LAUNCH -- Python for Data Science

Curriculum/Content Area: Mathematics/Data Science (LAUNCH)	**Course Length:** 2 Terms
Course Title: Python for Data Science	**Date last reviewed:** 2021
Prerequisites: No formal prerequisite.	**Board approval date:** February 2021

Secondary Resources and Teacher Reference Materials:

- DataQuest ... DataQuest Free Classroom Plan
- DataCamp ... DataCamp Free Classroom Plan
- How to Think Like a Computer Scientist: Learning with Python 3
- The Python Tutorial ... Documentation available from www.python.org.
- LinkedIn Learning -- Python Essential Training
- Think Python
- TowardsDataScience (blog)
- Sharp Sight
- w3schools.com Python Tutorials
- Python for Data Analysis [W. McKinney]
- Python Data Science Handbook [J. Vanderplas]
- Introduction to Computation and Programming Using Python [J. Guttag]
- Become a Python Data Analyst [A. Fuentes]
- Python Programming Language [D. Beazley]
- Memorable Python [J. Hale]
- The Quick Python Book [Cedar, Naomi]
- A Better Way to Learn Python [M. Myers]

Desired Results

Course Description and Purpose: This course introduces core features of the Python programming language, while demonstrating and utilizing fundamental concepts in computer science. It provides an in-depth discussion of data representation strategies, showing how data structures are implemented in Python along with demonstrating tools for data science and software engineering. While working on data analysis problems and data manipulation tasks, students will employ various programming paradigms, including functional programming, object-oriented programming, and data stream processing. Special attention is paid to the standard Python library and packages for analytics and modeling (Pandas, Numpy, Matplotlib, etc.).

Enduring Understandings:	**Essential Questions**:
❖ Mathematicians and Data Scientists make sense of problems and persevere in solving them. ❖ Mathematicians and Data Scientists	How can I use mathematics in data science to make sense of the world? What strategies and tools transcend all mathematical and data science problems, and how

reason abstractly and quantitatively. ❖ Mathematicians and Data Scientists embrace creative development as an essential process for creating computational artifacts ❖ Mathematicians and Data Scientists construct viable arguments and critique the reasoning of others. ❖ Mathematicians and Data Scientists model with mathematics. ❖ Mathematicians and Data Scientists use appropriate tools strategically. ❖ Mathematicians and Data Scientists attend to precision. ❖ Mathematicians and Data Scientists look for and express regularity in repeated reasoning.	can I apply those strategies/tools in unique settings? How can we as mathematicians and data scientists evaluate and question whether an argument is accurate? How can mathematics, computational models, and simulations help make predictions, generate new understandings, and solve problems? How can computing and the use of computational tools foster creative expression?

PRIORITY STANDARDS	MEANING- *(The Priority Standards help students construct understanding of…)*
The Python for Data Science Course Skills Priority Standards are distinct skills that are integrated throughout the course and derived from Elmbrook Mathematical Priority Standards & Progressions, Advanced Placement Calculus (APC) and Advanced Placement Computer Science Principles (APCS). These standards ensure our Elmbrook Scholars learn to think and act like data science modelers and problem solvers, and are authentically integrated in each unit through the instructional approach of problem-based, experiential learning.	
➔ APCS 1B- COMPUTATIONAL SOLUTION DESIGN: Determine and design an appropriate method or approach to achieve a purpose. ➔ APCS2.B- ALGORITHMS AND PROGRAM DEVELOPMENT: Implement and apply an algorithm ➔ APC1.D-IMPLEMENTING MATHEMATICAL PROCESSES: Identify an appropriate mathematical rule or procedure based on the relationship between concepts or processes to solve problems. ➔ APCS5.A- COMPUTING INNOVATIONS: Explain how computing systems work. ➔ APCS5.B- COMPUTING INNOVATIONS:	1. Developing a structured and conceptual understanding of the Python programming language along with incorporating best practice computer science methods. *Build a solid technology/coding skill base and programming foundation that will position students to:* ● *readily learn both new technologies and more advanced programming concepts* ● *effectively use coding as a complementary skill that can be applied to other disciplines and to a variety of scenarios* ● *efficiently earn and stack*

Explain how knowledge can be generated from data ➔ APCS5.C- COMPUTING INNOVATIONS: Describe the impact of computing innovation. ➔ APCS5.D- COMPUTING INNOVATIONS: Describe the impact of gathering data	*credentials in a number of data-related areas* *2.* Making coding skills more of a mainstream discipline. *Create a dynamic where students from a variety of disciplines -- not just computer science -- can transfer their coding skills in complementary ways to other topics and future courses. Treat coding as a gateway/lynchpin skill that opens up the floodgates of learning in many new and relevant ways.* *3.* Fostering the ability to find answers to questions and solutions to problems. *Learning how to figure out a solution when it's not in the textbook. Developing the capacity to identify and access resources to find answers and solutions is the biggest lesson. The answer is out there -- you just have to know how to find it.* *4.* Equipping students with the tools they will need to become effective data analysts. *Providing students with the nuts and bolts of how to manipulate, process, clean, wrangle, crunch, and visualize data in Python.* *5.* Leveraging skills across different domains *Use coding skills to solve domain area problems and answer/raise domain area questions.* *6.* Exposing students to the vast data-oriented Python library ecosystem. *Provide avenues for students to learn how to access and take advantage of the additional functionality that Python provides in several other data-related areas (e.g. modeling, reporting, machine learning, web scraping, etc.).*

Module #1 Python Installation and Introduction

Essential Unit Questions

1. How can I use mathematics in data science to make sense of the world?
2. How can computing and the use of computational tools foster creative expression?

Guiding Content Questions

1. What is the single most important skill for a computer scientist?
2. What is a program?
3. What is debugging and what different types of errors can occur when writing and executing a program?
4. What is the core philosophy behind Python?
5. What is Anaconda and what is the main advantage of using Anaconda?
6. What is Jupyter Notebook/IPython Notebook?

Learning Targets:

- I can install the Anaconda Distribution of Python.
- I understand the key features of the Anaconda Distribution of Python.
- I can launch Jupyter Notebook from within the Anaconda Distribution of Python.
- I can interact with Python using both the command prompt and Python shell.
- I can perform basic print commands and debugging techniques.
- I can describe the overall structure of Python and its benefits.
- I can explain the difference between a high-level programming language and a low-level programming language and describe the advantages of a high level language?
- I understand how to write comments and I know what they are used for.
- I can describe the key differences between Python 3 and Python 2.
- I can explain what debugging is.
- I can identify the different types of errors that can occur when writing and executing a program.

Assessment Evidence:

Performance Assessment Options *May include, but are not limited to the following:*	Other assessment options *May include, but are not limited to the following:*
• Problem Sets • Project-based/Problem-based activities • Unit Assessment • Coding Tasks • Feedback on Success/Professional Skills	• Project reflection

Digital Tools & Supplementary Resources:

Python software, Dataquest, DataCamp, How to Think Like a Computer Scientist: Learning with Python 3, The Python Tutorial, LinkedIn Learning -- Python Essential Training, Think Python, Python for Data Analysis [W. McKinney], Python Data Science Handbook [J. Vanderplas], Introduction to Computation and Programming Using Python [J. Guttag], Become a Python Data Analyst [A. Fuentes], Python Programming Language [D. Beazley], Memorable Python [J. Hale], The Quick Python Book [Cedar, Naomi], A Better Way to Learn Python [M. Myers], TowardsDataScience (blog), Sharp Sight, w3schools.com Python Tutorials

Module #2 Python Fundamentals and Basics

Essential Unit Questions

1. How can I use mathematics in data science to make sense of the world?
2. What strategies and tools transcend all mathematical and data science problems, and how can I apply those strategies/tools in unique settings?
3. How can mathematics, computational models, and simulations help make predictions, generate new understandings, and solve problems?

Guiding Content Questions

1. What is some of the basic vocabulary of programming?
2. What are some of the fundamental building blocks of Python?
3. What are some best practice debugging techniques?

Learning Targets:

- I understand what a value is in Python.
- I can determine and recognize the data type (class) of a value.
- I can convert values from one type to another.
- I understand what a Python variable is and I can determine if a variable name is legal or not.
- I am aware of the different types of statements that the Python interpreter can handle (e.g. assignment, while, for, if, import, etc.).
- I understand what a Python expression is and what it means to evaluate a Python expression.
- I can recognize the different Python operators and I understand the rules of precedence that determine the order of evaluation.
- I can perform mathematical calculations using operators.
- I understand variable assignment, variable reassignment, and how to update a variable.
- I understand what Boolean values and Boolean expressions are and I understand how they can be used.
- I can recognize and understand the purpose, structure, and syntax of the different types of conditional statements/selection statements that there are.
- I can write and execute conditional statements.
- I can discuss the algorithmic process of programs.
- I am aware of tips that will help me avoid debugging and tips that will help me debug.
- I know how to read and interpret the different types of error messages.

Assessment Evidence:

Performance Assessment Options
May include, but are not limited to the following:

- Problem Sets
- Project-based/Problem-based activities
- Unit Assessment
- Coding Tasks
- Feedback on Success/Professional Skills

Other assessment options
May include, but are not limited to the following:

- Project reflection

Digital Tools & Supplementary Resources:

Python software, Dataquest, DataCamp, How to Think Like a Computer Scientist: Learning with Python 3, The Python Tutorial, LinkedIn Learning -- Python Essential Training, Think Python, Python for Data Analysis [W. McKinney], Python Data Science Handbook [J. Vanderplas], Introduction to Computation and Programming Using Python [J. Guttag], Become a Python Data Analyst [A. Fuentes], Python Programming Language [D. Beazley], Memorable Python [J. Hale], The Quick Python Book [Cedar, Naomi], A Better Way to Learn Python [M. Myers], TowardsDataScience (blog), Sharp Sight, w3schools.com Python Tutorials

Module #3 Python Functions and Modules

Essential Unit Questions

1. What strategies and tools transcend all mathematical and data science problems, and how can I apply those strategies/tools in unique settings?
2. How can mathematics, computational models, and simulations help make predictions, generate new understandings, and solve problems?
3. How can computing and the use of computational tools foster creative expression?

Guiding Content Questions

1. What is the best way to get help about the Python programming environment?
2. What is a Python module?
3. What is a Python function?
4. What is functional decomposition and why is it important?
5. What is iteration and what are the different types of Python loops?

Learning Targets:

- I can use the Python Documentation site to get help about the Python programming environment.
- I understand what a Python module is and I know how to access a list of all the modules that are available as part of the Python standard library.
- I can create Python modules.
- I understand what a Python function is and I can explain the main purpose of a function.
- I can define/create new Python functions using proper function syntax and structure.
- I understand how functions can be used and called from other functions.
- I can perform calculations using built-in functions.
- I understand how to use functional decomposition to break a problem into smaller subproblems.
- I understand what the flow of execution of a function means and how it works.
- I can use composition to build functions from other functions.
- I can create and execute for, while, and nested loops.
- I understand the importance of comments within a program.

Assessment Evidence:

Performance Assessment Options *May include, but are not limited to the following:*	Other assessment options *May include, but are not limited to the following:*
• Problem Sets • Project-based/Problem-based activities • Unit Assessment • Coding Tasks • Feedback on Success/Professional Skills	• Project reflection

Digital Tools & Supplementary Resources:

Python software, Dataquest, DataCamp, How to Think Like a Computer Scientist: Learning with Python 3, The Python Tutorial, LinkedIn Learning -- Python Essential Training, Think Python, Python for Data Analysis [W. McKinney], Python Data Science Handbook [J. Vanderplas], Introduction to Computation and Programming Using Python [J. Guttag], Become a Python Data Analyst [A. Fuentes], Python Programming Language [D. Beazley], Memorable Python [J. Hale], The Quick Python Book [Cedar, Naomi], A Better Way to Learn Python [M. Myers], TowardsDataScience (blog), Sharp Sight, w3schools.com Python Tutorials

Module #4 Python Lists

Essential Unit Questions

1. What strategies and tools transcend all mathematical and data science problems, and how can I apply those strategies/tools in unique settings?
2. How can mathematics, computational models, and simulations help make predictions, generate new understandings, and solve problems?
3. How can computing and the use of computational tools foster creative expression?

Guiding Content Questions

1. What is a Python list?
2. How can Python lists be created and manipulated?
3. How can I read and write a file in Python?

Learning Targets:

- I can create lists in a variety of ways.
- I can determine the length of a list.
- I can access the elements of a list.
- I can append a list.
- I can concatenate lists and repeat items in a list a given number of times.
- I can delete elements from a list.
- I can clone a list.
- I can take slices from a list.
- I understand what a nested list is and I know how to extract elements from a nested list.
- I can use the split method to break a string into a list of words.
- I can use the dot operator to access built-in methods of list objects.
- I can perform list traversal using both iteration by item and iteration by index.
- I understand the similarities and differences between tuples and lists and I know how to perform tuple assignment.
- I can read data from various file types.
- I can find files on a disk.
- I can open and close files.
- I can use a variety of methods to read data from an input file.
- I can discuss the flexibility of reading files with Python.
- I can write data to an output file.

Assessment Evidence:

Performance Assessment Options
May include, but are not limited to the following:

- Problem Sets
- Project-based/Problem-based activities
- Unit Assessment
- Coding Tasks
- Feedback on Success/Professional Skills

Other assessment options
May include, but are not limited to the following:

- Project reflection

Digital Tools & Supplementary Resources:

Python software, Dataquest, DataCamp, How to Think Like a Computer Scientist: Learning with Python 3, The Python Tutorial, LinkedIn Learning -- Python Essential Training, Think Python, Python for Data Analysis [W. McKinney], Python Data Science Handbook [J. Vanderplas], Introduction to Computation and Programming Using Python [J. Guttag], Become a Python Data Analyst [A. Fuentes], Python Programming Language [D. Beazley], Memorable Python [J. Hale], The Quick Python Book [Cedar, Naomi], A Better Way

to Learn Python [M. Myers], TowardsDataScience (blog), Sharp Sight, w3schools.com Python Tutorials

Module #5 EDA Using Python

Essential Unit Questions

1. How can mathematics, computational models, and simulations help make predictions, generate new understandings, and solve problems?

How can computing and the use of computational tools foster creative expression?

Guiding Content Questions

1. What are the major steps in tackling a data science problem?
2. Why do data scientists need a defined, systematic methodology and approach?
3. What does it mean to understand data?
4. What does it mean to clean, prepare, and wrangle data?
5. What is an EDA (Exploratory Data Analysis) and why is it important?
6. How do you form a concrete business/research question upon which you will perform the data science analysis?
7. What are the most widely used Python libraries for data analysis?

Learning Targets:

- I can use EDA techniques for cleaning and wrangling data.
- I understand what the major steps are in tackling a data science problem?
- I understand why data scientists need a defined, systematic methodology and approach.
- I understand what it means to understand data.
- I understand what EDA (Exploratory Data Analysis) is and why it is important.
- I can form a concrete business/research question upon which I will perform a data science analysis?
- I understand how NumPy uses vectorized operations.
- I can create NumPy arrays from lists and from built-in NumPy functions,
- I can discuss the attributes of a NumPy array.
- I can perform basic math with NumPy arrays.
- I can index, slice, and reshape NumPy arrays.
- I understand the data types associated with Numpy arrays.
- I can transpose a Numpy array.
- I can create a pandas series and a pandas DataFrame.
- I understand the anatomy of a DataFrame.
- I can manipulate data to remove null, missing, duplicate, and irrelevant values.
- I can combine and merge data sets.
- I can select a column or subset of columns.
- I can analyze data to locate outliers and other anomalies.
- I can perform methods on data to determine its shape and structure.

Assessment Evidence:

Performance Assessment Options *May include, but are not limited to the following:*	Other assessment options *May include, but are not limited to the following:*
• Problem Sets • Project-based/Problem-based activities • Unit Assessment • Coding Tasks • Feedback on Success/Professional Skills	• Project reflection

Digital Tools & Supplementary Resources:

Python software, Dataquest, DataCamp, How to Think Like a Computer Scientist: Learning with Python 3, The Python Tutorial, LinkedIn Learning -- Python Essential Training, Think Python, Python for Data Analysis [W. McKinney], Python Data Science Handbook [J. Vanderplas], Introduction to Computation and Programming Using Python [J. Guttag], Become a Python Data Analyst [A. Fuentes], Python Programming Language [D. Beazley], Memorable Python [J. Hale], The Quick Python Book [Cedar, Naomi], A Better Way to Learn Python [M. Myers], TowardsDataScience (blog), Sharp Sight, w3schools.com Python Tutorials

Module #6 Python Data Visualization

Essential Unit Questions

1. How can mathematics, computational models, and simulations help make predictions, generate new understandings, and solve problems?
2. How can computing and the use of computational tools foster creative expression?

Guiding Content Questions

1. What are the most widely used Python libraries for data visualization?
2. What are the most common customizations that can be done using the Python data visualization libraries?
3. How can the Python data visualization libraries be used to examine variables and enhance an EDA?

Learning Targets:

- I understand Matplotlib terminology.
- I can perform common customizations (colors, setting axis limits, setting ticks and tick labels, legend, annotations, grids) using the Python visualization libraries.
- I can perform EDA with the Python libraries seaborn and pandas.
- I can create several different types of plots (e.g. scatterplots, box plots, histograms, etc.).
- I can apply Python visualization methods to recognize the distribution of data.
- I can analyze both numerical and categorical variables individually using the Python visualization libraries.
- I can analyze the relationship between variables using the Python visualization libraries.
- I can manipulate data as needed for EDA purposes.
- I can use the Python visualization libraries to tell a compelling data analysis story.

Assessment Evidence:

Performance Assessment Options
May include, but are not limited to the following:

- Problem Sets
- Project-based/Problem-based activities
- Unit Assessment
- Coding Tasks
- Feedback on Success/Professional Skills

Other assessment options
May include, but are not limited to the following:

- Project reflection

Digital Tools & Supplementary Resources:

Python software, Dataquest, DataCamp, How to Think Like a Computer Scientist: Learning with Python 3, The Python Tutorial, LinkedIn Learning -- Python Essential Training, Think Python, Python for Data Analysis [W. McKinney], Python Data Science Handbook [J. Vanderplas], Introduction to Computation and

Programming Using Python [J. Guttag], Become a Python Data Analyst [A. Fuentes], Python Programming Language [D. Beazley], Memorable Python [J. Hale], The Quick Python Book [Cedar, Naomi], A Better Way to Learn Python [M. Myers], TowardsDataScience (blog), Sharp Sight, w3schools.com Python Tutorials

Module #7 Python Strings and Dictionaries/Web-Scraping in Python

Essential Unit Questions

1. How can mathematics, computational models, and simulations help make predictions, generate new understandings, and solve problems?
2. How can computing and the use of computational tools foster creative expression?

Guiding Content Questions

1. How do you work with strings in Python?
2. What are Python dictionaries and why are they useful?
3. Why is Python good for scraping data from the web?
4. How do you use Python to scrape data from the web?

Learning Targets:

- Execute statements related to variables, expressions, and control flow
- Create formatted output and write to a file
- Manipulate text strings, convert input data, and perform calculations
- Investigate uses for extracting internet data
- I can perform concatenation with strings.
- I can use the index operator to work with characters of a string.
- I can determine the number of characters in a string.
- I can select a slice from a string.
- I can use string comparison to see if two strings are equal.
- I can use both the for loop and the while loop to perform a traversal on a string.
- I understand that a Python dictionary is Python's built-in mapping type which allows associations to be made from a key to a value.
- I can access, change, add, and remove items from a dictionary.
- I can loop through a dictionary and copy a dictionary.
- I understand what nested dictionaries are.
- I am aware of a number of powerful dictionary methods and the purpose for each of these methods.
- I can use the Beautiful Soup Python package to scrape data from the web.

Assessment Evidence:

Performance Assessment Options
May include, but are not limited to the following:

- Problem Sets
- Project-based/Problem-based activities
- Unit Assessment
- Coding Tasks
- Feedback on Success/Professional Skills

Other assessment options
May include, but are not limited to the following:

- Project reflection

Digital Tools & Supplementary Resources:

Python software, Dataquest, DataCamp, How to Think Like a Computer Scientist: Learning with Python 3, The Python Tutorial, LinkedIn Learning -- Python Essential Training, Think Python, Python for Data

Analysis [W. McKinney], Python Data Science Handbook [J. Vanderplas], Introduction to Computation and Programming Using Python [J. Guttag], Become a Python Data Analyst [A. Fuentes], Python Programming Language [D. Beazley], Memorable Python [J. Hale], The Quick Python Book [Cedar, Naomi], A Better Way to Learn Python [M. Myers], TowardsDataScience (blog), Sharp Sight, w3schools.com Python Tutorials

Module #8 Python Exceptions/Reading and Writing Files

Essential Unit Questions

1. How can mathematics, computational models, and simulations help make predictions, generate new understandings, and solve problems?
2. How can computing and the use of computational tools foster creative expression?

Guiding Content Questions

1. What is an exception?
2. What are some fundamental principles for using exceptions?
3. How do you manipulate data from files?
4. What is the purpose of data structures?
5. How do you build data structures from files?

Learning Targets:

- I can handle different types of exceptions.
- I understand the exceptions syntax.
- I can define new exceptions.
- I understand paths and pathnames.
- I know how to get information about files.
- I can process all files in a directory subtree.
- I can open files and file objects.
- I can open files in write and other modes.
- I know how to close a file.
- I know which abstract data types and data structures are built into the Python standard library.
- I know the different types of data structures -- both primitive and non-primitive.
- I know how the most common abstract data types map to Python's naming scheme.
- I know how to put abstract data types to practical use in various algorithms.

Assessment Evidence:

Performance Assessment Options
May include, but are not limited to the following:

- Problem Sets
- Project-based/Problem-based activities
- Unit Assessment
- Coding Tasks
- Feedback on Success/Professional Skills

Other assessment options
May include, but are not limited to the following:

- Project reflection

Digital Tools & Supplementary Resources:

Python software, Dataquest, DataCamp, How to Think Like a Computer Scientist: Learning with Python 3, The Python Tutorial, LinkedIn Learning -- Python Essential Training, Think Python, Python for Data Analysis [W. McKinney], Python Data Science Handbook [J. Vanderplas], Introduction to Computation and Programming Using Python [J. Guttag], Become a Python Data Analyst [A. Fuentes], Python Programming Language [D. Beazley], Memorable Python [J. Hale], The Quick Python Book [Cedar, Naomi], A Better Way

to Learn Python [M. Myers], TowardsDataScience (blog), Sharp Sight, w3schools.com Python Tutorials

Module #9 Classes and Object-Oriented Programming

Essential Unit Questions

1. How can mathematics, computational models, and simulations help make predictions, generate new understandings, and solve problems?
2. How can computing and the use of computational tools foster creative expression?

Guiding Content Questions

1. What is object-oriented programming?
2. What are the pillars of object-oriented programming?
3. What is a class?
4. What is Python inheritance?
5. What is an object?
6. What is a package?

Learning Targets:

- I can describe object-oriented programming.
- I can create classes and attach methods.
- I can define and manipulate objects.
- I can create a parent class and a child class to utilize Python inheritance.
- I can create an instance.
- I can add properties and methods to the child class.
- I can organize code bases into packages.

Assessment Evidence:

Performance Assessment Options
May include, but are not limited to the following:

- Problem Sets
- Project-based/Problem-based activities
- Unit Assessment
- Coding Tasks
- Feedback on Success/Professional Skills

Other assessment options
May include, but are not limited to the following:

- Project reflection

Digital Tools & Supplementary Resources:

Python software, Dataquest, DataCamp, How to Think Like a Computer Scientist: Learning with Python 3, The Python Tutorial, LinkedIn Learning -- Python Essential Training, Think Python, Python for Data Analysis [W. McKinney], Python Data Science Handbook [J. Vanderplas], Introduction to Computation and Programming Using Python [J. Guttag], Become a Python Data Analyst [A. Fuentes], Python Programming Language [D. Beazley], Memorable Python [J. Hale], The Quick Python Book [Cedar, Naomi], A Better Way to Learn Python [M. Myers], TowardsDataScience (blog), Sharp Sight, w3schools.com Python Tutorials