Computer Science 111
Discovering Computer Science: Scientific Data and Dynamics
Spring, 2019

Professor:	Jessen Havill	**E-mail:**	`havill@denison.edu`
Office:	Burton Morgan 401 (inside 403)	**Phone:**	587–6582
Web site:	`http://personal.denison.edu/~havill/`	**TA:**	Caileigh Marshall '19
Office hours:	MTW 2:00–3:00 or by appointment or any time my door is open		

Description

> *It has often been said that a person does not really understand something until after teaching it to someone else. Actually a person does not really understand something until after teaching it to a computer, i.e., expressing it as an algorithm.* – Donald Knuth, in *American Scientist:61(6)*, 1973

Computation has become a fundamental mode of discovery in many different areas, from laboratory sciences to social sciences to literary analysis. In this course, we still focus on how to model problems from the natural sciences, in particular, and design computational solutions that are both elegant and efficient. We will express these solutions as computer programs written in a programming language called *Python*. Absolutely no prior experience is necessary.

> *Computer science is no more about computers than astronomy is about telescopes.* – Edsger W. Dijkstra

Over the course of the semester, we will tackle a wide variety of problems involving modeling and simulation, Monte Carlo methods, cellular automata, bioinformatics, heuristics, networks, data analysis, and fractals.

Learning Goals

At the end of this course, you will be able to

1. identify ways in which computing has become a powerful mode of scientific inquiry
2. design simple algorithms to solve computational problems
3. implement algorithms as computer programs in the Python programming language
4. utilize abstraction (data and functional) to solve more complex problems
5. recognize the importance of efficiency in algorithm design and assess the asymptotic time complexity of simple algorithms

Required Text

Discovering Computer Science: Interdisciplinary Problems, Principles, and Python Programming by Jessen Havill, Chapman & Hall/CRC Press, 2016.

Online Resources

I will maintain all course materials this semester on Notebowl (`notebowl.denison.edu`). Here you will find your daily reading and homework assignments, projects, quiz dates, and links to useful resources. You will also hand in all assignments through Notebowl. Refer to `notebowl.denison.edu` *daily* for updates.

Outside of class, we will also use Notebowl for Q&A and discussion. When you have a question, instead of sending me an email, please post it to our class' Bulletin on Notebowl. This way, everyone can benefit from the answer. You are also strongly encouraged to answer your classmates' questions. I will answer them too; but if you see it first and know the answer, go for it!

The textbook sometimes refers to files and other resources on the "book web site." You can find this at `DiscoverCS.denison.edu`.

Attendance and Other Responsibilities

Your **active participation** both in class and in your preparation for class is absolutely essential to your success. I cannot emphasize this enough! By simply attending class and doing a minimal amount of work, you will both learn very little and earn a very poor grade.

It is very important that you keep up with the coursework (see Notebowl calendar) on a **daily** basis; **consistency is the key**. Like other classes at Denison, it is expected that you devote at least 2–3 hours outside of class for each hour of class time. Read your book and do the examples and exercises in front of a computer. We will use the reading as a starting point for each class discussion rather than rehash everything that you read the night before.

Your attendance is expected at each class meeting. Your grade will almost certainly suffer indirectly if you choose not to attend. In addition, I may consider attendance when assigning grades, especially in borderline situations. Of course, previously arranged and unavoidable absences (sickness, family emergencies, varsity athletic participation) will not be held against you. To the extent possible, absences should be communicated to me in advance. You are responsible for the content of reading assignments, lectures and handouts, as well as announcements and schedule changes made in class whether or not you are present. If you must miss a class, be sure to check with me or another student to get what you missed. Exams will be given in class on the day scheduled and may not be made up.

Course Work

I hear and I forget. I see and I remember. I do and I understand. – Confucius

Reading Notes To facilitate careful reading, I will provide you with a set of questions to answer for each reading assignment (on Notebowl). Some, but not all, of these questions are the "Reflection" questions in the textbook. You are encouraged, of course, to jot down additional notes as well.

Daily Homework Exercises The textbook contains hundreds of exercises that are designed to reinforce the concepts and give you extra practice solving problems. For each section that you read, I will generally choose 3 to 10 of these exercises for you to hand in (via Notebowl) before the next class. Due to time constraints, I will generally not be grading exercises for correctness unless you explicitly ask me to (which I am happy to do). Rather, I will only check that you have made a serious effort toward a solution. I may hand out solutions to these exercises and/or we may discuss them in class. Exercises are meant to provide both practice and diagnostic information about how well you are understanding the concepts. If you do not understand how to do an exercise, see me or a tutor right away.

It is never a bad idea to also do some unassigned exercises if you need more practice with a given topic.

Completed exercises and reading notes must be submitted through Notebowl by 8:30am on the day they are due.

Group Projects Much of the work in this class revolves around several larger projects that apply what you are learning to interesting problems in the social sciences. Each project will be due in class on the date specified. No late assignments will be accepted, unless arrangements have been made with me well in advance. Since it will most likely not be obvious how long an assignment might take, **you are well advised to start early**. You will be working in pairs on all but the first few programming projects. Pairs will be rotated every 2–3 weeks.

Keys to Success

1. **Consistent, daily work is the key.** The daily reading notes and homework assignments are meant to reinforce good habits. If you spend an hour or two every day, this course is quite manageable, and you will learn more effectively. Do not try to do the work for this class in one or two long sessions each week. Start thinking about projects the day they are assigned, and consistently work on them all week. Aim to be done a day early so you have time to polish your program and write your best report.

2. **Practice active reading.** Read the book in front of a computer and type in all of the code to see what it does. Be curious. If you're wondering "What would happen if ...", try it! The beauty of programming is that it is easy to test ideas and get instant feedback. Answer the reading note questions as you read. It is perfectly normal to have to read a section two or three times before you understand it. Many students find it very useful to read a section again after we have talked about the topic in class.

3. **Ask questions.** If you are having trouble understanding something, ask questions right away — before class, in class, after class, in my office, during tutoring sessions. There is plenty of help available, but it is up to you to seek it out. Everything in this class is cumulative; the further you get behind, the harder it becomes to catch up.

4. **Test yourself often.** On quizzes, you will be expected to write code on paper for problems that are similar to homework exercises. So it is important to practice this on your own. Choose some previous exercises, or some new ones from the book, and work out the answers on paper without looking at the solution. When you are done, you can check your answers by trying out your solution on the computer. Testing yourself often has also been shown scientifically to improve learning!

5. **Be patient but persistent.** Some of the concepts in this class take a while to sink in. Try not to get frustrated if it doesn't make sense right away. Keep doing examples and ask a lot of questions. Don't give up! With sufficient effort, it will eventually click.

6. **Use pair programming.** You will be working with a partner on each project. Listen to each other and hash out a plan (including meeting times) for how you are going to complete the project together. You and your partner should work together on all aspects of the project, from beginning to end. Do not practice "divide and conquer." Instead, I strongly encourage you to practice "pair programming," a well-established practice where one partner (the "driver") types and the other (the "navigator") watches for errors and focuses on the overall design. The "driver" and "navigator" roles should switch often so that both partners are contributing equally. See this (rather dorky) video for a tutorial: `https://www.youtube.com/watch?v=rG_U12uqRhE` .

Big Idea Tuesdays (BITs)

In an introductory programming-oriented class such as this, it is easy to get caught up in the daily routine and lose sight of the "big picture." To mix things up a bit and get us thinking about some fascinating new directions in computer science and applied computing, we will step back every other week for a Big Idea Tuesday. During this class period, we will have a discussion (or another activity) surrounding a different "big idea." In some cases, the next project may also be related to this topic.

Before each of these discussions, we will read an article on which to base the discussion. On the Monday before, you will post (on Notebowl) your thoughts on the reading and two well-constructed discussion questions for the next day. A half dozen or so of the class' questions, together with some of my own, will form the basis for our discussion.

Grade Determination

The following relative weights will be used to determine your final grade:

Projects	30%
Homework exercises	10%
Reading notes and BITs	10%
Biweekly quizzes	35%
Final exam (Sat, May 11, 9:00)	15%

Academic Integrity

Proposed and developed by Denison students, passed unanimously by DCGA and Denisons faculty, the Code of Academic Integrity requires that instructors notify the Associate Provost of cases of academic dishonesty, and it requires that cases be heard by the Academic Integrity Board. Further, the code makes students responsible for promoting a culture of integrity on campus and acting in instances in which integrity is violated. Academic honesty, the cornerstone of teaching and learning, lays the foundation for lifelong integrity. Academic dishonesty is 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 not only for quoted phrases but also for ideas and information that are not common knowledge. Neither ignorance nor carelessness is an acceptable defense in cases of plagiarism. It is the students responsibility to follow the appropriate format for citations. Students should ask their instructors for assistance in determining what sorts of materials and assistance are appropriate for assignments and for guidance in citing such materials clearly.

You can find further information about Denisons Code of Academic Integrity on Denisons web site at `denison.edu/academics/curriculum/integrity`.

In this class, you may discuss problems with other students in the class, but written (and typed) work must be your own. In other words, you may talk about problems with your peers, but when it comes time to write your solutions, you (and your partner) are on your own. You may have general conversations about problem strategies, but you must leave these conversations without having written anything down. Keep in mind that it is quite easy for me to tell when students have been working too closely. You may not get help from students outside the class, except for departmental tutors. If you have questions, come see me and I will be happy to help. You are also quite welcome to send me email or call if you would like to discuss an assignment.

Students found responsible for breaches of academic integrity may earn a failing grade for the course.

Accommdations

Any student who feels he or she may need an accommodation based on the impact of a disability should contact me privately as soon as possible to discuss his or her specific needs. I rely on the Academic Resource Center (ARC) in Higley 020 to verify the need for reasonable accommodations based on the documentation on file in that office.

Course Evaluations

At the end of the semester, you will be asked to evaluate this course and the instructor. These evaluations are an important tool for helping Denison faculty achieve and maintain excellence in the classroom; it will also help you reflect on your learning, participation, and effort in the course. A key purpose of course evaluations, then, is to constantly improve the level of teaching and learning at Denison by instructors and students. Your ratings and comments will also be included as one element of an instructor's overall teaching portfolio. Together with peer observations and other means of assessing teaching effectiveness, this portfolio will be considered by the instructor's colleagues and college administrators in making recommendations for contract renewal, tenure, promotion, and salary decisions.

Have a great semester! If you need anything, please let me know.