Computational Reasoning 2:

Representations of Data

INFO 2201; Spring 2017

Lecture: Monday, Wednesday, Friday 15:00–15:50; Humanities 1B80
Lab 021: Monday, 13:00–13:50; Armory 201
Lab 031: Monday, 14:00–14:50; Armory 201

Assistant Professor Brian Keegan
E-mail: brian.keegan@colorado.edu
Office: ENVD 201
Office hours: Monday 16:00–18:00 or by appointment

Course Description

This course surveys techniques for representing data and expressing relationships among data. It will introduce fundamentals of algorithm analysis, review differences between fundamental data structures, implement classic algorithms, and cover methods for querying relational databases.

Learning Objectives and Course Design

The course consists of four modules. The first module on *Fundamentals* focuses on developing familiarity with Python basics such as object-oriented programming, recursion, and algorithmic complexity. The second module on *Structures* introduces several basic data structures used across the computer and information sciences as well as syntax for manipulating elementary data structures in Python. The third module on *Algorithms* unit explores classic algorithms for manipulating and analyzing more advanced data structures like trees and graphs. The fourth module on *Tables* covers the essentials for querying and manipulating tabular data. By the end of the semester, students will be able to evaluate the trade-offs in different basic data structures, implement classic algorithms for manipulating and analyzing data, and query tabular data using SQL syntax.

Class will meet three times per week on Monday, Wednesday, and Friday from 15:00 to 15:50 (3:00pm to 3:50pm) in Humanities 1B80 as well as a required laboratory section on Monday afternoon before class. The format of each class will vary between lectures and tutorials depending on the learning objectives. In addition to weekly participation and peer grading, students will complete two kinds of assignments: (1) weekly lab assignments and (2) Friday quizzes. There are no exams or final projects.

Prerequisites

Students should have completed an introductory course like INFO 1201, CSCI 1300/1310, or ATLS 1710 before enrolling in INFO 2201. If you have questions about these prerequisites, please email me.

Requirements

Students' regular and sustained participation in all class activities as well as punctual and thorough completion of assignments are essential. If you need to be excused from attending a class session or need an extension to an assignment, please notify me via email at least 24 hours in advance.

Course Website and Materials
There are two "textbooks" required for the class, both available via `http://computationaltales.blogspot.com/p/book.html#ComputationalFairyTales` (but not the CU Bookstore).

- Kubica, J. (2012). *Computational Fairy Tales*
- Kubica, J. (2016). *The CS Detective: An Algorithmic Tale of Crime, Conspiracy, and Computation*

Additional readings, tutorials, and other material will be made available through Desire2Learn (D2L):

https://learn.colorado.edu/d2l/home/190526

Once the semester begins, this PDF version of the syllabus will be revised infrequently and any revised requirements will be posted as announcements and updated course schedule to D2L. The instructor reserves the right to make superseding changes to the course's schedule, evaluation criteria, policies, *etc.* outlined below by making announcements via D2L, so please check D2L regularly.

Statistical Computing
You will need to use statistical computing software in this class. We will be using a combination of the PyCharm Edu interactive development environment (IDE) and Jupyter notebooks written in Python 3 for all in-class examples and lab assignments. (We may also get early access to a very exciting collaborative coding learning platform called Elice later in the term) I recommend using the Anaconda distribution of Python 3.5 (or above) to provide all of these programs and other libraries. We will discuss how to download and use the Anaconda package on your local machine during the first week of class. If students cannot bring a laptop to class, they should email me to work out another arrangement.

Evaluation
Your final grade for the course will be based on my evaluation of the three kinds of assignments as well as your general participation within discussions during class and online.

Lab Assignments (45%) Weekly Lab Assignments are intended to develop students' skill, intuition, and confidence writing Python code to understand the trade-offs between different data structures and algorithms. The Lab Assignments will primarily be drawn from the Kubica books assigned above. Each of these 15 weekly Lab Assignments will be worth 3% of the final grade (45% cumulative). Lab Assignments will be due on Wednesday before class and will be evaluated by a random anonymous peer on their completeness, use of good documentation practice, and clarity in addition to their correctness. Students should be prepared to explain their implementation in a 1-on-1 code review if I request it. *In the absence of an approved excuse, late submissions will be docked 33% of their value for every 24 hours elapsed since the deadline*: late assignments submitted after Saturday will lose all credit.

Peer Grading (15%) Peer Grading will expose students to each other's diverse problem-solving approaches and socialize students into supportive code reviewing practices. Each of the 15 weekly Lab Assignments will be graded anonymously by other students in class after submission. The primary foci of peer grading are (1) ensuring the assignment is complete and passes all unit tests, (2) ensuring that Lab Assignments follow good coding style and documentation practice, and (3) on providing feedback and suggestions about alternative approaches, unanticipated unit tests, *etc.* Each on-time and substantive Peer Grade will be worth 1% of the final grade (15% cumulative). Peer Grades will be due on Friday before lecture and late Peer Grade submissions will forfeit this credit. To avoid abuse of this system, submitting students will be able to flag poor peer grading responses for my attention and I will also randomly spot-check submitted peer grades and deduct from the grader's Peer Grading credit if they fail to provide substantive feedback. Peer Graders will also be able to nominate creative Lab Assignments as well as identify unique unit tests for recognition during lecture.

Weekly Quizzes (24%) Students will complete an approximately 15 minute Weekly Quiz at the start of each Friday lecture. These quizzes will provide important and timely feedback for both the instructor and the students about the difficulty of materials covered in the previous two sessions. Each quiz will be worth 2% of the final grade but only the 12 highest Weekly Quiz scores will be calculated in the final grade, meaning students can miss three quizzes with no penalty. *There will be absolutely no make-up opportunities for missed quizzes beyond these "three strikes".*

Participation (16%) Participation will be evaluated in combination with attendance. Students' participation across the three lectures and lab section each week will be worth approximately 1% of the final grade (16% cumulative). Perfect attendance without engagement will be penalized as much or more than infrequent attendance with active engagement. Students demonstrating consistent attendance, high preparation, positive attitude, novel perspectives, and persuasive arguments will earn the most credit. If you have a disability, anxiety, or other issues that limit your ability to actively participate in a lecture format, please contact me to discuss alternative participation methods. *In the absence of an approved excuse, there will be no opportunities to make up for missed participation.*

Weekly Routine

The class has several moving pieces spread across multiple days for each topic, but will follow a regular routine throughout the term that students should anticipate. I have adapted this framework from Professor Ricarose Roque's course design for INFO 1201. Each week will introduce a new concept in lecture on Monday and dive deeper on this concept during Wednesday's lecture. The Friday lecture will include a quiz on those concepts, a discussion about difficulties with the concepts, and then introduce the lab assignment. Students will work on the lab assignment over the weekend and during the following Monday lab sessions before their following Wednesday deadline. Students will be receive their Peer Grading assignment after lecture on that Wednesday and will submit the Peer Grading score on Friday before lecture.

Activity	Monday	Wednesday	Friday
Theme	Introduction	Re-inforcement	Reflection
Lecture	Outline weekly goals, introduce new concepts, motivate their relevance	Reinforce concepts with additional details, working through applications	Quiz and discussion followed by weekly lab assignment introduction
Lab Assignment	Use sections to finish previous Lab Assignment and answer questions	Previous Lab Assignment is due before lecture	Next Lab Assignment and its grading rubric introduced during lecture
Peer Grading	Nothing	Receive peer's previous Lab Assignment after lecture, begin peer grading	Submit scores and nominate solutions/unit tests before lecture
Weekly Quiz	Nothing	Prepare with example problems in supporting materials	Quiz at start of class; reflection and feedback in lecture

Course Policies

In-Class Confidentiality

The success of this class depends on participants feeling comfortable sharing questions, ideas, concerns, and confusions about lab assignments, peer grading, research or entrepreneurial work-in-progress, and their personal experiences. These assignments and discussions should be considered confidential and generally should not be discussed outside of class. You may read, comment, and run on classmates' writing, code, and other class-related content for the sole purpose of use within this class. However, you may not use, run, copy, perform, display, distribute, modify, translate, or create derivative works of other students' work outside of this class without their expressed written consent or formal license. Furthermore, you may not create any audio, video, or other records during class time nor may you publicly share comments made in class attributable to another person's identity without that person's permission.

Faculty Interaction

In addition to teaching this class, I also (1) manage a research program; (2) advise students; (3) perform service for the academic community; and (4) live my life as a private citizen. Interestingly, my middle name is Christopher, my favorite band is Brand New, I played the cello in high school, and my first job after college was bartending. I will check e-mail between 8:00 and 18:00 on non-holiday business days and try to respond within 24 hours. I welcome online or offline interactions outside of class, however these are not appropriate spaces for discussing class matters. E-mailing me or coming to my office hours are the best ways to get feedback outside of class.

Accommodations for Disabilities
I am committed to providing everyone the support and services needed to participate in this course. If you qualify for accommodations because of a disability, please submit to your professor a letter from Disability Services in a timely manner (for exam accommodations provide your letter at least one week prior to the exam) so that your needs can be addressed. Disability Services determines accommodations based on documented disabilities. Contact Disability Services at 303-492-8671 or by e-mail at dsinfo@colorado.edu. If you have a temporary medical condition or injury, see Temporary Medical Conditions guidelines under Quick Links at Disability Services website and discuss your needs with me.

Religious Observance
Campus policy regarding religious observances requires that faculty make every effort to deal reasonably and fairly with all students who, because of religious obligations, have conflicts with scheduled exams, assignments or required assignments/attendance. If this applies to you, please email me directly as soon as possible at the beginning of the term.

Classroom Behavior
Students and faculty each have responsibility for maintaining an appropriate learning environment. Those who fail to adhere to such behavioral standards may be subject to discipline. Professional courtesy and sensitivity are especially important with respect to individuals and topics dealing with differences of race, color, culture, religion, creed, politics, veteran's status, sexual orientation, gender, gender identity and gender expression, age, ability, and nationality. Class rosters are provided to the instructor with the student's legal name. I will gladly honor your request to address you by an alternate name or gender pronoun. Please advise me of this preference early in the semester so that I may make appropriate changes. For more information, see the policies on class behavior and the student code.

Harassment and Discrimination
The University of Colorado Boulder (CU Boulder) is committed to maintaining a positive learning, working, and living environment. CU Boulder will not tolerate acts of sexual misconduct, discrimination, harassment or related retaliation against or by any employee or student. CU's Sexual Misconduct Policy prohibits sexual assault, sexual exploitation, sexual harassment, intimate partner abuse (dating or domestic violence), stalking or related retaliation. CU Boulder's Discrimination and Harassment Policy prohibits discrimination, harassment or related retaliation based on race, color, national origin, sex, pregnancy, age, disability, creed, religion, sexual orientation, gender identity, gender expression, veteran status, political affiliation or political philosophy. Individuals who believe they have been subject to misconduct under either policy should contact the Office of Institutional Equity and Compliance (OIEC) at 303-492-2127. Information about the OIEC, the above referenced policies, and the campus resources available to assist individuals regarding sexual misconduct, discrimination, harassment or related retaliation can be found at the OIEC website.

Honor Code
All students enrolled in a University of Colorado Boulder course are responsible for knowing and adhering to the academic integrity policy of the institution. Violations of the policy may include: plagiarism, cheating, fabrication, lying, bribery, threat, unauthorized access, clicker fraud, resubmission, and aiding academic dishonesty. All incidents of academic misconduct will be reported to the Honor Code Council (honor@colorado.edu; 303-735-2273). Students who are found responsible for violating the academic integrity policy will be subject to nonacademic sanctions from the Honor Code Council as well as academic sanctions from the faculty member. Additional information can be found at honorcode.colorado.edu.

Acknowledgements

The design and format of this course borrows from similar courses and textbooks. You are not required to buy these books or classes, but these can be very helpful resources if you find yourself struggling.

- Textbooks
  - Bhargava, A. (2016). *Grokking Algorithms: An illustrated guide for programmers and other curious people*. Manning Publications, Shelter Island, 1st edition
  - Goodrich, M. T., Tamassia, R., and Goldwasser, M. H. (2013). *Data Structures and Algorithms in Python*. Wiley, Hoboken, NJ, 1st edition
  - Guttag, J. (2016). *Introduction to Computation and Programming Using Python*. MIT Press, 2nd edition
  - Heineman, G. T., Pollice, G., and Selkow, S. (2016). *Algorithms in a Nutshell: A Practical Guide*. O'Reilly Media, 2nd edition
  - Hetland, M. L. (2014). *Python Algorithms: Mastering Basic Algorithms in the Python Language*. Apress, 2nd edition
  - Karumanchi, N. (2015). *Data Structure and Algorithmic Thinking with Python: Data Structure and Algorithmic Puzzles*. CareerMonk Publications, 1st edition
  - Lambert, K. A. (2014). *Fundamentals of Python: Data Structures*. Cengage, 1st edition
  - Lee, K. D. and Hubbard, S. (2015). *Data Structures and Algorithms with Python*. Springer, New York, 2015 edition. `http://www.springer.com/us/book/9783319130712`
  - Miller, B. N. and Ranum, D. L. (2013). *Problem Solving with Algorithms and Data Structures Using Python*. Franklin, Beedle & Associates, Sherwood, Oregon, 3rd edition. `http://interactivepython.org/runestone/static/pythonds/index.html`
  - Pilgrim, M. and Willison, S. (2009). *Dive Into Python 3*. Springer, 2nd edition. `http://www.diveintopython3.net/`
  - Stephens, R. (2013). *Essential Algorithms: A Practical Approach to Computer Algorithms*. Wiley, 1st edition

- Online courses
  - Balazs, H. (2016). Algorithms and Data Structures in Python. `https://www.udemy.com/algorithms-and-data-structures-in-python/`
  - Kane, D., Kulikov, A. S., Levin, M., Pevzner, P., and Rhodes, N. (2016). Algorithmic Toolbox. `https://www.coursera.org/learn/algorithmic-toolbox/home/welcome`
  - Nakhleh, L., Rixner, S., and Warren, J. (2016). Algorithmic Thinking (Part 1) – Rice University. `https://www.coursera.org/learn/algorithmic-thinking-1`
  - Portilla, J. (2016). Python for Data Structures, Algorithms, and Interviews. `https://www.udemy.com/python-for-data-structures-algorithms-and-interviews/`
  - Roughgarden, T. (2016). Online Courses. `http://theory.stanford.edu/~tim/videos.html`
  - Sentance, S. and McNicol, A. (2016). Python School. `https://pythonschool.net/`

This syllabus was typeset in LaTeX using ShareLaTeX with the fbb/Bembo font and is derived from the `memoir` styles adapted by Kieran Healy and Benjamin 'Mako' Hill.

Course Outline

I may adjust the list of readings or the schedule as needed throughout the semester, so please consult the schedule online at Desire2Learn for the most up-to-date information. We will meet in class a total of 44 times over the course of the the 16-week semester.

Week 1 – Fundamentals: Review

Wednesday, January 18; Friday, January 20

Configure computing environment and review concepts from INFO 1201.

Objectives

- Setup and introduce PyCharm Edu and Jupyter Notebook environments.
- Review concepts like loops, conditions, and functions.
- Introduce documentation standards and peer grading standards.

Assignments

Lab 1 – due Wednesday, January 25.

Week 2 – Fundamentals: Objects

Monday, January 23; Wednesday, January 25; Friday, January 27

Introduction to object-oriented programming within Python using classes.

Objectives

- Initializing and interacting with classes.
- Defining functions, attributes, and methods.
- Inheritance of properties across objects.

Assignments

Lab 2 – due Wednesday, February 1

Week 3 – Fundamentals: Recursion

Monday, January 30; Wednesday, February 1; Friday, February 3

Developing intuitions with recursive functions and identifying cases to apply them.

Objectives

- Defining functions calling themselves.
- Identifying base and recursive cases.
- Pushing and popping items from a call stack.

Assignments

Lab 3 – due Wednesday, February 8.

Week 4 – Fundamentals: Complexity

Monday, February 6; Wednesday, February 8; Friday, February 10

Measuring the time complexity of functions and reviewing efficiency of different design patterns.

Objectives

- Demonstrate importance of time complexity and asymptotic analysis.
- Review Big-O notation and complexity classes.
- Identify primitive operations affecting algorithm complexity.

Assignments
Lab 4 – due Wednesday, February 15.

Week 5 – Structures: Linear

Monday, February 13; Wednesday, February 15; Friday, February 17
Introduction to linear data structures like arrays, matrices, and linked lists.

Objectives
- Comparing 1-dimensional arrays, linked lists, and matrices.
- Inserting and deleting values in linear data structures.
- Indexing and swapping values in linear data structures.

Assignments
Lab 5 – due Wednesday, February 22.

Week 6 – Structures: Unique

Monday, February 20; Wednesday, February 22; Friday, February 24
Introduction to data collections where values are indexed by unique keys.

Objectives
- Operations, applications, and limitations of sets and dictionaries.
- Implementing hashing functions to generate key-value pairs.
- Comparing array and hashing implementations of dictionaries.

Assignments
Lab 6 – due Wednesday, March 1.

Week 7 – Structures: Trees

Monday, February 27; Wednesday, March 1; Friday, March 3
Accessing and analyzing hierarchical properties of trees.

NOTE: *Professor Keegan will be out-of-town attending the CSCW 2017 conference Monday and Wednesday. There will be a guest to cover the Monday and Wednesday lectures and labs.*

Objectives
- Implementing and altering trees with basic data structures.
- Methods for traversing and manipulating trees.
- Comparing depth- and breadth-first search algorithms.

Assignments
Lab 7 – due Wednesday, March 8.

Week 8 – Algorithms: Strings

Monday, March 6; Wednesday, March 8; Friday, March 10
Using algorithms to manipulate, compare, and find strings.

Objectives
- Implementing algorithms to manipulate and format strings.
- Using algorithms to measure string distance and similarity.
- Fundamentals and limitations of regular expressions for finding sub-strings.

Assignments
Lab 8 – due Wednesday, March 15.

Week 9 – Algorithms: Sorting

Monday, March 13; Wednesday, March 15; Friday, March 17
Using algorithms to efficiently order values.

Objectives
- Implementing and comparing exchange, selection, insertion, and merge sorting.

Assignments
Lab 9 – due Wednesday, March 22.

Week 10 – Algorithms: Graphs

Monday, March 20; Wednesday, March 22; Friday, March 24
Accessing and analyzing structural properties of graphs.

Objectives
- Types and representations of graphs.
- Algorithms for graph traversal and manipulation.
- Computing basic network science metrics.

Assignments
Lab 10 – due Wednesday, April 5.

Week 11 – Spring Break

Monday, March 27; Wednesday, March 29; Friday, March 31
Adding new objects into long-term memory.

Week 12 – Databases: `pandas`

Monday, April 3; Wednesday, April 5; Friday, April 7
Using a panel data library to reshape, manipulate, and visualize tabular data.

Objectives
- Loading data from CSV files into DataFrames.
- Converting between data shapes by melting, pivoting, and stacking.
- Visualizing tabular data with `matplotlib` and `seaborn`.

Assignments
Lab 12 – due Wednesday, April 12.

Week 13 – Databases: Selecting

Monday, April 10; Wednesday April 12; Friday, April 14
Applying basic SQL syntax for selecting, filtering, sorting, and counting data.

Objectives
- Selecting columns and sorting results.
- Filtering results with `where` syntax.
- Summarizing results with functions and aliases.

Assignments
Lab 13 – due Wednesday, April 19.

Week 14 – Databases: Joining

Monday, April 17; Wednesday, April 19; Friday, April 21
Combining data from two different tables and dealing with missing or repeated data.

Objectives
- Joining on single and multiple keys.
- Comparing left, right, inner, and outer joins.
- Managing joins of multiple tables and duplicate columns.

Assignments
Lab 14 – due Wednesday, April 26.

Week 15 – Databases: Grouping

Monday, April 24; Wednesday, April 26; Friday, April 28
Dealing with repeated observations and computing aggregated values.

Objectives
- Identifying and grouping repeated data values.
- Specifying aggregation functions to summarize grouped values.
- Using having clause to subset group by results.

Assignments
Lab 15 – due Wednesday, May 3.

Week 16 – Databases: Exotic

Monday, May 1; Wednesday, May 3; Friday, May 5
Relational databases alternatives like NoSQL and graph databases as well as map-reduce frameworks.

Objectives
- Storing and querying JSON data in a NoSQL database like MongoDB.
- Querying complex networks in a graph database like Neo4j using Cypher.
- Parallelizing data mining with map-reduce frameworks like Hadoop.

Assignments
Lab 16 – due Wednesday, May 10.

Module	**Week**	**Dates**	**Theme**	**Topics**
Fundamentals	1	Jan 18 – Jan 20	Review	loops, conditions, functions
	2	Jan 23 – Jan 27	Objects	classes, namespaces, inheritance
	3	Jan 30 – Feb 3	Recursion	base case, head/tail, stack diagram
	4	Feb 6 – Feb 10	Complexity	Big-O, asymptotic behavior
Structures	5	Feb 13 – Feb 17	Linear	arrays, linked lists, matrices
	6	Feb 20 – Feb 24	Unique	sets, hashes, dictionaries
	7	Feb 27 – Mar 3	Trees	construction, traversal
Algorithms	8	Mar 6 – Mar 10	Strings	similarity, searching, matching
	9	Mar 13 – Mar 17	Sorting	insertion, bubble, merge
	10	Mar 20 – Mar 24	Graphs	formats, search, connectivity
	11	Mar 27 – Mar 31	Spring Break	
Tables	12	Apr 3 – Apr 7	`pandas`	dataframes, indexing, plotting
	13	Apr 10 – Apr 14	Selecting	aliases, order, where, functions
	14	Apr 17 – Apr 21	Joining	keys, inner, outer, left, right
	15	Apr 24 – Apr 28	Grouping	groupby, aggregation, having
	16	May 1 – May 5	Exotic	NoSQL, graph, map-reduce