CPSC150A
Scientific Computing
Syllabus

Instructor: Dr. Durell Bouchard

Office Hours: MWF: 1:40-2:40, TTH: 2:40-3:40, also by appointment or open door

Office: Trexler 365-C

E-Mail: bouchard@roanoke.edu (mailto:bouchard@roanoke.edu)

Phone: 375-4901

Course Objectives

This course is designed to introduce students to fundamental concepts, methods, and tools of scientific computing. The course begins with an introduction to the Python programming language in the context of solving scientific computing problems from a variety of disciplines, including physics, chemistry, biology, and mathematics. The course concludes with a series of guest lectures by experts in scientific computing in various disciplines where students can apply their programming and problem-solving skills.

Intended Learning Outcomes: At the end of the course the successful student will be able to

1. write programs, in the Python programming language, to solve scientific computing problems. In particular, the student will be able to design, implement, and test algorithms.
    - use the basic control structures (conditionals and loops), data structures (lists and dictionaries), and modules (math and random) of the Python language.
    - use the SciPy scientific computing software library.
2. apply scientific computing methods, such as the Monte Carlo method, principal component analysis, or Fourier transforms, to scientific computing problems.
3. create visualization of and analyze scientific data.

Course Content

Prerequisites: There are no formal prerequisites for this course. Prior experience with programming is not necessary.

Text: *How to Think Like a Computer Scientist: Learning with Python: Interactive Edition 2.0*, by Bradley Miller and David Ranum, Runestone Interactive, 2015.

Activities: Learning to program is easy, but becoming a skilled programmer requires a lot of practice. Inclass activities give the student an opportunity to practice the programming and problem-solving skill that are integral to scientific computing in an environment where help from the instructor is easily accessible. Unless otherwise specified, the activities must be done during class and turned in before leaving.

Assignments: In addition to regular reading and in-class activities, there will be programming assignments. These assignments are designed to give the student the opportunity to put into practice the problem-solving and programming skills they have learned. As such they are one of the most important aspects of the course both for student learning and for assessment. The assignments will vary in length and difficulty. You are encouraged to start on them immediately when assigned and get help from the instructor as needed.

Quizzes, Tests, and Exams: Two tests and one comprehensive final exam will be given.

Test Dates:	Test #1	Tuesday, February 9
	Test #2	Tuesday, March 17
	Final Exam	Thursday, April 28 (2:00PM-5:00PM)

Co-curricular: The Department of Mathematics, Computer Science, and Physics is offering a series of lectures designed to engage the campus community in discussions of ongoing research, novel applications, and other issues that face these disciplines. You are invited to attend all of the events but participating in at least two is mandatory. Within one week of attending an event you must submit a one page, single-spaced, paper (to Inquire) reflecting on the discussion. If you do not turn the paper in within the one week time frame you may not count that event as one you attended.

Grading: Course grades are assigned based on the following weights and scale:

Grade Weights: quizzes………….10% tests………..30% final exam…….20% assignments……26%
activities…….12% co-curricular….2%

Grade Scale:	93-100….A	83-86……B	73-76……C	63-66……D
	90-92……A-	80-82……B-	70-72……C-	60-62……D-
	87-89……B+	77-79……C+	67-69……D+	0-60……..F

Course Policies

Attendance Policy: Class attendance is vital to your success in this course; material covered during missed sessions is the responsibility of the student. Conversations held in class illuminate the published class materials and are subject to evaluation on subsequent tests and quizzes. Moreover, quizzes and inclass assignments are not available for make-up. Students with excused absences will not be penalized for missing quizzes or in-class activities. Students with unexcused absences will receive no credit for missed quizzes and in-class activities.

Late Assignment Policy: Unless otherwise specified, assignments are to be turned in before the start of class on the due date. If you anticipate being unable to meet a deadline, talk to me at least 24 hours before the deadline. In extenuating circumstances we may be able to make special arrangements. Please note that this must be discussed – just sending an email does not automatically grant you extra time. If you have not been granted extra time ten percent per calendar day (24 hours) will be deducted for late work (including weekends and holidays); work more than 2 days late will receive no credit. Electronic “glitches” do not waive your responsibility to submit your work in a timely manner.

Make-up Policy: Everyone is expected to take the exam at the scheduled time. Make-ups will be given for legitimate, documented absences. Make-up tests, if given, may be oral.

Academic Integrity: It is accepted that you have read and understood the standards for academic integrity at Roanoke College. All tests and exams are to be the work of the individual student. You are encouraged to get help from the instructor if you need help with any aspect of the course including programs and assignments. Student assistants, tutors, and classmates may help you understand course concepts but may not show you how to do any particular aspect of an assignment. Students may discuss in-class activities and assignments and help each other out but in all cases the work you turn in must be your own. Copying someone else's work or turning in someone else's work is NEVER allowed. Using someone else's work or ideas as your own is plagiarism and an academic integrity offense. Examples of academic integrity violations include copying a program or part of a program (even one line) from someone else, writing code for someone else, telling someone else how to solve a problem or having someone tell you how to solve a problem. Discussion among students about programming projects should be limited to general concepts, not specific aspects of how to complete the work.

Electronic Devices: All cell phones must be turned off prior to entering the classroom. The use of any electronic device during a test or quiz is prohibited. This includes cell phones, personal media players, personal digital assistants, and laptops. Any use of such a device during a test or quiz will be considered a breach of academic integrity.

Disability Support Services: The Office of Disability Support Services, located in the Goode-Pasfield Center for Learning and Teaching in Fintel Library, provides reasonable accommodations to students with identified disabilities. Reasonable accommodations are provided based on the diagnosed disability and the recommendations of the professional evaluator. In order to be considered for disability services, students must identify themselves to the Office of Disability Support Services. Students requesting accommodations are required to provide specific current documentation of their disabilities. Please contact Dr. Bill Tenbrunsel, Director of the Center for Learning & Teaching, at 540-375-2247 or e-mail tenbruns@roanoke.edu (mailto:tenbruns@roanoke.edu).

If you are on record with the College's Office of Disability Support Services as having academic or physical needs requiring accommodations, please schedule an appointment with Dr. Tenbrunsel as soon as possible. You need to discuss your accommodations with him before they can be implemented. Also, please note that arrangements for extended time on exams, testing, and quizzes in a distraction-reduced environment must be made with the Center for Learning & Teaching at least 2 business days (M-F) *before every exam*.

Course Schedule

This course expects you to spend at least 12 hours of work each week inside and outside of class.

Dates	**Topics**
Jan 19 – Feb 9	Expressions, Loops, and Functions
Feb 11 – Mar 17	Branching, Collections, and Numpy
Mar 22 – Apr 21	Guest Lectures