ME369P (18169) / ME397 (18365): Application Programming for Engineers
Walker Department of Mechanical Engineering
University of Texas at Austin

Lectures:
Time: TTh, 3:30 - 4:45 pm
Location: ETC 3.108

Instructor: Mitch Pryor
Email: mpryor@utexas.edu
Phone: (512) 471-5182
Office: AHG 106 (this may change)
Office Hours: Tues 10-12ME

Teaching Assistants:
Name: Barak Lidsky
Email: baraklids@utexas.edu
Office Hours: tbd

Course Objectives:
- Learn and use Python to solve problems in the engineering domain;
- Introduce data structures and encapsulation which are instrumental to developing modular, re-useable software applications;
- Introduce students to software tools that will help them to solve challenging engineering problems;
- Develop problem solving skills for writing, integrating, debugging and testing software; and
- Learn to collaborate and participate with teams and the open-source community so students can utilize – and contribute to – its wealth of resources.

Course Overview: This is a second course in computer programming designed to provide a broad overview of fundamental programming concepts and workflow for mechanical engineers. Where ME318 (a course prerequisite) focused on common programming structures, programming numerical algorithms, and debugging; this course will review these concepts while learning to use Python. Students then explore the fundamental concepts related to data structures and encapsulation via object-oriented design.

While the topics taught are applicable to most modern programming languages, this course will focus on Python. Python is an open-source, flexible, intuitive, cross-platform, and easy-to-debug language. Furthermore, multiple libraries useful to engineers exist including:

- SciPy/NumPy: scientific computing
- Matplotlib: plotting library

- Simpy: Symbolic mathematics
- IPython: GUIs and interactive computing
- ROS: Robot Operating System
- Pyglet: 3D Animation and game creation engine
- pyQT: GUI toolkit for python.
- Torch/pyTorch: Machine learning / deep learning
- OpenCV bindings: Computer vision

Students will also learn to use the tools that enable programmers to work in teams or within the open source community including software repositories, versioning control, debugging, and open source practices/etiquette.

Official Course Catalog Description: Designed for students who have some experience in programming and are interested in the sharing and development of open source software applications. Introduces the Python Programming language, an open source, flexible, and intuitive debug programming language, with an emphasis on system modeling, simulation, data analysis, and software/data management. Students will create mini projects in Python that demonstrate software design and organization, debugging, open source practices, and data visualization. Three lecture hours a week for one semester. Prerequisite: For engineering majors, Mechanical Engineering 318M *or equivalent* with a grade of at least C-; for others, upper-division standing and written consent of instructor.

Course Outline:

- **Section I – Programming & Python Fundamentals**
- **Section II – Data Structures**
- **Section III – Encapsulation and Object-oriented programming**
- **Section IV – Collaborative Programming**

Course Prerequisites: Undergraduates at UT must have completed ME318 with a grade of at least C- or equivalent. All students will be expected to understand and have experience with common programming scripting structures. If you have any concerns, or feel that you have acquired the perquisite skills through other avenues, please contact the instructor.

Homework and Exams: In this class there will be approximately 4-6 homework assignments; 4-6 in-class puzzles/quizzes, a paper/resource presentation, and team project with a final presentation. Attendance will be taken during class and will be considered as part of the class participation in the weighting below. **Late homework cannot be accepted as homework results are discussed in future lectures and solutions are posted.** Note that one homework assignment, I have found that attention to the homework is the best way to learn to program. Programming is NOT a spectator sport. Learning is best facilitated by doing. Thus, the weighting will be as follows:

- Homework: 30%
- Puzzles/Quizzes/Exams: 15%
- Paper/Resource/Package Presentation: 15%

- Final: Project/Presentation: 35%
    - Project proposal, requirements document: 5%
    - Final report/application/documentation: 15%
    - Peer evaluation: 5%
    - Final presentation: 10%
- Class participation: 5%

Final projects will be collaborative team projects. The scope of the project will depend on course standing (graduate or undergraduate), and may be proposed by a team, university research group, or an external party or company.

Text and References: There are numerous free online books and resources. Some that will be utilized in the course.

- Official Python documentation
    - https://www.python.org/doc/
- A Byte of Python
    - https://python.swaroopch.com/
- Learn Python the Hard Way (Ver 3)
    - https://learnpythonthehardway.org/python3/intro.html

If you prefer a book with a cover and more detail on the concepts, then consider:

- Introduction to Programming using Python, by Y. Daniel Liang. (ISBN-13: 978-0132747189)

Course Policies: *Collaboration* with other students is encouraged. All work submitted for credit, however, must be your own. This includes any code required for the assignments. Late homework assignments will not be accepted. ***Any evidence of plagiarism or other forms of scholastic dishonesty will be grounds for a failing grade in the course.***

Homework assignments are significant portion of the class grade as understanding controls is facilitated by creating controllers. Assignments must be neat and easy for the grader to follow. ***Unclear, undocumented, or unorganized assignments may be returned without credit.***

Class Attendance is in your best interest and sign-up sheets will be passed around at the beginning of each lecture. Also note, ***some material not specifically covered in the text will be presented in lecture.***

Disabilities: The University of Texas at Austin provides, upon request, appropriate academic adjustments for qualified students with disabilities. The University of Texas at Austin provides, upon request, appropriate academic adjustments for qualified students with disabilities. For more information, contact the Office of the Dean of Students at 471-6259, 471-4241 TDD or the College of Engineering Director of Students with Disabilities at 471-4382.