), two midterms (20% each), and a final exam (30%). Homework is roughly due every week, though please pay attention to the schedule since there are exceptions to this (and every) rule.

Homework Policy: Work you submit should be as stand alone Python files or in Jupyter notebooks per the request of the problem. Late work is not accepted unless you make arrangements with me in advance.

Students with Disabilities: If you are a student with a disability and believe you will need accommodations for this class, it is your responsibility to contact Student Disability Services at (619) 594-6473. To avoid any delay in the receipt of your accommodations, you should contact Student Disability Services as soon as possible. Please note that accommodations are not retroactive, and that accommodations based upon disability cannot be provided until you have presented your instructor with an accommodation letter from Student Disability Services. Your cooperation is appreciated.

Week	Date	Sections
Week 1	08/28	Jupyter Notebooks, Approximation
	08/30	Approximation
Week 2	09/04	No Lecture, Labor Day
	09/06	Taylor Series, HMWK 1 Due
Week 3	09/11	Taylor Series, Taylor's Remainder Theorem
	09/13	Taylor's Remainder Theorem, HMWK 2 Due
Week 4	09/18	3n+1 Sequence, Conditionals, While Loops (Ch. 12, 13)
	09/20	Bessel Functions, While Loops (Ch. 13), HMWK 3 Due
Week 5	09/25	Fibonacci Sequence, For Loops (Ch. 19)
	09/27	Functions (Ch. 18), HMWK 4 Due
Week 6	10/02	Floating Point Arithmetic
	10/04	Review, HMWK 5 Due, Midterm I Assigned
Week 7	10/09	Floating Point Arithmetic
	10/11	Floating Point Arithmetic, Midterm I Due
Week 8	10/16	Zeros of Functions
	10/18	Zeros of Functions, HMWK 6 Due
Week 9	10/23	Zeros of Functions
	10/25	Zeros of Functions, HMWK 7 Due
Week 10	10/30	Quadrature
	11/01	Quadrature, HMWK 8 Due
Week 11	11/06	Quadrature
	11/08	Quadrature HMWK 9 Due
Week 12	11/13	Matrices
	11/15	Review, Midterm 2 Assigned
Week 13	11/20	Solving Linear Systems
	11/22	No Lecture, Midterm 2 Due
Week 14	11/27	Lagrange Interpolation
	11/29	Lagrange Interpolation, HMWK 9 Due
Week 15	12/04	Splines
	12/06	Splines, HMWK 10 Due
Week 16	12/11	Splines
	12/13	Review
Week 17		Final is due, Merry Christmas