CS 111: Scientific Data and Dynamics

Fall 2018

Course Description: This course is an introduction to computational problem solving. Motivated by real-world problems, we will develop the ability to break complex problems down into simple pieces and produce efficient algorithmic solutions to such problems. We will practice these skills by developing computer programs in a programming language called *Python*, using an interface called *Anaconda* and a text editor *Atom*. The course will cover programming fundamentals, as well as the development of algorithms and data manipulation techniques. No prior experience is necessary.

Course Goals:

1. Practice how to break a complex problem into simple pieces. Develop analytic thinking skills.
2. Develop problem-solving skills. Build confidence in your abilities. Learn how to troubleshoot without becoming frustrated.
3. Develop an algorithmic way of thinking, including attention to the design and complexity of a solution. Learn to mold theory-based approaches to solve messy real-world problems.
4. Develop time management skills, metacognitive skills, and the habit of thinking intentionally about your learning and your goals.
5. Implement algorithmic solutions in Python as a proof of concept.
6. Develop group work skills, test-taking skills, the ability to write clear programs with helpful comments, and the ability to write to a general audience about technical material.
7. Develop comfort with the idea of writing as a way of thinking and understanding. Become proficient at re-writing as a way of communicating to your reader.

Class Details:

Instructor:	David White
Office:	Olin 202
Extension:	6644
Email:	david.white@denison.edu
Class:	11:30-12:20 MTWF in Olin 217
Office Hours:	Olin 202, Times (fill in):
TA Sessions:	Olin 217, Times (fill in):
TA:	Available to help via e-mail and in person (fill-in):
Final:	6:30-8:30pm, Wednesday, December 19, Olin 217
Web Resources:	http://personal.denison.edu/~whiteda/cs111fall2018.html
	http://codingbat.com/home/david.white@denison.edu/cs111fall2018
	NoteBowl: CS 111-01

Textbook: *Discovering Computer Science: Interdisciplinary Problems, Principles, and Python Programming*, by Jessen Havill.

Course Evaluation:

Homework	7%	Quizzes	30%
Labs	30%	Final Project	8%
Participation	5%	Final Exam	20%

Keys to Success

- Prior to exams, be able to solve every quiz and homework problem quickly and with pen and paper rather than in Atom/Anaconda.
- Review the material from class the same day it is given. Find a way to attach this new knowledge to things you already understand.
- Read the textbook slowly and carefully, at your desk, with a notebook nearby to write down questions. Have Atom/Anaconda open while reading. Type code in, play around, and experiment to figure out what Python can do. Answer all reflection questions.
- Start labs early. Give yourself time to get stumped and to get past these difficulties. Finish the programming part of the lab a full day before it is due so as to have time to write the lab report.
- Study computer science a bit every day rather than in bursts just before an exam.
- Keep a list of key definitions, built-in functions, and syntax, and commit them to memory throughout the course. Test your memory each week.
- Have a perfect, hand-written copy of each homework and quizzes within one day after the problems have been discussed in class. Use this to study for exams.

Exams and Quizzes will cover material presented in lecture, homework, and the textbook. We will have a 20 minute quiz every 2 weeks, each worth approximately 5% of your final grade. The best way to study is to attend all class meetings and to do all homework problems carefully. We will have a cumulative final exam on Wednesday, December 19, 6:30-8:30pm. **Please mark your calendars!**

Homework and Labs: We will have 7-8 labs which allow you to synthesize the skills of the previous unit in order to solve a real-world problem. Each lab is worth about 4% of your final grade in the course, and that grade is split evenly between the code and the lab report. The lab report applies your model to answer real-world questions. You should write the introduction and methods part of your lab report, then write your code following the blueprint in your methods section, then answer the questions and re-write your lab report for clarity. For each lab you will have a lab partner, with whom you do the coding. Partners should hand in identical code, and will receive the same grade on the coding part of the lab, except in cases of extreme free-riding. To give you time to spend on writing you should finish your code a full 24 hours before writing the lab report. I will not answer questions about code within 24 hours of the lab deadline.

Homework is meant to give you daily practice with Python. The coding exercises will be submitted through a website called CodingBat, which ensures that your code is correct before accepting your submission. In addition there will be practice problems which will not be collected

but which you are encouraged to solve. Quiz and exam questions will often be drawn from homework and practice problems. The homework is meant to help you identify where you need more help. Homework is not meant to be stressful: all together the homework is worth less than 2 labs. If you find yourself spending more than 45 minutes per night on the homework, you should stop and ask for help (e.g. by emailing the course TA).

Collaboration on homework and labs is strongly encouraged, but you should write up your lab report yourself, and you should never share code with your classmates (except for assigned lab partners). Part of the goal of the course is practicing your writing skills, so you are expected to hand in clear and readable submissions in which you show all work for written problems and leave detailed and useful comments on code. I am happy to answer questions, and I encourage you to come to office hours if you are confused about anything. You will get the most out of this time if you attempt the homework or lab first and come with questions already prepared. Late submissions will not be accepted. In consideration for sickness, personal emergencies, etc. I will drop the lowest lab grade.

Reading: You are expected to keep up with the reading, and to complete the reading before class. This will allow us to cover examples and applications in class that are different from the book, rather than rehashing what you have already read. While you read, work through the "reflections" included in the text, and practice implementing the concepts in Atom. The best way to read the book is with an Atom console open so that you can play with the new commands you're learning and see how they work. It's usually best to try to learn one new thing at a time, and working out the kinks in Atom before starting the homework will deepen your understanding of the material. I'll start class by answering questions, so please ask if anything in the reading was unclear, or if you did not know how to answer any reflection questions. Sometimes we will have pop quizzes (worth less than the main quizzes) to check that you are doing the reading. If students are not reading, we might implement reading quizzes, as part of the daily homework grade. Expect to spend an average of 10-12 hours per week on average outside of class.

Final Project There will be a final project worth 2 labs, where students will apply the techniques learned in this class to a real-world problem of their choosing.

Course Format: Each class meeting is 50 minutes. Please arrive on time or even early, as we will begin promptly. Class will begin with a review of old material and an introduction of new material. Please take advantage of this and use this time to ask questions on things which are confusing you. We're all in this together and don't want to leave anyone behind. We will then have a block of time for in-class programming to make sure we understand the implementation of the concepts discussed in the lecture and reading. Working on problems in class will give you a chance to identify things which may cause confusion on the homework and ask for clarification while we are all in the same room.

Communication: It cannot be stressed enough how essential communication is to succeeding in this course. After identifying topics that may be giving you trouble, please communicate this information to me. There's no such thing as a bad or unwelcome question. Additionally, please communicate with each other. I view the class as a team trying to learn the material together. Collaboration will help all parties achieve this goal, as explaining concepts and examples to each other is a great way to learn.

Participation: Class meetings will be highly interactive and our goal will be to involve all

participants. Attending class, answering questions, and asking questions is therefore essential and will be a significant factor in determining your participation grade. There will also be periodic group exercises which will factor into your participation grade, as will your ability to work well with your partner(s). Each day participation will be graded on a scale from 0-3, with 0 signifying an absence, 2 attending attentively, and 3 active participation such as asking or answering a question.

Grading Scale: A standard 10% grading scale will be used. Therefore, 60% is required to pass the class, 70% will be a C-, 80% will be a B-, and 90% will be an A-. At every moment, you will have the tools to compute your own grade. Since this is a course on computation, I will often ask you to compute it yourself, rather than computing it for you upon request.

Disability: Any student who feels he/she may need an accommodation based on the impact of a disability should contact me privately as soon as possible to discuss his/her specific needs. I rely on the Academic Support & Enrichment Center to verify the need for reasonable accommodations based on documentation on file in that office.

Academic Integrity: The students and faculty of Denison University and the Department of Mathematics and Computer Science are committed to academic integrity and will not tolerate any violation of this principle. Academic dishonesty is, in most cases, intellectual theft. It includes, but is not limited to, providing or receiving assistance in a manner not authorized by the instructor in the creation of work to be submitted for evaluation. This standard applies to all work ranging from daily homework assignments to major exams. Students must clearly cite any sources consulted, including classmates who have been collaborators on the homework and online sources of aid. As is indicated in Denison's Student Handbook, instructors must refer every act of academic dishonesty to the Associate Provost, and violations may result in failure in the course, suspension, or expulsion.

I expect that you will all abide by the honor code in this course. Please do not use resources outside of me, your fellow students, the tutors, and the textbook. Students are not allowed to copy code from online sources, and should not search for solutions online. Collaboration on homework and projects is permitted, but you are not allowed to share code, except with your assigned lab partner. You may discuss the problems with each other and the tutors, but when you leave you should not have any written or typed notes from those you collaborated with. Collaboration on quizzes and exams is not permitted. Violations of the honor code will be reported and may result in severe penalties, including failure of the course.

Appropriate Use of Course Materials: The materials distributed in this class, including the syllabus, exams, handouts, study aides, and in-class presentations, may be protected by copyright and are provided solely for the educational use of students enrolled in this course. You are not permitted to re-distribute them for purposes unapproved by the instructor; in particular you are not permitted to post course materials or your notes from lectures and discussions online. Unauthorized uses of course materials may be considered academic misconduct.

Email: I will frequently contact you via email. Please check your email regularly. I will also check my email regularly, but often not after 7pm. Experience has shown that students who begin labs early are much more successful, while those who try to do it the night before are unable. Thus, I will not respond to questions about labs within 24 hours of the due date, because I do not want to enable students who wait till the last minute to begin.