MATH 260: Python programming in math
Fall 2020 syllabus

Instructor/Office: Jeffrey Wong (office online: see piazza for Zoom link)

Physical office: Physics 029B (not likely to be used)

Times/location: WF 3:30-4:45, online (Zoom), classes Aug. 19 to Nov. 13

Course Website: Piazza (main site) and course Sakai site (grades/submissions)

Textbook: No required text.

Course description: This is an introductory programming course in python that will provide a foundational background for programming in a mathematical setting. Students learn the basics of object oriented programming: memory storage and variable scoping, recursion, objects and classes, and basic data structures. A variety of numerical methods are introduced, with a focus on practical implementation, through a series of modules covering a variety of subjects such as physical modeling, genetics, and optimization.

Prerequisites: Students should have taken at least linear algebra (Math 216, 218 or 221), while no programming background is required. Not open to students who have taken CS 201.

Technology: Access to the following (free) resources is required: Zoom (for lectures), github (to share and turn in work), Python 3.7 or 3.8 and an editor (e.g. an IDE like Spyder/PyCharm). Information for getting started will be provided.

Online matters:: In addition to accessibility issues experienced in the typical academic year, I recognize that remote learning may present additional challenges such as unreliable internet, conflicts that prevent attendance of synchronous meetings, or lack of quiet study spaces. If you are experiencing such difficulties or any others, please feel free to contact me to discuss accommodations.

Grading: Course grades will be based on the following components:

- Final Project (32%)
- Homework: Assigned (roughly) once per week, lowest score dropped (40%)
- Midterm Exam: One 'take home' exam covering programming (20%)
- Participation (8%): includes active participation in activities such as posting questions/answers on Piazza, attending office hours and final project discussions

Note that attendance in synchronous class meetings is strongly encouraged but, if missed, is not penalized in the grade.

Some words on grading: The percentages indicate, roughly, the relative weights of each part and a **baseline score**. However, the point of the course is to learn something by the end, and this 'average' may not reflect that. Thus, your final grade may depend on a more detailed assessment of the graded items. For instance, a low grade on the midterm can be mitigated by demonstrating progress elsewhere.

Exams: One midterm exam that covers programming fundamentals and **no final exam**. The midterm will be 'take-home' and open notes (with some restrictions), to be taken during a subset of a continuous three hour period of your choice.

Final project: Details will be provided around mid-way through the course. You will select a topic involving some application of computational mathematics. The project will involve learning and implementing algorithms, exploring the results and writing a brief discussion.

Homework:

- Programming requires paying attention to specifications! Read all instructions carefully and make sure you are doing what is requested. Make sure to format and submit assignments as specified in the assignment.
- Homework, up to a certain point in the course, will be turned in **to Sakai** Later, it will be turned in **through github**. You will be responsible for making sure your homework on github is accessible / easy to find.
- Homework will be assigned approximately once per week; typically assigned and due on Wednesdays (the first meeting of the week).
- Homework should be turned in by the deadline to ensure full credit. If late, the homework should still be turned in for feedback and full or partial credit. If you need an extension on a homework, feel free to ask.
- Working and studying in groups is strongly encouraged, as well as discussing code (writing code to be read by others is an excellent way to improve coding skills!). However, the final product - solutions and code - should be your own.
- Code should be functional 'out of the box' (i.e. it runs and produces the required outputs). Explanatory comments, when asked for, should be concise and in the appropriate place.

Ethics: Students are expected to follow the Duke Community Standard. If a student is found responsible for academic dishonesty through the Office of Student Conduct, the student will receive a score of zero for that assignment. If a student's admitted academic dishonesty is resolved directly through a faculty-student resolution agreement approved by the Office of Student Conduct, the terms of that agreement will dictate the grading response to the assignment at issue.

Overview of course material

The goal is to cover most if not all of the topics in the main areas listed. The details may change; see the updated schedule on Piazza for details.

- **Part I: Programming fundamentals:**
  - The basics (language function; syntax)
  - Control structures (loops, ifs, etc.)
  - Data types, memory management basics
  - Functional programming and recursion
  - Input/Output, plotting
  - Object oriented basics: classes, objects, OO programming ideas
  - Data structures (trees, arrays, etc.)
- **Part II: Introduction to scientific computing**
  - An intro to using computers to solve mathematical problems
  - Errors, convergence, and ‘numerical’ considerations
  - Translating algorithms to code - efficiency, organization
  - A few numerical methods (integration...)
  - Applications to modeling and solving ODEs
- **Part III: Modules in computational mathematics**
  - Each module will cover a topic, including: context, the application and the practical problem to solve, numerical methods and implementation.
  - The goal is to illustrate how the programming principles are applied, and to learn to code through meaningful example (plus, to learn some numerical methods and interesting mathematics along the way).
  - Modules will span approximately two to four class periods.
  - See the schedule on Piazza for tentative topics for this part.