

Course Syllabus

Course Information

Course: BUAN 6340 – Programming for Data Science
Section: 001
Term: Fall 2017
Meets: Saturday, 10:00 am to 12:45 pm, JSOM 2.717

Professor Contact Information

Instructor: Kevin R. Crook
Email: Kevin.Crook@utdallas.edu
Office Hours: Office hours will be following class in classroom JSOM 2.717
(as a part-time adjunct, I'm only on campus Friday nights and Saturday mornings)

MIS Lab with Python Help

MIS Lab – JSOM 2.711 - <http://jindal.utdallas.edu/student-resources/mis-lab/>

The JSOM MIS Lab will have a special Python expert on hand to help students. The lab will have numerous open hours over six days per week. It is strongly recommended that students who need help with this course use this MIS lab as much as they can this semester.

(next page)

Course Pre-requisites, Co-requisites, and/or Other Restrictions
--

- Formal prerequisites (one of the following):
 - BUAN 6356 – Business Analytics with R
 - MIS 6323 – Object Oriented Programming
- Recommended prerequisites

In prior semesters, students with these knowledge areas had a distinct advantage over students learning these for the first time and usually came out on top in rank grading:

- Knowledge of statistics
- Prior programming experience with an object-oriented language (if not Python, then Java, C++, etc.)
 - Variables, data types, bitwise manipulation, control structures, looping, file I/O including binary files, procedures, functions, classes, objects, inheritance, etc.
 - Data structures: stacks, queues, trees, heaps, etc.
 - Algorithms: searching, sorting, graph, etc.
 - Computational complexity – Big O Notation, etc.
- Debugging skills – as is the standard with any advanced programming course, the students must be able to debug any syntax errors, semantic errors, stack overflow, core dumps, etc. independently on their own.
- Software installation skills – as is the standard with any advanced programming course, the students must be able to install and configure software packages independently on their own and work through and resolve any issues independently on their own.
- Database skills – as a secondary activity, programming assignments may require students to read and create ERDs, create database tables, and insert, update, delete, and select data using SQL.
- Source code control skills – students will need to use GitHub for source code assignments and to store their code for the course for protection against loss

(next page)

Course Description

This course covers many aspects of programming for data science and analytics, including syntax, handling data, data visualization, and implementation of statistical analysis models. The course will be taught using Python language and may use a different programming language as applicable.

Specifics include:

(time permitting – we will cover as much as we can depending on how well prepared the class is in terms of prior programming skills)

- Python Programming Fundamentals
- Data Formats and Conversions: XML, JSON, CSV, UTF-8, Unicode, Code Pages, etc.
- Graphs and Networks
- Interfacing Python with Relational Databases using SQL
- Statistical Programming
- Data Wrangling, Data Munging, Data Cleansing
- Data Visualization
- Web Page Screen Scraping / Parsing HTML
- Web APIs / Representational State Transfer (REST) / Service Oriented Architecture (SOA)
- Social Media Analytics
- Machine Learning
 - from libraries
 - from source code
 - Scale Out architectures
- Natural Language Processing
- Deep Learning / Neural Networks
- Graphical Processing Unit (GPU) processing on graphics cards and in the cloud

Programming Labs:

- Scheduling Analytics
 - Programmatically creating and optimizing an airline schedule
- Healthcare Analytics
 - Recommender system for hospitals on Medicare ratings and patient surveys
- Market Basket Analytics
 - Given a purchasing history of products purchased together, recommend additional product to customers making purchases
- Social Media Analytics / Natural Language Processing / Geographic Data Visualization
 - Collecting Tweets from Twitter, performing sentiment analysis using natural language processing, and displaying the results as a geographic data visualization

Student Learning Objectives/Outcomes

- To understand the fundamental programming concepts of data science environments including their solution design, implementation and presentation by mapping technology strategy of an enterprise to its business requirements.
- Data Extraction and filtering from a large dataset stored in a variety of different formats.
- Perform complex statistical analysis on datasets.
- Predictive modeling on datasets using advanced analytical techniques as well as using machine learning algorithms and natural language processing techniques.
- Interactively visualize large datasets and present the results as a driver for adaptive business intelligence.

(next page)

Required Textbooks and Materials

Laptop:

Students are required to have a Windows 10 (64-bit) laptop, legitimately licensed copy, which they need to bring to class. Some of the software may run on older versions of Windows, but it will be at your own risk and responsibility. The software will run on 4 GiB of RAM, but more memory is better.

If you have a Mac, you can run the software used in this class on the Mac, but please be advised that students in the past have had major issues with the Mac and the Mac lack of competent technical support. If you use a Mac, you do so at your own risk and must resolve any technical issues on your own.

Software:

This course will use the Anaconda managed data science distribution of Python and associated libraries. We will use Python 3.x. It can be downloaded and installed at the following link: <https://www.continuum.io/downloads>

Textbooks:

Textbooks for this course are available online for free from the UTD library

<http://www.utdallas.edu/library>

They are also available if you have a Safari Books account, safaribooksonline.com

Data Science Essentials in Python

By Dmitry Zinoviev, Pragmatic Bookshelf
(we will use almost all of this book)

Hands-On Machine Learning with Scikit-Learn and TensorFlow

By Aurelien Geron, O'Reilly Media, Inc.
(we will use all of the Machine Learning chapters in this book and as much of Neural Net as we can time permitting)

Whirlwind Tour of Python - <https://github.com/jakevdp/WhirlwindTourOfPython>

By Jake VanderPlas
(it's a GitHub repo of Jupyter Notebooks that make it really easy to learn Python basics)

Introducing Python

By Bill Lubanovic, O'Reilly Media, Inc.
(students will be assigned chapters in this book to work through outside of class)

Assignments & Academic Calendar

Class Date	Tentative Agenda <ul style="list-style-type: none"> • Subject to change without prior notice at the instructor's discretion • This course will more adaptive in nature than most students are used to due the variation of programming backgrounds and experience among students • Below represents the best case where we move the fastest
August 26, 2017	Python Programming Fundamentals
September 2, 2017	Data Formats and Conversions, Graphs and Networks
September 9, 2017	Interfacing with Relational Databases using SQL, Statistical Programming
September 16, 2017	Data Wrangling, Data Munging, Data Cleansing, Data Visualization
September 23, 2017	Web Page Screen Scraping, Parsing HTML, Web APIs, Representational State Transfer (REST), Service Oriented Architecture (SOA)
September 30, 2017	Social Media Analytics, Natural Language Processing
October 7, 2017	Exam 1
October 14, 2017	Machine Learning
October 21, 2017	Machine Learning
October 28, 2017	No Class (Instructor will be presenting at a machine learning conference at UC Berkeley)
November 4, 2017	Machine Learning
November 11, 2017	Deep Learning / Neural Nets
November 18, 2017	Deep Learning / Neural Nets
November 25, 2017	No Class – Thanksgiving / Fall Break
December 2, 2017	Exam 2

Grading Policy

Rank Grading System

- Students will first be ranked based on the number of programming labs they have completed on time with a grade of pass
- Students will then be ranked based on the average of their scores for the exams with each exam counting equal weight
- Ties (if any) will be broken:
 - first ties: based on the average turn in time of the exams
 - further ties: based on the average turn in time for programming labs
- The rank grading scale will be no harsher than the following:
 - A = top 15 %
 - A- = next 10 %
 - B+ = next 10%
 - B and below for the remaining rankings at the instructor's sole discretion
- Students will be given a grade of F for any one of the following:
 - Not taking exam 1
 - Not taking exam 2
 - Not turning in at least 3 of the programming labs on time

Programming Labs

- Each programming lab will be graded as pass / fail
- The criteria for passing a lab is 65% correct completion of the lab objectives
- A grade of fail will be given if a lab is late (1 second late is late)

Exams

- 2 Exams
- Will be rigorous and very difficult written exams
- Closed book, closed note, no "cheat sheets", no electronic devices of any kind, no communication in any form between a student and anyone other than the instructor or proctor
- Content will be taken from:
 - Programming Labs
 - Reading assignments, even if not covered in class
 - Expect 30% of exam material to be from reading assignments not covered in class to ensure students are rigorously studying the materials assigned
 - Anything covered in class, even material only discussed verbally
 - Any material from the required prerequisites will be assumed and may be covered on exams to ensure that students have not received waivers based on false or misleading information

Course & Instructor Policies

- **Professional Conduct** - expected at all times from all students.

Examples of unprofessional conduct includes, but is not limited to:

- Trying the “game the system” in any manner
 - When told “no” not accepting “no” for an answer and not dropping the matter
 - Making special requests for favors unless every student is given that same favor
 - Attempting to negotiate for grades – grades are not negotiable
 - Coming late to class
 - Leaving early from class
 - Talking in class
 - Working on other subjects while in class
 - Using electronic devices for other than the course materials
- **Class Attendance** – Required, must be on-time, instructor may refuse entry to late arrivals after class has started
 - **Make-up Programming Labs** – Not permitted
 - **Make-up Exams** – Not permitted
 - **Extra Credit** – Not permitted
 - **Late Work** – Not permitted
 - **Instructor Response Policy** - Instructor is only part time and has limited time to answer emails. Instructor will check email on Fridays and respond time permitting, so it’s best to ask questions in class, visit the MIS Lab, or stay after class for the instructor’s office hours.
 - **Regrade Requests** – Regrade requests will involve the instructor regrading the entire semester. Incorrect items that were previously overlooked or waived will be applied. The rank grading scale published in the syllabus will be applied even if a more lenient rank grading scale was applied. Students in many cases will receive a lower grade than their original grade, so regrade requests should be saved for legitimate items and not as an attempt to “game the system” for a higher grade.

(next page)

Comet Creed

This creed was voted on by the UT Dallas student body in 2014. It is a standard that Comets choose to live by and encourage others to do the same:

“As a Comet, I pledge honesty, integrity, and service in all that I do.”

UT Dallas Syllabus Policies and Procedures

The information contained in the following link constitutes the University’s policies and procedures segment of the course syllabus.

Please go to <http://go.utdallas.edu/syllabus-policies> for these policies.

The descriptions and timelines contained in this syllabus are subject to change at the discretion of the Professor.