

Course Objectives. This course has been designed to help you succeed in our Master of Data Science Online program. In particular, we have three subgoals. The first is to develop good programming skills, including testing, debugging, and some basics of programming methodology. The second goal is to introduce fundamental concepts in data structures and algorithms. And the third is to encourage you to think critically; for example, there will be some quiz questions that do not simply ask you to regurgitate things that you were told in the lectures but will instead require you to think to figure out the answer.

Prerequisites. This course assumes at least a college semester (or a high school year) of previous programming experience with an object-oriented programming language such as Python, Java, or C++. With this background, you should pick up Python fairly quickly, even if you have not used it before.

To help students ease into Python, the first assignment focuses on basic Python programming constructs. We have also included in the first two weeks some optional lectures whose titles begin with the prefix “Python Bootcamp.” These lectures discuss some of the basics of Python, but they are not intended to be comprehensive or to replace a good Python textbook or reference manual.

Workload. To accomplish our objectives, the course has a set of six modest-sized programming assignments, to be completed using Python. There will be no exams, but there will be small quizzes roughly every other week. These quizzes will contain a variety of questions, some of which assess your ability to deeply understand the lecture material.

It’s difficult to estimate the amount of time that you will spend on this course. Last semester, some students spent as little as 5 hours per week, while some spent as many as 25 hours per week. There are two reasons for this wide range. First, students will enter the course with very different programming backgrounds. Second, programmer productivity has a notoriously high variance, so some of you will spend much less time, while others will spend much more time.

Assignment	Start Date	Due Date (Central Time)	Mode
Friends	January 19	February 4	Group
Evil Hangman	February 2	February 18	Group
Boggle	February 16	March 10	Group
Treaps	March 1	March 24	Individual
B-Trees	March 22	April 14	Individual
WikiRacer	April 12	May 5	Individual

Textbook. There is no required textbook, but there are three supplementary textbooks that might be useful, two of which are freely available.

The first provides a good presentation of many of the data structures that we discuss in this class; unfortunately, the code examples are presented in Java instead of Python, but the prose is quite clear. *The chapters mentioned in the syllabus refer to chapters in this book:*

M. Weiss. *Data Structures and Problem Solving using Java*. 4th Edition. Addison-Wesley, 2006.

The second textbook describes just the most basic data structures, but it also provides some background on Python, so this book would be useful for those of you who are not comfortable programming in Python:

B. Miller and D. Ranum, *Problem Solving with Algorithms and Data Structures Using Python*. Franklin, Beedle, and Associates, 2011.

A digital version of this book is available for free under the Creative Commons License:

<https://www.freetechbooks.com/problem-solving-with-algorithms-and-data-structures-using-python-t11>

Finally, the third book, which we highly recommend, is *the Hitchhiker’s Guide to Python*, by Kenneth Reitz and Tanya Schlusser. O’Reilly, 2016. Our Ed Discussion page will contain a link to the online version of this book, and we recommend that you minimally read Chapter 4.

Lecture Contents. We'll cover the following topics, roughly in the following order, although some of these topics are sprinkled throughout the course. (The chapters listed here refer to chapters in the first recommended text by Weiss.) Except for the first week, each week's lectures will be made available on the preceding Friday.

1. **Introduction.** Course overview, administrative matters.
2. **Programming Skills.** Xtreme Programming, debugging, testing. (These topics are sprinkled throughout.)
3. **Algorithm Analysis.** (Chapter 5)
4. **Basic Data Structures.** Arrays, Linked Lists, Stacks, and Queues. (Chapters 16-17)
5. **Sorting Algorithms.** Quicksort, Mergesort, Radix Sort. (Chapter 8)
6. **Recursion.** (Chapter 7)
7. **Trees.** Tries, Binary Trees, Binary Search Trees (Chapters 18-19)
8. **Balanced BSTs.** AVL Trees, Red-Black Trees, Splay Trees, B-Trees. (Chapter 22)
9. **Priority Queues.** Heaps. (Chapter 21)
10. **Hash Tables.** (Chapter 20)
11. **Amortized Analysis.** (Chapter 22)
12. **Graphs and Graph Algorithms.** Breadth-first search, depth-first search, Dijkstra's algorithm. (Chapter 14)
13. **Dynamic Programming.**

Course Policies

Group Communication. We will use the Ed Discussion website to communicate with one another. Feel free to use this medium to ask and answer questions, and look here for class announcements. You should receive an email message from Ed Discussion asking you to join the class; if you do not, you can visit the Ed Discussion web site through Canvas and request to join yourself.

Course Etiquette. We expect each of you to treat online discussions as though you were having a civil, respectful discussion with your fellow classmates in a physical classroom. Please refrain from using profanity or any euphemisms for profanity. Please do not bait others or personally attack others. Please do not use sarcasm in a way that can be misinterpreted negatively. And please do not belabor the same points repeatedly. In short, please respect the right of your colleagues to ask questions and to discuss their opinions about the course subject matter on the discussion board. Violators of these discussion rules will be shut out from all class communications—email, Ed Discussion, and office hours.

Office Hours. Professor Lin will hold office hours through Zoom by appointment only. The TAs will hold regularly scheduled office hours as posted on the Ed Discussion page.

Grading.

Programming assignments	70%
Quizzes	30%

This course will be graded on a generous curve. Note also that the programming assignments are not weighted equally. For example, the first assignment will be worth significantly less than the others.

Curved Grading. For those of you who are unfamiliar with curved grading, the idea is that your grades are not determined by any pre-determined cutoffs. Instead, at the end of the semester, we examine the distribution of scores and only then determine the cutoffs, and we can place the cutoffs wherever we wish.

This process may sound horribly mysterious and unfair, but it has the advantage that it allows us to be generous in placing the cutoffs, and that is our intent. A large downside to curved grading is that you as a student have no idea what your grade is until the end of the semester. While we cannot predict either the distribution that we'll see at the end of the semester or the specific cutoffs for the various grades, we can tell you that last semester, students whose overall grade was greater than 80.1% received an A–. Of course, last year's curve guarantees you nothing about this year's curve, so the cutoff for an A– may well be different this year, but the point is that the curve will in fact be generous.

The point here is that for this course, you should remove from your brain the thought that a score of 90 is an A, 80 is a B, and 70 is a C.

Late Policy for Assignments. Unless otherwise specified, assignments are due by **11:59pm Central Time** of the due date. Late assignments will be penalized 20% per day, but you are given **six slip days**, where each slip day removes this penalty for one day. Since the different assignments have different weights, we will at the end of the semester automatically assign slip days to your assignments to that maximize your grades. *We will assign at most two slip days for any one assignment.*

Quizzes. Quizzes are open-note, so you may consult any publicly available resource that you wish, as well our lectures and any notes that you've taken. Once you begin a quiz, you will have a limited amount of time to complete it.

Late Policy for Quizzes. Quizzes can only be taken inside the specified time frame.

Karma. Our goal is to challenge each of you, so most of the programming assignments provide ideas on optional extensions to the assignment. We refer to this optional work as Karma, because while it will generally not affect your grade (and will not be graded by the TAs), it is good for you. **Under rare circumstances, students who have done truly significant amounts of Karma may see their grade increased if they are close to a grade boundary.**

Do Not Share Course Materials with Outsiders. Except for the syllabus, the material used in this class, including, but not limited to, quizzes, lecture notes, and assignments, may not be shared with anyone outside of the class without the instructor's explicit permission. The problem with unauthorized sharing of materials is that it promotes cheating.

Academic Integrity in This Course

Programming assignments and quizzes are to be done individually. Thus, while you are encouraged to discuss the lectures, readings, and assignment specifications with others, the work that you submit, including both programming assignments and quizzes, must be entirely your own (or if it is a group assignment, then it must be entirely your group's).

Intellectual dishonesty can end your career, and it is your responsibility to stay on the right side of the line. If you are not sure about something, ask.

We encourage you to use code repositories such as github and bitbucket to provide version control for your code, but **it is absolutely critical that you keep your repositories private**. As mentioned below, you are not allowed to look at other solutions or to make your solutions publicly available. Violators will be referred to the Dean of Students office.

Our standard penalty for cheating is to assign a grade that is –50% of the maximum score for that assignment or quiz. So if an assignment is graded out of 100 points, then the penalty for cheating would be to receive a score of –50 for that assignment.

Violations of the course policy include (but are not limited to) the following:

- Providing your UT EID to any other person.
- Accessing another student's computer account.

- Collaborating or sharing information with another person regarding the material on any quiz or assignment, before, during and/or after any quiz or assignment.
- Helping another student write or debug their code.
- Recording any quiz or assignment material in any format.
- Failing to properly cite language, ideas, data, or arguments that are not originally yours.
- The public (such that it can be viewed by more than one person) posting of any form of any quiz questions or any assignment.
- Consulting forbidden materials or sources of information, including code repositories such as github.com or bitbucket.org.
- The use of code-producing tools, such as Co-Pilot.
- Maintaining your own source code in public repositories at sites such as github.com or bitbucket.org. Such repositories must be marked private.

If you have any doubts about what is allowed, ask the instructional staff.

The University of Texas at Austin Academic Integrity principles call for students to avoid engaging in any form of academic dishonesty on behalf of yourself or another student. If you have any questions about what constitutes academic dishonesty, please refer to the Dean of Students website or contact the instructor for this course.

You must agree to abide by the Honor Code of the University of Texas. You will not work with or collaborate with others in any way while completing any of the graded course assignments or quizzes.

We will use various tools to detect illicit collusion, collaboration, or plagiarism.

Final Thoughts

How to Succeed in this Course. Read this syllabus, start the assignments early, and ask questions via Ed Discussion. If you get behind, it can be difficult to catch up, so if you have any problems or questions, please contact us as soon as *soon as possible* so that we can help.

Religious Holidays. If religious holidays will affect your ability to turn in a programming assignment on time, please let us know as soon as possible (the deadlines are all available now on Canvas) and no later than 48 hours after an assignment has been published.

Students with Disabilities. The University of Texas at Austin provides upon request appropriate academic accommodations for qualified students with disabilities. For more information about the parameters for appropriate accommodations, contact the Office of the Dean of Students at 471-6259, 471-6441 TTY. To issue your request for this course, please send a message to the course email address: onlinedatastructures@austin.utexas.edu.

Academic Advisor Support. If you have additional questions or require support from an academic advisor, please contact the program coordinator at msdsgradcoordinator@utexas.edu.

No Whining. We welcome feedback and comments about the course, and we will correct legitimate grading errors that are identified in a timely fashion, but whining about grades is counter-productive and will only irritate the instructional staff.