

## CDS DS 110: Introduction to Data Science with Python

Instructor Names: Dr. Scott Ladenheim (A1)      Course Dates: 1/20-5/8  
Dr. Kevin Gold (B1)

Times & Locations: A1 MWF 12:20-1:10 CGS 505      Course Credits: 4  
B1 MWF 1:25-2:15pm CGS 511

Office Locations: CCDS 1545 (Ladenheim)  
CCDS 1406 (Gold)

Contact Information: saladenh@bu.edu (Ladenheim)  
klgold@bu.edu (Gold)

Ladenheim Drop-in Office Hours: M 4:30-5:30PM, W 10-11:30AM  
Gold Drop-in Office Hours: M 3-4:30pm, F 10-11:30am

### TAs

Apoorv Verma apoorv@bu.edu  
Nikhila Kuntla nikhilak@bu.edu  
Arnaldo Ariston Paguio adpaguio@bu.edu

### CAs

Angelo Amato gamato23@bu.edu  
Brian Capozza bfc0503@bu.edu  
Xiang Fu xfu@bu.edu  
Amanda Eng engak@bu.edu

### **Course Description:**

DS 110 is the first in a two-course sequence (including DS 210) that builds students' skills in computing techniques central to data science.

In DS 110, students will use Python to explore many of the fundamental computer science concepts and processes used in data science. This begins with core computer science topics such as data structures, development of functions and recursion, and leads to topics including object oriented programming and data visualization.

In terms of Python tools for data science, students will learn how to work with numpy, pandas, and matplotlib to analyze real-world data. (See the more comprehensive list of topics below in this syllabus.)

The sequence of DS 110 and 210 works in concert with the 120-121-122 series: as students develop their expertise in the mathematical foundations covered in 120-121-122, they build their abilities to implement those tools and manage actual datasets in the 110-210 sequence.

## Instructional Format, Course Pedagogy, and Approach to Learning

The course will be primarily lecture-based, but with additional exercises for the audience designed to engage and check understanding, including weekly for-credit quizzes. Homework will play a key role in moving from hypothetical concepts to applied problems. The discussion section will serve as additional review. The optional project will be a chance to let the students guide their own learning toward a personally meaningful goal.

Because of the prevalence of LLMs and other online resources that may mask gaps in a student's knowledge, understanding, and skills, two in-class midterms and a final will be administered in which other resources are not allowed.

### AI Policy

While AI in the hands of an experienced programmer can lead to productivity gains, in the hands of a student learning to program, it can prevent the student from learning key programming skills. Therefore we adopt a policy similar to the computer science program's AI policy, in the hopes of keeping students from failing to learn to program.

**AI generated code is prohibited on assignments.** If we believe submitted code was written by an AI, the assignment as a whole will get a 0, and an academic integrity case for plagiarism may follow. *If you are an advanced programmer, please constrain yourself to using the tools we cover so that your code isn't mistakenly flagged.*

**Using AI at all during discussion section is prohibited.** Using AI during section will result in a 0 for the section. This time is meant to provide AI-free practice programming for the exams. If you have a question, ask the TA.

**You *may* consult AI for answers to general questions about how to program.** However, you may not copy any code from the answer, so keep your questions general.

Using AI to help debug programs is a gray area - the more you do this, the less you will improve as a programmer, but we don't have the tools to stop you.

We provide an AI that generates problems and comments on their solutions, the DS110 Practice Problem Generator, and this AI can be used freely.

### Books and Other Course Materials

**Recommended Text:** Deitel & Deitel, *Intro to Python for Computer Science and Data Science*, ISBN-13 978-0135404676

Recommended readings by day are in the schedule at the end of this syllabus. This book can be helpful in providing an alternate take on the material, as well as in providing practice problems to help students prepare for the midterms. But, it isn't required.

**Recommended Hardware:** You need a basic laptop that has a web browser for this course. If you don't own one, you can [borrow one from BU](#). The heavy lifting of running programs will happen in the cloud, with very little done locally on your machine unless you prefer to work that way (see Anaconda below).

## Courseware

We'll be using Blackboard for distributing assignments, collecting assignments, grading, and the distribution of course notes: <https://learn.bu.edu>

We will mostly be using Google Colab for assignments and course notes:  
[colab.research.google.com/](https://colab.research.google.com/)

Students who wish to work offline could download and install Anaconda:  
<https://www.anaconda.com/>

Students wanting additional practice programming problems before the exams can try our practice problem generator:  
<https://huggingface.co/spaces/klgold/ds110probs>  
(your 4-digit PIN is provided as a "grade" in Blackboard)

We'll use Piazza for Q&A and posting office hours:  
<https://piazza.com/bu/spring2026/ds110/>  
<https://piazza.com/bu/spring2026/ds110/staff>

## Assignments and Grading

Course assignments will consist mostly of programming, but will also contain some thought questions.

The individual assignments are meant to teach programming skills and other concepts. Students will be tested on their abilities in the in-class exams, so they should try to not take AI shortcuts. The staff can help get students unstuck if they aren't making progress. AI can be helpful as well if the questions are kept general, but students who rely on it for programming will find themselves lost during the exams.

In general, assignments are about all the topics that were covered in lecture since the last assignment - see the day-by-day topic list at the end of the syllabus for more details. All the assignments are weighted equally, even if maximum point values differ.

There will be one possible source of extra credit in the course - a mini-project that is part of assignment 7, where students can earn additional points by using a different dataset and/or performing additional analysis of the dataset. The best submissions will be invited to present a "lightning talk" near the end of the course for further extra credit on the final.

There will be two in-class midterms and one final exam during the semester. These exams will be partly multiple choice, partly free response coding. These exams are meant to encourage learning all the material rather than just what is emphasized on the homework, and also to

discourage using generative AI in any way that impedes the learning process. Students should therefore practice on their own until they can produce basic programs quickly and without external aids.

You are expected to participate in lecture at least once every three weeks, either by asking a question, answering a question, or saying something else that contributes to the discussion. When you contribute, you will turn in a card with your name on it. At the end of the three week period, the cards will be scored and returned.

Grading:

- Lecture Participation (5%)
- In-class quizzes (10%, drop 2 lowest)
- Midterm exam 1 (15%)
- Midterm exam 2 (15%)
- Individual assignments (15%, lowest dropped)
- Recitation participation (5%)
- Final exam (35%)

### **Resources/Support/How to Succeed in This Course:**

1. To succeed in this course students should attend all lectures, come to discussion sections prepared with questions, and complete all assignments on time without relying on generative AI to produce answers.
2. Students may work in groups but their code should be written separately, and the submission should mention which students provided collaboration or help.
3. Students are welcomed and encouraged to visit office hours to ask questions about the material covered in class, the homework, or any relevant topics.
4. The [Education Resource Center](#) offers free individual and group tutoring.
5. Accommodations for Students with Documented Disabilities: If you are a student with a disability or believe you might have a disability that requires accommodations, please contact the Office for Disability Services (ODS) at (617) 353-3658 or [access@bu.edu](mailto:access@bu.edu) to coordinate any reasonable accommodation requests. ODS is located at 25 Buick Street on the 3rd floor. Please give me notice of any special needs at least 2 weeks before an exam that requires special accommodations, so that we can schedule space if necessary.

### **Community of Learning: Class and University Policies**

1. **Courtesy expectations.** Students are responsible for supporting a courteous learning environment. Please show respect for other students' questions, and maintain an attentive attitude in class. Please participate in the in-class exercises and activities. Using screens for activities other than following lecture is discouraged.
2. **Attendance & Absences.** Attending lecture is effectively mandatory because of the in-class quizzes that could be delivered on any non-exam day. Attending discussion sections also carries some participation credit. However, attendance or quizzes in either lecture or section can be waived with a reasonable request for accommodation. We will

in particular waive for religious holidays; see the University [Policy on Religious Observance](#).

3. **Assignment Completion & Late Work.** Assignments will be submitted as PDF and .ipynb files on Blackboard. Students have 5 late days which can be used over the course of the semester, no more than 2 per assignment, which allow work to be turned in late with no penalty. Using 1 late day extends the deadline 24 hours for that student; late days can't be used in fractions. You don't need to ask to use a late day; we will just deduct it if you submit late.
4. **Academic Conduct Statement**

Students are expected to abide by the guidelines and rules of the Academic Code of Conduct. <https://www.bu.edu/academics/policies/academic-conduct-code/>

In particular, **if code is submitted that is heavily based on another source, the source must be cited in the comments or a text box.** This is true for all submitted code (e.g., including the project). **Failure to cite a source may be considered plagiarism and reported to CDS for a hearing.**

Note as well that **using AI to generate code for an assignment is completely prohibited** and may be subject to a 0 and an academic integrity case.

In lieu of the CDS GAIA policy, we ask that students simply attest that they did not use AI to generate code at the end of each assignment. We are no longer requiring transcripts of AI interactions.

You can consult the CS department's helpful webpage about code plagiarism if you have any doubts about what is considered ethical: <https://www.bu.edu/cs/undergraduate/undergraduate-life/academic-integrity/>

### **Hub Learning Outcomes - Quantitative Reasoning I (QR1):**

Students will be introduced to the Python programming language. Students will learn how to use pure Python to write basic functional and object-oriented programs. Students will also learn about core data science libraries such as numpy and pandas and, by the end of the course, students will be able to use these tools to produce descriptive analyses and visualizations of different datasets. (All assignments, especially Assignment 7).

In parallel to learning Python for data science, students will develop and demonstrate their understanding of core concepts in computer science, and their ability to use them to manipulate different types of data. Students will learn about basic CS topics such as algorithms, big-O notation and algorithm runtime (Assignment 8), object oriented programming methods (Assignment 5), and parallel computing (Assignment 9).

Students will master the fundamentals of programming in Python for data science, including learning how to use common libraries such numpy, pandas, and matplotlib (Assignment 4).

Using these tools, students will build basic models of the data to uncover and interpret interesting patterns in real-world data. (Assignment 7).

## **Outline of Class Meetings: Topics by Day, Optional Readings, Assignments Due**

**Homework goes out and is due on *Wednesdays*.**

### **Week of 1/19 (BU classes start 1/20)**

**W** Introduction

**F** Programming basics - expressions and types (Deitel 1.10, 2.3, 2.4)

### **Week of 1/26**

**M** Variables and conditional branching (Deitel 2.2, 2.6, 2.7)

**W** while and lists (Deitel 3.7, 5.2, 5.8) *HW1 OUT*

**F** boolean operators, importing, and writing code (Deitel 3.14, 3.16)

### **Week of 2/2**

**M** foreach loops; range (Deitel Ch 3.7-3.13)

**W** More on loops *HW1 DUE, HW2 OUT*

**F** Defining a function; typical commenting; refactoring; writing a test (Deitel Ch 4)

### **Week of 2/9**

**M** More on functions

**W** Hash tables, sets, and dictionaries; pass by reference (Deitel Ch 6), *HW2 DUE, HW3 OUT*

**F** numpy

### **Week of 2/16**

**Tu** matplotlib (**President's Day, Tu has M schedule**)

**W** Midterm 1 review *HW3 DUE*

**F** Midterm 1

### **Week of 2/23**

**M** Reading data into Pandas; escape characters; easy analysis in Pandas (Deitel Ch 7)

**W** More Pandas examples *HW4 OUT*

**F** Strings, regexes, and data cleaning (Deitel Ch 8)

### **Week of 3/2**

**M** Files and Exceptions (Deitel Ch 9)

**W** Object-oriented programming: creating our own objects (Deitel Ch 10) *HW4 DUE, HW5 OUT*

**F** Common methods to override; other details of OO Python

### **Week of 3/9 - SPRING BREAK**

### **Week of 3/16**

**M** Recursion

**W** Data structure implementation: trees, linked lists, dynamic arrays *HW5 DUE, HW6 OUT*

**F** scikit-learn and k-nearest neighbors (Deitel Ch 15)

**Week of 3/23**

**M** Decision Trees

**W** Midterm 2 review *HW6 DUE*

**F** Midterm 2

**Week of 3/30**

**M** Decision Forests

**W** Regression *HW7 OUT (with miniproject)*

**F** Using ML with Natural Language

**Week of 4/6**

**M** Complexity and big-O (Deitel Ch 11)

**W** Analysis of algorithms: sorts *HW7 DUE, HW8 OUT*

**F** Graphs and centrality

**Week of 4/13**

**M** SQL

**W** Advanced Pandas *HW8 DUE, HW 9 OUT*

**F** Other ways to speed up code

**Week of 4/20**

**M** Patriot's Day

**W** Visualization (**Monday Schedule**) *HW9 DUE*

**F** Effective communication

**Week of 4/27**

**M** Lightning talks

**W** Final exam review

**May 4-8 FINAL EXAM DAY TBD**

*~Last updated 1/7/26~*