Course #: Cloud Data Engineer

**course description & Syllabus**

Name of College or University

Semester Year

# Overview

**Title:** Data Engineer
**Credits/Units:** 6-credit hours
**Pre-requisites:** Practical Programming with Python, Cloud Administrator

**Institution LMS access:** [https://lms.univ-or-college.edu](https://canvas.cmu.edu)

**Recitation:**

1. **Tuesday, 8:00 AM – 8:50 AM ET, GHC 4307 (Videotaped)**

**Teaching Staff:**

| [**Prof. Best Faculty**](http://www.xyz.edu/...)faculty-email@xyz.eduOffice Number, +1-555-555-5555*Office hours:*Tuesday, 3-4pm (time zone) |  |
| --- | --- |
| TAs typically hold office hours in building/room. The TA office hours are posted on the LMS:* TA Name<ta-email@xyz.edu>
 | * TA Name<ta-email@xyz.edu>
 |

# Course Description

Students will gain knowledge and develop hands-on experience solving real-world problems in the field of data engineering. This includes ingesting, egressing and transforming data from multiple sources using various technologies, services and tools. Students will develop the skills needed to identify and meet data requirements of an organization by designing and implementing systems and data pipelines that manage, monitor and secure the data using the full stack of cloud services to satisfy business needs.

# Students will explore and experiment with various storage abstractions such as SQL and NoSQL databases, data lakes and data warehouses to store, transform and draw insights from data. Students will utilize MapReduce and Spark frameworks and provision clusters on public cloud infrastructure to clean, prepare and analyze large datasets. In addition, students will architect and develop different analytics applications and pipelines using batch, iterative and stream processing frameworks.

# All projects in the course utilize existing data storage and data processing technologies and services available on cloud. Specifically, students will be exposed to real-world scenarios, infrastructure, and data. It is our goal that students will develop the required skills needed to become a data engineer.

# Course Goals

At the conclusion of the course, the successful student should be able to:

1. Identify and suggest appropriate storage abstractions to store, transform and draw insights from data;
2. Architect, orchestrate and deploy a variety of analytics applications and pipelines using batch and stream processing frameworks.
3. Identify and construct useful features using well-known feature engineering methods to improve the accuracy of ML based predictors.
4. Implement and automate data pipelines using cloud-based data lakes, data warehouses and workflow orchestration services to manage, monitor and secure data.

Through this process, we aspire for our students to become sophisticated, independent, and resilient problem solvers who are able to overcome challenges and learn.

# Learning Outcomes

In this project-based course, we have project and conceptual learning objectives.

## Conceptual Learning Objectives

The **conceptual learning objectives** (LOs) are the following. Students will be able to:

1. Describe the data science life cycle and define the role of a data engineer.
2. Explain the importance of data modeling, data representation and describe the relational model.
3. Define and compare the storage abstractions such as SQL and NoSQL databases, Datalakes and Data Warehouses to store, transform and draw insights from data.
4. Describe the process of data extraction, transformation and loading (ETL) for data preparation and analysis.
5. Distinguish between data preparation, data wrangling and data crunching.
6. Describe the iterative process of data exploration, feature extraction and feature selection.
7. Describe and distinguish between the various feature engineering methods.
8. Explain the advantages of parallel and distributed processing frameworks such as MapReduce and Spark for big data analytics and data mining.
9. Discuss the characteristics of real-time stream data and the general requirements of stream processing systems.
10. Describe the process of architecting, scheduling and monitoring data pipelines in production.
11. Discuss how data pipelines can be configured to meet the data security, privacy and compliance requirements.

## Project Learning Objectives

The **project learning objectives** (LOs) are the following. Students will be able to:

1. Utilize Python libraries such as Pandas to solve data science problems with interactive programming using Jupyter Notebook.
2. Provision, configure, load and query data from a managed relational database services on the cloud.
3. Extract, load, and query data from NoSQL databases and multi-model databases on the cloud.
4. Identify and construct discriminating features using feature engineering methods to improve the accuracy of ML based predictors.
5. Implement an iterative batch processing solution using Apache Spark and Azure DataBricks to gather insights from a large dataset.
6. Create a stream processing pipeline to process, analyze and represent real-time stream data using managed stream processing frameworks on the cloud.
7. Perform interactive analytics on annotated big data stored in a DataLake using cloud-based data ingestion services.
8. Provision, configure, load and query data using the massive parallel processing capabilities of a SQL Data Warehouse.

## Course organization

Your participation in the course will involve several forms of activity:

1. Reading the conceptual content for each unit.
2. Completing the graded weekly assessments after each unit.
3. Complete projects, which are hands-on training and automated feedback.

# Getting help

Students are encouraged to ask questions about content and projects through **the Q & A forum**. The course link for the forum is:

[http://Q&A-forum.edu/](http://piazza.com/cmu/spring2020/1531915619/home)

# Policies

###### Working Alone on Projects

Projects that are assigned to single students should be performed individually.

###### Handing in Projects

All assessments are due at 11:59 PM EST (one minute before midnight) on the due dates specified on the Sail() Platform. All hand-ins are electronic.

###### Appealing Grades

After each project module is graded, you have seven calendar days to appeal your grade. All your appeals should be provided by email to the professor.

# Assessment

Inline activities (“Learn by Doing” and “Did I Get This”), which are available in most pages in the OLI course, are simple, non-graded activities to assess your comprehension of the material as you read through the course material. You are advised to complete all of the inline activities before proceeding through to the next page or module. If you missed many of the activities, it is recommended that you review the material again.

The conceptual units consist of modules of content on OLI, each week has a Checkpoint Quiz that you must complete before the deadline posted on OLI. Each weekly Checkpoint Quiz will be worth N% of your total grade. It is your responsibility to ensure that the quiz is submitted prior to the deadline. You will have only a single attempt to complete each Checkpoint Quiz on OLI.

This course includes several individual projects. Each project module has to be completed based on the deadlines posted on Sail(). The write-up required to complete each project module is available on Sail(). Each module has a submission process that is specific to the project module that is due. It is the students’ responsibility to make sure that all project work is completed and that the project module is submitted prior to the deadline. Students typically have multiple attempts to submit the project module on Sail().

| Type | Number | Weight |
| --- | --- | --- |
| Conceptual Content Quizzes | 8 | XX% |
| Projects | 8 | YY% |
| Total Grade |  | **100%** |

# Cheating

We urge each student to carefully abide by the university policy on academic integrity, which outlines the policy on cheating, plagiarism or unauthorized assistance. It is the responsibility of each student to produce her/his own original academic work. Collaboration or assistance on academic work to be graded is not permitted unless explicitly authorized by the course instructor. Each unit checkpoint quiz or project module submitted must be the sole work of the student turning it in. Student work on the cloud is logged, submitted work will be closely monitored by automatic cheat checkers, and students may be asked to explain any suspicious similarities with any piece of code available. The following are guidelines on what collaboration is authorized and what is not:

###### What is cheating?

1. Sharing code or other electronic files by either copying, retyping, looking at, or supplying a copy of any file. Copying any code from the internet (stackoverflow.com or github or others). No code can be used to “test” the auto-grader. Anything you submit to the auto-grader must be your code.
2. Copying answers to any checkpoint quiz from another individual, published or unpublished written sources, and electronic sources.
3. Collaborating with another student or another individual on checkpoint quizzes or project modules.
4. Sharing written work, looking at, copying, or supplying work from another individual, published or unpublished written sources, and electronic sources.
5. Collaboration in team projects is strictly limited to the members of the team.

###### What is **not** cheating?

1. Clarifying ambiguities or vague points in class handouts.
2. Helping others use computer systems, networks, compilers, debuggers, profilers, or system facilities.
3. Helping others with high-level design issues.
4. Guiding others through code debugging but not debugging for them.

Cheating in projects will also be strictly monitored and penalized. Be aware of what constitutes cheating (and what does not) while interacting with students. You cannot share or use written code, and other electronic files from students. If you are unsure, ask the teaching staff.

Be sure to store your work in protected directories. The penalty for cheating is severe, and might jeopardize your career – cheating is simply not worth the trouble. By cheating in the course, you are cheating yourself; the worst outcome of cheating is missing an opportunity to learn. In addition, you will be removed from the course with a failing grade. We also place a record of the incident in the student’s permanent record.

# Conceptual Topics

The course content will be structured into the following modules:

| Module | Title | Learning Objectives |
| --- | --- | --- |
| 1 | **Data Landscape** | * Describe the evolution of data types and data’s economic value.
* Explore the impact of data on various industries and the importance of data-driven decision making (Data science, ML/AI, Visualizations, etc.).
* Explore the concepts of structure, dynamicity and granularity of data.
* Explore the various types of data and data access patterns.
* Describe the data science life cycle.
* Discuss data security, privacy and compliance requirements in data science.
* Describe the role of a data engineer and the increasing necessity, demand for the role in various industries.
 |
| 2 | **Data Modeling, Representation, and basic storage abstractions for data science** | * Describe the differences between structured and unstructured data.
* Explore the importance of data modeling and data object representation.
* Describe the relational database model. Explore the concept of schema, schema enforcement, ACID and normalization in relational databases.
* Compare the advantages and disadvantages of using data storage models (Files, Databases, BLOB, Data Lake, Warehouses) in the data science life cycle.
* Explore the various types of database systems. Describe the importance of various database systems in data modeling and data object representation.
* Describe the data model, operations and use cases of various NoSQL database types.
* Explore various storage abstractions for solving data science problems.
 |
| 3 | **Extraction, Transformation, and Loading** | * Describe the process of data extraction, transformation and loading (ETL) for data preparation and analysis.
* Explore the various components of a typical ETL lifecycle.
* Describe the various stages of data transformation: data cleaning and preparation, data wrangling, data crunching.
* Describe the popular approaches and tools for data cleaning and data processing.
* Distinguish between data preparation, data wrangling and data crunching.
 |
| 4 | **Data Exploration, Visualization, and Feature engineering** | * Describe the importance of feature engineering and feature selection for predictive modeling.
* Emphasize the need to inspect and visualize data to identify discriminating features.
* Describe the iterative process of data exploration, feature extraction and feature selection.
* Describe the various approaches to exploratory data analysis and explore the various data visualization tools on the cloud.
* Describe and distinguish between the various feature engineering methods.
* Explore the basic ETL pipeline for a real world application.
 |
| 5 | **Big data and batch processing frameworks** | * Define batch processing and discuss several popular use-cases.
* Explore the advantages of parallel and distributed processing frameworks (MapReduce, Spark, etc.) for big data analytics and data mining.
* Compare big data processing frameworks such as MapReduce and Spark.
* Explain the concept of RDDs, transformations and actions, and their roles in the execution of a Spark job.
* Discuss the differences between RDD, DataFrame, and Datasets.
 |
| 6 | **Real-time stream data and stream processing frameworks** | * Discuss the characteristics of real-time stream data.
* Identify real-world scenarios with real-time data streams where stream processing is adopted.
* Describe the concept of message queues, partitioning and fault tolerance.
* Explain the general requirements of stream processing systems.
* Describe the data flow model and the characteristics of contemporary stream processing frameworks such as Kafka and Samza.
 |
| 7 | **Advanced data storage abstractions for data science** | * Recall the various types of data and data access patterns.
* Describe the properties of binary data and explore object or BLOB storage on the cloud.
* Describe the data model, access pattern and use cases of data lake storage.
* Compare and contrast the functionality and the feature differences between object storage and data lake storage.
* Design a common workflow of data ingestion and data analytics on a data lake.
* Discuss data lake solutions and managed analytical engines on the cloud.
* Compare relational database systems and data warehouses. Identify the use-cases of data warehouses in the data science life cycle.
* Discuss data warehouse solutions on the cloud.
 |
| 8 | **Schedule, Automate and Monitor Data Pipelines in Production** | * Describe the benefits of adopting data pipelines in production systems.
* Describe the process of architecting, scheduling and monitoring data pipelines in production.
* Discuss how data pipelines can be configured to meet the data security, privacy and compliance requirements.
* Compare data pipeline orchestration tools like Apache Airflow, AWS Glue, Azure Data Factory.
* Industry Use Case 1 - Netflix Recommendation System.
* Industry Use Case 2 - Google Search Indexing Pipeline.
 |

# Projects

The projects are geared towards providing hands-on experience. Students will learn to develop all projects using various public cloud services. For each project, students are expected to work within a specified budget otherwise they risk being penalized, and fulfill the following learning objectives.

## Project 1: Data Exploration

* Inspect and visualize data to gain measurable insight using Linux tools such as awk and grep.
* Adopt Python libraries such as Pandas to solve data science problems with interactive programming using Jupyter Notebook.
* Experiment using Terraform to orchestrate and manage resources on the cloud.

## Project 2: Relational Databases

* Explore and perform data extraction, transformation and loading into an Azure database for MySQL.
* Apply the relational database model and design a MySQL database schema a real world application scenario.
* Practice the basics of Structured Query Language to gather insights from data.
* Practice using Structured Query Language and the JDBC API to gather insights from SQL databases.
* Implement and evaluate indexing in SQL databases to improve the performance of queries.
* Configure managed SQL database services on Azure as per the best-practices for data security and fault tolerance.
* Practice using Azure Resource Manager to provision and manage database instances on Azure.

## Project 3: ETL using NoSQL Databases

* Perform ETL and load the resulting data into a MongoDB database.
* Compare the advantages and disadvantages of SQL and NoSQL databases and their suitable applications in the data science life cycle.
* Implement and evaluate MongoDB indexing to improve the performance of MongoDB queries
* Practice writing efficient MongoDB queries using the Mongo API to gather insights from data.
* Explore and compare multi-model databases with those primarily supporting only one model.
* Deploy, configure and populate a fully managed multi-model database on Azure.
* Practice using Azure Resource Manager to orchestrate and manage Cosmos DB instances.

## Project 4: Feature Engineering

* Adopt Python libraries such as Pandas to solve data science problems with interactive programming using Jupyter Notebook.
* Explore the characteristics of a real-world dataset and recognize outliers and missing data.
* Transform features in a given dataset to lend itself better to machine learning.
* Construct new discriminating features using feature engineering methods to improve the accuracy of ML based predictors.
* Train a machine learning model (XGBoost) predictor utilizing the constructed features and training data. Evaluate impact of different features on the accuracy of the predictor.
* Experiment using Terraform to orchestrate and manage resources on the cloud.

## Project 5: Iterative Batch Processing using Spark

* Explore the concept of batch processing using MapReduce and Spark.
* Compare the difference between non-iterative processing (Hadoop) and iterative processing (Spark).
* Explain the concept of Resilient Distributed Dataset (RDD), transformations and actions, and their roles in the execution of Spark job.
* Discuss the differences between RDD and Dataframe. Identify the application scenarios for each of the APIs.
* Develop Spark applications progressively with interactive programming using Azure Databricks.
* Analyze a Spark application to identify potential bottlenecks through monitoring. Practice performance tuning of Spark jobs given limited resources.

## Project 6: Real Time Stream Processing

* Explore and define the characteristics of stream data.
* Identify scenarios with real-time data streams where stream processing is adopted.
* Distinguish between general purpose open-source stream processing frameworks and fully managed stream processing engines in cloud.
* Implement a real-time message ingestion system through Azure Event Hub framework.
* Implement a stream processing pipeline to process streams of data using Stream Analytics.
* Use managed ML services on the cloud to analyse stream data and influence decision making in real-time.
* Deploy stream processing engines such as EventHub and Stream Analytics using Infrastructure as Code

## Project 7: Big Data Analytics on a Data Lake

* Explore and compare Data Lake Storage with Blob Storage.
* Evaluate the advantages and disadvantages of using a Data Lake in a given scenario.
* Deploy data pipelines that collect and transform data from multiple sources into a Data Lake.
* Implement data transformations to preprocess unstructured or semi-structured data in a Data Lake for querying.
* Inspect, analyze and visualize data using Azure Kusto.
* Automate the process of data ingestion, analysis and visualization on a Data Lake by deploying an end-to-end data pipeline using cloud-based workflow management services.

## Project 8: Massive Parallel Processing using a Data Warehouse

Provision, configure, load and query data using MPP capabilities of Azure SQL Data Warehouse.

#  Schedule

The tentative schedule is as follows (specific deadlines are posted on OLI and Sail()):

##### 15-week option:

| **Week** | **Conceptual Content on OLI** | **Quiz** | **Hands-on Projects on Sail()** |
| --- | --- | --- | --- |
| 1 |  M1. Data Landscape | Quiz 1 | P0. Getting Started with Cloud & Jupyter Notebooks |
| 2 | No New Content Week (NNCW) |  | P1. Data Exploration |  |
| 3 | M2. Data Modeling, Representation, and basic storage abstractions for data science | Quiz 2 | P2. Relational Databases  |
| 4 | No New Content Week (NNCW) |  |  |
| 5 | M3. Extraction, Transformation, and Loading | Quiz 3 | P3. ETL using NoSQL Databases  |  |
| 6 | NNCW |  | P4. Feature Engineering |
| 7 | M4. Data Exploration, Visualization, and Feature engineering | Quiz 4 |  |
| 8 | NNCW |  | P5. Iterative Batch Processing using Spark |  |
| 9 | M5. Big data and batch processing frameworks | Quiz 5 |
| 10 | NNCW |  |  | P6. Real Time Stream Processing |
| 11 | M6. Real-time stream data and stream processing frameworks | Quiz 6 |  |
| 12 | NNCW |  | P7. Big Data Analytics on a Data Lake |  |
| 13 | M7. Advanced data storage abstractions for data science | Quiz 7 |
| 14 | NNCW |  | P8. Massive Parallel Processing using a Data Warehouse |
| 15 | M8. Schedule, Automate and Monitor Data Pipelines in Production | Quiz 8 |  |

##### 8-week option:

| **Week** | **Conceptual Content on OLI** | **Quiz** | **Hands-on Projects on Sail()** |
| --- | --- | --- | --- |
| 1 |  |  | P0. Getting Started with Cloud & Jupyter Notebooks |
| 2 | M1. Data Landscape | Quiz 1 | P1. Data Exploration |
| 3 | M2. Data Modeling, Representation, and basic storage abstractions for data science | Quiz 2 | P2. Relational Databases  |
| 4 | M3. Extraction, Transformation, and Loading | Quiz 3 | P3. ETL using NoSQL Databases  |
| 5 | M4. Data Exploration, Visualization, and Feature engineering | Quiz 4 | P4. Feature Engineering |
| 6 | M5. Big data and batch processing frameworks | Quiz 5 | P5. Iterative Batch Processing using Spark |
| 7 | M6. Real-time stream data and stream processing frameworks | Quiz 6 | P6. Real Time Stream Processing |
| 8 | M7. Advanced data storage abstractions for data science | Quiz 7 | P7. Big Data Analytics on a Data Lake |
| 9 | M8. Schedule, Automate and Monitor Data Pipelines in Production | Quiz 8 | P8. Massive Parallel Processing using a Data Warehouse |

# Accommodations for Students with Disabilities

If you have a disability and have an accommodations letter from the Disability Resources office, I encourage you to discuss your accommodations and needs with me as early in the semester as possible. I will work with you to ensure that accommodations are provided as appropriate. If you suspect that you may have a disability and would benefit from accommodations but are not yet registered with the Office of Disability Resources, I encourage you to contact them at access@xyz.edu.

#  Take care of yourself

Do your best to maintain a healthy lifestyle this semester by eating well, exercising, avoiding drugs and alcohol, getting enough sleep and taking some time to relax. This will help you achieve your goals and cope with stress.

All of us benefit from support during times of struggle. You are not alone. There are many helpful resources available on campus and an important part of the college experience is learning how to ask for help. Asking for support sooner rather than later is often helpful.