***ICS 340-01: Data Structures and Algorithms***

**Spring 2021 4 credit (mostly synchronous) online course**

**Mon-11-Jan-2021 🡪 Mon-03-May-2021**

# **Welcome to ICS 340!**

If you like Math and Programming you should have a fun time. If you’ve done well in your previous math courses (especially Discrete Math and – to a lesser degree - Calculus), and in the programming portions of your previous Computer Science courses – especially in Java programming, you should have a reasonably easy time in this course (but I hope not *too* easy, I hope you still learn a lot.)

If you’ve struggled in the past with Math and/or Programming, plan now how you will make the extra time for this course so that you do well enough to pass. Please see the section “Preparation” below for more information.

# **Instructor and Department Contact Information**

| Instructor Information | Department Information |
| --- | --- |
| Name: Mike Stein | Name: Computer Science & Cybersecurity |
| Email: michael.stein@metrostate.edu | Email: csc@metrostate.edu |
| Zoom Sessions: 6:00 PM, Mondays  [**https://minnstate.zoom.us/j/91751002245**](https://minnstate.zoom.us/j/91751002245)  **Passcode 608536** | Mailing address:  Dept. of Computer Science & Cybersecurity  Metropolitan State University  700 7th Street East  Saint Paul, MN 55106-5000 |
| Phone: 612-802-6727 (Cell)  612-793-1476 (Ofc, voicemail only please) | CSC Dept. Phone: 651-793-1471 |

* Office hours: The best way to contact me is via email. I am happy to make Zoom appointments to talk to people during reasonable hours of the day or evening. Please ask at least 24 hours before you want to meet. My schedule varies weekly, but normally I’ll be available some times during 2-5 weekday afternoons, and 12-4 weekend afternoons.

# **About the Course**

## Course Description

This course explores the design and implementation of data structures and algorithms fundamental to computer systems and applications and to software engineering. Topics include trees, graphs, basic analysis of algorithmic complexity, fundamental questions of computability, and introduction to the algorithmic basis of intelligent systems. Programming projects.

## Prerequisites

Prior to taking this course, you must have successfully completed the equivalents of ICS 240 (Introduction to Data Structures) and Math 215 (Discrete Mathematics). In particular, based on the MinnState Computer Science Transfer Pathway, you should be able to do the following after having passed these courses:

1. Implement recursive and iterative algorithms over appropriate data structures (e.g., tree traversal, pre-order, and post-order)
2. Analyze recursive and iterative algorithms for space-time complexity
3. Identify appropriate use of and implement statically allocated data structures.
4. Identify appropriate use of and implement dynamically allocated data structures.
5. Develop and implement a dictionary data type incorporating hashing algorithms and collision avoidance.
6. Write programs that use each of the following data structures: arrays, records/structs, strings, linked lists, stacks, and queues
7. Implement algorithms and underlying data structures utilizing standard object oriented design principles.
8. Describe how symbolic logic can be used to model real-life situations or applications, including those arising in computing contexts such as software analysis (e.g., program correctness), database queries, and algorithms.
9. Examine the logical validity of arguments and proofs as they apply to Boolean expressions.
10. Apply mathematical induction and other techniques to prove mathematical results.
11. Perform computations using recursively defined functions and structures.
12. Solve problems involving sets, relations, functions, and congruences.
13. Illustrate the basic terminology and properties of graphs and trees.
14. Use graphs and trees to solve problems algorithmically.
15. Use methods of combinatorics to solve counting and basic probability problems.

To do this, you should be familiar with at least 80% of the following topics, and able to pick up the remaining 20% on your own:

1. Logic and proof
2. Mathematical induction, including both strong and weak induction
3. Elementary set theory
4. Relations and functions
5. Recurrence relations
6. Elementary number theory and applications
7. Elementary graph theory and applications
8. Combinatorics and probability
9. Algorithm analysis
10. Object-oriented programming: encapsulation, polymorphism (including dynamic binding), inheritance
11. Abstract Data Types
12. Static allocation vs. dynamic allocation of memory
13. Linked Lists: static implementation, dynamic implementation, and doubly linked
14. Stacks: static and dynamic implementation
15. Queues: static and dynamic implementation, circular, and priority
16. Generic Objects
17. Recursion
18. Algorithm Analysis
19. Sorting: quicksort, merge sort, heap sort, shell sort, and radix sort
20. Hashing with collision strategies
21. Trees
22. Graph theory
23. Traversals
24. Heaps

I assume that you know the Java programming language well enough to write the programs in this class.

## Preparation

In an upper-division 4-credit university Computer Science course, students who are up-to-date on their prerequisite knowledge should expect to spend on average 4 hours per credit per week in their studies, preparation activities, and assessments. In a 4 credit course, a median student who is up-to-date on their math and Java programming should expect to spend 16 hours per week completing readings, assignments, discussions, class time, and research for this course. Student feedback from previous semesters indicates that this is approximately how long they have spent working on this course, on average.

Please look at the video lecture, the textbooks, and the resources listed under “Topic Intro” before class. If you find a video that helps you understand some topic better than the textbook, please share this on the discussion board!

This course covers a lot of material at a very shallow level. Many videos were prepared in the days prior to the latest revision of the ACM’s recommended computing curriculum, where the curriculum was narrower and deeper. Thus, most videos will go into topics in more detail than covered in this class. This is especially true of AI, NP-Completeness, and Theory topics.

# **Texts and Learning Outcomes**

## Textbooks

Cormen, Leiserson, Rivest, and Stein[[1]](#footnote-1), *Introduction to Algorithms, 3rd ed.*, ISBN 978-0262033848. a.k.a. “CLRS”. Perhaps the single most widely used textbook for upper-division Computer Science courses in the country, you should have no trouble finding it, new or used, if you want a hard copy.

CLRS is available as a Metro State library eBook. <https://ebookcentral.proquest.com/lib/metrostate/detail.action?docID=3339142>

This book has no simultaneous user restrictions. Upon clicking the link above, you will be taken to the Library's EZproxy login page, where you should login with your StarID (no @my.metrostate.edu) and password. After successfully logging in, you will see the title page of the book, with links to specific chapters which then can always be read online and sometimes downloaded as PDFs.

Poole, et. al. *Artificial Intelligence, 2nd ed*, ISBN 978-1107195394. This textbook is available legally and freely on the Web at <http://artint.info/index.html>. Be sure to look at the 2nd edition, both the 1st and 2nd editions are available at this URL.

For every chapter that we cover in either textbook, I upload a “Chapter Intro” indicating what I want you to get from that chapter, and a set of other resources (web pages, youtube videos, ,etc.) that cover the same material.

## Suggested Supplementary Readings, Databases, and Sources

See my “Topic Intros” on the D2L site for supplemental information, listed by topic.

## Course Learning Outcomes

1. Analyze asymptotical complexities (O(n), O(n2), O(log n), and O(n log n), etc.) of algorithms.

3. Design and implement non-linear abstract data types such as trees and heaps, and apply them to appropriate problems.

4. Implement and apply different graph algorithms (such as depth first, breadth first, shortest path, and minimum spanning tree, etc.) to solve problems.

5. Select, implement, and apply appropriate sorting algorithms to solve problems.

6. Use pseudo-code to describe algorithms at an intermediate/advanced level.

7. Write programs using recursion at an intermediate/advanced level.

8. Design and implement algorithmic strategies such as Dynamic Programming and Greedy Strategies to solve problems.

## Learning Methods

1. Textbook readings, from classic texts that cover our fields of study in depth.
2. Videos and web pages covering the same material as the textbooks.
3. Videotaped lectures covering the material. These will be available before class starts.
4. Zoom sessions on Monday evenings. I will answer questions and will have you work on problems in groups in these sessions.

I also encourage people to share other sources of information that they find (videos, etc.). I may add them to the resources for future offerings of this course.

# **Course Policies**

## Attendance

Each week, there will be one or more assignments for people to do during class, most likely in groups. This will be part of the graded work in the course, so class attendance is strongly recommended (if you don’t come at least sometimes, you’ll fail the course – see the grading distribution below).

## Interview

**Fall semester was rough because I didn’t get to really connect with anybody, since the class was totally online. I am going to try a suggestion from other instructors – I’ll make the first assignment a meeting ≤ 15 minutes, by zoom, with video on, in which I’ll interview you and ask you a few questions about your background related to topics that I think are relevant for the course. This is the required first assignment, and it will count as attendance for the course. Because the deadline for course attendance is the end of the second class, this needs to be completed no later than January 25th – before the second class section. I have a doodle poll, listing times I am available and set up so that only one person can sign up for any timeslot. I hope to complete the interviews by the January 18th.**

**Link to Doodle Poll:** <https://doodle.com/poll/xsgerih6nbcqcwxp?utm_source=poll&utm_medium=link>

**It is set up so that you can sign up for only one time, and only one person can sign up for any given time.**

Assignment Types

**Note: All assignments (except in-class work and possibly the final exam) are due at high noon of the due date, according to D2L.[[2]](#footnote-2) See the next section for late policies.**

1.Pre-Quizzes require that you read the material and/or view the recorded lecture before the class session.

2. In-class exercises give you practice applying the material covered, and/or planning your programming project work.

3. Homeworks have you explore the theoretical material in depth.

4. Programs have you implement the algorithms we study, since ultimately to be useful, computer algorithms must be implemented.

5. The Final Exam (in two parts) will be a way to summarize for me what you have learned.

## Late Policy for Assignments

* For homeworks (which are due at noon on the day listed), you may be up to 6 hours late. You lose one point off the sum of all homework grades for each hour late, as computed by D2L (< 1 our late counts as 1 hour late). The solutions will appear at 6 PM on the due date.
* No late quiz submissions are allowed. The solutions appear at the quiz deadline.
* For Programs, late work is accepted for 24 hours after the deadline. The late penalty is one point per hour, and is deducted from the grade on that submission of the program. It is calculated as the number of hours D2L reports the assignment as late (again, < 1 hour late counts as 1 hour late).

## Covid-19 Policy

University Policy: If you contract COVID-19 (or if someone you care for does), please contact me as soon as you are able and then complete Metro's COVID-19 [reporting form](https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fwww.metrostate.edu%2FCOVID&data=02%7C01%7Cmichael.stein%40metrostate.edu%7Cdb3c769ab00341a9a68908d842c31c7c%7C5011c7c60ab446ab9ef4fae74a921a7f%7C0%7C0%7C637332753474153208&sdata=S7MlUT%2BS63GhUllJSRsrjew83WrLUcXemLUhoNPGxh8%3D&reserved=0). All COVID-19 related class absences will be excused, and we will accommodate your re-entry into the course when you feel better. You will be given appropriate extensions for your required work, but will still be responsible for completing *all* assignments. If you are unable to complete the requirements for the course following a COVID-19 illness, please complete a [registration appeal](https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fwww.metrostate.edu%2Facademics%2Fregistration%2Fappeals&data=02%7C01%7Cmichael.stein%40metrostate.edu%7Cdb3c769ab00341a9a68908d842c31c7c%7C5011c7c60ab446ab9ef4fae74a921a7f%7C0%7C0%7C637332753474163201&sdata=%2Bu5hXeTG1uhMz8TvgqNrqwPIHHyhtq0F8atmgpI5b2Y%3D&reserved=0).

What that means for this course: Since I give out solutions ASAP after the assignment due date, if you miss time because of Covid-19, I can create new assignments (similar to the old ones) for you to do with a negotiated due date, to replace assignments you missed.

**Assignments and Grading**

Assignment due dates are located in the Course Schedule of the Content area. The Late policy applies to all assignments. Following are the numbers of assignments, points assigned and grade points. I discuss each type of assignment in a subsection later. Note that there may be more points possible than the “max”, for quizzes, in-class exercises, and homeworks. This serves as a form of extra credit.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type** | **Num** | **Pts Each** | **Tot Pts** | **Comments** |
| Pre-Quizzes | 14 | 15 | 210 |  |
| In-Class Work | 14 | 7-12 | 130 |  |
| Homeworks | 14 | 15 | 210 |  |
| Program Deliverables | 4 | 25-75 | 250 | Two chances to hand in each program, the second with maximum 80% credit. |
| Exam | 1 | 200 | 200 | Multi-part |
| **TOTAL** |  |  | **1000** |  |

## Grade Scale

My grading philosophy is based on my educational philosophy that a college education should produce a well-rounded graduate who is prepared to interview for, and go into the workforce as, a software developer of complex systems. So I try to make assignments that will be preparatory for “coding interviews” and types of assignments that you might have in your career.

On an implementation level in a junior-level course, that means there are a few things that I want everybody to be able to do to some extent. To pass this course, you must have at least a bare minimum of competence at each of these phases.

* You should be able to figure out the basics of material you don’t know by using the resources available from books and the internet, given sufficient time (tested by pre-quizzes).
* As an upper division student, you should be able to operate at the higher levels in the “cognitive domain” of Bloom’s Taxonomy. That is, you should be able to take the things that you know how to do (from previous courses, pre-quizzes, and in-class work), and deal with them abstractly by doing things like proving properties about them, and reasoning about them in the context of novel problems (homeworks).
* You should be able to program Java at a high-intermediate level (program deliverables).
* You should be able to answer questions based on your available knowledge in a limited amount of time (exams).

***The basic requirement to pass the course is to get at least 40% total in each of the five areas listed above. If you don’t do that, if your overall score in the class is 40% or more you will get a “D”; if less, you will get an “F”.[[3]](#footnote-3)***

The following percentages will guarantee you the given grade, assuming you earn at least a “D” overall in each element of the course as noted above.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Grade | A | A- | B+ | B | B- | C+ | C | C- |
| Min. pct. | 90% | 85% | 80% | 75% | 70% | 65% | 60% | 55% |

Historically, 60-85% of the students attending the first lecture will pass the course. *Most who do not had problems with the programs.* One person in the last 14 classes I’ve taught has failed to meet the exam criterion.

## More Information on Assignments

You are responsible for reading and understanding all the documents in the Assignments section of the Content tool. Check the schedule for assignment due dates. Late work is not allowed, except for resubmission of program deliverables.

Please note that the topics in the course are numbered sequentially from 0x0 through 0xe. Quizzes, exercises, lectures, and homeworks are numbered in line with the topic they cover. Thus quiz *n* covers the same material as is discussed in lecture *n* and which will form the basis for the in-class exercise set *n*, and which will be the topic of homework *n*.

### Pre-Quizzes

Each week, there is an online pre-quiz over the material to be covered in the next class, released by noon Tuesday, and due noon Monday.[[4]](#footnote-4) Quizzes use the D2L quiz tool. Grades will be released before class Monday, so auto-grading errors can be fixed. You get to submit the quiz twice. Upon each quiz submission, the quiz tool will tell you what questions you got wrong, so that you can look at the material to try to figure out the answers before class.

The purpose of the quizzes is to motivate people to read the material, and watch the video lecture, before the class. The quiz questions will be easiest to answer if you read/watch the material and understand it at a basic level before coming to class. When you submit the quiz, you can see which questions you got wrong, so you can review those topics.

### In-Class Work

In-class work gives you a chance to solve relatively easy problems related to homeworks or programs, alone or as a member of a group. If you can understand how to do these problems, it will go a long way toward helping you pass the course. I am trying new experiments with in-class work in a Zoom course this semester, the format may change throughout the semester.

### Homeworks

Homeworks ask questions that are more in-depth than quizzes. They must be turned in to the correct D2L homework assignment folder, in MS Word or PDF format. Assignments in the wrong folder will lose one point. To minimize the chance of this happening, I only leave one homework folder open at any one time.

Some homeworks involve drawing pictures of graphs. Use any tool you wish to draw them (even freehand is OK), but they must be embedded in the homework solution. It is fine to write a program to help you do homework problems. Please include your source code with your submission in the folder.

You may work together to try to understand the problems and solutions better. Everyone must write up and turn in their own homeworks. This means that any *n* papers with the exact same wording (or nearly so) will have the points divided by *n* and split among them. (Obviously, this doesn’t apply to fill-in-the-blanks type of problems). Remember that even though the exam is under 20% of the total credit fort he class, you must get at least 40% on it to pass the course. If you don't get to the point where you really understand the problems and their solutions, you may have difficulty passing the final exam.

Homeworks will be uploaded on Tuesdays, and will be due on Fridays (10 days later) at noon. Graded homeworks will be returned before class on Monday so we can talk about them in class.

### Program

There is one large Program in this course. It is divided into 4 Deliverables, A, B, C, D. Program submission is 12:00 PM (noon) on Saturdays, D2L Central Time. I give you an initial codebase to build on. Each deliverable will build on this codebase. You are expected to write program in such a way that they don’t break the previous deliverable.

Programs will be graded by the second class session after the program was due (9 days later). If your program did poorly, I will try to get it back to you sooner, with comments. Notice that one or two deliverables may have reports.

The tentative points per program, and program section, are as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Category | Deliv A | Deliv B | Deliv C | Deliv D |
| Correctness | 25 | 55 | 55 | 55 |
| Regression | 0 | 5 | 10 | 15 |
| Design | 0 | 5 | 5 | 5 |
| Documentation & Style | 0 | 5 | 5 | 5 |
| TOTAL | 25 | 70 | 75 | 80 |
| **GRAND TOTAL** | **250** | | | |

Your programs need to run on my Windows 10 computer. If your program doesn’t run there, it needs to be revised. In fact, the main purpose of Deliverable A is to be sure that it does run on my computer. Program I/O is usually the place where incompatibilities show up. Since I have gone to giving people an Initial Codebase which reads input files, there are few incompatibilities.

If you develop your program on a Mac, you definitely need to check it on a Metro State University Windows system to be sure it runs properly there. ***Programs must run without importing anything not in the Java Standard Edition.*** The most common – but not the only – things that people try to import are from the Apache project or NetBeans.

# **Schedule of Topics and Due Dates**

***The full course schedule and due dates are listed in a separate document in the course content section. This is an overview.***

Except for the first and last week, every week will look the same as far as quizzes, inclass exercises, and homework goes. This is for a hypothetical “Week *n*”:

Monday, noon: The quiz over Week *n*’s material (simple questions about the stuff we are going over that evening) is due.

Monday, 6 PM: Class begins on zoom. We do lots of problems, probably in groups. Note: The first week is special, since I don’t have a lecture for you to watch.

Tuesday: Homework problems concerning Week *n*, for 10 days hence, and the lecture notes for Week *n+1* quiz for Week *n+1* material due next Monday are uploaded. The Zoom lecture covering next week’s material is available.

Wednesday: The Quiz for Week *n+*1is uploaded.

Friday, noon: Homework problems from Week *n-1* are due.

On some Saturdays, a program deliverable is due at noon.

# **Miscellaneous**

## Incompletes for Course

If you choose to seek an incomplete grade in this course, you must provide a request through email to the instructor no earlier than the last date to withdraw from the course, and preferably at least 7 days prior to the end date of the term. The instructor may refuse to grant a grade of incomplete for students who have not sustained a passing grade (C- or better) in the course up to the date of the student's emailed request for an incomplete.

Remember, if you take an incomplete in a class, you have not passed the class until you’ve cleared the incomplete.

## Technical Assistance

**IT Helpdesk** ([it.desk@metrostate.edu](mailto:it.desk@metrostate.edu); 651-793-1240) provides general computer assistance.

**Center for Online Learning** ([online.learning@metrostate.edu](mailto:online.learning@metrostate.edu); 651-793-1650) provides general assistance with online learning and course access. Please include your tech id number and course name and number.

Appendix C contains a discussion of the technology requirements for students in this course. I’ve never known an upper-division computer science major who has had a problem with these.

## Plagiarism and Copyright

Plagiarism is defined as the act of claiming another person’s work as one’s own. This can be copying or copying even parts of sentences from an article, journal, internet site, another student’s work, or other written work. It can also mean using another student’s assignment and making minor changes. Changing a few words around in content is still plagiarism.

Sometimes students are not sure if they are committing plagiarism. Please consult the instructor regarding more information on plagiarism and how to avoid it. The instructor reserves the right to assess through <http://www.turnitin.com/> and/or asking for resources used. Student are encouraged to consult the Metropolitan State University Writing Lab and resources for assistance as well.

For a first offense, if multiple students submit the same assignment (homework or program), the points will be split among them and they will be written up as an Academic Integrity violation.

Future work that is found to be plagiarized will be given a grade of zero (0). This will also be written up as an Academic Integrity violation.

As part of academic honesty, students are expected to respect intellectual property laws, including copyright law. No parts of this site, your textbook, other publisher materials, or contribution from other class members can be used or distributed outside of this class without the permission of the author(s). Copyright law provides that any creative thought fixed into tangible form is copyrighted the moment it is created, whether or not a copyright notice is affixed.

Please review this excellent article on plagiarism to learn more. Parts of the article are posted below: <https://www.plagiarism.org/article/what-is-plagiarism>

Acts of plagiarism may be reported to the Associate Provost for Student Success.

Exceptions: If you use publicly available solutions for your homework or as part of the code for your program, and you document that resource that you used, this is perfectly OK and is not plagiarism.

# **Additional Resources for Students**

## Center for Accessibility Resources

### Diversity and Disability Statement:

Students with disabilities who experience barriers in this course are encouraged to contact the instructor. The Center for Accessibility Resources is available to facilitate the removal of barriers and ensure reasonable accommodations.

The Center for Accessibility Resources is located in New Main, Suite L223. Phone number is 651-793-1549 and email is [accessibility.resources@metrostate.edu](mailto:accessibility.resources@metrostate.edu).

### Tutoring:

The Center for Academic Excellence provides tutoring services free of charge in most academic areas. They can be reached at [centerfolk@metrostate.edu](mailto:centerfolk@metrostate.edu) or 651-793-1460. Tutoring is available this semester by Zoom only. Note: CSC department tutors concentrate on lower-division courses. It is possible that limited tutoring will be available for ICS 340, but check with the CAE to find out.

### Make up Exams:

Make-ups for the take-home portion of the final exam will be negotiated on an individual basis.

### Veterans and Military Student Services:

Programs and support for veterans and students connected to the military. They can be reached at [veterans.services@metrostate.edu](mailto:veterans.services@metrostate.edu) or 651-763-1561.

### TRIO Student Support Services:

Support for first-generation students, low-income students and students with disabilities. They can be reached at [trio.center@metrostate.edu](mailto:trio.center@metrostate.edu) or 651-793-1525.

### Counseling Services:

Mental health support for individuals and groups. They can be reached at [counseling.services@metrostate.edu](mailto:counseling.services@metrostate.edu) or 651-763-1568.

### Library and Information Services:

The university's hub for information, research help, study rooms and more. You may reach them online through the [Library pages](https://www.metrostate.edu/library) on the Metropolitan State University website or at 651-793-1616.

### LGBTQ+ Student Services:

Programs and advocacy for Metropolitan State University’s LGBTQA community. They can be reached at 651-793-1300 or through the [LBGTQ pages](https://www.metrostate.edu/students/support/glbtq) on the Metropolitan State University website.

# **University Academic Policies & Information**

## University Non-Attendance and Reporting Policy and Procedure

The purpose of the Non-Attendance and Reporting Policy is to ensure Federal Title IV regulations are adhered to with respect to a student’s enrollment level for the purpose of calculating and paying financial aid.  While Metropolitan State University is not required to take attendance, Federal Title IV financial aid regulations require a procedure to establish that students have attended, at a minimum, one day of class for each course in which the student’s enrollment status was used to determine eligibility for the Pell Grant Program.  In addition, the university needs to determine a last date of attendance for those students who receive all failing grades or unofficially withdraw.

Attendance is defined based on course delivery mode. A student is “in attendance” if he or she meets the following conditions before the end of the second week of the course:

• Classroom Courses – the student is present in the classroom.

• Web-Enhanced (Reduced Seat Time Courses) – the student is present in the classroom or submits at least one academically relevant assignment.

• Online Courses –the student submits at least one academically relevant assignment

• Independent Studies – the student contacts the instructor or submits at least one academically relevant assignment.

**Note: The above description is the University Policy, but for this course, in accordance with department policy, participation must occur during the first week of class or the student is dropped from the course.**

## Academic Appeals

The university has written procedures for appealing decisions concerning grades. The student should first attempt to resolve an appeal issue informally with the instructor. To file a formal appeal, the student must begin the formal appeal process within 60 calendar days of the posting of the grade or evaluation. A staff member in Student Affairs serves as ombudsperson to work with students in preparing formal appeals. For details, refer to [Procedure 202, Academic Appeal Procedure](https://www.metrostate.edu/about/policies/7111).

Generally, the next step is to appeal to the department chair. However, I’m the department chair. So for this course, if we can’t reach an agreement, you will need to appeal directly to the Dean.

## Email

Metropolitan State University has designated e-mail as an official method of communication with students. **The university expects** students to be responsible for all information sent to them via their official university email account.  Refer to [Policy 1050, University E-mail](https://www.metrostate.edu/about/policies/6771), for further information.

## Student Code of Conduct

Students at Metropolitan State University deserve the opportunity to pursue an education, and it is the responsibility of the university to provide an environment that promotes learning and protects the safety and well-being of the university community. Therefore, the university establishes this Student Conduct Code. Any action by a student that interferes with the education of any other student or interferes with the operations of the university in carrying out its responsibility to provide an education will be considered a violation of this code.

The Student Code of Conduct balances individual student due process rights with the broader interests of the safety, wellbeing and academic integrity of the university community. The Office of Judicial Affairs operates with the philosophy of balancing the need for student accountability with the opportunity for education and making amends. Students are encouraged to review the [Student Conduct Code University Policy #1020](https://www.metrostate.edu/about/policies/6746) and the [Student Conduct Code Procedure #112](https://www.metrostate.edu/about/policies/6876)to understand their rights and responsibilities under the Code.

## Academic Integrity

The University does not accept knowingly copying the work of others without attribution (plagiarism), or colluding with other students to share answers unless permitted by the instructor (e.g. group project). At my discretion, the consequence of these activities may include failure for the assignment, failure for the entire course. You should be aware that the university subscribes to plagiarism detection software, and that your papers may be selected for plagiarism checking. In instances of plagiarism or other forms of academic dishonesty, instructors may impose academic sanctions. Allegations of plagiarism or other forms of academic dishonesty are also subject to investigation and additional conduct sanctions under the [Student Academic Integrity Policy #2190](https://www.metrostate.edu/about/policies/7081), and [Procedure #219: Student Academic Integrity](https://www.metrostate.edu/about/policies/7156).

If you have questions about the use of footnotes or other notations, talk to your instructor, consult the Library and Information Services website, or seek assistance in the proper way of writing a paper by contacting a tutor in the [Center for Academic Success](https://www.metrostate.edu/academics/success/tutoring). Students who believe that they have been falsely accused of plagiarism should request assistance from the Ombudsperson at (651) 793-1517.

# **Appendix A – Why Are Assignments Due at Noon?**

I’f often asked why assignments are due at noon instead of 11:59 PM. My reason is because I want people to be able to hand in late work for credit without pulling all-nighters.

I want to try to hold to a strict deadline, for a couple reasons:

* I want to publish the solutions to homework problems shortly after they’re due, so you can see what the right answers were while the problems are still fresh in your mind.
* I want to be able to start grading homeworks and programs shortly after they are due, so that I can get graded work back to you in a timely fashion. (I find it horrendous that some instructors, here and elsewhere, take up to a month to return submitted work.)

If my deadline is noon, that’s easy to remember, and I can penalize people one point per hour (not a big penalty), and still have the assignments all available to me that evening for me to start grading (no, I don’t always manage to start grading that evening, but I sure try to).

# **Appendix B – Why Require 40% on Everything to Pass the Course?**

A Bachelor’s degree is a general degree, not a specialty degree. I believe it should give you an education that is “a mile wide and (at least) an inch deep”. I think that it’s important that a graduate with a Computer Science major be able to do the wide range of tasks that a Computer Science graduate might be asked to do, at least a little bit. For this course, in particular:

* Analyze algorithms to determine how fast their time (and perhaps space) requirements grow with problem size.
* Program, in at least some high level language (we use Java) well enough to do something.
* Work together, at least a little, with other professionals.
* Submit things in a timely manner.

There are three experiences that I’ve had that have led me to this 40%-on-everything requirement.

* I’ve had calls from employers over the years – very few (but even one is too many) – asking about a certain graduate “How could you possibly give this person a Computer Science degree? They can barely write a `Hello, World’ program?” I really, really don’t want to ever get a call about one of our graduates like this again. You’ve got to be able to write programs that will actually run and do something to get through this course.
* I’ve had a really bright student in this class years ago who didn’t do any of the programs, and they pointed out that since they had a job and all they needed to was pass, and since they were getting virtually 100% on the non-programming assignments, they didn’t need to program. This did not, and does not, seem right to me.
* When I taught the Capstone, which I used to do shortly after leaving industry to teach full time, I had a student do absolutely nothing on the Ethics portion of the course (covered at that time in Capstone), and say “I don’t like ethics, and it’s only a quarter of the course, so I’m not going to do it.” I didn’t like that either.

# **Appendix C - Technology Requirements and Expectations**

This is a required part of the syllabus, but I’ve never had a student in this class who had a problem with the technology, so I’ve relegated this to an appendix.

## Computer Hardware and Software

For this course, your computer must fully pass the System Check found on the [D2L login page](https://metrostate.learn.minnstate.edu/). Students must have internet access in order to log into (this is required) the D2L system at least two days per week in order to check for updates and complete required work. Your skills should include the ability to add browser plug-ins for viewing files and content presented within the course or be able to get such assistance from non-campus sources at your own expense.

## File Management

Intermediate or higher level skills at file management (ability to create folders, move and rename folders and files, identify type of file by its file-extension, attach files to emails, etc.)

## Anti-Virus Software

Updated virus scanning software for all files sent and received (such as McAfee Antivirus, Norton Internet Security, etc.) is required.

## Computer Software

Computer skills include proficiency in Microsoft’s Internet Explorer (or other web browser) and in using the Internet to access online resources and sites as well as competence at using Microsoft Office products such as Word and Power Point.

## Email

This course requires students to use their campus email account for all communication for related to this class. Emails originating from outside the campus email servers may be deleted without review of the instructor. This policy prevents viruses and spam. Please include in the subject line of your emails the course number and a brief description that summarizes the content of the email as well. It is very important to at least include your course and course number in the subject line to assure proper feedback and credit for you work.

1. No, not the instructor, another Stein [↑](#footnote-ref-1)
2. See *Appendix A, Why Are Assignments Due at Noon?* For an explanation of why I chose this time. [↑](#footnote-ref-2)
3. See Appendix B, *Why Require 40% On Everything To Pass*, if you’re interested in learning why I do this. [↑](#footnote-ref-3)
4. Exception: Quiz W0 covers prerequisite material, and Quiz W1 covers the material of January 14th. [↑](#footnote-ref-4)