ECE-203 Programming for Engineers
Course Syllabus and Policies

Textbook

The textbook for this course is *Think Python* by Allen Downey, O'Reilly, 2015 (ISBN-13: 978-1449330729). The official webpage for the textbook is:
`http://www.greenteapress.com/thinkpython`
where the author has made the most up-to-date version available in PDF format. We will follow the Downey textbook for preliminary introductions to subject matter and we may occasionally use external resources to supplement textbook material.

Optional References

For a complete *A to Z* language reference, *Learning Python* by Mark Lutz is highly recommended and very approachable. Lutz's book is, however, extremely comprehensive and provides detailed coverage across all aspects of the Python programming language (which is the reason for its hefty 1600 page count). If you need clarification of a specific aspect of the Python language, Lutz provides good, down to earth, clarification accompanied with plenty of useful examples.

Syllabus

The following outlines the course coverage by term week:

- **Introduction & The Python Execution Model (Jan 5)**: Course introduction, general review of computer organization, program execution, control flow, basic python program anatomy, mutable and immutable types, name binding, and brief overview of fundamental types. *Reading: Chapters 1 and 2*

- **Python Fundamentals (Jan 12)**: **Typing:** dynamic typing, explicit and implicit type casting, immutable and mutable types; **Strings:** immutability of strings, the index operator, negative (wraparound) indexing, slicing, substring matching using the `in` operator, string concatenation using the `+` operator, string formatting using the `%` operator; **Boolean Types**: introduction to conditional `if` statements; **Lists:** mutability of lists, list heterogeneity, list indexing and slicing, appending to lists, removing from lists, sorting lists, reversing lists, using the `range()` builtin to generate lists; **Iterators:** iterables (iterable types), using `iter()` to manually retrieve an iterator, the `StopIteration` exception, introduction to `for`-loops and their use of iterators.

  Reading: Chapter 8

- **Writing Pythonic Code (Jan 19)**: Discussion of pythonic and non-pythonic for-loops on list types (iterator vs C-style index based access). Review of name binding/re-binding and the associated dangers when working with iterators and loops. The `enumerate()` builtin for working with lists that need to be modified in a loop. In-depth look at `enumerate()` with respect to iterators. Introduction to the immutable `tuple` type. Tuple packing and unpacking. The pythonic swap. The use of underscore

(_) when unpacking unwanted items in a for-loop. Unpacking nested iterable types in for-loops. *Reading: Chapters 10 and 12 (skip 12.6)*

- **Anatomy of "Real" Python Programs (Jan 26)**: Introduction of functions, function docstrings, global variables, the module docstring, function parameters, using python's interactive `help()`, local variables, return values, introduction to namespaces, the local, global, and built-in namespaces, namespace search and name resolution, using the `import` builtin for code reuse, managing namespace pollution when using `import`, introduction of the `__name__` "magic" variable, using `__name__` as an import guard. *Reading: Chapters 3 and 11*

- **Generators and List Comprehensions (Feb 2)**: Review for Midterm. Introduction of generator function and the consideration of memory consumption when working with large lists, the `xrange()` generator, the `yield` keyword, the link between generator functions and iterators, introduction to list comprehensions *Reading: None*

- **Midterm Exam (Feb 9)**: *Reading: None*

- **Dictionaries and Basic Exception Handling (Feb 16)**: Review of the midterm solutions and performance. **Dictionaries:** Introduction of (key, value) pairs, python dictionary syntax, iterating over dictionaries, using the `items()` method, key indexing; **Exception Handling:** the `try` keyword, the `except` keyword, the `else` keyword, using manual iterator `StopIteration` exception handling to implement for-loop mechanics manually. *Reading: Chapter 14*

- **File I/O Fundamentals (Feb 23)**: Introduction to file descriptors and OS support for file I/O. The newline control character vs null termination byte, escape sequences, Opening files, using `readlines()` to read an entire file into memory as a list, using `readline()` for more memory efficient file reading, using the `read()` function to read `n`-bytes from a file, using `seek()` to manipulate the file cursor, `os.SEEK_SET` vs `os.SEEK_END` vs `os.SEEK_CUR`, file modes (`r, r+, w, w+, a, a+`), closing files, using exception handlers to ensure file descriptors get `close()d`, introduction to contexts managers, the `with` statement, the `__enter__` and `__exit__` magic methods *Reading: None*

- **Object Oriented Programming (Mar 1)**: Class definitions as a blueprint, objects as instances of a class, the use of `self` to refer to the current instance, methods, instance attributes, object instantiation, object instantiation using `__init__`, explicit instance passing to the class methods, implicit instance passing to methods (traditional method calling) *Reading: None*

- **More on Classes and Instances (Mar 8)**: Object interface design and best practices; constructor, mutator, and accessor methods; private attributes and encapsulation; the benefits of indirect attribute access vs direct access; private methods; class attributes and communication amongst instances; memory layout of class attributes/members vs instance attributes; static methods; the @staticmethod decorator *Reading: None*

- **Final Exam (Friday, March 18th @ 1–3pm; Location: CAT 61)**

Grading Policy

The grading in this course is based on homework assignments, special projects, the mid-term exam, and the final exam. The cumulative grade is based on the following:

- 10% In-lab Programming Assignments
- 10% Take-Home Programming Assignments
- 35% Mid-term Exam
- 45% Final Exam

The class rank in both the exam scores and the overall scores will be considered for the final grade.

Homework Assignment Policy

Homework is always to be submitted on the date for which it is due. Homework submitted after the due date will not be accepted and will be graded at 0 points. Homework should be submitted using Black Board Learn unless otherwise instructed. In the event a student is unable to attend class, he/she should make alternate arrangements to deliver the homework to the instructor before the time it is due (e.g., by faxing the homework to the instructor at 215-895-1695, or sending the homework as an e-mail attachment to the teaching assistant).

Examination Policy

All exams in the course are closed-textbook and closed-“reading assignment material.” Use of other books or any other material (such as solutions to homework assignments), however, is not permitted. Use of cell phones, laptops, tablets, PDAs, or any other device capable of wireless communication is prohibited. Exams will cover material discussed in the lectures, homework assignments, or textbook sections given as reading assignments. For example, the exams may include questions on material covered in class lectures or homework but not specifically covered in the textbook. Similarly, the exams may include any material covered in a section of the textbook given as a reading assignment but not specifically covered in the lectures or homework.

Absentee Policy

Absence from examinations will be excused only under extraordinary circumstances such as medical or family emergencies. A missed examination without prior approval and without legitimate reason will be graded at zero points. An absence will be excused only if the student is able to provide legitimate documentation (such as a physician note). An absence from an examination with prior approval will require the student to take an alternate exam at a later time. Special examinations will not be held earlier or on later dates to accommodate, for example, flight schedules for overseas vacations.

Special Accommodations Policy

Students with disabilities requesting special accommodations must inform the professor within the first two week of the course. Such students requesting accommodations and services at Drexel University must present a current accommodation verification letter (AVL) to the instructor at least five days prior to receiving testing accommodations. AVLs are issued by the Office of Disability Services (ODS). For additional information, contact ODS at www.drexel.edu/ods, 3201 Arch St., Street, Suite 210, Philadelphia, PA 19104, 215.895.1401 (V), or 215.895.2299 (TTY).

Academic Honesty Policy

Each student is expected to complete weekly assignments independently; it is not acceptable to copy another student's work or to copy solutions from any other source. Barring action on flagrant violations, and honor system will be assumed.

The follow is a partial list of activities that will be considered to constitute academic dishonesty:

- Presenting the work of another person (fellow student or not) as your own
- Cheating in an examination such as through conversations with other students, sharing textbooks, calculators, or other materials with another student, using unauthorized books not approved by the instructor in an open book examination, or by inappropriate or unauthorized use of technology such as laptops and cell-phones during an examination
- Using or attempting to use the work of another student or providing solutions to other students
- Failing to take reasonable measures to protect your work from use by other students in assignments, projects, or examinations.

Penalties for academic dishonesty will be strictly enforced and will include a lowering of the grade or a failing grade in the course.

Instructor

Dr. James Shackleford
shack@drexel.edu
Bossone 211
Office Hours: Tuesday 3–5 pm