CSC316 - Data Structures & Algorithms

Sections 001
Spring 2020
3 Credit Hours

Course Description

The course will cover the following topics: Abstract data types; abstract and implementation-level views of data types. Linear and branching data structures, including stacks, queues, trees, heaps, hash tables, graphs, and others at discretion of instructor. Best, worst, and average case asymptotic time and space complexity as a means of formal analysis of iterative and recursive algorithms.

The course will cover a wide range of data structures and associated algorithms, including:

- Properties of programs, running time, and asymptotics
- Array and linked-memory implementations of lists, stacks, and queues
- Searching using lists, unbalanced tree structures (binary search trees, Splay trees) and balanced trees (2-3 trees, randomized binary search trees)
- Up-trees as sets with union-find operations
- Graphs and graph algorithms (traversals, shortest paths, minimum spanning trees)
- Sorting (heap sort, merge sort, insertion sort, selection sort, quick sort)
- Hash tables and hashing techniques

Learning Outcomes

The purpose of this course is to introduce the principles and underlying concepts of algorithm design, and enhance your problem solving and software development skills. To this end, a wide range of practical techniques for manipulating data in digital computers will be presented, along with a mathematical analysis of their performance.

Upon successful completion of this course, a student will be able to...

characterize the worst-case running time and space usage of algorithms and data structure operations as a function of input size;
identify when recursion is useful, and design and implement complex recursive and iterative algorithms, including sorting algorithms;
construct and use a number of data structures, including a stack, queue, linked list, array, tree, heap, graph, and hash table;
explain how abstract data types (e.g., sequences or graphs) can be represented as different data structures (e.g., adjacency lists or adjacency matrices);
describe and implement algorithms for binary search trees;
describe and implement algorithms on graphs, including breadth-first and depth-first search, constructing minimum spanning trees, and finding shortest paths;
describe and implement hashing functions and hash tables.
Course Structure

Overview

We will meet as a class two times per week, 75 minutes each. Class periods will consist of lectures accompanied by in-class exercises in which you apply the concepts of the course. Outside of class, you will also complete course projects and labs. You are expected to spend on average 6 to 12 hours per week outside of class preparing and working on assignments.

Course Website

To help you navigate all of the tools and materials in the course, the instructor has provided a navigation page at https://gs.ncsu.edu/csc316.

We will be using Moodle for the academic integrity quiz, for extra credit reflections, and for the course gradebook. When you log into the Moodle system, this course section will be listed.

Most of the semester’s submission lockers will be within the Gradescope system or through NCSU GitHub. The message board will be within Piazza. To access ALL course information, use the following URL: https://gs.ncsu.edu/csc316.

Student Concerns

You must inform your instructor as soon as possible of anything that may prevent you from completing coursework and exams as well as any other concerns that you may have.

Academic Integrity Contract

At the beginning of the semester, the instructor will review academic integrity policies. Before 11:59PM on the current semester's Census Date (see: https://studentservices.ncsu.edu/calendars/academic/), you must complete an online Academic Integrity quiz to show that you have reviewed and understood the course academic integrity policies. If you do not complete the academic integrity quiz, earn a 100%, and sign the contract by the stated deadline, none of your semester assignment submissions will be graded, and you will receive a 0 on each.

Development Environment

For both labs and projects, all code must be pushed to NC State's GitHub to an instructor-provided repository. All coding activities and project implementations must be completed using Java and by reviewing feedback from Jenkins. We will be using a continuous integration server (Jenkins) to automatically compile and test your program (both with your tests and the teaching staff tests) and provide style and efficiency feedback. Your grade for coding activities and project implementations will be calculated from the last GitHub submission you make before the deadline (even if Jenkins runs after the deadline for that submission) plus additional points for acceptance tests, SpotBugs issues, and other related rubric items. The style deductions as derived from Jenkins feedback may be modified by the teaching staff when manually inspecting your comments. For programming portions of labs and projects, use of the Eclipse Integrated Development Environment (IDE) is required.

Testing and Debugging

Since CSC216 is a prerequisite course for CSC316, that means you have successfully completed both CSC116 and CSC216 (or have equivalent course credit). Recall the following learning objectives for the CSC116 course:

Correct syntax errors and distinguish between syntax, logic, and runtime errors
Find and correct logical programming errors using debugging printout, pencil-and-paper tracing, and systematic search (to locate where an incorrect decision or value first appears)
Verify and validate programs using white- and black-box testing
Similarly, recall the following learning objectives for the CSC216 course:

- Describe the utility of inheritance, abstract classes, interfaces, and polymorphism in object-oriented systems, and design, implement, and test programs which use these language features;
- Identify and compare the basic kinds of software testing, describe when to use each method, and design and implement test code;
- Use software engineering best practices like pair programming, test-driven development, code coverage, static analysis, version control, continuous integration, and documentation with supporting tooling to design, implement, and test object-oriented systems.
- Identify when recursion is useful, and design, implement, and test recursive algorithms and simple recursive data structures;
- Implement, test, and use a stack, queue, array-based list, and linked list.

At this point in the computer science program, you are expected to test and debug your own software! Teaching staff test cases will not provide any hints on assignments this semester. If you fail any teaching staff test cases, that means you should:
- consider writing additional test cases so that all requirements are covered
- consider testing additional boundary cases
- double-check that your understanding of data structure operations is accurate and that your expected results are correct

---

**Testing and Debugging are YOUR Responsibility**

CSC216 Instructors and TAs will not help debug your code for you. Office hours should be focused on understanding algorithms, algorithm analysis, and data structure operations. We are happy to walk through examples of data structures and algorithms to help with your understanding of course material, but debugging code is your responsibility.

If you are “stuck” on an assignment and are having trouble making adequate progress, please visit office hours or describe your situation on the message board so that we can give you suggestions for moving forward and making progress.

Teaching staff test cases give you limited feedback before each deadline for your convenience so that you can estimate your progress:

- Teaching staff test cases are not a substitute for writing your own thorough test cases.
- You should not rely solely on teaching staff test cases to debug your software!

In industry, there are no “teaching staff test cases” – it is your responsibility to test and debug your software!

---

**Labs**

There are 12 labs this semester. Labs will be submitted electronically by the due date. All programs must be completed using Java, NCSU GitHub, and by reviewing feedback from Jenkins.

Labs are organized by topics – some labs require less effort than others. Please plan ahead and use your time wisely. **Do NOT wait until the last minute to complete labs!!!**

Labs consist of two components:

**Written Activity**

- **Description:** Written activities involve a short set of 2-5 questions about theory of algorithms and data structures. Responses will be submitted in Gradescope.
- **Collaboration Policy:** Written activities may be completed individually, though we encourage you to work in pairs to help facilitate timely grading and feedback.
- **Late Submission Policy:** Written activity deliverables will NOT be accepted late!

**Coding Activity**
**Description:** Coding activities involve implementing and testing data structures and algorithms using guided steps that will be provided by the teaching staff. **Lab coding activities are required for projects to be completed:** projects will use your data structure implementations from labs.

**Collaboration Policy:** Coding activities must be completed individually.

**Late Submission Policy:** Lab coding activity deliverables will receive a late deduction of -1 points for every 2 hours late, up to 48 hours late. The timestamp of your last lab build on [Jenkins](https://jenkins) will determine any late penalty.

## Projects

There are 2 course projects this semester. Projects will be submitted electronically by the due date.

Please plan ahead and use your time wisely. **Do NOT wait until the last minute to complete projects!!!**

### Project Part 1s

**Description:** During Part 1 of each project (you will have at least 4 days to work on Part 1 after the description is posted), you will submit a proposal document in which you will

- Develop and analyze an algorithm for solving a given problem
- Select and justify choice of data structures for efficiently solving the given problem
- Develop a software design.

Project Part 1s will also undergo a peer review process, in which your draft proposal will be reviewed by another pair of students.

**Collaboration Policy:** Project Part 1s must be completed in pairs. Both team members are responsible for the final deliverables. Any team-related issues must be communicated to the instructor at least 72 hours before the deliverable deadline. No changes to teams will be made within 72 hours of the deliverable deadline.

**Late Submission Policy:** Project Part 1 deliverables will NOT be accepted late!

### Project Part 2s

**Description:** During Part 2 of each project (you will have at least 7 days to work on Part 2 after the description is posted), you will submit

- All code to NCSU [GitHub](https://github)

**Collaboration Policy:** Coding activities must be completed individually.

**Late Submission Policy:** Project Part 2 deliverables will NOT be accepted late! The timestamp of your last project build on [Jenkins](https://jenkins) will be used for grading.

### Project Part 3s

**Description:** During Part 3 of each project (you will have at least 4 days to work on Part 3 after the description is posted), you will submit

- A project experiment report, which must include results of the experiment (including tables and charts) to demonstrate experimental analysis and performance comparisons of different data structures that could be used to solve the problem.
- The project report will also include reflection and discussion, based on the results of the experimental analysis.

**Collaboration Policy:** Project Part 3s must be completed in pairs. Both team members are responsible for the final deliverables. Any team-related issues must be communicated to the instructor at least 72 hours before the deliverable deadline. No changes to teams will be made within 72 hours of the deliverable deadline.

**Late Submission Policy:** Project Part 3 deliverables will NOT be accepted late.

## In-Class Exercises

During each lecture period, you will complete several small in-class exercises through TopHat. These exercises will be applications and review of new course concepts. Exercises will be graded on correctness and participation. You will only receive credit for these exercises if you are present in class. Attendance may be taken at any time during the class period (at the beginning, middle, or end). Attendance may also be taken multiple times during the same class period. **If you miss more than 4 lectures without a documented excused absence, your exercise/participation grade used to calculate your course average will automatically be 0.**
You are strongly encouraged to work in groups to complete and discuss in-class exercises, but each student must submit answers in the response system. If you are absent from class with an excused university absence, you will not be penalized for missing the in-class exercise.

At the end of the semester, your in-class exercise score must be at least 75% in TopHat to earn full exercise/participation in your course average. Otherwise, if your in-class exercise score is less than 75%, the 4 points allotted for in-class exercises in your course average calculation will be multiplied by your in-class exercise score percentage. For example:

- if your in-class exercise score is between 75-100%, then the exercise/participation component used to calculate your course average will be 100% * 4 = 4 points
- if your in-class exercise score is 74%, then the exercise/participation component used to calculate your course average will be 74% * 4 = 2.96 points
- if your in-class exercise score is 61%, then the in-class exercise component used to calculate your course average will be 61% * 4 = 2.44 points.
- if your in-class exercise score is 30%, then the in-class exercise component used to calculate your course average will be 30% * 4 = 1.2 points.

**Exams**

There will be three exams in this course: two exams during the semester and one final exam. Each exam will be comprehensive and cumulative. Missed exams cannot be made up without an official university excuse. Exams will have randomly assigned seating, and seating assignments will be posted within 24 hours of the exam period.

**Exam Policies**

Be on time for exams! Any student who arrives to the exam after another student has already turned-in an exam and left the classroom will not be allowed to take the exam and will receive a 0%.

Your exam must be placed in the designated location at the front of the classroom before the specified end time of the exam. The instructor will immediately collect the stack of exams at the end of the exam time period. If your exam is not placed in the designated location by the end of the exam time, your exam will not be graded, and you will receive a 0%.

**Extra Credit Videos**

You have the opportunity to earn up to 0.5 point extra credit on your final course average. **Final course averages are not curved**, and 0.5 point can easily bump you above a grade threshold (for example, from B+ to A-). During the semester, 5 videos of notable computer scientists or course topics will be posted in Moodle. After watching the video, you will be presented with a series of multiple choice questions to ensure you watched the video, followed by a reflection question for you to document your thoughts about the video and how it connects to course topics. Each video is worth up to 0.1 points. Deadlines for the video activities will be spread throughout the semester. No late submissions will be accepted.

Since deadlines for the video activities will be spread throughout the semester, you cannot wait until the last week of the semester to decide to earn this extra credit opportunity!

**Posting Code**

While your deliverable is your work, the project requirements and design are the intellectual property of the instructors and the university. You may not post a project solution publicly during or after the semester. See the syllabus section on [Academic Integrity](#) for more information.

**Course Policies**

**General Course Policies**

Count on last minute system failures, power outages, laptop issues, internet connectivity issues, etc. Make sure you are backing-up your work and frequently pushing to your remote NCSU [GitHub](#) repository so that you are able to at least receive partial credit. A broken computer, power outage, etc. are not acceptable excuses for requesting extensions on assignment deadlines!

**Regrade Requests**
If at any time you feel an assignment was graded improperly, you must submit a request for regrade via Gradescope. For assignments not submitted in Gradescope, submit the Regrade Request Form (http://go.ncsu.edu/regrade-csc316), and explain why you believe the assignment was graded improperly. You will first discuss the grade with the TA who graded the assignment. If you are still unsatisfied with the answer, submit the assignment to the instructor for a regrade. All regrade requests must be submitted to the instructor no later than 1 week after the assignment was returned to you. Assignments returned within one week of the final exam must have all regrade requests submitted by the start of the final exam.

Computers and Electronic Devices

Students are encouraged to use computers and other electronic devices like tablets during class. The teaching staff asks that students respect their neighbors and keep their focus on course materials rather than games, social media, working on course assignments (for this course or any other course), etc. Electronic devices are required for submission of exercises. You may not record the lecture without express written permission from the instructor.

Electronic Communication

The teaching staff looks forward to receiving emails and message board posts about any questions you have about the class, materials, exams, and assignments. Below are several rules for electronic communication.

Higher education provides you with a training ground prior to entry into the work environment for your chosen career. You will use many of the following rules of professional communication when you are communicating with colleagues, your supervisor, or clients once you are in the work world. Although many of the rules of etiquette for electronic communication will be similar in the work environment, we have some specific to this course.

Please observe the following etiquette when communicating with the teaching staff and your peers. The teaching staff receives many emails on a daily basis and the instructor teaches several courses. Please note that a member of the teaching staff will respond to an email or message board post within 24 hours on a business day and within 48 hours on a weekend or holiday. Most of the time, we will respond more quickly, but it is not guaranteed.

Also, before sending an email, try to find the answer to the question by using various references already available to you:

- If the question is related to class administration, check the syllabus
- If the question is related to recent information, check previous emails from the teaching staff
- If the question is project or exam related, check the message board to see if it has already been answered. Also, read your textbook.

For emails, please identify your course, section, and your name in the subject line (first and last name) along with the subject of the message. For example: “CSC316-001 Jenny Smith - Question about Project 1 Part 1”

Email should include a salutation to identify the recipients of the email. For example, begin an email to your instructor with a salutation such as “Hi Dr. Schmidt;” or “Dr. Schmidt:” For emails to the entire course teaching staff, consider a salutation like “Greetings Teaching Staff;” You now have the attention of the email recipients.

The tone of the email message should be professional. Re-read your email before you press Send and make a judgment as to how you would respond if you were a recipient of the email you are planning to send.

If you have a question that is beyond the scope of an email, consider coming to office hours or scheduling an appointment with a member of the teaching staff.

If you have several questions or items, please number them for ease of reading. The response will also be easier to understand.

Please spell check and correct mechanical/grammar errors. Avoid emails written only in lowercase and lacking punctuation. Close your email with your name. If you have a general question about a homework, post your question to Piazza. If you have a question that is more specific, that involves snippets of code, or that involves a grade question, make a private Piazza post or email it to the support list for your section.

All communication must follow professional norms. That includes emails, message board posts, in-person communication, and peer evaluations. Any threatening language or behaviors will be reported to the appropriate authorities. Repeated instances of unprofessional communication may result in a 2 point deduction (per instance) from your overall course average, at the discretion of the instructor.

Instructor
Instructor: Dr. Jessica Young Schmidt
Email: jessica_schmidt@ncsu.edu
Web Page: https://people.engr.ncsu.edu/idyoung2
Phone: N/A
Office Location: EB2 2258 (Mondays/Wednesdays)
Office Hours: (Posted on Course website)

Course Meetings

Section CSC316-001 Lecture

Days: Mondays & Wednesdays
Time: 8:30AM - 9:45AM
Campus: Centennial
Location: 1103 James B Hunt Jr Library

This meeting is required

Course Materials

Textbook

Data Structures and Algorithms in JAVA - M. T. Goodrich, R. Tamassia
ISBN: 978-0470383261
Web Link: http://bcs.wiley.com/he-bcs/Books?action=index&itemId=1118771338&bcsId=8635
Cost: $135.41

This textbook is required

Expenses

None.

Materials

None.

Requisites and Restrictions

Prerequisites

CSC 216 and CSC 228 with a grade of C or better;
CSC, CSU Majors and Minors and CPE Majors

Co-requisites
General Education Program (GEP) Information

GEP Category
This course does not fulfill a General Education Program category.

GEP Co-requisites
This course does not fulfill a General Education Program co-requisite.

Transporation
This course will not require students to provide their own transportation. Non-scheduled class time for field trips or out-of-class activities is NOT required for this class.

Safety & Risk Assumptions
None.

Grading

Grade Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Weight</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Projects</td>
<td>26</td>
<td>Two projects will be weighted at 13% each</td>
</tr>
<tr>
<td>Labs</td>
<td>24</td>
<td>The average of all lab assignments during the semester. Your lowest lab grade will be dropped and not used when calculating your final course average.</td>
</tr>
<tr>
<td>Exam 1</td>
<td>14</td>
<td>Exam 1 is cumulative</td>
</tr>
<tr>
<td>Exam 2</td>
<td>14</td>
<td>Exam 2 is cumulative</td>
</tr>
<tr>
<td>Final Exam</td>
<td>18</td>
<td>The final exam is cumulative</td>
</tr>
<tr>
<td>Exercises/Participation</td>
<td>4</td>
<td>Your in-class exercise score must be at least 75% to earn full exercise/participation in your course average. Otherwise, if your in-class exercise score is less than 75%, the 4 points allotted for in-class exercises in your course average calculation will be multiplied by your in-class exercise score percentage. Students are allowed to miss up to 4 lectures throughout the semester without penalty. If you miss more than 4 lectures without a documented excused absence for each, your exercise/participation grade used to calculate your course average will automatically be 0.</td>
</tr>
</tbody>
</table>

Minimum Grade Requirement
To pass CSC316, you must meet the following two requirements:

1. a weighted average of 60% or higher on the following components: Exam 1, Exam 2, Final Exam, and Lab Coding Activities.

\[
MinimumGradeRequirement_1 = \frac{(0.14 \times \text{Exam1}) + (0.14 \times \text{Exam2}) + (0.18 \times \text{FinalExam}) + (0.12 \times 100 \times \sum_{n=1}^{13} \text{LabCodingActivity}_n)}{600} \]

2. a weighted average of 60% or higher on the following components: Lab written activities and Projects.

\[
MinimumGradeRequirement_2 = \frac{(0.13 \times \text{Project1}) + (0.13 \times \text{Project2}) + (0.12 \times 100 \times \sum_{n=1}^{13} \text{LabWrittenActivity}_n)}{600} \]

**Letter Grades**

This course uses standard NCSU Letter Grading:

<table>
<thead>
<tr>
<th>Grade</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>A+</td>
<td>97 ≤</td>
</tr>
<tr>
<td>A</td>
<td>93 ≤</td>
</tr>
<tr>
<td>A-</td>
<td>90 ≤</td>
</tr>
<tr>
<td>B+</td>
<td>87 ≤</td>
</tr>
<tr>
<td>B</td>
<td>83 ≤</td>
</tr>
<tr>
<td>B-</td>
<td>80 ≤</td>
</tr>
<tr>
<td>C+</td>
<td>77 ≤</td>
</tr>
<tr>
<td>C</td>
<td>73 ≤</td>
</tr>
<tr>
<td>C-</td>
<td>70 ≤</td>
</tr>
<tr>
<td>D+</td>
<td>67 ≤</td>
</tr>
<tr>
<td>D</td>
<td>63 ≤</td>
</tr>
<tr>
<td>D-</td>
<td>60 ≤</td>
</tr>
<tr>
<td>F</td>
<td>0 ≤</td>
</tr>
</tbody>
</table>

**Requirements for Credit-Only (S/U) Grading**

In order to receive a grade of S, students are required to take all exams and quizzes, complete all assignments, and earn a grade of C- or better. Conversion from letter grading to credit only (S/U) grading is subject to university deadlines. Refer to the Registration and Records calendar for deadlines related to grading. For more details refer to [http://policies.ncsu.edu/regulation/reg-02-20-15](http://policies.ncsu.edu/regulation/reg-02-20-15).

**Requirements for Auditors (AU)**

Information about and requirements for auditing a course can be found at [http://policies.ncsu.edu/regulation/reg-02-20-04](http://policies.ncsu.edu/regulation/reg-02-20-04).

The grade of “AU” will be awarded to students who take all exams and earn a 60% or higher average on all of the exams. Auditors are required to meet with the instructor during the first two weeks of the course.

**Policies on Incomplete Grades**

If an extended deadline is not authorized by the instructor or department, an unfinished incomplete grade will automatically change to an F after either (a) the end of the next regular semester in which the student is enrolled (not including summer sessions), or (b) the end of 12 months if the student is not enrolled, whichever is shorter. Incompletes that change to F will count as an attempted course on transcripts. The burden of fulfilling an incomplete grade is the responsibility of the student. The university policy on incomplete grades is located at [http://policies.ncsu.edu/regulation/reg-02-50-3](http://policies.ncsu.edu/regulation/reg-02-50-3).
Late Assignments

Count on last minute system failures, power outages, laptop issues, internet connectivity issues, etc. Make sure you are backing-up your work and frequently pushing to your remote NCSU GitHub repository so that you are able to at least receive partial credit!

- **Lab Coding Activity Late Submission Policy:** Lab coding activity deliverables will receive a late deduction of -1 points for every 2 hours late, up to 48 hours late. The timestamp of your last push to NCSU GitHub will determine any late penalty.
- **Lab Written Activity Late Submission Policy:** Written activity deliverables will NOT be accepted late!
- **Project Part 1 Late Submission Policy:** Project Part 1 deliverables will NOT be accepted late!
- **Project Part 2 Late Submission Policy:** Project Part 2 deliverables will NOT be accepted late!
- **Project Part 3 Late Submission Policy:** Project Part 3 deliverables will NOT be accepted late!

Attendance

For complete attendance and excused absence policies, please see [http://policies.ncsu.edu/regulation/reg-02-20-03](http://policies.ncsu.edu/regulation/reg-02-20-03)

Attendance Policy

Attendance in CSC316 is mandatory. In-class exercises and participation account for 4% of your course average, and exams may contain questions on material presented during the lecture period.

Absences Policy

Excused absences are defined in the NC State Academic Policy on Attendance Regulations [http://policies.ncsu.edu/regulation/reg-02-20-03](http://policies.ncsu.edu/regulation/reg-02-20-03). Documentation of the absence is required to excuse an absence.

- Exam makeups will only be given with a documented excused absence.
- Lab and Project extensions will only be given with a documented excused absence. If the project solution has already been released, an alternative assignment may be given.
- Exercise and homework waivers will only be given with a documented excused absence.

All anticipated absences must be presented to the instructor no later than one week before the absence. All emergency absences must be turned in no later than one week after the student's return date. All other absences will be unexcused.

- You are allowed to miss up to 4 in-class exercises throughout the semester without penalty. **If you miss more than 4 lecture days without a documented excused absence for each, your exercise/participation grade used to calculate your course average will automatically be 0.**

Makeup Work Policy

All projects and exams must be made up within one week of the absence, and the timeframe will be determined through discussion between the instructor and student. If a project has moved forward in such a way that the missed project cannot be completed, the instructor may request the student to complete an alternative assignment. No exercises will be made up; students with documented excused absences will have exercises waived.

Additional Excuses Policy

None.

Academic Integrity

Students are required to comply with the university policy on academic integrity found in the Code of Student Conduct found at [http://policies.ncsu.edu/policy/pol-11-35-01](http://policies.ncsu.edu/policy/pol-11-35-01)

All members of the University community, students, faculty and other employees, have the responsibility to report academic misconduct to the appropriate authority.
The Computer Science department uses software that detects cheating violations for programming projects. Do not use other student's code, do not share your code, do not copy or use code from someone who took the class X semesters ago, do not use code from online. Start on assignments early so that you do not feel tempted to cheat!

**All work that you turn in for grading must be your own!** This means that all work must be an independent and individual creation by you or in the case of paired/team assignments; all work must be an independent and individual creation by you and your assigned partner or assigned teammates. Any attempt to gain an unfair advantage in grading, whether for yourself or another, is a violation of academic integrity. You may only work on an assignment with another student(s) in the class if explicitly stated in the assignment.

### Why is Academic Integrity Important?

(Adapted from Matt Stallmann and Mitchell Wand)

Would you want to fly in a plane whose controller software was designed and implemented by a group of people who had never demonstrated the persistence, attention to detail, and ability to deal with negative feedback from compilers, linkers, etc., that it takes to design, implement, and debug a program on their own?

Academic misconduct affects you, your peers, the CSC department, the university, all students who have ever graduated from NCSU with a CSC degree, and all users of software products to which you contribute. When you receive a degree from NCSU:

- The degree represents the university's certification that you have demonstrated certain skills and knowledge in your degree program.
- Your grade in a course represents the instructor’s certification that you have demonstrated certain skills and knowledge in the specific course.
- When an employer sees your degree from NCSU, they expect you to be able to demonstrate certain skills and knowledge. If a student graduates with a CSC degree and performs poorly, the value and reputation of a CSC degree from NCSU is negatively affected.

In industry, intellectual property rights are crucial in software and product development. Rules regarding intellectual property are similar to rules outlining academic integrity. Employees who “cheat” or violate copyrights or other intellectual property rights can cost the employer large sums of money. In addition, even though you will likely work on a team in industry, completion of the CSC degree program includes demonstrating skills to work effectively on teams. For example, students should demonstrate well-developed individual skills, integrity to take responsibility for one's own work, and the ability to recognize clear boundaries between one's own contributions and those of others.

As a future computing professional, you should review and abide by the ACM Code of Ethics and Professional Conduct.

### What are the Consequences of Academic Misconduct?

**Students who commit an academic integrity violation on any course deliverable will receive a -100% for the assignment!**

Violating the Academic Integrity Policy is worse than not turning in the item.

**Students who commit academic misconduct on more than one assignment during the course will receive no credit for the course (resulting in an F as the final grade on the student's transcript), making the course ineligible for Grade Exclusion under University Regulation 02.20.16.**

All cases of academic misconduct will be reported to the Office of Student Conduct.

A first offense will place the student on Academic Probation for the remainder of their academic career. Academic Probation is not visible on a student's transcript or other educational record, but the Office of Student Conduct does supply this information for various campus agencies running checks for disciplinary standings. If the student is suspended, the Office of Student Conduct may notify many other departments on campus, such as Registration & Records, Housing, Campus Health, Counseling, and Financial Aid. In addition, administrators of some scholarships routinely ask the Office of Student Conduct to confirm whether the student is in good standing.

### Resources you ARE Allowed to Use

You must cite your use of the approved resources in your assignment submissions. If you do not cite your use of the approved resources, you may be committing plagiarism.

The only people that you MAY receive help from:

- your instructor,
- the TAs for CSC316,
- for Project Part 1s, you may receive help from your assigned partner or your assigned teammates,
for Project Part 3s, you may receive help from your assigned partner or your assigned teammates, for Lab Written Activities, you may receive help from your one selected partner, and for exercises, you may work with any of your neighbors that are physically present in class.

The only external resources that you MAY also reference:

- your required textbook,
- the textbook website,
- the JAVA API HTML pages, and
- other third-party API HTML pages as appropriate for an assignment (for example, you may use the JUnit API HTML pages to help you with writing JUnit tests).

Resources you ARE NOT Allowed to Use

You are not allowed to receive help from anyone or anything else that is not in the list of approved resources (above).

If you think a resource should be added to the list of approved resources (above), you must first receive written permission from the instructor so that the instructor can share the resource with all students.

Examples of Academic Misconduct

Note: this list is not exhaustive.

It is aiding & abetting and cheating to give any student access to any of your work which you have completed for individual class assignments.

It is cheating and plagiarism to use another person's work and claim it as your own. You are expected to complete all assignments on your own, unless otherwise specified in the assignment.

It is cheating to interfere with another student's use of computing resources or to circumvent system security.

It is aiding & abetting and cheating to email, ftp, post on the Internet, bulletin boards, message boards, etc. your work for others to obtain. Do NOT use sites that allow you to “anonymously” post code. Those sites are searchable, and others may find your code (like the teaching staff).

It is cheating to ask or pay another person or persons to complete an assignment for you.

It is cheating and plagiarism to decompile any compiled code and use the decompiled source code as your own. You may also break the law by decompiling code.

It is cheating and plagiarism to use code that you find online, including code behind the Java API webpages.

It is aiding & abetting and cheating to give another student access to your account (NC State account or others that you use for university work) or to give them your account password.

It is aiding & abetting and cheating for you and another student to work collaboratively on an assignment, unless otherwise specified by the assignment.

It is cheating to circumvent the intention of the assignment and/or the automated grading system (e.g., by hardcoding test case solutions, by copying/pasting code provided in the Java libraries to fulfill an assignment objective, to implement extra lines of code to achieve higher statement coverage, etc.).

It is aiding & abetting to allow another student to copy from your written or electronic assignment submissions (e.g., it is the student's responsibility to cover his or her exam answers to help prevent others from copying answers)

It is cheating and plagiarism to copy from another student's written assignment (e.g., exams or homeworks).

It is cheating to submit identical or similar assignment submissions from an assignment submitted in a previous course, or a previous attempt of the current course.

It is cheating to reuse your code from previous semesters if retaking the course. Start over to focus your learning this semester.

It is aiding & abetting to leave your computer unlocked and/or unattended (whether intentional or accidental) such that others could access your assignments.

It is cheating to submit in-class exercises if you are not physically present in class.

It is aiding & abetting and cheating to give attendance codes to students who are not physically present in class.

Examples of NOT Cheating

Note: this list is not exhaustive.

Using the code from the class website (with citations in the comments).

Using code from other programs YOU wrote in this course during this semester (with citations in the comments).

Using code from other programs that YOU and a partner wrote as part of assigned exercises in this course during this semester (with citations in the comments).
Example Citations

```java
/* Citing Help from another Person: (In method or class level comments)
1 * I received help from Dr. Schmidt on DATE during his office hours. We discussed X.
2 */
3
/* Citing Help from other Assignments
4 * The code for this method is based on Exercise Y that I completed with Z on DATE.
5 */
6
/* Citing Help from the Textbooks
7 * The code for this method is based on the mergesort algorithm on page X in the course textbook
8 * "Data Structures & Algorithms" by Goodrich, Tamassia, Goldwasser.
9 */
```

Policies for Protecting Yourself

- Do not leave papers lying around your workstation.
- Cover your written exam responses with a cover sheet to prevent others from copying your responses.
- Do not dispose of important papers in the lab recycling bins and trash cans until after the assignment is graded.
- Do not give out your password.
- Do not leave your workstation unattended or forget to log yourself out.
- Do not leave your laptop unattended.
- Do not give other students access to any of your workspace or email them any code.
- Do not give other students access to your course materials or your personal computer.
- Do not email, ftp, or post your code on the Internet, message boards, etc.
- Keep all copies of final an intermediate work until after the assignment is graded.
- Keep all graded assignments until after you receive the final semester grade for the course.
- Do not discuss implementation details of the assignment with your peers.
- Do not discuss the contents of a course exam with other students, especially those students who have not taken the exam yet.

Ask the instructor for clarification of any questions or concerns about academic integrity policies before submitting an assignment.

Forum Use

The forum is available to ask questions about assignments and tests. **Do NOT post any code to the forum unless the post is private!** The teaching staff reserves the right to edit any student's forum post for inappropriate content. Additionally, use of the forum is a privilege. Improper use for the forum may result in a ban from posting or reading.

Posting Assignment Artifacts Online

While your deliverable is your work, the assignment artifacts (homework questions, project requirements, project design, etc.) are the intellectual property of the instructors and the university. You may not post any assignment artifacts (including assignment descriptions) or solutions to a publicly accessible website or public code repository during or after the semester.

Academic Honesty

See [http://policies.ncsu.edu/policy/pol-11-35-01](http://policies.ncsu.edu/policy/pol-11-35-01) for a detailed explanation of academic honesty.

Honor Pledge

Your name on any test or assignment or the electronic submission of an assignment through Moodle or other class courseware system indicates, “I have neither given nor received unauthorized aid on this test or assignment.”
Electronically-Hosted Course Components

Students may be required to disclose personally identifiable information to other students in the course, via electronic tools like email or web-postings, where relevant to the course. Examples include online discussions of class topics, and posting of student coursework. All students are expected to respect the privacy of each other by not sharing or using such information outside the course.

**Electronically-hosted Components:** The following materials are electronically-hosted for use by students through a combination of Moodle, TopHat, Google Docs (through NC State), GitHub, Jenkins, Piazza, and Gradescope: lecture notes, message boards, electronic submission of assignments, electronic submission of exercises, electronic evaluation of exams.

Accommodations for Disabilities

Reasonable accommodations will be made for students with verifiable disabilities. In order to take advantage of available accommodations, student must register with the Disability Services Office (http://www.ncsu.edu/dso), 919-515-7653. For more information on NC State's policy on working with students with disabilities, please see the Academic Accommodations for Students with Disabilities Regulation at http://policies.ncsu.edu/regulation/reg-02-20-01.

Non-Discrimination Policy

NC State University provides equality of opportunity in education and employment for all students and employees. Accordingly, NC State affirms its commitment to maintain a work environment for all employees and an academic environment for all students that is free from all forms of discrimination. Discrimination based on race, color, religion, creed, sex, national origin, age, disability, veteran status, or sexual orientation is a violation of state and federal law and/or NC State University policy and will not be tolerated. Harassment of any person (either in the form of quid pro quo or creation of a hostile environment) based on race, color, religion, creed, sex, national origin, age, disability, veteran status, or sexual orientation also is a violation of state and federal law and/or NC State University policy and will not be tolerated. Retaliation against any person who complains about discrimination is also prohibited. NC State's policies and regulations covering discrimination, harassment, and retaliation may be accessed at http://policies.ncsu.edu/policy/pol-04-25-05 or http://www.ncsu.edu/equal_op. Any person who feels that he or she has been the subject of prohibited discrimination, harassment, or retaliation should contact the Office for Equal Opportunity (OEO) at 919-515-3148.

In an effort to affirm and respect the identities of transgender students in the classroom and beyond, please contact the instructor if you wish to be referred to using a name and/or pronouns other than what is listed in the student directory.

Tentative Course Schedule

**NOTE** that the course schedule may change throughout the semester due to weather, illness, etc. All deadlines will be updated in Google Calendar throughout the semester.