Cover Sheet: Request 14464

MAD 2XXX Introduction to Computational Mathematics

Info

Process	Course|New|Ugrad/Pro
Status	Pending at PV - University Curriculum Committee (UCC)
Submitter	Jason Harrington mathguy@ufl.edu
Created	11/20/2019 5:45:09 PM
Updated	12/9/2019 6:31:27 PM
Description of request	This new course will serve as an intro course for data science and give the students an introductory experience with using Python.

Actions

Step	Status	Group	User	Comment	Updated
Department	Approved	CLAS - Mathematics 011613000	Kevin Knudson		11/20/2019
No document changes					
College	Approved	CLAS - College of Liberal Arts and Sciences	Joseph Spillane		11/21/2019
No document changes					
University Curriculum Committee	Pending	PV - University Curriculum Committee (UCC)			11/21/2019
No document changes					
Statewide Course Numbering System					
No document changes					
Office of the Registrar					
No document changes					
Student Academic Support System					
No document changes					
Catalog					
No document changes					
College Notified					
No document changes					

Course|New for request 14464

Info

Request: MAD 2XXX Introduction to Computational Mathematics
Description of request: This new course will serve as an intro course for data science and give the students an introductory experience with using Python.
Submitter: Jason Harrington mathguy@ufl.edu
Created: 11/21/2019 2:54:21 PM
Form version: 8

Responses

Recommended Prefix MAD
Course Level 2
Course Number xxx
Category of Instruction Introductory
Lab Code None
Course Title Intro to Computational Math
Transcript Title Intro Comp Math
Degree Type Baccalaureate

Delivery Method(s) On-Campus
Co-Listing No

Effective Term Earliest Available
Effective Year Earliest Available
Rotating Topic? No
Repeatable Credit? No

Amount of Credit 3

S/U Only? No
Contact Type Regularly Scheduled
Weekly Contact Hours 3
Course Description Is an introduction to mathematical computation and the Python programming language. Emphasis is on using mathematical algorithms to solve problems in analysis, number theory, combinatorics, algebra, linear algebra, numerical analysis, and probability.
Prerequisites MAC 2311 (C) or MAC 3472 (C)
Co-requisites N/A
Rationale and Placement in Curriculum In the new field of data science, Python is the primary language of choice due its versatility. This course serves as an introduction to Python and Computational Mathematics at the start of the of the student's degree so they have the necessary tools needed for more advanced courses on topics of data science.
Course Objectives A student who successfully completes this course will be able to:
-be able to analyze problems from a computing perspective, propose and evaluate solutions to problems;
-understand the importance of and consistently use data and process abstraction;
-understand the importance of and consistently use good programming practices including good documentation;
-write simple programs in Python to solve computational problems from different areas within mathematics.
-analyze and test programs against a set of requirements;
-be able to use packages in Python that are commonly used in data science.
This is a course on how to think about and solve problems using Python and Mathematics, not a course on merely how to write programs in the Python language.
Course Textbook(s) and/or Other Assigned Reading Introduction to Computation and Programming Using Python, Revised and Expanded Edition; ISBN-13: 978-0262525008
Weekly Schedule of Topics Week 1- Chapter 2:Python Basics
Week 2- Chapter 3: Numerical Programs

Week 3- Chapter 4: Functions in Python
Week 4- Chapter 4: Floating Point Arithmetic
Week 5- Chapter 4.3: Recursion
Week 6- Chapter 5: Euclidean Algorithm
Week 7- Functional Programming Tools in Python (map, sum, max, min, etc), List Comprehensions, Gauss quadrature, Vectors as lists, Dot products of vectors
Week 8- Matrices as Lists of Lists, Product of Matrices, Determinants, Determinants from expansions by minor/major
Week 9- Inverse of a square matrix
Week 10- Chapter 10: Sorting Algorithms, Section 10.2 Selection Sort, Bubble Sort, Chapter 9: Complexity and Big-O notation
Week 11- Recursive sorting, section 10.2.1 Merge sort, The Master Theorem
Week 12- Quick Sort
Week 13- Using python packages for data science, like Numpy, Padas, Matplotlib, Scipy
Week 14- Downloading data from the internet , Store and manipulate data from the internet
Week 15- Miscellaneous topics within data science applications

Grading Scheme Homework 20%
Projects 20%
3 Exams 40%
Final Exam 20%

>=93.00 A
90.00 - 92.99: A-
87.00 - 89.99: B+
83.00 - 86.99: B
80.00 - 82.99: B-
77.00 - 79.99: C+
73.00 - 76.99: C
70.00 - 72.99: C-
67.00 - 69.99: D+
63.00 - 66.99: D
60.00 - 62.99: D-
<59.99: E
Instructor(s) Jason Harrington
Attendance & Make-up Yes
Accomodations Yes
UF Grading Policies for assigning Grade Points Yes
Course Evaluation Policy Yes

Introduction to Computational Mathematics
MAD 2*** Section 15387
Monday, Wednesday, Friday, 8nd Period (3:00pm-3:50pm)
Location: Little Hall 223
Spring 2020

Dr. Harrington
mathguy@ufl.edu
https://people.clas.ufl.edu/mathguy/
Little 378
Office Hours: TBD
(352) 294-2320

Course Description: The course provides an introduction to the use of computers for solving mathematical problems. For this, the basics of the Python programming Language (see http://www.python.org) are introduced, and are demonstrate how a programming language can enable the solution of mathematical problems. The course does not assume prior programming experience and does not aim at an in-depth understanding of the details of Python. Rather the focus is on understanding concepts and techniques of how programming can help to expand the spectrum of tractable mathematical problems. After completion of the course, you should be acquainted with the use of different data types and programming concepts. You should be able to write simple programs in Python to solve computational problems from different areas within mathematics, including analysis, number theory, combinatorics, algebra, linear algebra, numerical analysis, and probability. Finally, this course should enable you to read more advanced material on Python, and prepare you to learn other programming languages or packages that are commonly encountered in computational mathematics.
Prerequisite(s): MAC 2311 or MAC 3472 with minimum grade of C.
Credit Hours: 3

Text: *Introduction to Computation and Programming Using Python, Revised and Expanded Edition*; **ISBN-13:** 978-0262525008
Author: John V. Guttag;

Course Objectives:

- be able to analyze problems from a computing perspective, propose and evaluate solutions to problems;
- understand the importance of and consistently use data and process abstraction;
- understand the importance of and consistently use good programming practices including good documentation;
- write simple programs in Python to solve computational problems from different areas within mathematics.
- analyze and test programs against a set of requirements;
- be able to use packages in Python that are commonly used in data science.

This is a course on how to think about and solve problems using Python and Mathematics, not a course on merely how to write programs in the Python language.

Grade Distribution:

Homework	20%
Projects	20%
3 Exams	40%
Final Exam	20%

Letter Grade Distribution:

>= 93.00	A	73.00 - 76.99	C
90.00 - 92.99	A-	70.00 - 72.99	C-
87.00 - 89.99	B+	67.00 - 69.99	D+
83.00 - 86.99	B	63.00 - 66.99	D
80.00 - 82.99	B-	60.00 - 62.99	D-
77.00 - 79.99	C+	<= 59.99	E

Course Policies:

- **General**
  - Projects will be be done using the computer language Python.
  - Exams are closed book, closed notes.
- **Grades**
  - Grades in the **C** range represent performance that **meets expectations**; Grades in the **B** range represent performance that is **substantially better** than the expectations; Grades in the **A** range represent work that is **excellent**.
  - Grades will be maintained in the Canvas course shell. Students are responsible for tracking their progress by referring to the online gradebook.
  - Your grade is your responsibility. You have exactly one week once your assignment has been returned to you to discuss that grade. After that week, the grade is final. You can discuss the content of the assignment anytime but grade disputes must be resolved within one week of the graded assignment.
- **Homework**
  - Homeworks will be short assignments given throughout the semester. A homework set will be given with each unit and then told when it will be collected.
  - Homeworks are handwritten assignments where we work the mathematics which includes: analysis, number theory, combinatorics, algebra, linear algebra, numerical analysis, and probability. For example, an assignment may ask you to use numerical analysis to determine which algorithm is more efficient. This is not only a class on how to learn Python, but to the learn the mathematics behind data science.

- **Projects**
  - For the purpose of this course, we will use Anaconda (See:https://www.anaconda.com/) which has a free platform that manages Python with the data science packages needed for this class. You may also access Anaconda for free via UFapps (see: https://info.apps.ufl.edu/).
  - Students are required to submit their work as a .py file and must use Python 3.7 or greater and provide documentation within their file. We will **not** use Python 2.x since the Python 2 series is set to expire and will eventually be obsolete.
  - Once we learn a topic, we will learn how to implement algorithm in Python. For example, we will investigate different sorting techniques in this course, and you will learn to implement sorting data sets and determine how fast it runs on your machine. Some techniques are written in a way that is easier for a person to understand, and there are ways that are optimized for computational speed. Most of the projects are to simply learn how to code what we have learned so far in the course. These types of projects are to be submitted individually.
  - You will be given a project every two weeks (as long as it doesn't conflict with an exam day) and you will have 2 weeks to work on it before receiving the next assignment.
  - An example of a project, the student will learn how to download data sets from free and reliable sources, like from government sites that house free data or the many free resources for stock information, just to name two. Then manipulate the data based on what we have done in class, like sorting, cleaning or analyzing the data just to name a few topics we will learn.
  - Grading is based on not only the program but how you comment within and the way code is implemented. The instructions will be clear in each assignment as to what technique is needed for that assignment. For example, if the assignment states you are expected to use "Bubble Sort" then that is how you are to sort the data set. Using a different technique than the one mentioned in the assignment details will be penalized.
  - The last project will be one that students will work in a group and present it to the class. You are to use any techniques learned this class to create a program in Python that interests you. You may be as creative as you want. For example, you can create a Python game, mine Twitter data, or automate a laborious process that you do on an computer. The main objective of this last project is to use the skills you learned in this class and create a program that is useful, interesting or fun. There is a set lower level of complexity which will be explained as we approach this project. In essence, it can not be so simple that it renders the program ineffective. (Like download the top Twitter feed for the day, when you can just open up your Twitter feed and see that automatically by changing settings.) Students are in groups of at most 3 people and typically have 15 minutes to present to the class. Your grade is determined by your presentation and the submitted program.

- **Attendance and Absences**

  Registration in this course obligates the student to be regular and punctual in class attendance. All late work will be penalized. Students will NOT be given the opportunity to complete old assignments at the end of the semester to improve their grades. Excused absences are defined consistently with university policies stated in the undergraduate catalog (https://catalog.ufl.edu/UGRD/academic-regulations/attendance-policies/) and require appropriate documentation. Attendance is expected but not required.

Academic Honesty Policy Summary:

Introduction

The University of Florida aims to teach students not just skills and knowledge, but appropriate ethical and professional standards of conduct as well. The Academic Honesty Policy exists to inform students and faculty of their obligations in upholding the highest standards of professional and ethical integrity. All student work is subject to the Academic Honest Policy. Any attempt to deceive a faculty member or to help another student to do so will be considered a violation of this standard.

Instructor's Intended Purpose

The student's work must match the instructor's intended purpose for an assignment. The instructor will establish the intent of each assignment, but it is up to each student to obtain clarification from the instructor when there is any question concerning that assignment's intent.

Authorship

The student must clearly establish authorship of a work. Referenced work must be clearly documented, cited, and attributed, regardless of medium or distribution. Even in the case of work licensed as public domain, the student must provide attribution of that work in order to uphold the standards of intent and authorship. (See, for example, http://creativecommons.org/)

Declaration

Online submission of, or placing one's name on, an exam, assignment, or any course document constitutes a statement that the student has complied with the Academic Honesty Policy in completing that work; in particular, that the student has not received or given inappropriate assistance.

Honor Pledge

We, the members of the University of Florida community, pledge to hold ourselves and our peers to the highest standards of honor and integrity by abiding by the Honor Code. On all work submitted for credit by students at the University of Florida, the following pledge is either required or implied: "On my honor, I have neither given nor received unauthorized aid in doing this assignment."

Software Use

All faculty, staff and student of the University are required and expected to obey the laws and legal agreements governing software use. Failure to do so can lead to monetary damages and/or criminal penalties for the individual violator. Because such violations are also against University policies and rules, disciplinary action will be taken as appropriate. We, the members of the University of Florida community, pledge to uphold ourselves and our peers to the highest standards of honesty and integrity.

Consequences of violations

The webpage https://sccr.dso.ufl.edu/policies/student-honor-code-student-conduct-code/ specifies a number of behaviors that are in violation of the Student Honor Code and the possible sanctions. Furthermore, students are obligated to report to appropriate personnel any condition that facilitates academic misconduct. If you have any questions or concerns, please consult with the instructor or TAs in this class.

Students with Disabilites:

Students with disabilities requesting accommodations should first register with the UF Disability Resource Center (352.392.8565) by providing appropriate documentation. Once registered, students will receive an accommodation letter which must be presented to the instructor when requesting accommodations. Students with disabilities should follow this procedure as early as possible in the semester.

Student Feedback:

"Students are expected to provide professional and respectful feedback on the quality of instruction in this course by completing course evaluations online via GatorEvals. Guidance on how to give feedback in a professional and respectful manner is available at https://gatorevals.aa.ufl.edu/students/. Students will be notified when the evaluation period opens, and can complete evaluations through the email they receive from GatorEvals, in their Canvas course menu under GatorEvals, or via https://ufl.bluera.com/ufl/. Summaries of course evaluation results are available to students at https://gatorevals.aa.ufl.edu/public-results/."

Health and Wellness:

- *U Matter, We Care*: If you or a friend is in distress, please contact umatter@ufl.edu or 352 392- 1575 so that a team member can reach out to the student
- *Counseling and Wellness Center*: https://counseling.ufl.edu/, 392-1575; and the University Police Department: 392-1111 or 9-1-1 for emergencies.
- *Sexual Assault Recovery Services (SARS)*: Student Health Care Center, 392-1161

Tentative Course Outline:

The weekly coverage might change as it depends on the progress of the class. However, you must keep up with the reading assignments.

Week	Content
Week 1	• Chapter 2: Python Basics
Week 2	• Chapter 3: Numerical Programs
Week 3	• Chapter 4: Functions in Python
Week 4	• Chapter 4: Floating Point Arithmetic
Week 5	• Chapter 4.3: Recursion
Week 6	• Chapter 5: Euclidean Algorithm
Week 7	• Functional Programming Tools in Python – map, sum, max, min • List Comprehensions • Gauss quadrature • Vectors as Lists • Dot products of vectors
Week 8	• Matrices as Lists of Lists • Product of two matrices • Determinants • Determinants from expansion by minors/cofactors • Expansion by minors video
Week 9	• Inverse of a square matrix • Least squares regression
Week 10	• Chapter 10: Sorting Algorithms • Section 10.2 Selection sort • Bubble sort • Chapter 9: Complexity and Big-O
Week 11	• Recursive sorting • Section 10.2.1 Merge sort • The master theorem
Week 12	• Quick sort
Week 13	• Using python packages for data science, like Numpy, Padas, Matplotlib, Scipy
Week 14	• Downloading data from the internet • Store and manipulate data from the internet
Week 15	• Miscellaneous topics within data science applications