Course Syllabus: **DS 310 Data Mechanics**

**Instructor Name**: Chris Seferlis, MBA [seferlis@bu.edu](mailto:seferlis@bu.edu)

**Office:** CDS 1609

**Instructor Office Hours:** See Piazza

**Lectures**: Tue/Thur 5:00pm - 6:15pm

**Discussions**: Wed **8:00-8:50, 9:05-9:55, 10:10-11:00, 11:15-12:05**

**Teaching Assistants:** David E. Kim [dk98@bu.edu](mailto:dk98@bu.edu) and Ashwin Daswani ashwind@bu.edu

**Course Assistants:** Emma Steel [evs17@bu.edu](mailto:evs17@bu.edu), Sindhuja Kumar [sins@bu.edu](mailto:sins@bu.edu), Anup De [anupde@bu.edu](mailto:anupde@bu.edu), Bella Davies [bjdavies@bu.edu](mailto:bjdavies@bu.edu), and Josh Yip [joshyip@bu.edu](mailto:joshyip@bu.edu)

**DS310 Team Office Hours: See Piazza**

**1. Overview**

**1.1 Course Description:** This course is focused on developing students’ capacity to design and implement the data flows and the associated workflows that are meant to inform online and offline decision-making within large systems. To that end, in supervised group projects, students explore the data science lifecycle, including question formulation, data collection and cleaning (data wrangling), exploratory data analysis and visualization, and decision-making. The course applies tools and methods for data collection, retrieval, integration, and interpretation, using relational (SQL), non-relational (noSQL) and Big Data paradigms to assemble analysis, optimization, and decision-making algorithms to track and scale data. Topics covered include: consolidation, synchronization, and summarization of multiple data streams; data maintenance, and availability; optimization, and analytics that can operate on large amounts of static or streaming data; and online and offline interactive visualizations platforms for presenting and examining data. Projects and assignments in this course will leverage problems in real-world settings, especially those related to CDS Impact Labs and co-Labs focusing on equity, sustainability, health, and civic tech.

**1.2 Course Objectives:**

At the end of the course, successful students will have gained skills and hands-on experience on the following methods and technology:

* Design and implementation of data processing pipelines
* Complex data modeling
* Architectural considerations for various data requirements
* Relational query optimization
* Dataflow programming abstractions
* Data stream processing concepts
* System support for distributed workloads
* Statistical analysis

Further, students will be exposed to recent developments in distributed data processing systems such as Hadoop, Apache Spark, Databricks, Massively Parallel Processing and more through paper assignments and presentations. The collaborative semester-long project will prepare them for the practical aspects of their future careers and expose them to project management tools and software engineering best practices.

**1.3 Prerequisites:**

**DS 110 and 210 or consent of the Instructor.**

**2. BU Hub:**

This course satisfies **Teamwork/Collaboration**, **Quantitative Reasoning II** and **Critical Thinking**.

**Teamwork/Collaboration Learning Outcomes:**

* **Learning Outcome #1:** As a result of explicit training in teamwork and sustained experiences of collaborating with others, students will be able to identify the characteristics of a well-functioning team.

Students will be assigned to teams of 5-6 students to complete a final project. Students will be provided with feedback and guidance on teamwork by the course instructor and the Teaching Assistants, including how to promote team and individual accountability, set expectations, and how to schedule meetings regularly and make meetings effective. Students’ teamwork skills may also be developed through peer-programming experiences.

* **Learning Outcome #2:** Students will demonstrate an ability to use the tools and strategies of working successfully with a diverse group, such as assigning roles and responsibilities, giving, and receiving feedback, and engaging in meaningful group reflection that inspires collective ownership of results.

Students will assign each other roles and responsibilities and will be expected to provide equal contributions during the project planning, development, and presentation phases, which will encourage team and individual accountability. Students will be provided with troubleshooting strategies for dealing with problems that may arise in their team. At the conclusion of the project, team members will perform a peer evaluation exercise where they will reflect upon their own contributions as well as the contributions of their teammates. This feedback can be used by the instructor to (in most likely few cases) upgrade or downgrade project components of the grade of individual students. Peer-, self- and team-evaluations will be collected, and students will be encouraged to regularly discuss team roles and to plan for upcoming tasks and reports. These evaluations will be used to help direct students in techniques for troubleshooting and problem solving within the team. Peer ratings will also be used to adjust scores for each team member.

**Quantitative Reasoning II Learning Outcomes:**

1. Students will frame and solve complex problems using quantitative tools, such as analytical, statistical, or computational methods.
2. Students will apply quantitative tools in diverse settings to answer discipline‐specific questions or to engage societal questions and debates.
3. Students will formulate and test an argument by marshaling and analyzing quantitative evidence.
4. Students will communicate quantitative information symbolically, visually, numerically, or verbally.
5. Students will recognize and articulate the capacity and limitations of quantitative methods and the risks of using them improperly.

**How this course achieves the Quantitative Reasoning II learning outcomes:**

* **Outcome 1:** Throughout this course, students will design and implement techniques for managing complex data pipelines.
* **Outcome 2:** Students will face a breadth of problems ranging from classic computer science applications to everyday occurrences and determine which tools to apply.
* **Outcome 3:** Students will “test” the correctness of their algorithmic solutions by writing unit tests.
* **Outcome 4:** Students will learn to communicate clear and logically correct arguments using a combination of mathematical language, symbols, and numbers.
* **Outcome 5:** The course will specifically cover the capacity of each algorithmic technique and when the tool is not appropriate. Students will need to determine the appropriate tool for each problem in their homework, or whether algorithmic approaches are even suitable for a problem.

**Toolkit Critical Thinking Learning Outcomes:**

Students will be able to identify key elements of critical thinking, such as habits of distinguishing deductive from inductive modes of inference, recognizing common logical fallacies and cognitive biases, translating ordinary language into formal argument, distinguishing empirical claims about matters of fact from normative or evaluative judgments, and recognizing the ways in which emotional responses can affect reasoning processes.

Drawing on skills developed in class, students will be able to evaluate the validity of arguments, including their own.

**How this course achieves the Toolkit Critical Thinking learning outcomes:**

* **Outcome 1:** Students will learn algorithmic reasoning—intuition for how to consider any possible input instance and the requisite steps needed to transform the instance into the desired output. This form of thinking is not only essential in computing and data sciences, but transformative in understanding a world shaped by algorithmic technology.
* **Outcome 2:** Lecture, collaborative problem-solving, and homework will include designing and implementing data processing pipelines, all honing the students’ skills to evaluate the validity of these solutions through experimental benchmarking.

**Feedback on Learning Outcomes:**

Students will complete weekly problem sets that require them to solve data processing tasks and analyze the pros and cons of their solutions. They will receive prompt grading with feedback on their solutions, and thus on their quantitative reasoning and critical thinking. During lectures, students will learn and practice both algorithmic techniques and system design principles. In-class exercises will include opportunities to observe ideas from other students as well as to hear feedback from other students and from the instructor.

**3. Instructional Format, Course Pedagogy, and Approach to Learning**

**3.1 Courseware**

* We will use **Piazza** for announcements, questions, discussions, and all other communication: <https://piazza.com/bu/spring2024/ds310/info> access code: 8v2wfcfm677 Each student will create a Microsoft Azure account for homework assignments. Project related work
* We will use **Gradescope** for homework assignments: Entry Code **WB2DEE**
* Each Student will create a GitHub account for storing code and assets
* We may use other resources throughout the semester as required: TBD

**3.2 Lectures:**

Lectures will be held during the assigned time slots. Section 4 of the syllabus provides the topic and assigned readings for each lecture. For some lectures there will be readings and other content that you are expected to complete and actively participate in class discussions. Many concepts and anecdotes that will appear on exams will be covered during lecture, so attendance is highly recommended. Lecture slides will be made available on Piazza shortly after each lecture.

**3.3 Discussions:**

Students are expected to attend the weekly discussion section they have been assigned to. The Teaching Assistant will lead the discussion sessions. The objectives are to present material on the required tools that reinforce the concepts covered in the lectures, and answer questions (or provide clarifications) regarding the assignments and projects. The Teaching Assistant will post information to Piazza as necessary. In addition to the discussions, the Teaching Assistant will hold weekly Office Hours.

**3.4 Classroom recordings:**

Class sessions might be recorded for the benefit of registered students who are unable to attend live sessions (either in person or remotely). Recorded sessions will be made available to registered students ONLY via their password-protected BU account. Students may not share such sessions with anyone not registered in the course and may certainly not repost them in a public platform. Students have the right to opt-out of being part of the class recording.

**3.5 Course Materials:**

The reference material we will use for the class is a combination of book and provided materials. Please acquire the book:

Fundamentals of Data Engineering: Plan and Build Robust Data Systems by Joe Reis and Matt Housley - https://a.co/d/g4JQ0c0

Slides, lecture notes, and other publicly available resources will be published on Piazza.

**4. Assignments and Grading Criteria**

The course consists of lectures, hands-on assignments, a midterm exam, and a final project. Final projects must be done in teams of 3-5 students. Your final grade will be determined as follows:

* 8 assignments: 24% (3% per assignment)
* Discussion and attendance: 8%
* Quizzes and in-class assignments: 8%
* 2 Exams: 25%
* Final project: 35%

To be considered complete, code deliverables must be accompanied by sufficient documentation. The exams will include open-ended questions on all topics we will discuss in the lectures. The exams will not include programming exercises.

Individual contributions to assignments and the final project will be assessed by taking into account the following:

* The quality of individual task deliverables outlined in the project design document
* The individual’s ability to answer questions about the assignments and project during office hours and demo presentations
* The individual’s participation during lecture discussions
* The individual’s contribution to the project’s GitHub repository

**Final Project:**

In the first month, you will be assigned to a project team with 5-6 students. You will work on the concepts covered in the final project throughout the semester and rubric(s) will be provided for guidance on expectations. Each team will be working together to deliver:

* A **design document** outlining (1) the project goals, (2) an implementation and evaluation plan, (3) the task distribution among team members.
* A **final demo** to be presented near finals
* The project’s **GitHub repository**, or related service, including code, tests, automation and plotting scripts, and documentation.

**5. Class and University Policies**

**5.1 Homework submission:**

All assignments and the project deliverables will be submitted via the course resources. All deliverables are **due by 11:59pm ET on the day of the respective deadline**.

**5.2 Attendance:**

Students are expected to attend each class session unless they have a valid reason for being absent. Acceptable excused absences include observing religious holidays and illness. In such cases, students are advised to contact the instructor as soon as possible, so that reasonable accommodations can be provided. Please review the **BU attendance policy** and the **BU Policy on Religious Observance** for more information.

**5.3 Late work policy:**

Students who submit homework late will only be eligible for up to 6**7% of the original score, unless otherwise approved by the professor, and may only submit homework no later than 7 days after due date, unless extenuating circumstances are discussed with a member of the DS310 Class Staff**.

**5.4 Academic conduct:**

Academic standards and the code of academic conduct are taken very seriously at our university. Please take the time to review the CAS Academic Conduct Code: http://www.bu.edu/academics/resources/academic-conduct-code/ and the GRS Academic Conduct Code: http://www.bu.edu/cas/students/graduate/grs-forms-policies-procedures/ academic-discipline-procedures/. Please review the sections regarding plagiarism and cheating carefully. Copies of the CAS Academic Conduct Code are also available in room CAS 105. A student suspected to violate this code will be reported to the Academic Conduct Committee, and if found culpable, the student will receive a grade of "F" for the course.

All assignments must be completed individually, unless instructed otherwise. Discussion with fellow students via Piazza or in-person is encouraged but presenting the work of another person as your own is expressly forbidden. This includes “borrowing”, “stealing”, copying programs/solutions or parts of them from others. Note that we may use an automated plagiarism checker. Cheating will not be tolerated under any circumstances.

Any resources, including material from other students (current or past), that are used, beyond the text or that provided by the TA or professor must be clearly acknowledged and attributed. Using such material may at the discretion of the TA or professor result in a lower grade. However, if such material is used and not acknowledged and 12 attributed, it will automatically be considered as possible academic misconduct.

**6. Accommodations:**

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. Any grants of accommodation will be provided with a letter, this letter must be submitted to the professor within 2 weeks of receipt, and cannot be used for retroactive requests, per the ODS policy.

**7. Course Schedule:**

The rest of the syllabus is tentative and might be updated during the semester. We will be keeping you informed of any changes made to the readings or assignment deadline via Piazza.

Make sure to become familiar with the **Official Semester Dates**. Some of the critical Semester Dates are:

* The Last Day to DROP Classes (without a ‘W’ grade) February 22, 2024
* The Last Day to DROP Classes (with a ‘W’ grade) March 29, 2024

|  |  |
| --- | --- |
| Class Weeks | Topics |
| Week 1 - 2 | Data Concepts, Structure, Schemas, Governance, Lifecycle |
| Week 3 - 5 | Structured Data, DB Fundamentals, Data Models, Relational DB |
| Week 6 - 7 | Data Warehousing, ETL/ELT, Exam 1 |
| Week 8 - 9 | Semi-Structured Data, NoSQL, Project Overview |
| Week 10 - 11 | Big Data Fundamentals, Implementations, Processing, Architectures |
| Week 12 - 15 | Data Analysis, Security/Privacy, Business Intelligence, Responsible AI |
| Week 16 - 17 | Exam 2, Project Work |
| Final Exam | Project Presentations |