Syllabus

COURSE INFORMATION

Course: INSC 360: Programming for Information Professionals, 3 hours
Term: Spring 2020
Campus: University of Tennessee, Knoxville
Course Mode: Online, synchronous
Class Meetings: 4:40 – 5:55 pm, Monday & Wednesday
ZOOM Course ID/Link Info: https://tennessee.zoom.us/j/546715588

Or Telephone:
+1 646 876 9923 (US Toll)
+1 669 900 6833 (US Toll)
Meeting ID: 546 715 588
International numbers: https://tennessee.zoom.us/u/ab0AMuTzO

Faculty Contact Information
Monica Ihli
monica@utk.edu
145E, First Floor, Hodges Library (Keypad entry by appointment)
Office Hours by appointment.

SIS Office Information
450 Communications Bldg.
1345 Circle Park Drive
Knoxville, TN 37996-0341
SIS Office: 865.974.2148
Fax (SIS): 865.974.4667

Welcome Statement
Welcome to INSC 360! In Programming for Information Professionals, we will think about the different strategies for getting at data, moving it, and cleaning it up so that data can become information. Our course will be taught using Python. You will find, however, that in order to make Python work for us we will sometimes have to invest a little in understanding how it interfaces with related technologies such as HTTP and XML. This is because our goal is, first and foremost, to gain skills that you can take straightaway out of the classroom and into the workforce. Our course intended to be practical and applied. If at any time you find yourself overwhelmed or feeling lost, please reach out for help.
**COURSE INFORMATION**

**Catalog Description**
This course is an introduction to object-oriented programming using Python with emphasis on data structures, in-built functions, user-defined variables, design, syntax, and control structures for processing and visualizing datasets. *Prerequisite(s): 210.*

**Student Learning Outcomes**
Students who complete this course will be capable of demonstrating awareness, knowledge, and/or understanding of:

- Capable of independently setting up and configuring their own Python development environment on their operating system of choice.
- Familiarity with different data types and containers in Python.
- How to develop simple but robust programs that leverage flow control, exception handling, and logging features.
- Identifying and implementing appropriate solutions for accessing data under a variety of circumstances, such as data on disk, breaking up very large files, using web services, and web scraping.
- Using Python libraries for plotting and visualization of data.

**Course Design**
This course will be delivered synchronously and online. Most class meetings will feature a combination of traditional lecture, in which concepts are sketched out and discussed using slides or whiteboard, and live code demonstrations. You are expected to follow along with code demonstrations by typing out and/or executing the code yourself, when indicated by the instructor. Most classes will also feature in-class activities or polls to test retention of knowledge, which will contribute to your participation grade. The course will be managed in Canvas, though assignments will be submitted in GitHub.

**Required Texts**
None

**Recommended Texts**
In lieu of recommended texts, students are advised to become familiar with the Python documentation, for which a link is provided under the Course Resources section.
COMMUNICATION

Email
I am required to communicate with you through your UTK email address. If you prefer to use another address, consult the OIT Helpdesk to obtain directions for forwarding your UTK mail to your preferred address if you don’t wish to check both accounts.

Instructor Availability
The instructor is available to meet most working days during regular business hours. But if work or other obligations mean that the traditional M-F, 9-5 is not manageable for you, do not let that stop you from getting the same level of support as anyone else. Hodges Library is open 24-7 most of the time, Zoom is also an option and we will find a way to make it happen.

COMPUTING REQUIREMENTS AND RESOURCES

Requirements

• **Laptop or Desktop Computer Requirements:**
  Do not drop the class if you don’t have a personal computer that meets these requirements! Contact the instructor and we will find a solution!
  ◦ 5 GB disk space available
  ◦ 4 GB RAM minimum, 8+ GB RAM recommended
  ◦ 1024x768 minimum screen resolution.
  ◦ Operating System:
    ▪ **Windows:** 64-bit versions of Microsoft Windows 10, 8, 7 (SP1)
    ▪ **Mac:** 64-bit versions of macOS 10.11 or later
    ▪ **Linux:** Any 64-bit Linux distribution with Gnome, KDE, or Unity

• **Python** - 3.5 or higher for the libraries we will be using.

• **PyCharm** – (Integrated Development Environment). If you really want to use a different IDE, I am open to this but it must include minimal variable watching and debugging features. Technical support will not be provided for other IDEs.
  [https://www.jetbrains.com/pycharm/download](https://www.jetbrains.com/pycharm/download)

01/06/20
• **Google Chrome** - There is a lecture and assignment on web scraping that requires Google chrome.

• **Internet Access** - For attending class, but also for installing packages and for web scraping activities. Otherwise, most programming can be done offline, although an internet connection is still valuable for referencing documentation.

• **Monitors** - Dual monitors are not required, but you are highly encouraged to use this setup if at all possible. Doing so allows you to watch the instructor’s desktop on one screen and follow along on the other.

**Technical Support**
OIT will help with basic services like managing Windows or Mac OS, email, and with any software that they license for campus use, as found at [https://oit.utk.edu/software-hardware/software/](https://oit.utk.edu/software-hardware/software/). They will help you setup Linux in the context of the public Linux accounts they provide on their servers, but not for desktop OS. You can access these services at [https://help.utk.edu](https://help.utk.edu).

See your instructor to obtain support for installing/configuring Python, PyCharm, and any other libraries or tools used in this course. OIT will not be able to assist you with these.

**Course Resources**

• Python Download: [https://www.python.org/downloads/](https://www.python.org/downloads/)
• Latest Release Python Documentation: [https://docs.python.org/3/](https://docs.python.org/3/)
• Pycharm Download: [https://www.jetbrains.com/pycharm/download/](https://www.jetbrains.com/pycharm/download/)
• Pycharm Support Documentation: [https://www.jetbrains.com/support/](https://www.jetbrains.com/support/)
• GitHub: [https://github.com/](https://github.com/)
• GitHub User Guides: [https://guides.github.com/](https://guides.github.com/)
• Stack Overflow: [https://stackoverflow.com/](https://stackoverflow.com/)
• Chromedriver: [https://chromedriver.chromium.org/downloads](https://chromedriver.chromium.org/downloads)

**COURSE ATTENDANCE AND PARTICIPATION POLICIES**

**Learner Expectations**

• Turn in homework on time.
• Attend lecture.
- Participate in in-class activities.

**Instructor Expectations**
- Assignments graded within two weeks.
- Provide meaningful feedback on assignments that help you understand how to grow as a programmer.
- Emails responded to within 24-48 business hours.

**Attendance and Participation**
Attendance is required. If illness, family emergency, or other crisis happens, please notify the instructor as soon as possible, watch the class recording, and follow any other instructions provided at the time.

**Inclement Weather**
“The chancellor (or appointed representative) may officially close or suspend selected activities of the university because of extreme weather conditions. When a decision to close is made, information is distributed to the campus community, shared with local media, and posted on the front page at [http://utk.edu](http://utk.edu). SIS will cancel classes when UT is closed. Please check the SIS student listserv ([UTKSIS-L@LISTSERV.UTK.EDU](mailto:UTKSIS-L@LISTSERV.UTK.EDU)) for messages about closing.

**ADDITIONAL POLICIES AND POINTS OF INFORMATION**

**Disabilities that may Impede Learning**
Any student who feels s/he may need an accommodation based on the impact of a disability should contact Student Disability Services in 2227 Dunford Hall at 865-974-6087, or by video relay at 865-622-6566 to coordinate reasonable academic accommodations.

**Civility**
Civility is genuine respect and regard for others: politeness, consideration, tact, good manners, graciousness, cordiality, affability, amiability and courteousness. Civility enhances academic freedom and integrity, and is a prerequisite to the free exchange of ideas and knowledge in the learning community. Our community consists of students, faculty, staff, alumni, and campus visitors. Community members affect each other’s well-being and have a shared interest in creating and sustaining an environment where all
community members and their points of view are valued and respected. Affirming the value of each member of the university community, the campus asks that all its members adhere to the principles of civility and community adopted by the campus: [http://civility.utk.edu/](http://civility.utk.edu/).

**CCI Diversity Statement**
The College of Communication and Information recognizes that a college diverse in its people, curricula, scholarship, research, and creative activities expands opportunities for intellectual inquiry and engagement, helps students develop critical thinking skills, and prepares students for social and civic responsibilities. All members of the College benefit from diversity and the quality of learning, research, scholarship and creative activities is enhanced by a climate of inclusion, understanding and appreciation of differences and the full range of human experience. As a result, the College is committed to diversity and equal opportunity and it recognizes that it must represent the diversity inherent in American society. The College is acutely aware that diversity and fairness are foundations that unite the College’s faculty, staff, students, and the larger communication and information community.

**Instructor Status as a Title IX Mandatory Reporter**
University of Tennessee faculty are committed to supporting our students and upholding gender equity laws as outlined by Title IX. Please be aware that if you choose to confide in a faculty member regarding an issue of sexual misconduct, dating violence, or stalking, we are obligated to inform the University’s Title IX Coordinator, who can assist you in connecting with all possible resources both on- and off-campus. If you would like to speak with someone confidentially, the Student Counseling Center (865-974-2196) and the Student Health Center (865-974-3135) are both confidential resources. For additional resources and information, visit [titleix.utk.edu](http://titleix.utk.edu).

**ASSIGNMENTS, ASSESSMENTS, AND EVALUATIONS**

**Academic Integrity**
Students should be familiar and maintain their Academic Integrity described in [https://hilltopics.utk.edu/academics/](https://hilltopics.utk.edu/academics/), p. 15 as: “Study, preparation and presentation should involve at all times the student’s own work, unless it has been clearly specified that work is to be a team effort. Academic honesty requires that the student present his or her own work in all academic
projects, including tests, papers, homework, and class presentation. When incorporating the work of other scholars and writers into a project, the student must accurately cite the source of that work.”

Students should abide by the Honor Statement (https://hilltopics.utk.edu/student-code-of-conduct/ Section X. Honor Statement)

“As a student of the University, I pledge that I will neither knowingly give nor receive any inappropriate assistance in academic work, thus affirming my own personal commitment to honor and integrity.”

Plagiarism

Statement about plagiarism and penalties, e.g.,

Plagiarism in any of its several forms is intolerable, and attention to matters of documentation in all written work is expected and required. Inadvertence, alleged lack of understanding, or avowed ignorance of the various types of plagiarism are not acceptable excuses.

Specific examples of plagiarism are:

1. Copying without proper documentation (quotation marks and a citation) written or spoken words, phrases, or sentences from any source;
2. Summarizing without proper documentation (usually a citation) ideas from another source (unless such information is recognized as common knowledge);
3. Borrowing facts, statistics, graphs, pictorial representations, or phrases without acknowledging the source (unless such information is recognized as common knowledge);
4. Collaborating on a graded assignment without the instructor's approval;
5. Submitting work, either in whole or in part, created by a professional service and used without attribution (e.g., paper, speech, bibliography, or photograph).

Students who may be unsure of the nature of plagiarism should consult the instructor or a guide for writing research reports. (Additional resources are available at http://www.lib.utk.edu/instruction/plagiarism.)

Infractions of academic integrity are penalized according to the severity of the infraction but may include a course grade of "F."

Assignments and Grading

In-Class Participation (20%):

At some point in most lectures, the instructor will either ask some questions to gauge how well the class is picking up on a concept, give a poll, or give an in class activity. Participation credit will be tracked in Canvas, the same as any other part of the grade, and updated regularly.
Homework Assignments (50%):

- **Number of assignments:** There are ten homework assignments throughout the semester. The homework instructions, due date, and all other information is posted in Canvas under the Assignments navigation link. The homework average is weighted as 50% of the overall course grade in Canvas.

- **Individual Work:** You are encouraged to study together and help each other learn the concepts, but the code that is submitted must be your own work and individually written. There are tools for testing the similarity of two documents, and they will be used if it is suspected that two students are submitting the same code.

- **Grading:** Each homework assignment will include an explanation for how point values are broken up. Coding assignments will clearly delineate the point values for each part in the assignments. You will receive a grade that represents the percentage of total points earned. (0-100%).

- **Commenting:** Commenting refers to adding text to your programming code for informational purposes that doesn’t get interpreted or compiled as part of the program. Adding comments will be a graded component of every assignment. The top of any submitted script should include notes about the author, date, and an explanation of its purpose. In line commenting should be included as appropriate, but don’t forget that commenting is not a substitute for well-named functions, classes, and variables.

- **Submission:** Due to the job-market value of familiarity with using a repository for managing code and version control, this class will teach you how to submit all coding assignments using GitHub. Obviously your own GitHub repository will let you submit content any time you want, unlike Canvas. But only code pushed to the repository prior to the Due date/time associated with the assignment on Canvas will be graded.

- **ONE FREE PASS:** Life doesn’t always go as planned. Everyone gets ONE free pass to turn in a single assignment up to 1 week late without penalty. You are advised not to use it without good reason, as there won’t be another one.

- **Re-submission:** If you lose points on any coding assignment, you can make a new attempt within 2 weeks of the due date to recover up to half of the lost points. The new attempt must be pushed as an update to GitHub. Turning in late work (with the exception of your
one free pass for late work) forfeits the opportunity for re-submission.

- **Note**: Homework in a programming class can be different than many other classes! There are usually multiple solutions to the same problem. For example, one person might use a string literal to find a desired segment of text, while another person might use a regular expression. These are both valid ways to solve the same problem. So focus on understanding the requirements, and then crafting a solution which meets those requirements.

**Exams (20%)**:  
- **Number of tests**: The exams are weighted to be worth 20% of the overall course grade. These will contain a few multiple choice questions, short answers, and small programming tasks to be completed with in a set duration of time. We will have a review day before each exam and the reviews will very explicitly prepare you for what’s on the exam.  
- **Programming Tasks**: If you’ve stayed caught up with homework, then you already know how to do any task that might come up. These will ask you to produce manageable little snippets of code, not complete programs. For example, you might be asked to write a function that performs some simple task, or to write a line of code that shows how to create an instance of a class.  
- **Multiple Choice / Short Answers**: Multiple choice questions will be taken straight from lectures. Short answers will also be taken directly from lecture, and will be related to making good decisions as a developer. For example, you can expect to explain the problem that is solved by using virtual environments, or an example of the circumstances under which you would need to use Selenium instead of requests for web scraping.
Coding Project in Lieu of Final Exam (10%)

Instead of a final exam, this class will have a final programming project, due on the last day of the exam period. Working on a team with other developers is good experience, so unlike your homework, you’ll be free to work either individually or in teams of up to three people. You or your team will be allowed to choose among several available mini-projects for a Python program. The program requirements will be in the scope of topics covered in class. You’ll come up with a strategy, diagram out a process, and then meet with your Team Lead / Senior Developer (your instructor) to get feedback. You’ll also create documentation as you are writing code, and submit the finished product for feedback. Full instructions and a timeline for completion will be posted later in the semester, but the final due date for submission will be the last day of the official university exam period (May 4th).

Preparation of Written Work
Not Applicable.

Due Dates and Late Assignments
Late assignments automatically receive 50% credit reduction, with the exception of your Free Pass to turn a single assignment in up to a week late. Late assignments may not be turned in at all after 2 weeks past the due date. One minute after two weeks, it’s a zero.

Incompletes
Based on adopted University of Tennessee-Knoxville and SIS policy, a grade of I (Incomplete) is reserved for emergencies that prevent the student from completing the course on time. Incompletes are granted only under “the most unusual of circumstances” and solely at the discretion of the instructor. Plan your semester’s course of study carefully to insure sufficient time to complete the required work. For students who simply “disappear” without contacting the instructor and without completing the required form, an "F" is submitted.

Assigning Grades
All grades will be reported as percentages, for example 92% (92 out of 100 points). Assignments will be graded within 2 weeks. Although your work consists of coding assignments to be submitted in GitHub, grades will be entered and maintained in Canvas. There will be no accommodation of individual requests for extra credit at the end of the semester.
## Evaluation

Semester grades will be assigned according to the following scale:

<table>
<thead>
<tr>
<th>Grade</th>
<th>Grade Range</th>
<th>Quality Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>93 ≤</td>
<td>(4 quality points per semester hour) superior performance.</td>
</tr>
<tr>
<td>A-</td>
<td>90-92.99</td>
<td>(3.7 quality points per semester credit hour) intermediate grade performance.</td>
</tr>
<tr>
<td>B+</td>
<td>88-89.99</td>
<td>(3.5 quality points per semester hour) better than satisfactory performance.</td>
</tr>
<tr>
<td>B</td>
<td>83-87.99</td>
<td>(3 quality points per semester hour) satisfactory performance.</td>
</tr>
<tr>
<td>B-</td>
<td>80-82.99</td>
<td>(2.7 quality points per semester credit hour) intermediate grade performance.</td>
</tr>
<tr>
<td>C+</td>
<td>78-79.99</td>
<td>(2.5 quality points per semester hour) less than satisfactory performance.</td>
</tr>
<tr>
<td>C</td>
<td>70-77.99</td>
<td>(2 quality points per semester hour) performance well below the standard expected of graduate students.</td>
</tr>
<tr>
<td>D</td>
<td>60-69.99</td>
<td>(1 quality point per semester hour) clearly unsatisfactory performance and cannot be used to satisfy degree requirements.</td>
</tr>
<tr>
<td>F</td>
<td>59.99 ≥</td>
<td>(no quality points) extremely unsatisfactory performance and cannot be used to satisfy degree requirements.</td>
</tr>
<tr>
<td>I</td>
<td>(no quality points) a temporary grade indicating that the student has performed satisfactorily in the course but, due to unforeseen circumstances, has been unable to finish all requirements. An I is not given to enable a student to do additional work to raise a deficient grade. The instructor, in consultation with the student, decides the terms for the removal of the I, including the time limit for removal. If the I is not removed within one calendar year, the grade will be changed to an F. The course will not be counted in the cumulative grade point average until a final grade is assigned. No student may graduate with an I on the record.</td>
<td></td>
</tr>
<tr>
<td>S/NC</td>
<td>(carries credit hours, but no quality points) S is equivalent to a grade of B or better, and NC means no credit earned. A grade of Satisfactory/No Credit is allowed only where indicated in the course description in the Graduate Catalog. The number of Satisfactory/No Credit courses in a student’s program is limited to one-fourth of the total credit hours required.</td>
<td></td>
</tr>
<tr>
<td>P/NP</td>
<td>(carries credit hours, but no quality points) P indicates progress toward completion of a thesis or dissertation. NP indicates no progress or inadequate progress.</td>
<td></td>
</tr>
<tr>
<td>W</td>
<td>(carries no credit hours or quality points) indicates that the student officially withdrew from the course.</td>
<td></td>
</tr>
</tbody>
</table>

## Course Evaluation

You will be invited to evaluate the course at the end of the term. Please participate in this valuable process. I also invite your comments throughout the course and read all comments, suggestions, and recommendations.

## BSIS Program Outcomes

It is our vision to provide a quality educational program, and for students to have the very best educational experience possible. By the end of their time in the BSIS program, each student should be able to:

1. Understand and communicate information ethics and interdisciplinary approaches to information.
2. Demonstrate a combination of technical and professional skills for a variety of application domains.
3. Apply socio-technical, and humanistic perspectives to implement/create information science artifacts for diverse settings.
4. Evaluate, implement and manage information technologies, systems, sources, and services to enhance user experiences.
5. Plan life-long learning skills to respond to challenges facing information professionals in a culturally diverse and dynamic global society.

Assignments: Descriptions, Due Dates, and Program Outcomes

Homework assignments for this course meet BSIS Program Outcome #2: *Demonstrate a combination of technical and professional skills for a variety of application domains.*

This table provides a brief summary of assignment names, due dates, and grade distribution.

<table>
<thead>
<tr>
<th>Homework (50% of Grade)</th>
<th>Value</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Homework 1</strong>: Standard and third party libraries, virtual environments, and first GitHub submission.</td>
<td>100</td>
<td>Mon, Jan 27</td>
</tr>
<tr>
<td><strong>Homework 2</strong>: String processing, regular expressions.</td>
<td>100</td>
<td>Mon, Feb 03</td>
</tr>
<tr>
<td><strong>Homework 3</strong>: Numeric types, Mathematical Operators, Booleans, Dates and Times</td>
<td>100</td>
<td>Mon, Feb 10</td>
</tr>
<tr>
<td><strong>Homework 4</strong>: Conditional Logic, Flow Control, Containers</td>
<td>100</td>
<td>Mon, Feb 17</td>
</tr>
<tr>
<td><strong>Homework 5</strong>: Files, Error Handling, Logging</td>
<td>100</td>
<td>Mon, Feb 24</td>
</tr>
<tr>
<td><strong>Homework 6</strong>: Functions, Recursion, Classes</td>
<td>100</td>
<td>Wed, Mar 04</td>
</tr>
<tr>
<td><strong>Homework 7</strong>: Processing XML</td>
<td>100</td>
<td>Wed, Mar 11</td>
</tr>
<tr>
<td><strong>Homework 8</strong>: Web Scraping and APIs</td>
<td>100</td>
<td>Mon, Mar 30</td>
</tr>
<tr>
<td><strong>Homework 9</strong>: Dataframes and Arrays</td>
<td>100</td>
<td>Mon, Apr 06</td>
</tr>
<tr>
<td><strong>Homework 10</strong>: Plotting &amp; Visualization</td>
<td>100</td>
<td>Mon, Apr 20</td>
</tr>
</tbody>
</table>

01/06/20
### Exams (Weighted for 20% of Grade)

<table>
<thead>
<tr>
<th>Exam</th>
<th>Value</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exam 1</td>
<td>100</td>
<td>Wed, Feb 26</td>
</tr>
<tr>
<td>Exam 2</td>
<td>100</td>
<td>Mon, Apr 13</td>
</tr>
</tbody>
</table>

### Programming Project in Lieu of Final (Weighted for 10% of Grade)

<table>
<thead>
<tr>
<th>Project</th>
<th>Value</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming Project in Lieu of Final</td>
<td>100</td>
<td>Mon, May 4</td>
</tr>
</tbody>
</table>

### COMPLETE SCHEDULE

<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wed, Jan 08</td>
<td><strong>Syllabus Overview &amp; Intro to Python</strong></td>
</tr>
<tr>
<td></td>
<td>• <em>Homework 1 posted to Canvas.</em></td>
</tr>
<tr>
<td></td>
<td>• Review syllabus.</td>
</tr>
<tr>
<td></td>
<td>• Broad discussion of languages.</td>
</tr>
<tr>
<td></td>
<td>• Go over installing Python and IDE.</td>
</tr>
<tr>
<td></td>
<td>• Basic Python syntax: whitespace, print statements, commenting.</td>
</tr>
<tr>
<td></td>
<td>• Running a script</td>
</tr>
<tr>
<td>Mon, Jan 13</td>
<td><strong>Libraries, Packages, &amp; Virtual Environments</strong></td>
</tr>
<tr>
<td></td>
<td>• Check in with progress on installing Python.</td>
</tr>
<tr>
<td></td>
<td>• Standard versus third party libraries.</td>
</tr>
<tr>
<td></td>
<td>• Installing libraries.</td>
</tr>
<tr>
<td></td>
<td>• How and why to implement packages.</td>
</tr>
<tr>
<td></td>
<td>• main program vs imported module</td>
</tr>
<tr>
<td></td>
<td>• Purpose of a virtual environment.</td>
</tr>
<tr>
<td></td>
<td>• How to implement virtual environments.</td>
</tr>
<tr>
<td>Date</td>
<td>Topic</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>Wed, Jan 15</td>
<td>Intro to Version Control &amp; GitHub</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Mon, Jan 20</td>
<td>No class. Martin Luther King holiday.</td>
</tr>
<tr>
<td>Wed, Jan 22</td>
<td>Strings Part I</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Mon, Jan 27</td>
<td>Strings Part II</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Wed, Jan 29  **Dates and Times**
- datetime and time modules
- timezone naive vs aware objects
- available types
- difference between points in time
- operators
- strings to datetime objects and vice versa
- Getting current date / time from system.
- Building time delays into scripts.
- Meaning of epoch time

Mon, Feb 03  **Numeric Types, Comparison Operators, Booleans**
- *Homework 2 due before start of class.*
- *Homework 3 posted to Canvas.*
- ints versus floats
- mathematical operators
- comparison operators
- testing for type and changing type
- None types
- Boolean types
- Boolean logic

Wed, Feb 05  **Conditional Logic & Flow Control**
- diagramming
- review evaluating logical expressions
- if/elif/else
- for & range()
- while & while true
- break
- continue
- pass
Mon, Feb 10  Containers: Lists, Dictionaries, Tuples, Sets
   • Homework 3 due before start of class.
   • Homework 4 posted to Canvas.
   • dictionaries
   • lists
   • tuples
   • sets & uses of set logic
   • slicing
   • iteration: for loops, range(), enumerate(), iteritems(), list comprehension
   • methods to add, remove, sort

Wed, Feb 12  Files & Directories + Strings Part III
   • Extending things we learned about strings to working with files
   • os module
   • csv module
   • Accessing directories and files
   • File handlers, opening/closing files, modes
   • Reading entire file to memory vs line by line (and why you would choose one or the other)
   • Nuances of reading/writing as binary vs text

Mon, Feb 17  Error Handling & Logging
   • Homework 4 due before start of class.
   • Homework 5 posted to Canvas.
   • Mechanics of exceptions
   • Built in exceptions
   • Custom Exceptions
   • logging module
   • Combining exception handling and logging
   • Constructing meaningful log file format
Wed, Feb 19  Functions + Classes
- defining functions
- keyword arguments, concepts of reference & value
- returning a value or multiple values
- recursion
- how and why of classes
- objects vs instances
- scope
- name spaces
- attributes
- functions

Mon, Feb 24  Exam 1 Review Day
- Homework 5 due before start of class.
- Homework 6 posted to Canvas.

Wed, Feb 26  Exam 1

Mon, Mar 02  Python & XML Part I - XML Refresher
- Content standards versus data structure standard / metadata standard
- Basic syntax.
- Conceptual structure.
- Comments
- Elements & Attributes
- Well formed XML
- Namespaces
- Schema & Validation
- Document Type Definition
- Comparison to structure/format of JSON.
Wed, Mar 4

**Python & XML Part II - Parsing & Writing XML + Quick JSON Tutorial**
- **Homework 6 Due at start of class.**
- **Homework 7 posted to Canvas.**
- Libraries: Beautiful Soup, ElementTree, lxml, xml.dom
- Notes on character encoding.
- General strategies for search. Methods for searching based on library.
- Brief intro to xpath & varying levels of xpath support for different modules.
- pretty printing
- Parsing XML with namespaces
- Parsing attribute values & searching based on attribute
- Building XML documents with ElementTree
- Comparing module parsing speeds.
- Strategies for dealing with huge files.
- Quick intro to json module

Mon, Mar 09

**Python & HTTP - Web Servers & HTTP Refresher**
- IP Addresses and DNS
- Understanding web servers & basic server system architectures
- HTTP transaction processes
- requests: methods, headers, body/payload
- responses: response codes, headers, body.
- HTTP vs HTTPS. What’s a certificate?
- First notes on trouble shooting
<table>
<thead>
<tr>
<th>Date</th>
<th>Subject</th>
<th>Details</th>
</tr>
</thead>
</table>
| Wed, Mar 11 | **Web Scraping Part I - requests Library & APIs.** | - *Homework 7 Due before start of class.*  
- constructing requests  
- different methods  
- accessing headers  
- writing response content to file  
- Defining APIs & common practices.  
- APIs: Reading response content as XML or JSON.  
- Using what we’ve learned about conditional logic, error handling, and logging to robustly respond to different error codes.  
- Beautiful Soup for parsing HTML |
| Mon, Mar 16 | **No Class - Spring Break**                   |                                                                                                  |
| Wed, Mar 18 | **No Class - Spring Break**                   |                                                                                                  |
| Mon, Mar 23 | **Recap & Web Scraping Part 2 - Selenium**   | - *Homework 8 posted to Canvas.*  
- Review HTTP / web scraping part I.  
- Installing Selenium and correct version of chromedriver.  
- Use cases for Selenium instead of requests.  
- Demo Selenium and parsing with Beautiful Soup. |
Wed, Mar 25  **Pandas Dataframes**
- Use cases & basics
- Installing with pip
- Brief note on conda/miniconda.
- Creating dataframes from file and from other containers such as lists, dictionaries.
- Properties, describing, shape, head
- accessing elements
- Filtering
- Aggregate functions
- slicing & subsetting
- merging & joining
- date/time functions

Mon, Mar 30  **Numpy Arrays**
- *Homework 8 due before start of class.*
- *Homework 9 posted to Canvas.*
- Use cases & basics
- creating arrays from file or other python containers
- Indexing
- slicing
- reshaping
- copying
- joining & splitting

Wed, April 01  **Plotting & Visualization Part I**
- Installing & configuring matplotlib + dependencies.
- Creating first basic plots.
- Various workflows for creating plots starting from lists, dictionaries, arrays, dataframes.
- customizing plot appearance: labels, colors, axes, figure size, bin sizes, etc.
Mon, Apr 06  | **Plotting & Visualization Part II**
| • *Homework 9 due before start of class.*
| • *Homework 10 posted to Canvas.*
| • More advanced statistical graphics using seaborn.
| • scatter plots, estimations, categorical plots
| • multi-figure plots
| • customizing plot appearance: labels, colors, axes, figure size, bin sizes, etc.

Wed, Apr 08  | **Exam 2 Review Day**

Mon, Apr 13  | **Exam 2**

Wed, Apr 15  | **Advanced Topics: Text Analysis Demo with NLTK**
(May be removed from schedule if other class days are canceled due to inclement weather)
| • Installing NLTK
| • Downloading NLTK datasets
| • Preprocessing & text cleaning workflows:
  ◦ case normalization
  ◦ punctuation & number removal
  ◦ stopword removal
  ◦ lemmatization
  ◦ custom cleaning steps with regular expressions
| • Term frequency reports
| • Word cloud visualizations
Mon, Apr 20  Advanced Topics: Social Network Analysis Demo  
(May be removed from schedule if other class days are canceled due to inclement weather)
  • Homework 10 due before start of class.
  • Brief overview of theory.
  • networkx
  • Workflows for restructuring data for import into more sophisticated, dedicated network visualization tools such as Cytoscape.

Wed, Apr 22  Advanced Topics: Python & Linux  
(May be removed from schedule if other class days are canceled due to inclement weather)
  • Brief Linux / Apache refresher
  • Automating Python scripts with crontab.
  • Introducing web application frameworks (Flask, Django).

--

DISCLAIMER

Please be aware revisions may be made to this syllabus over the course of the semester, and as such, the content contained within may be subject to change.