**IDS 400: Programming for Data Science in Business (Spring 2023)**

Department of Information and Decision Science

UIC Business

Instructor: Tengteng Ma

Teaching Assistant: Aida Sanatizadeh

Lectures: Thursday 6:30pm-9:00pm (online: 01/09/2023-03/17/2023; in-person: 03/18/2023-05/05/2023, subject to change)

Contact: Piazza private message (you can log into Piazza through Blackboard)

**Textbooks:**

While we do not have any required textbook for the course, here are a few recommended books/online resources to learn Python:

* [Think Python—How to Think Like a Computer Scientist](https://greenteapress.com/wp/think-python-2e/), by Allen B. Downey,

(1st edition for Python 2, 2nd edition for Python 3). You can download it [here](http://greenteapress.com/thinkpython2/thinkpython2.pdf) or [read it in HTML](http://greenteapress.com/thinkpython2/html/index.html).

* Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython, 2nd Edition, by Wes McKinny, O'Reilly Media, 2017. E-book is available at UIC library website.
* [Python Data Visualization Cookbook.](http://blaqueyard.com/download/Python%20Data%20Visualization%20Cookbook.pdf) This is a book for those who want to explore more in visualization in Python.
* [MIT Open Coursware: A Gentle Introduction to Programming Using Python](https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-189-a-gentle-introduction-to-programming-using-python-january-iap-2011/index.htm)
* [Google’s Python class](https://developers.google.com/edu/python)

**Course Description**

Data volume grows exponentially in recent years due to the development of technology. Companies are also increasingly employing data analytics techniques to leverage the benefit of large-scale dataset. It has already been well-recognized that companies can significantly improve their profitability through data analytics. This class, IDS 400, Programming for Data Science in Business is an introductory class to help you learn the basics of programming for data analytics, and how to program in Python to solve real-world problems. This will be very helpful for your jobs in the future. This course will first cover the basics of data science, then move to foundations in Python programming, and gradually to data analysis using Python and external packages.

The (tentative) list of topics we will cover in this course is:

• Basic programming concepts

• Install and run Python programs

• Variables and expressions

• Control structures: sequence, branch, iteration

• List, tuple, dictionary in Python

• Functions, parameters, and recursion in Python

• String operations

• File operations and modules

• Visualization

• Data manipulation: Pandas

• Optimization and statistics in Python: SciPy

• Machine learning package: Mlpy, scikit-learn

While this course does not require any prior programming knowledge of python, students enroll in the class are expected to have some familiarity with programming in other languages, at the introductory level (Such as Java, R and etc.).

**Learning Outcomes:**

By the end of this class, students are expected to:

1. Understand the basics of data science in programming.
2. Understand the basic of Python, such as syntax, semantics, iterative control, data structure etc.
3. Understand how to use existing Python package
4. Apply Python package for data storage, manipulation, and cleaning
5. Understand how to perform machine learning in Python
6. Be able to use Python to solve real-world problems.

**Software**

We will mainly use Jupyter Notebook as the programming environment. You can download it through [Anocaonda](https://problemsolvingwithpython.com/01-Orientation/01.03-Installing-Anaconda-on-Windows/).

**Attendance**

I strongly encourage you to attend all lectures and labs. There might be some random attendance checks. I will deduct 5 points if you are absent for all checks. Failing to attend midterm or final project will receive an “F” for the course except extreme reasons.

**Face Masks**: Masks covering both the mouth and nose must be worn at all times by all students, faculty, and staff while inside any campus building regardless of vaccination status. If you do not wear a mask, you will be asked to leave the classroom and will not be allowed back in class unless or until you wear a mask. If you have forgotten your mask, you may pick one up from one of the student information desks on campus during the first two weeks of classes. Students who do not comply with the mask-wearing policy will be reported to the Dean of Students.

**Piazza**

We will use the Piazza (you can log into this system through Blackboard) for a discussion board and for posting updates. Participation on the discussion board counts towards your grade. A constructive, relevant, non-redundant post earns a Piazza point. Here are some examples:

* A question that elicits a substantive, correct answer. In other words, this cannot be a superficial question which you can easily find answer on google. This also suggests that you should provide enough details in this question.
* A substantive, correct answer to another student's question.
* A substantive, correct refinement or clarification of another student's answer to a question.
* A correction to anything substantive that I get wrong!

No anonymous questions will be answered by the instructor. If there is an anonymous post, I will ask the author to reveal the ID on the platform. This is because you are expected to form connection with your classmates. You can also form groups with the help of Piazza.

**Assignments**

There are tentatively 7 assignments across this semester. While you are allowed to submit your assignment late, you will receive 10% grade penalty for every additional day after the deadline. As this is a generous late submission policy, penalties are strictly enforced, and I will not grant extensions. Please plan accordingly, and do not leave submission for the last minute.

**Plagiarism Policy**: I have seen plagiarisms in my past years of experience teaching programming classes, and it seems that a few people will turn assignments that are not their own. You should understand that it is very easy to detect copy & paste for programs, even when a program has been changed to cover where it comes from. Copying code from others, or letting others copy your code is a violation of student disciplinary policy and academic integrity at UIC, and penalty for such behavior follows related school policy.

**Lab Sessions via Zoom**

There are tentatively 11 labs**.** Each lab session lasts one hour.Within each lab, you need to finish some given tasks. Typical tasks are reviewing/practicing what we learned in lectures for that week. For some labs, you need to submit your code. Lab attendance is not mandatory, **but you are highly encouraged to attend all the Lab sessions**. Without submitting 6 or more labs, you will automatically fail this course. Note that lab submissions will not be graded.

**Course project**

You can only learn programming through coding by yourself. That is why I emphasize a lot on the final project. You can form a group **up to four students** to do the project. The deliverables of the course project include a project report, Python code plus a class presentation.  The presentation constitutes 50% of the grade and project report and Python code deliverable 50%.  The presentation is graded on the basis of content, structure, design, and delivery.  By content we mean that the students have been able to include all the necessary pieces of information needed to achieve the objective.  By structure we mean that the students have included in the presentation introductory, main body, and conclusion slides which follow each other in logical succession.  By design we mean that the students have been able to use layouts, graphics, charts, videos, and other visual aids effectively and which enhance the understanding of the topic presented.  The second deliverable of the project, the project report and Python code, is graded for quality and completeness (whether this application achieve the goal you claim).

I expect two things from the project presentation: 1. business insight. Why your project is important for the business, and how your project help the business improve their business decisions. 2. The analytics component. Explaining why you choose different models and which model performs the best (and why). If visualization is the focus of your project, you should explain what business insight you can derive from this specific type of visualization technique. Given the limited time of presentation, you don’t need to show your python code in the presentation. The final presentation will be live format. Please inform the instructor if you are not able to present on the scheduled day. Every student is supposed to grade other groups’ projects, and the final score of your project presentation is the combination of scores from your classmates and the instructor.

Project report essentially is a written version of the presentation, which should also include the five important components mentioned above. Notice that given the report is due one week after the presentation, you can keep improving your project in the report after the presentation.  A recommended format of the report is 12-point, single-spaced, Times New Roman font with one-inch margins on all four sides, 5-8 pages.

**If any one of the team members requests a peer evaluation within the team, I will conduct the evaluation and adjust the project grade accordingly.**

**Course Assessment Measures and Grading Criteria**

**A** (> 90); **B** (80-89); **C** (70-79); **D** (60-69); **F** (< 60).

|  |  |
| --- | --- |
| Component | Weight |
| Attendance/Piazza | 10% |
| Assignment | 50% |
| Quiz | 20% |
| Final project | 20% |

**Acknowledgements**

Much of the structure of this course was derived from previous iterations of courses taught by Professors Yingda Lu, Kunpeng Zhang, and Yixin Lu. I appreciate their help and permission to build on their good work.

**Additional Readings for Each Topic:**

Intro to Python:

[How to Think Like a Computer Scientist, Chapter 1](http://www.greenteapress.com/thinkpython/thinkCSpy/html/chap01.html)

The Python Tutorial, Chapter [1](https://docs.python.org/3/tutorial/appetite.html), [2](https://docs.python.org/3/tutorial/interpreter.html), and [3](https://docs.python.org/3/tutorial/introduction.html)

Variables, Expressions, Statements:

[How to Think Like a Computer Scientist, Chapter 2](http://www.greenteapress.com/thinkpython/thinkCSpy/html/chap02.html)

[Interactive Python: variables, expressions, and statements](http://openbookproject.net/thinkcs/python/english3e/variables_expressions_statements.html)

Conditions

[How to Think Like a Computer Scientist, Chapter 4](http://www.greenteapress.com/thinkpython/thinkCSpy/html/chap04.html)

[Tutorialspoint](http://www.tutorialspoint.com/python/python_decision_making.htm)

Loop

[How to Think Like a Computer Scientist, Chapter 6](http://www.greenteapress.com/thinkpython/thinkCSpy/html/chap06.html)

[Tutorialspoint](http://www.tutorialspoint.com/python/python_loops.htm)

Lists, Tuples and Dictionaries

How to Think Like a Computer Scientist, Chapter [8](http://www.greenteapress.com/thinkpython/thinkCSpy/html/chap08.html), [9](http://www.greenteapress.com/thinkpython/thinkCSpy/html/chap09.html), [10](http://www.greenteapress.com/thinkpython/thinkCSpy/html/chap10.html)

[Tutorialspoint](http://www.tutorialspoint.com/python/python_decision_making.htm): [Lists](http://www.tutorialspoint.com/python/python_lists.htm), [Tuples](http://www.tutorialspoint.com/python/python_tuples.htm) and [Dictionaries](http://www.tutorialspoint.com/python/python_dictionary.htm)

Functions and Parameters

How to Think Like a Computer Scientist, Chapter [3](http://www.greenteapress.com/thinkpython/thinkCSpy/html/chap03.html)

String Operations

How to Think Like a Computer Scientist, Chapter [7](http://www.greenteapress.com/thinkpython/thinkCSpy/html/chap07.html)

Files

How to Think Like a Computer Scientist, Chapter [11](http://www.greenteapress.com/thinkpython/thinkCSpy/html/chap11.html)

TutorialsPoint: [Files](http://www.tutorialspoint.com/python/python_files_io.htm), [Modules](http://www.tutorialspoint.com/python/python_modules.htm)

Pandas

[10 Minutes to Pandas](http://pandas.pydata.org/pandas-docs/stable/10min.html)

[Pandas Tutorials](http://pandas.pydata.org/pandas-docs/stable/tutorials.html)

SciPy

[NumPy Tutorial](https://docs.scipy.org/doc/numpy/user/quickstart.html)

[SciPy Tutorial](https://docs.scipy.org/doc/scipy-0.18.1/reference/tutorial/index.html)

Machine Learning

[mlpy](http://mlpy.sourceforge.net)