W200 Introduction to Data Science Programming – Syllabus

University of California-Berkeley, School of Information
Master of Information & Data Science (MIDS) Program, Spring 2020

Course Developers:
Paul Laskowski: paul@ischool.berkeley.edu
William Chambers
Kay Ashaolu

Course Coordinator:
Rob Foster: fosterrj@berkeley.edu

Section Instructors:

Christopher Llop:	christopher.llop@berkeley.edu	Sec 1: Tue 4:00-5:30pm PT
Gunnar Kleemann:	gunnarkl@berkeley.edu	Sec 2: Wed 6:30-8:00pm PT
Gerald Benoit:	gbenoit@berkeley.edu	Sec 3: Tue 4:00-5:30pm PT, Sec 4: Thu 4:00-5:30pm PT, Sec 5: Sat 8:00-9:30am PT

Teaching Assistants:
Tutor: Mark Barnett: markbarn@berkeley.edu
Reader Sec 1 & 2: Brian Ament: brian.ament@ischool.berkeley.edu
Reader Sec 3, & 4: Christopher Wen: chrishcwen@berkeley.edu
Readers Sec 5: Both

Office Hours:

Mark Barnett:	Monday: 12:30-1:30 pm PT (Homework Review OH) (or by appointment)
Christopher Llop:	Monday: 2:00-3:00 pm PT (or by appointment)
Gerald Benoit:	Tuesday: 3:00-4:00 pm PT (or by appointment)
Gunnar Kleemann:	Wednesday : 8:00-9:00 pm PT (or by appointment)

Prerequisites: There are no prior courses required, but due to the fast pace of the course material, previous experience in a general-purpose programming language is required.

Course Description:
The Python programming language is an increasingly popular tool for the manipulation and analysis of data. This fast-paced course aims to give students the fundamental Python knowledge necessary for more advanced work in data science. The course structure provides students with frequent opportunities to practice writing code, gradually building to an advanced set of skills focused on data science applications. We begin by introducing a range of Python objects and control structures, then build on these with classes and object-oriented programming. A major programming project will reinforce these concepts, give students insight into how a large piece of software is built, and give students experience in managing a full-cycle development project. The last section of the course is devoted to two popular Python packages for data analysis, Numpy and Pandas. The course ends with an exploratory data analysis, in which students apply a script-style of programming to describe and understand a dataset of their own choosing. Aside from Python, the course also spends time on several other technologies that

are fundamental to the modern practice of data science, including use of the command line, Jupyter notebooks, and source control with Git and GitHub.

Learning Objectives:
After completing this course, students will:

- Be able to navigate a file system, manipulate files, and execute programs using a command line interface
- Understand how to manage different versions of a project using Git and how to collaborate with others using Github
- Be fluent in Python syntax and familiar with foundational Python object types
- Be able to design, reason about, and implement algorithms for solving computational problems
- Understand the principles of object-oriented design and the process by which large pieces of software are developed
- Be able to test and effectively debug programs
- Know how to use Python to extract data from different types of files and other sources
- Understand the principles of functional programming
- Know how to read, manipulate, describe, and visualize data using the NumPy and Pandas packages
- Be able to generate an exploratory analysis of a data set using Python
- Be prepared for further programming challenges in more advanced data science courses

Methods of Instruction:
Students log into the Data Science homepage (https://learn.datascience.berkeley.edu/) and then select their course sections. From there, select Coursework to view the *asynchronous* lessons. Each week there are a number of videos, references to required readings, and additional readings/videos. After the asynchronous lessons, students attend an online video chat live section to review the materials and participate in breakout sessions to work with other students on activities. There are a number of weekly activities, projects, and in-class breakouts that require a computer, Python3, and Jupyter. Homework for that unit is assigned after class and due the following week (see weekly assignments below).

Required Textbook and other Readings:

- Lubanovic, B. (Edition 1: 2015 or Edition 2: 2019). *Introducing Python: Modern computing in simple packages*. Sebastopol, CA: O'Reilly Media. [can be viewed using the UC Berkeley library VPN]
  - If using Edition 1 - chapters are noted in reading below as Ed1; if using Edition 2 - chapters are noted in reading below as Ed2
- Goodrich (2016). *Data Structures and Algorithms in Python* (chapter 3). [Available from study.net]
- McKinney, W (2017). *Python for Data Analysis*. Sebastopol, CA: O'Reilly Media. [can be viewed using the UC Berkeley library VPN]

Online Tools:

- Study Net: http://www.study.net
  Study.net has some of the reading materials required for the course.
- GitHub, Git, & Bash: https://github.com; class github: https://github.prod.oc.2u.com/UCB-INFO-PYTHON
  Install and configure Git & Bash.
- Anaconda with Jupyter Notebooks & Python3: https://anaconda.org/anaconda/python
- I School Virtual Campus (ISVC): https://learn.datascience.berkeley.edu/
- Google Group Email: w200-python-2020-spring@googlegroups.com
  Google Group web link
  Use Google Group Email to formally communicate with your instructors and classmates, ask questions, share problems and your successes (but don't share code).

- Instructor group email: mids-python-instructors@googlegroups.com
- Slack: https://ucbischool.slack.com/messages/C5AL99BU6/
  Use Slack to informally communicate with your classmates, ask questions, share problems and your successes (but don't share code).

Calendar:

Week	Date	Topic & Activities
Week 1:	Jan 7-11	**Introduction to Programming, Command Line, Source Code**
	Readings	Lubanovic, B. *Introducing Python: Modern computing in simple packages*. Sebastopol, CA: O'Reilly Media. Ed 1 or Ed 2: chapter 1 Command Line Notes Merge Conflict Resolution
	Homework	Assignments are made available via ISVC github link after the live section class
Week 2:	Jan 14-18	**Starting out with Python**
	Readings	Lubanovic: Ed1 - chapter 2 and beginning of chapter 4, up to but not including the section labeled "cancel with break" [pp. 69-75]. Ed2 - chapters 2, 3, 4, 5 & chapter 6 "Repeat with While" section String Formatting Cookbook
	Homework unit 1	Due 11:59 PM Pacific Time - the day before your live section
Week 3:	Jan 21-25	**Sequence Types & Dictionaries**
	Readings	Lubanovic: Ed 1 - chapter 3. Ed2 - chapters 7 & 8
	Homework unit 2	Due 11:59 PM Pacific Time - the day before your live section
Week 4:	Jan 28-Feb 1	**More About Control and Algorithms**
	Readings	Lubanovic: Ed1 - chapter 4, start with "cancel with break" (p. 75) through "comprehensions" (p. 85) Ed2 - Rest of chapter 6 after "Repeat with While" section; review chapter 7 & 8 comprehensions material
	Homework unit 3	Due 11:59 PM Pacific Time - the day before your live section
Week 5:	Feb 4-8	**Functions**
	Readings	Lubanovic: Ed1 - chapter 4, start with Functions, p. 85. Ed2: chapter 9
	Homework unit 4	Due 11:59 PM Pacific Time - the day before your live section
Week 6:	Feb 11-15	**Complexity**
	Readings	Goodrich et al - chapter 3 (2016) *Data structures* (available from study.net)
	Project 1 start	Project 1 is assigned
	Homework unit 5	Due 11:59 PM Pacific Time - the day before your live section

Week 7:	Feb 18-22	**Classes**
	Readings	Lubanovic: Ed1 - chapter 6 through “Define a Class with class” and next “In self Defense” through “Method Types” Ed2 - chapter 10 the first two sections “What are Objects” & “Simple Objects” and next “In self Defense” through “Method Types”
	Homework unit 6	Due 11:59 PM Pacific Time - the day before your live section
Week 8:	Feb 25-29	**Object-Oriented Programming**
	Readings	Lubanovic: Ed1 - rest of chapter 6: “inheritance” through “In self defense” and “Duck typing” to the end of chapter Ed2 - rest of chapter 10: “inheritance” through “In self defense” and “Duck typing” to the end of chapter “Learn Python The Hard Way” Chapter 43 (linked in github-week 08 folder)
	Project 1 proposal	Due 11:59 PM Pacific Time - the day before your live section
	Homework unit 7	Due 11:59 PM Pacific Time - the day before your live section
	Exam 1 start	Window starts; distributed on ISVC; you have 24 hours to complete from the time that you start the exam
Week 9:	Mar 3-7	**Working with text and binary data**
	Readings	Lubanovic: Ed1 - chapters 7 and 8 through JSON Ed2: chapters 12, 14 and 16 through JSON
	Exam 1	Due before your live section
Week 10:	Mar 10-14	**NumPy**
	Readings	There are no readings assigned this week
	Project 1 presentation	Present your project 1 in your live section. Submit your code and presentations materials by 11:59 PM Pacific Time the day AFTER your live section
Week 11:	Mar 17-21	**Data analysis with Pandas**
	Readings	There are no readings assigned this week
	Project 2 start	Project 2 is assigned
	Homework unit 9	Due 11:59 PM Pacific Time - the day before your live section
Mar 24-28	Spring Break	
Week 12:	Mar 31 - Apr 4	**Plotting and Visualization**
	Readings	There are no readings assigned this week
	Project 2 proposal	Due 11:59 PM Pacific Time - the day before your live section
	Homework unit 10	Due 11:59 PM Pacific Time - the day before your live section

Week 13:	Apr 7-11	**Pandas Aggregation and Group Operation**
	Readings	McKinney, W. (2017). Python for data analysis: Data wrangling with Pandas, NumPy, and IPython. O'Reilly Media, Inc. Chapter 11 (Chapter 10 in 2012 version)
	Exam 2 start	Window starts; distributed on ISVC; you have 48 hours to complete from the time that you start the exam
	Other	A make-up class will be scheduled for the Thursday section
	Homework units 11,12,13	Due 11:59 PM Pacific Time - the day before your live section
Week 14:	Apr 14-18	**Testing**
	Readings	Lubanovic Ed1 - chapter 12: just the section titled "Test with Unitest" Ed2 - ch19: just the section titled "Test with Unitest"
	Project 2 presentation	Present your project 2 in your live section
	Project 2 report	Due 11:59 PM Pacific Time - the day AFTER your live section
	Exam 2	Due by 11:59 PM PST, Monday April 20th

Congratulations!

See Calendar_Spring_2020 for latest updates, if any, to calendar dates.

Course Outline:

1. **Programming Languages, the Command Line, and Version Control (1 lecture)**: The course begins with an overview of programming languages and an introduction to some of the lower-level tools that support the work of a data scientist.

    - Programming Language Characteristics: Interpreted Versus Compiled, Low Level Versus High Level, General Purpose Versus Specialized
    - Using the Command Line
    - Version Control with Git
    - Collaboration with GitHub

2. **Python Objects and Basic Control Structures (5 lectures)**: We continue with a vocabulary-building survey of basic Python syntax, object types, and control structures. These elements are common to virtually all programming applications. Students will gain experience designing algorithms and organizing code logically into functions and modules.

    - Important Python Object Types
    - Iteration and Conditionals
    - Functions
    - The Design of Algorithms
    - Writing and Presenting Code in Jupyter Notebooks
    - Python Modules and Packages
    - Big-O Notation

3. **Classes and Object-Oriented Programming (3 lectures)**: We will spend three weeks discussing classes, as well as the larger practice of object-oriented programming. This part of the course will give students a view into how large production systems are organized and developed. At the end, students will complete a significant coding project that will reinforce their understanding of object-oriented development.

    - Classes and Attributes
    - Class Inheritance
    - Object-Oriented Programming
    - Project 1

4. **Using Python's Packages for Data Analysis (6 lectures):** We introduce the basics of data analysis using Python's system of scientific programming packages. Students learn the common tools that form the basis of the PyData ecosystem and gain experience with programming in a functional style. The final two weeks of the course give students time to work on a final data analysis project, while emphasizing good practices for developers.

    - File Input-Output
    - Text Encoding
    - Common Structure File Formats
    - Functional Programming
    - NumPy Arrays
    - Pandas Series and DataFrames
    - Basic Data Set Manipulation with Pandas
    - Plotting with Matplotlib
    - Descriptive Statistics
    - Test-Driven Development
    - Resources for Further Development
    - Project 2

Grading:

- 10 Weekly Assignments - 40% (4% each)
- 2 Projects - 30% (15% each)
- Midterm Exam - 10%
- Comprehensive Final Exam - 10%
- Participation - 10%

Weekly Assignments:

The weekly assignments are designed to reinforce and extend the programming concepts covered in each live section. A typical assignment consists of several programming exercises of varying difficulty. While some exercises can be completed in a single line of code, others may require students to combine commands in innovative ways, to design their own algorithms, and to navigate common sources of documentation. Students may consult with each other about the assignment but must write their own code and list their collaborators in their submissions. The expectation is that these assignments will take around 10-20 hours to complete each week. Depending on your experience with coding, this time estimate might be more or less. The weekly assignments are due at 11:59 PM PST (or PDT) the day before the next live section (e.g., if you have live section on Tuesday, then the assignments are due the following Monday night at 11:59 PM PST). Weekly assignments will be released via GitHub.

Projects:
There are two large coding projects. The first is an individual project that comes at the end of the discussion of object-oriented programming and allows students to practice designing a multi-class program using best coding practices. The second is a group project that comes at the end of the course and involves the analysis of an actual data set using Python's system of data analysis packages. Further details about the projects will be given during the school term.

Exams:
The midterm and final exams are cumulative. Unlike the weekly assignments, students must do all of their work independently. Both exams include multiple-choice and short-answer questions, as well as short programming tasks, designed to test each student's grasp of important programming concepts. Further details about the exams will be given during the school term.

Participation:
Students are expected to participate in class activities, to contribute to discussions held in live section and on other platforms, to behave professionally towards classmates, and to help maintain a supportive atmosphere for education. Participation scores will be assigned based on these criteria.

Academic Integrity:
Please read UC Berkeley's policies around academic integrity: http://sa.berkeley.edu/conduct/integrity

Avoiding Plagiarism:
Plagiarism is a serious academic offense, and students must take care not to copy code written by others. Beginning students sometimes have trouble identifying exactly when plagiarism takes place. Remember that it is generally fine to search for examples of code (e.g., on forums such as stackexchange). This is a normal part of programming and can help you learn. However, it is important that you understand the code you find and use what you learn to write your own statements. It is okay if a single line of code happens to match an example found on the internet, but you should not copy multiple lines at once. If in doubt, simply document the place you found your example code and ask your instructor for further guidance.

Accommodations:
We make every effort to address the needs of students with physical, medical, and learning disabilities. See https://disabilitycompliance.berkeley.edu and https://dsp.berkeley.edu. UC Berkeley's Center for Teaching & Learning maintains the academic calendar and student accommodations policies and guidelines. Here you will find the policy for accommodating religious creeds, religious holidays calendar, the honor code, in which all students will "act with honesty, integrity, and respect for others."
[https://teaching.berkeley.edu/academic-calendar-and-student-accommodations-campus-policies-and-guidelines]

Late/Extension Policy: We recognize that sometimes things happen in life outside the course, especially in MIDS where some of us have full time jobs and family responsibilities to attend to. To help with these situations, we are giving you 6 "late days" to use throughout the term as you see fit. Each late day gives you a 24 hour (or any part thereof) extension to any assignment in the course except the project presentations or exams. (UC Berkeley needs grades submitted very shortly after the end of classes.) Once you run out of late days, each 24 hour period (or any part thereof) results in a 10 percentage point deduction on the grade for that assignment. You can use a maximum of 2 late days on any single assignment. We will not be accepting any submissions more than 48 hours past the original due-date, even if you have late days (for example on the assignments due 11:59 PM PST Monday, the latest that assignment will be accept is 11:59 PM PST Wednesday). If you have **very** extenuating circumstances (for

example, an emergency medical situation), please reach out to your section instructor. If you know an event is coming up that will make doing the assignment difficult, talk to your section instructor to get the assignment slightly early. Plan your time accordingly!

Tutoring: If you are finding it difficult to get the programming concepts in the course please reach out to the tutor TA: markbarn@berkeley.edu to setup a 1 on 1 (or 1 on small group) tutoring session. This is a free resource available to all students who might need extra help with the course material.

Assignment Help:
If you are stuck on a problem for more than an hour please reach out to get some help. There are many avenues to aid students:

1) Search for the error message or problem - stack overflow is a good coding help resource.
2) Google Group email with fellow students - please don't share code though!
3) Use Slack to chat among yourselves.
4) Come to Instructor office hours.
5) Email the instruction team at mids-python-instructors@googlegroups.com please send to this email as it contains all of the instructors so we can respond faster; you can also email us your code so we can have a look.
6) Request a 1-on-1 meeting with the tutor TA: markbarn@berkeley.edu

Where to Find Things:

Syllabus

- Syllabus (this document): https://docs.google.com/document/d/1hGcho789MpkcQw5a9yW27so_l8tZGJWvI5GctLrTAgI/edit?usp=sharing

Calendars

- MIDS Program Calendar: https://www.ischool.berkeley.edu/intranet/students/mids/calendar
- W200 Course Calendar: https://docs.google.com/spreadsheets/d/1PYz286UPN0sCRsRII5YEGwU5b0w6dZpDEuhjrSwef7M/edit?usp=sharing

Online Tools

- Study Net: http://www.study.net
  Study.net has some of the reading materials required for the course.
- GitHub, Git, & Bash: https://github.com
  Install and configure Git & Bash.
- Anaconda with Jupyter Notebooks & Python3: https://anaconda.org/anaconda/python
- I School Virtual Campus (ISVC): https://learn.datascience.berkeley.edu/
- Google Group Email: w200-python-2020-spring@googlegroups.com
  Use Google Group Email to formally communicate with your instructors and classmates, ask questions, share problems and your successes (but don't share code).
- Instructor group email: mids-python-instructors@googlegroups.com

- Slack: https://ucbischool.slack.com/messages/C5AL99BU6/
  Use Slack to informally communicate with your classmates, ask questions, share problems and your successes (but don't share code).

Repositories

- ISVC: https://learn.datascience.berkeley.edu/login >
  Courses > 2020-0106 DATASCI W200 Introduction to Data Science Programming (your section)

Materials

- Video Lectures: ISVC > Coursework
- Drills: ISVC > Coursework
- Books: https://www.study.net
- Class slides: Github Upstream > week_*xx*
- Class companion notebooks: Github Upstream > week_*xx*
- Class activities & solutions: Github Upstream > week_*xx*/Activities
- Other Resources: Github Upstream > resources

Meetings

- Live section: ISVC > Meetings > *live section*
  On the left bar is a calendar icon, click on this to bring up the Meeting page. Below the Upcoming Meetings heading, mouseover the section and click on the video icon on the right to start Zoom for the session's live class.
- Office Hours: ISVC > Meetings > *office_hour_session*
- Impromptu: ISVC > Meetings > Start Instant Meeting

Assignments

- Homework: GitHub Upstream > week_*xx*
- Exam 1: ISVC > Coursework > Assessments > Midterm Exam
- Exam 2: ISVC > Coursework > Assessments > Final Exam
- Project 1: GitHub Upstream > project_1
- Project 2: GitHub Upstream > project_2

Communication

- Announcements: ISVC > Wall
- Announcements (Email): w200-python-2020-spring@googlegroups.com
- Questions to All Instructors & All Students: Email w200-python-2020-spring@googlegroups.com
- Responses from an Instructor to All Students: Email w200-python-2020-spring@googlegroups.com
- Instructor group email: mids-python-instructors@googlegroups.com
- Questions to a Specific Instructor: Email
- Informal Discussion: Slack Channel ucbischool.slack.com > w200-python

Feedback

- Homework, exam, & project grades: ISVC > Gradebook