Practical Computational Methods for Economists (496-05 and 895-012)
Innovation Hall, Room 129; 3:00 – 4:15 (Tuesday, Thursday)

Instructor: Professor Kevin McCabe, kmccabe@gmu.edu, (o) 703-993-9441.
All EMAIL correspondence should use the following subject lines.
Undergraduates: ECON 496 FALL 2018
Graduate: ECON 895 FALL 2018.

Office Hours: Wednesdays, 2:00-4:00, Room 5016, Vernon Smith Hall, Arlington.

Prerequisite: Intermediate Microeconomics or Graduate Microeconomics

Course Description: The field of economics is becoming increasingly computational. If you want to be competitive in the job market, you must learn how to think in terms of algorithms and be able to program algorithmic computations on computers. This class introduces you to computational methods using Python and the Python Scientific stack, and mTree, a GMU Python library for running computational economics experiments.

This course is divided into sixteen instructional notebooks (called primers) designed to teach you basic Python programming and computational methods. In addition, there are five project notebooks which have you apply the instructional primers to study the economics of risky decision making. We will be working with the standard Arrow-Pratt utility theory that assumes people evaluate risky prospects using expected utility. We will then extend the model to a behavioral model by introducing errors into the decision making process and studying how these errors will affect our ability to elicit people's risk preferences.

By the end of the course, you will have a solid understanding of Python, the Python Scientific Libraries and mTree, and you will have built a simulation model of risky decisions in mTree.

Course Materials

Laptops: Students should bring their laptops to class. On the first day we will download the software you will use for the class.

Textbook for Beginners: If you are a complete beginner to programming you should buy:

Eric Matthes, Python Crash Course.

Textbook for Programmers: If you know how to program in a computer language, you should get the following book:

John Guttag, Introduction to Computation and Programming using Python.

Blackboard: All additional materials will be made available on Blackboard.

Jupyter Notebooks

In addition to the textbook, we will be using Jupyter notebooks. Notebooks are the standard approach to doing interactive Python computations in the sciences. The course notebooks will be available on blackboard. You should have the appropriate notebook on your computer and ready to go before each class.

High Value Websites

Socratica You-Tube Tutorials **website.**

Tutorials Point **website**.

PythonProgramming **website.**

SciPy **website.**

NumFocus **website.**

QuantEcon **website.**

PythonChallenge **website.**

ProjectEuler **website.**

High Value Newsletters

Pycoders Weekly **website.**

Python Weekly **website.**

ImportPython **website.**

O'Reilly Programming Newsletter **website.**

How to Take This Class

This class uses a hands-on approach to learning how to program in Python and how to build algorithms that perform practical computations on a computer. The main hint I can give you is: *don't fall behind*. Learning to program is a cumulative process. You cannot cram. This class assumes you have no previous computer programming experience. The first seven primers will bring you up to speed. By the end of this class you will be on your way to being a solid intermediate level programmer, but more importantly you will be able to apply computational thinking to your study of economics.

The primers and projects are the main way to learn. If you are having problems understanding a programming concept, the beginner's textbook, Python Crash Course, is considered by programming professionals as one of the best books for novice programmers.

Grading

During the class you will be given 16 notebook primers. These notebooks will be completed both in class and as homework. Primers are calibrated to take roughly 5 to 7 hours to compete including in class time. Each primer has up to eight checkpoints, with answers provided in the back of the notebook, and two exercises with no answer provided. If you are taking the class as a 496 class you only have to complete exercise one in each primer. If you are taking the class as an 895 class you have to complete both exercises in each primer.

Tuesday notebooks are due Friday of the same week and Thursday notebooks are due Monday of the following week. A notebook must be submitted by 9 p.m. EST on its due date. No exceptions. When you complete a notebook, submit it by email to kmccabe@gmu.edu with the appropriate subject line (Econ 496 Fall 2018 or Econ 895 Fall 2018). Your notebook will be graded as complete (4 points), checkpoints completed but not exercises (2 points), or incomplete (0 points). To be complete, checkpoint code or exercise code must run correctly and follow the style guidelines explained in class. Your lowest scoring four notebooks will be dropped giving you a maximum score of 48 (i.e., 4 x (16-4)).

You also have five projects worth up to eight points each. Projects are due on Monday of the following week. Projects will be graded from 0-8 with 0 being no work completed and 8 being all your work is completed correctly and follows the style guidelines explained in class. If you are taking this class as a 496 class you only have to complete part one of each project. If you are taking the class as an 895 class you have to compete part one and part two of each project.

The remaining 12 points are given out as participation points to insure you make class and are prepared to answer questions in class.

Students with Disabilities: If you have a learning or physical difference that may affect your academic work, you will need to furnish appropriate documentation to the Office of Disability Services. If you qualify for accommodation, the ODS staff will give you a form detailing appropriate accommodations for your instructor. In addition to providing your professors with the appropriate form, please take the initiative to discuss accommodation with them at the beginning of the semester and as needed during the term. Because of the range of learning differences, faculty members need to learn from you the most effective ways to assist you. If you have contacted the Office of Disability Services and are waiting to hear from a counselor, please tell me.

Honor Code: George Mason University is an Honor Code university; please see the Office for Academic Integrity for a full description of the code and the honor committee process. What does academic integrity mean in this course? Essentially this:

(1) When you are responsible for a report, presentation, or case study, you will perform that task to the best of your ability. Case studies and quizzes are to be done completely independently. Any interaction with others during these times is in violation of the honor code.

(2) When you rely on someone else’s work in your reports, presentations, or case studies you will give full credit in the proper, accepted form.

(3) Another aspect of academic integrity is the free play of ideas. Vigorous discussion and debate are encouraged in this course, with the firm expectation that all aspects of the class will be conducted with civility and respect for differing ideas, perspectives, and traditions.

Course Schedule: This schedule might undergo some further revision.
Blank dates continue last topic.

T.	Aug. 28	Primer 1: Jupyter Notebooks
TH.	Aug. 30	Primer 2: Python Coding I (comments, data types, expressions)
T.	Sept. 4	Primer 3: Python Coding II (print, input, format, import, if statements)
TH.	Sept. 6	Primer 4: Python Coding III (functions, loops, doc strings, keyboard input)
T.	Sept. 11	Primer 5: Python Coding IV (strings, lists, and dictionaries)
TH.	Sept. 13	
T.	Sept. 18	Project 1: Risk Preferences and Elicitation
TH.	Sept. 20	
T.	Sept. 25	Project 2: Simulating Behavioral Agents
TH.	Sept. 27	
T.	Oct. 2	Primer 6: Python Coding V (object oriented programming)
TH.	Oct. 4	
T.	Oct. 9	No Class
TH.	Oct. 11	Primer 7: Primer Python Coding VI (file I/O)
T.	Oct. 16	Primer 8: PyCharm Projects
TH.	Oct. 18	Primer 9: PyCharm Debugging
T.	Oct. 23	Primer 10: Python Logging
TH.	Oct. 25	Primer 11: Primer mTree
T.	Oct. 30	Project 3: Risk Sharing Institutions in mTree
TH.	Nov. 1	
T.	Nov. 6	Primer 12: SciPy Stack I (numpy)
TH.	Nov. 8	Primer 13: SciPy Stack II (pandas)
T.	Nov. 13	Project 4: Monte-Carlo Simulation of Risk Sharing Institution
TH.	Nov. 15	
T.	Nov. 20	No Class
TH.	Nov. 22	No Class
T.	Nov. 27	Primer 14: SciPy Stack III (matplotlib)
TH.	Nov. 29	Primer 15: SciPy Stack IV (seaborn)
T.	Dec. 4	Primer 16: SciPy Stack V (statsmodels)
TH.	Dec. 6	Project 5: Analysis of Simulations of Risk Sharing