CS 104 - Topics in Introductory Programming: Social Inquiry
Spring 2017

Professor: Sorelle Friedler
sorelle@cs.haverford.edu

TTh 1 - 2:30, KINSC H110
Labs: W 9:30 - 10:30 or W 10:30 - 11:30, KINSC H110
Course website: http://www.cs.haverford.edu/courses/cmsc104

Topics in Introductory Programming is designed to give a general introduction to programming as related to data analysis across many fields. Students will be introduced to standard introductory programming imperative and object oriented techniques as well as data structures necessary to create efficient and understandable algorithmic solutions to problems. Data for analysis will be drawn from a single discipline that will vary per semester, forming a theme for topical study. Topical investigations will include the ethics of data use in that field, how data is commonly generated and used, and implementation of important discipline-specific algorithms.

Lab: Includes a weekly programming lab section. Class sessions will also be taught in a computer lab.

Prerequisites: none
This course serves as a prerequisite for CS 107. Students who have taken this course *may not* receive credit for CS 105 and students who have taken CS 105, CS 106, CS 107, or Bryn Mawr's CS 110 *may not* take this course. Students with past experience in programming may be able to place out of this course.

Enrollment Limit: 24. The course will be divided into two lab sessions of at most 12 each.

Lab Instructor: Suzanne Lindell

TAs: Katherine Lee and Dylan Emery. Office hours TBD (see the class Google Calendar).

Schedule of Topics

This schedule is *tentative*. Labs are due **by 11:59pm on Thursday** in the week listed. Students should expect ***at least* 10 hours of work each week.** For the most up-to-date dates and deadlines see the CS 104 Google Calendar.

Week 1. Introduction. The importance of documentation. File operations and functions.

- Programming topics: functions, parameters, return statements, variables, list basics.
- Data analysis topics: Unix basics (cd, ls, less, ssh), file reading / writing, git basics.
- Ethics: reproducibility, reading and writing documentation.

- **Reading**: Introduction from *the Signal and the Noise* by Nate Silver (on hold in the science library)

Week 2. For loops, lists, and other programming basics.

- Programming topics: for loops, list indices, list length operations, types and type conversion, basic use of libraries.
- Data analysis topics: random number generation.
- Ethics: reproducibility, documentation.

Week 3. Data analysis and files.

- **Lab 1 due**. Covers basics in documentation, comments, files, functions, for loops, and lists. It comes in three parts - *pace yourself!* There are three weeks for this lab and you should be finishing *at least* one part per week.
- Programming topics: if statements, boolean operators, arithmetic operators (including integer division and mod).
- Data analysis topics: csv file parsing, calculation of basic statistics (mean, min, etc.).
- Social inquiry: discussion of NYC Stop and Frisk data.

Week 4. Where does data come from? Creating and storing data.

- Programming topics: dictionaries, combined data structures (e.g., lists of lists).
- Data analysis topics: data integrity and validation, missing data.
- Ethics: analysis of the choices inherent in data creation - categories, missing values, duplicates, etc.
- Social inquiry: discussion of NYC Stop and Frisk data.
- Reading: White House Report: *Big Data: A Report on Algorithmic Systems, Opportunity, and Civil Rights*, May 2016.

Week 5. Visualizing data, using libraries.

- **Lab 2 due**.
- Programming topics: visualization libraries, date / time, tuples, multiple return values.
- Data analysis topics: basics of visualization, histogram bin size choices.

Week 6. Remaining basic programming topics.

- Programming topics: while loops, basic uses of recursion, arrays and matrices (numpy), variables and memory.

Week 7. Review week and midterm.

- **Lab 3 due *Monday* at midnight**.
- Wednesday: Midterm exam during class time.

Week 8. **Spring Break!**

Week 9. Complexity, including environmental impacts of complexity and basics of internet infrastructure.

- Programming and computational topics: complexity (counting steps based on input size, measuring timing), nested for loops.
- Data infrastructure topics: the internet and the cloud.
- Reading: What People Mean When They Talk About 'The Cloud' and The Environmental Toll of a Netflix Binge.
  Optional additional reading: take a look at the rest of a series on internet infrastructure by Ingrid Burrington.

Week 10. Sorting and data de-duplication.

- Programming and computational topics: quicksort (with random pivot), sorting library usage.
- Data analysis topics: data de-duplication methods (dictionaries, sorting, all-pairs) and goals.

Week 11. Graphs and object-oriented programming.

- **Lab 4 (Complexity and de-duplication) due**.
- Programming and computational topics: classes, objects, methods on objects, fields, graph basics (nodes, edges, weights, neighbors, degree, directed vs. undirected).
- Data analysis topics: uses of graphs.

Week 12. Graphs and object-oriented programming.

- Programming and computational topics: adjacency list graph representation, trees, inheritance, interfaces, libraries.
- Social inquiry: networks, human classification, and privacy / re-identification.
- Reading: Simple Demographics Often Identify People Uniquely by Latanya Sweeney, Sections 1-3, 6.
  Optional additional reading: *Sorting Things Out: Classification and Its Consequences* by Geoffrey C. Bowker and Susan Leigh Star (available digitally via the library), the Introduction and Chapter 6.

Week 13. Social network graph analysis measures and algorithms.

- **Lab 5 (Graphs and Object-oriented programming) due.**
- Social inquiry: degree centrality, betweenness centrality, and the shortest path problem (using libraries).

Week 14. Becoming a self-sufficient programmer.

- Programming topics: creating and using scripts and command-line options, reading documentation.
- Data analysis topics: hypothesis formation and experimental design.

- Ethics: reproducibility and reading and writing documentation.
- Reading: How to Hold Algorithms Accountable

Week 15. Review.

- **Lab 6 (Network centrality script) due.**

Programming Topics

- Programming basics:
  - if statements
  - for / while loops
  - variables
  - basic uses of recursion (coding from a self-referential definition)
  - functions, parameters, returns
  - memory
  - types (differences and uses of):
    - strings
    - ints
    - floats
    - booleans
- Basic data structures:
  - arrays
  - lists
  - matrices
  - dictionaries
  - trees (basics only)
- Object-oriented programming basics:
  - classes
  - objects
  - methods on objects
  - fields
  - class variables
  - inheritance (basic)
  - interfaces (basic)
  - libraries (use of)
- Data management and access basics (scripting):
  - Unix basics: cd, ls, less, edit, ^Z, bg, top, gnome-open, eclipse, ssh, scp
  - File reading / writing programmatically
  - Creating and using command-line options

Data Analysis Topics

- files as data containers
  - open, closing, creating, appending
  - file formats (e.g., csv, text, xlsx)
- (if time) working with APIs
- data parsing and formats
  - string, numeric, other
- data integrity and validation
  - duplicate, missing, outlier values
- potential consequences
  - ethical, context, objectivity assumptions
  - hypothesis formation and experimental design
  - raw data and category choice

Data and Infrastructure

- Internet and cloud
  - physical internet
  - environmental impacts
- data representation

Social Inquiry-specific Topics

- data-driven discrimination
- reproducibility of data analysis
- explainability of data analysis
- privacy/(de-/re-)identification
- basic network analysis (degree and betweenness centrality)

Labs and Exams

A general outline of the labs is given below. Lab starter code will be distributed to the class and can be found at: https://github.com/sorelle/dataintrocourse

1. File basics and basic python programming.
2. File parsing and basic data analysis.
3. More advanced data analysis.
4. Sorting and complexity.
5. Graph data and basic object-oriented programming.
6. Social network graph analysis.

There will be both a midterm and a final exam.

Labs	40%
Midterm	25%
Final Exam	35%

Total grade breakdown

Grades will be awarded based on the number of points earned and according to the percentage breakdowns shown. Students will not be graded on a curve.

Attendance and Participation

Attendance at, and active participation in, all class and lab sessions is expected of all students. Participation will be taken into account in awarding of final grades for students who are "on the edge" between two grades. For example, a student with a B+/A- average and a strong attendance and participation record would receive an A-, while a student with a weak record would receive a B+.

Late work policy

All extensions must be requested **at least 24 hours in advance** of the deadline. Extensions will be granted based on individual circumstances. Work handed in late without a previously granted extension may not be accepted (i.e., may receive zero credit).

Rules and Pet Peeves

- **Be on time.** This includes class, lab, office hours, and appointments.
- **Expect 24 hours before an email response** and read all emails within 24 hours.
- **Attend all classes and labs.**

Collaboration

Please see the full version of the departmental collaboration policy here. A summary is below, but students are encouraged to read the full linked document.

Work done in collaboration should never be copied from another student (e.g., from their computer or from joint work on the board). Work from previous semesters should never be shared with current students, or looked at by students in the current semester, though it is fine to share notes you make about lectures or the textbook. Code and other material should never be copied from another student or outside sources unless permission is explicitly given in advance by your professor and the code is cited.

Learning Accommodations

Haverford College is committed to supporting the learning process for all students. Please contact me as soon as possible if you are having difficulties in the course. There are also many resources on campus available to you as a student, including the Office of Academic Resources (`https://www.haverford.edu/oar/`) and the Office of Access and Disability Services (`https://www.haverford.edu/access-and-disability-services/`). If you think you may need accommodations because of a disability, you should contact Access

and Disability Services at hc-ads@haverford.edu. If you have already been approved to receive academic accommodations and would like to request accommodations in this course because of a disability, please meet with me privately at the beginning of the semester (ideally within the first two weeks) with your verification letter.