CSC316 Course Syllabus

Data Structures & Algorithms

CSC316 Course Information

Spring 2024

Instructor Information

Name	Office Phone	Email	Office Location
Dr. Jason King	919-515-8954	jtking@ncsu.edu	EB2 2258

Dr. Jason King

Virtual Office Hours

Office hours are posted on the course website (https://wolfware.ncsu.edu). To join office hours:

1. Open https://beta.mydigitalhand.org
2. Join the CSC316 course (if you haven't already) using code `{page.officehourcode}`
3. During scheduled office hours, submit a help ticket.

4. Wait for the instructor to send you a Zoom link in MyDigitalHand.

Preferred Method of Communication

For emergencies, contact the instructor at the office phone number listed in the instructor information above. Leave a voicemail if the instructor is not available.

For all other communications that are not emergencies, contact the instructor at the email listed in the instructor information above.

Response Time

Please allow 24 hours for a response to any communication (i.e., email or message board), or up to 48 hours on weekends, holidays, or breaks.

Course Information

Course Website: course website (https://wolfware.ncsu.edu)

Course Credit Hours: 3

Course Meetings

Section	Description	Day & Time	Location
CSC316-001	On-Campus	10:15-11:30am Mondays and Wednesdays	Hunt Library Auditorium
CSC316-002	On-Campus	11:45AM - 1:00PM Tuesdays and Thursdays	EB2 1231
CSC316-601	Distance Education	Asynchronous	Online

Prerequisites/Corequisites

Prerequisites

- CSC 216 and CSC 226 with a grade of C or better;
- CSC, CSU Majors and Minors and CPE Majors

Corequisites

- none

General Education Program (GEP) Information

None.

GEP Category

None.

GEP Co-requisites

None.

Course Overview

Catalog Description

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 (2-4 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

Structure

- This course delivers all learning materials, activities, and assignments through **Moodle**, a secure and easy-to-use learning platform.
- For on-campus students, the course may be completed **in-person**, **asynchonously**, or a mixture of **both**.
- Learning activities include daily lecture exercises, weekly homeworks ("workshops"), 5 extra credit opportinities, 1 course project, 2 midterm exams, and a final exam.
- The course consists of 3 units, each lasting approximately 4-5 weeks. Each unit consists of 4 workshops and ends with an exam.
- Lectures will typically follow a two lectures-per week schedule.
- Lecture videos will typically require 2-3 hours of videos and exercises each week.
- Outside of class, students are expected to spend an additional 6 to 12 hours per week (on average) outside of class preparing and working on assignments. Exact expected times each week will depend on coding activities and your individual ability to effectively and efficiently debug your code.

> Do not wait until the last minute!

Count on last minute system failures, system maintenance, power outages, laptop issues, internet connectivity issues, NCSU GitHub maintenance, illness, etc. Make sure you are backing-up your work and frequently pushing to your remote NCSU GitHub (https://github.ncsu.edu) repository *often* 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!

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.

Learning Outcomes

Learning Outcomes

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

A. Data Structures

- A1. explain how abstract data types (e.g., lists, stacks, queues, maps, trees, priority queues, sets, and graphs) can be represented as different data structures;
- A2. construct and use linear data structures, including array-based lists, linked lists, list-based stacks, and array-based queues;
- A3. construct and use tree data structures, including general trees, binary trees, binary search trees, and balanced search trees
- A4. construct and use hash tables that use complex hash functions and collision resolution strategies, including chaining and open addressing;
- A5. construct and use priority queue data structures, including heaps;
- A6. construct and use union-find data structures, including up-trees;
- A7. construct and use graph data structures, including adjacency lists and adjacency matrices;
- A8. employ phases of software development to design, implement, and test a software solution that uses efficient data structures and algorithms to solve a given problem;

B. Algorithms

- B1. characterize the worst-case running time and space usage of iterative algorithms as a function of input size;

- B2. characterize the worst-case running time and space usage of recursive algorithms as a function of input size;
- B3. design and implement complex iterative algorithms
- B4. design and implement complex recursive algorithms
- B5. describe and implement sorting algorithms, including bubblesort, insertion sort, selection sort, mergesort, quicksort, heapsort, counting sort, and radix sort;
- B6. describe and implement tree algorithms, including tree traversals;
- B7. describe and implement graph algorithms, including breadth-first and depth-first search, constructing minimum spanning trees, and finding shortest paths;
- B8. describe algorithmic design paradigms, including divide-and-conquer, brute-force search, dynamic programming, and greedy algorithms;

Course Materials

Required Textbook and/or Software

Data Structures and Algorithms in JAVA - M. T. Goodrich, R. Tamassia

Edition: 6th edition

ISBN: 978-0470383261

Web Link: http://bcs.wiley.com/he-bcs/Books?action=index&itemId=1118771338&bcsId=8635 (http://bcs.wiley.com/he-bcs/Books?action=index&itemId=1118771338&bcsId=8635)

Cost: $130.05 ($39 for All-in Access through VitalSource) *This textbook is required*

Course lectures will be based on topics in the course textbook. The textbook contains code samples that will be helpful for homeworks/workshop coding activities.

Technology Requirements

Hardware

NC State's Online and Distance Education provides technology requirements and recommendations (https://www.engineeringonline.ncsu.edu/technical-support/technical-requirements/){:target="_blank" for computer hardware.

Software

- Moodle (https://delta.ncsu.edu/learning-technology/instructional-tools/moodle/) and Wolfware (https://delta.ncsu.edu/learning-technology/instructional-tools/wolfware/) for accessing course materials

  - Moodle Accessibility Statement (https://moodledev.io/general/development/policies/accessibility)
  - Moodle Privacy Policy (https://moodle.com/privacy-notice/)
  - NC State Privacy Policy (https://www.ncsu.edu/privacy/)
- Adobe Reader (https://get.adobe.com/reader/) for reading PDF files
  - Adobe Accessibility Statement (https://www.adobe.com/accessibility.html)
  - Adobe Privacy Policy (https://www.adobe.com/privacy/policy.html)
- Zoom (https://delta.ncsu.edu/learning-technology/instructional-tools/zoom-at-nc-state/) for virtual office hours
  - Zoom Accessibility Statement ()
  - Zoom Privacy Policy (https://zoom.us/privacy)
- Gradescope (https://delta.ncsu.edu/learning-technology/instructional-tools/gradescope/) for submitting project written reports
  - Gradescope Privacy Policy (https://www.gradescope.com/privacy)
  - Gradescope Accessibility Standards (https://help.gradescope.com/article/lpvwbzvqy2-faq-accessibility)
- Panopto (https://delta.ncsu.edu/learning-technology/instructional-tools/panopto/) for lecture viewing
  - Panopto Accessibility Features (https://support.panopto.com/s/article/Learn-About-Accessibility-Features)
  - Panopto Privacy Policy (https://www.panopto.com/privacy/)
- Google (https://google.oit.ncsu.edu/) for communication, collaboration, and more
  - Google Accessibility Statement (https://www.google.com/accessibility/customers-partners/)
  - Google Privacy Policy (https://edu.google.com/k-12-solutions/privacy-security/?modal_active=none)
- PlayPosit (https://delta.ncsu.edu/learning-technology/instructional-tools/playposit/) for interactive video exercises
  - PlayPosit Accessibility Statement (https://knowledge.playposit.com/article/82-accessibility)
  - PlayPosit Privacy Policy (https://app.playpos.it/privacy/)
- Ed Discussion (https://edstem.org/)
  - Ed Accessibility Statement (https://edstem.org/us/help/accessibility)
  - Ed Privacy Policy (https://edstem.org/au/privacy)
- MyDigitalHand (https://beta.mydigitalhand.org)
- NCSU GitHub (https://it.engr.ncsu.edu/services/github/) for submitting coding activities
- NCSU CSC316 Jenkins Countinuous Integration Server (https://go.ncsu.edu/jenkins-csc316) for reviewing feedback on teaching staff test cases

Minimum Computer and Digital Literacy Skills

- Obtain regular access to a reliable internet connection.

- Check NCSU email inbox at least once per day.
- Proficient typing and word processing skills (Google Docs)
- Ability to use online communication tools, such as email (create, send, receive, reply, print, send/receive attachments), discussion boards (read, search, post, reply, follow threads)
- Ability to use software development tools (including Eclipse, Eclipse Debugger, SpotBugs, PMD, CheckStyle, Git version control, Jenkins continuous integration server)
- Ability to write code using Java with JUnit test cases.
- Knowledge of copy/paste and use of spell and grammar check.
- Use computer networks to locate and store files or data.
- Properly cite information sources (see the Academic Integrity section of the course syllabus for examples of citations).

Course Assignments

Academic Integrity Contract

At the beginning of the semester, the instructor will review academic integrity policies. You must complete an Academic Integrity Quiz with a 100% score (with unlimited attempts) before you can access materials for the 2nd lecture of the course and beyond. As an absolute deadline, before 11:59PM on January 22, 2023, 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. If you do not complete the academic integrity quiz by the stated deadline, then we will deduct -1 point from your semester course grade for every 6 hours late until you contact the instructor via email to request an extension and complete the quiz.**

Development Environment

For both workshops and projects, all code must be pushed to NC State's GitHub (https://github.ncsu.edu) to an instructor-provided repository. All coding activities and project implementations must be completed using Java and by reviewing feedback from Jenkins (https://go.ncsu.edu/jenkins-csc316). We will be using a continuous integration server (Jenkins (https://go.ncsu.edu/jenkins-csc316)) 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 (https://github.ncsu.edu) submission

Didn't take CSC216 at NCSU?

If you did not take CSC216 at NCSU, please let the instructor know within the first week of class so that we can provide you with the instructions and materials (https://pages.github.ncsu.edu/engr-csc316-staff/Workshops/development-environment/) needed for you to you make before the deadline (even if Jenkins (https://go.ncsu.edu/jenkins-csc316) 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 (https://go.ncsu.edu/jenkins-csc316) feedback may be modified by the teaching staff when manually inspecting your comments. For programming portions of workshops and projects, use of the Eclipse Integrated Development Environment (IDE) is required

install and use the required Eclipse development environment and testing tools.

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

> **CSC316 Instructors and TAs will not 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!

Workshops (Weekly Homework)

There are 12 workshops (weekly homeworks) this semester. Workshops will be submitted electronically by the due date. All programs must be completed using Java, NCSU GitHub (https://github.ncsu.edu), and by reviewing feedback from Jenkins (https://go.ncsu.edu/jenkins-csc316).

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

Workshops consist of two components:

Written Activity

- **Description**: Written activities involve a short set of questions about theory of algorithms and data structures. Responses must be submitted in Moodle (https://wolfware.ncsu.edu) before the expected completion date for full credit, or by the quiz close date for credit with a late penalty. **NOTE: if a student does not attempt a written workshop activity in Moodle, then the student will not be able to access the solutions or questions for that activity after the deadline to review for exams.**
- **Collaboration Policy**: Written activities may be completed individually, though we encourage you to work in *study groups of no more than 3 total students*. Each student may partitipate in at most 1 study group for each workshop activity. Each student must submit their own written activity separately in Moodle.
- **Late Submission Policy**: Workshop written activity deliverables submitted after the stated deadline will receive a late deduction of-1 point for every 2 hours late, up to 48 hours late. The timestamp from when you submit your written activity attempt in Moodle will determine any late penalty.

Coding Activity

- **Description**: Coding activities involve implementing and testing data structures and algorithms using guided steps that will be provided by the teaching staff.
- **Collaboration Policy**: Coding activities must be completed individually.
- **Late Submission Policy**: Workshop coding activity deliverables submitted after the stated deadline will receive a late deduction of-1 point for every 2 hours late, up to 48 hours late. The timestamp of your last push to NCSU GitHub (https://github.ncsu.edu) will determine any late penalty.

Project

There is 1 course project 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 project deliverables!!!**

Project Part 1

- **Description**: During Part 1 of the 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 1 will also include a peer review process, in which your draft proposal will be reviewed by another pair of students.
- **Collaboration Policy**: Project Part 1 can be completed in optional pairs. If 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. **To be eligible to work on a team, you must have a workshop coding activity average >=60% and a workshop written activity average >=60%**
- **Late Submission Policy**: Project part 1 deliverables submitted after the stated deadline will receive a late deduction of-1 point for every 2 hours late, up to 48 hours late. The timestamp from when you submit your proposal document in Gradescope will determine any late penalty. No peer review submissions will be accepted late.

Project Part 2

- **Description**: During Part 2 of the 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.ncsu.edu)
- **Collaboration Policy**: Project Part 2 must be completed individually.
- **Late Submission Policy**: Project coding activity deliverables submitted after the stated deadline will receive a late deduction of-1 point for every 2 hours late, up to 48 hours late. The timestamp of your

last push to NCSU GitHub (https://github.ncsu.edu) will determine any late penalty.

Project Part 3

- **Description**: During Part 3 of the 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 3 can be completed in optional pairs. If 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 submitted after the stated deadline will receive a late deduction of-1 point for every 2 hours late, up to 48 hours late. The timestamp from when you submit your experiment report document in Gradescope will determine any late penalty.

Lecture Exercises

During each lecture, you will complete several small exercises through Moodle. These exercises will be applications and review of new course concepts. Exercises will be graded on participation and correctness. **Earning 0 points total across all exercises/reviews for a given topic block will count as a missed topic. If you miss more than 4 lecture topics, your exercise/participation grade used to calculate your course average will automatically be 0. **

At the end of the semester, 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 alloted 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

- **Description**: 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 NCSU Absence Verification.

- **Collaboration Policy**: Exams must be completed individually, with no assistance from any other person.
- **Late Submission Policy**: Exams cannot be submitted late. Makeup exams require NCSU Absence Verification and must be completed within 3 calendar days of the student's return from the absence.

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!

Grading

Grading Policy

Component	Weight	Details
Project	20	One course project, which contains 3 Parts: (1) project proposal, (2) implementation & testing, and (3) experiment report. Rubrics are provided for each part of the project.
Workshops	30	The average of your workshop assignment grades during the semester. Your lowest workshop grade from Workshops 1-11 will be dropped (Workshop 12 will not be dropped) when calculating the workshop average at the end of the semester. All 12 workshop grades still factor into calculations for meeting the minimum grade requirements for the course. In addition, if a student does not attempt a written workshop activity in Moodle, then the student will not be able to access the solutions or questions for that activity after the deadline to review for exams.
Exam 1	14	Exam 1 covers topics from "Problems, Algorithms, Programs" through "Recursive Algorithm Analysis"
Exam 2	14	Exam 2 covers topics from "Map Abstract Data Type" through "Hash Tables & Collision Resolution"

Component	Weight	Details
Final Exam	18	The final exam covers all course topics. During the final exam, all students will have an opportunity to improve Exam 1 and Exam 2 scores.
Exercises/Participation	4	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 lecture topics throughout the semester without penalty. **If you miss more than 4 lecture topics without a documented excused absence for each, your exercise/participation grade used to calculate your course average will automatically be 0.**

Minimum Grade Requirements

To pass CSC316, you must meet the following three requirements:

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

$$MGR_{Exams} = \frac{(0.14 \times Exam1) + (0.14 \times Exam2) + (0.18 \times FinalExam)}{0.46}$$

2. a weighted average of 60% or higher on the following components: workshop and project coding activities.

$$MGR_{Coding} = \frac{(0.039 \times 100 \times \frac{Project_{Part2}}{30}) + (0.12 \times 100 \times \frac{\sum_{i=0}^{12} WorkshopCodingActivity_i}{600})}{0.159}$$

3. a weighted average of 60% or higher on the following components: workshop and project written activities.

$$MGR_{Written} = \frac{(0.078 \times 100 \times \frac{Project_{Part1}}{30}) + (0.104 \times 100 \times \frac{Project_{Part3}}{40}) + (0.12 \times 100}{0.302}$$

Grading Scale

This course uses this grading scale:

Minimum Numeric Grade		Letter		Maximum Numeric Grade
97	≤	A+	≤	100
93	≤	A	<	97
90	≤	A-	<	93
87	≤	B+	<	90
83	≤	B	<	87
80	≤	B-	<	83
77	≤	C+	<	80
73	≤	C	<	77
70	≤	C-	<	73
67	≤	D+	<	70
63	≤	D	<	67
60	≤	D-	<	63
0	≤	F	<	60

Requirements for (S/U) or Audit

Requirements for Credit-Only (S/U) Grading

In order to receive a grade of S, students are required to meet the minimum grade requirements 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.

Course Policies

Netiquette

Students should be aware that their behavior impacts other people, even online. I hope that we will all strive to develop a positive and supportive environment and will be courteous to fellow students and your instructor. Due to the nature of the online environment, there are some things to remember when taking a course and engaging with others.

Tips for Success:

- **Do**: Follow the same standards of behavior that you subscribe to offline. Keep in mind that all online communication is documented and therefore permanent.
- **Don't**: Flame others in discussion forums. Flaming is the act of responding in a highly critical, sarcastic, or ridiculing manner – especially if done on a personal level. Remember that these discussions are meant for constructive exchanges and learning!
- **Do**: Ensure you are responding to emails in a timely manner, in order to leave time for peers to comment on your response.
- **Don't**: Go for long periods of time without communicating to your instructors or classmates. It is important to stay a part of the online community!
- **Do**: Remember to read over your posts or emails before selecting "Submit" or "Send"
- **Don't**: Use slang, poor grammar, and other informal language in discussion forums or email messages to instructors or classmates.

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: "CSC316-601 Jenny Smith- Question about Project 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. King," or "Dr. King". 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 the course message board. If you have a question that is more specific, that involves snippets of code, or that involves a grade question, make a private message board 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 threating 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.**

Students may be required to disclose personally identifiable information to other students in the course, via digital tools, such as 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

Policies on Incomplete Grades

If an incomplete grade is authorized by the instructor, the instructor will prepare a contract that outlines the assignments already completed, the assignments that need to be completed to resolve the incomplete, and the deadlines/timeline for the completion of these assignments. The student must sign and return the contract to the instructor no later than noon on the day course grades are due for the semester. No additional extensions will be granted beyond the updated deadlines outlined on the contract prepared by the instructor.

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).

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.

Late Assignments

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

- **Workshop Coding Activity Late Submission Policy**: Workshop 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 (https://github.ncsu.edu) will determine any late penalty.
- **Workshop Written Activity Late Submission Policy**: Written activity deliverables will receive a late deduction of-1 points for every 2 hours late, up to 48 hours late. The timestamp of your activity submission in Moodle will determine any late penalty.
- **Project Part 1 Late Submission Policy**: Project Part 1 deliverables will receive a late deduction of-1 points for every 2 hours late, up to 48 hours late. The timestamp of your latest submission in Gradescope will determine any late penalty.
- **Project Part 2 Late Submission Policy**: 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 (https://github.ncsu.edu) will determine any late penalty.
- **Project Part 3 Late Submission Policy**: Project Part 3 deliverables will receive a late deduction of-1 points for every 2 hours late, up to 48 hours late. The timestamp of your latest submission in Gradescope will determine any late penalty.

Late Penalty Waivers

If you are having a rough week or especially busy week, you may request a waiver of the late penalty for a Workshop or Project activity by submitting the Late Penalty Waiver request form (https://go.ncsu.edu/csc316-waiver-request) **before** the activity's on-time deadline. The late penalty will *not* be waived if the form is submitted **after** the assignment on-time deadline. Once a waiver request has been submitted, it cannot be canceled or unsubmitted.

The instructor will waive no more than 6 late penalties during the semester per student.

Assignment Corrections

Any mistakes in assignment solutions will be corrected and updated for all students because of Moodle autograding. If corrections are made to Moodle quiz questions solutions, Moodle will regrade all submissions. Your grade on an assignment may increase, decrease, or stay the same as a result of a regrade or correction to the solution for a question.

Reviewing Final Exam

If you wish to review your final exam submission, you must schedule an appointment with the instructor prior to Census Date of the upcoming/next semester. No final exam reviews will be allowed after Census Date of the semester following the completion of the final exam.

Regrade Requests

If at any time you feel an assignment was graded improperly, you must submit a request for regrade via Gradescope (https://www.gradescope.com). For assignments not submitted in Gradescope (https://www.gradescope.com), 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.

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)

To remain on schedule, students who complete lecture topics asynchronously must follow the topic schedule provided in the Google Calendar linked from the course Moodle page. Access restrictions in Moodle will prevent students from accessing/attempting assignments until the related lecture topics have been fully completed. No access restrictions for Moodle activities will be waived or removed for any students.

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 academic absence verification (https://dasa.ncsu.edu/support-and-advocacy/find-help/absence-verification/) and must be completed within 3 calendar days of the student's return from the absence.
- Workshop and Project extensions are not granted since workshop and project activities have an existing 48 hour late submission deadline. If you need an extension on a workshop or project activity (other than peer reviews, which cannot be accepted late), consider using one of your late penalty waivers. At the instructor's discretion, any additional workshop or project extension will only be given with academic absence verification (https://dasa.ncsu.edu/support-and-advocacy/find-help/absence-verification/) for lengthy absences. If the project solution has already been released, an alternative assignment may be given.

All anticipated absences must be presented to the instructor no later than one week before the absence. All emergency absences require NCSU Absence Verification, which must be received by the instructor no later than 7 calendar days after the student's return date. All other absences will be unexcused.

Makeup Work Policy

All makeup exams or activities must be made up within three calendar days 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.

Final exam reschedule requests (due to other exam conflicts) must be approved by the university and communicated with the instructor at least 1 week before the scheduled final exam start time. If you have a conflict with your final exam schedule (e.g., three or more consecutively scheduled final examinations within any 24-hour period), you can access the Exam Reschedule tool in MyPack by navigating through the Planning & Enrollment tile.

University Policies

Personally Identifiable Information

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.

Student Services

Accessing University Resources

There are additional links to student services available to you in the Student Services block located in the block drawer on the right side of the course site. These services include: Academic Support, community, health and wellness, basic needs, financial assistance, and other campus resources. To open the block drawer click on the button that looks like an arrowhead on the right side of the main course page. You can close the block drawer by clicking on the X.

Disability Resources

Reasonable accommodations will be made for students with verifiable disabilities. In order to take advantage of available accommodations, students must register with the Disability Resource Office at Holmes Hall, Suite 304,Campus Box 7509, 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 (REG02.20.01)

Students should send DRO accomodation letters to the instructor in the DRO portal by Census Date. Any new or updated accommodation letters during the semester must be sent to the instructor before the assignment begins, or accommodations may not be granted/available for that assignment.

Trans-Inclusive Statement

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

Basic Needs Security

Any student who faces challenges securing their food or housing or has other severe adverse experiences and believes this may affect their performance in the course is encouraged to notify the professor if you are comfortable in doing so. Alternatively, you can contact the Division of Academic and Student Affairs to learn more about the Pack Essentials program https://dasa.ncsu.edu/pack-essentials/

Course Evaluations

ClassEval is the end-of-semester survey for students to evaluate instruction of all university classes. The current survey is administered online and includes 12 closed-ended questions and 3 open-ended questions. Deans, department heads, and instructors may add a limited number of their own questions to these 15 common-core questions.

Each semester students' responses are compiled into a ClassEval report for every instructor and class. Instructors use the evaluations to improve instruction and include them in their promotion and tenure dossiers, while department heads use them in annual reviews. The reports are included in instructors' personnel files and are considered confidential.

Online class evaluations will be available for students to complete during the last two weeks of the semester for full semester courses and the last week of shorter sessions. Students will receive an email directing them to a website to complete class evaluations. These become unavailable at 8am on the first day of finals.

Resources:

- Contact ClassEval Help Desk: classeval@ncsu.edu (mailto:classeval@ncsu.edu)
- ClassEval website (http://go.ncsu.edu/cesurvey)
- More information about ClassEval (http://oirp.ncsu.edu/surveys/classeval)

Other University Policies

Students are responsible for reviewing the NC State University PRR's which pertains to their course rights and responsibilities:

- Equal Opportunity and Non-Discrimination Policy Statement (https://policies.ncsu.edu/policy/pol-04-25-05) and additional references (https://oied.ncsu.edu/equity/policies)
- Code of Student Conduct (https://policies.ncsu.edu/policy/pol-11-35-01)
- Grades and Grade Point Average (https://policies.ncsu.edu/regulation/reg-02-50-03)
- Credit-Only Courses (https://policies.ncsu.edu/regulation/reg-02-20-15)
- Audits (https://policies.ncsu.edu/regulation/reg-02-20-04)

Academic Integrity and Honesty

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). Therefore, students are required to uphold the university pledge of honor and exercise honesty in completing any assignment.

Please refer to the Academic Integrity (https://studentconduct.dasa.ncsu.edu/academic-integrity-overview/) web page for a detailed explanation of the University's policies on academic integrity and some of the common understandings related to those policies.

Honor Pledge

> Your name on any test or assignment **or** the electronic submission of an assignment through Moodle (https://wolfware.ncsu.edu) or other class courseware system indicates, “I have neither given nor received unauthorized aid on this test or assignment.”

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 (https://docs.google.com/document/d/1unqxqiBbPW0STiiLg83NTR_vXKb5USHITgCmw1_aSM4/edit) and Mitchell Wand (https://course.ccs.neu.edu/cs5010sp16/Slides/Lesson%200.3%20Academic%20Honesty.pptx.pdf)

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 (https://www.acm.org/code-of-ethics).

What are the Consequences of Academic Misconduct?

All cases of academic misconduct will result in the following recommended sanctions when being referred to the Office of Student Conduct:

- **the default grade sanction for academic misconduct is 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. (https://policies.ncsu.edu/regulation/reg-02-20-16/)**
- **Students will also be required to complete an Academic Integrity Module provided by the NCSU Office of Student Conduct**
- **Students will also be required to complete Wellness Coaching (https://go.ncsu.edu/wellnesscoaching)**

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

All cases of academic misconduct will be reported to the Office of Student Conduct (https://studentconduct.dasa.ncsu.edu/wp-content/uploads/sites/39/2017/05/RAIV-Updated.pdf).

A first offense will place the student on **Academic Probation**. 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.

Academic misconduct that involves aiding & abetting (e.g., sharing course materials, assignment solutions, and/or assignment deliverables that were produced while enrolled as a student in CSC316; providing unauthorized assistance to currently-enrolled students as they complete course assignments; etc.) will result in a grade sanction of no credit for the course, even if the behavior of academic misconduct occurs when the student is not currently enrolled in the course.

If evidence of academic misconduct is discovered for a course activity, then the teaching team also reserves the right to review your prior (and future) assignment submissions in CSC316 for additional evidence of academic misconduct that will also be submitted to the NC State Office of Student Conduct.

CSC316 Policies

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* team member(s),
- for Project Part 3s, you may receive help from your *assigned* team member(s),
- for Workshop Written Activities, you may receive help from your *selected* study group member(s), and
- for exercises, you may work with any currently enrolled CSC316 students **who are completing the exercises at the same time as you**.

The only **external resources** that you MAY also reference **and CITE**:

- 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).
- With the exception of study groups for written workshop activities, **you are not allowed** to discuss course assignment details on any electronic communication platform other than the course message board and during office hours with a member of the CSC316 teaching staff.
- 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.

Policies for Protecting Yourself

- Do not leave papers lying around your workstation.
- Do not discuss course assignment details outside of the course message board (e.g., do not discuss course assignments on Discord or GroupMe)
- Cover your exam responses to prevent others from copying your responses.

- Do not dispose of important papers in the workshop recycling bins and trash cans until after the assignment is graded.
- Do not give out your password.
- Do not use prohibited resources like Chegg, CourseHero, etc.
- Do not upload course materials to online sites like Chegg, CourseHero, GitHub, etc.
- 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 or send 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 and 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.
- Do not post any assignment artifacts (including assignment descriptions) or solutions to a publicly accessible website, public code repository (e.g. GitHub), or assignment repository (e.g., Chegg, CourseHero, etc.) during **or after** the semester.
- **Ask the instructor for clarification of any questions or concerns about academic integrity policies before submitting an assignment.**

Examples

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 (Chegg, CourseHero, etc.), bulletin boards, message boards (including services like Slack, Discord, etc.) your work for others to obtain. Do NOT use sites that allow you to "anonymously" post materials. Those sites are searchable, and others may find your solutions or code (like the teaching staff).
- It is **cheating** to ask or pay another person or persons to complete an assignment for you (this includes websites like Chegg and StackOverflow).
- It is **cheating** and **aiding & abetting** for you to share your screen or assignment information with another student while completing a non-collaborative electronic assignment; or for you to share

information about a non-collaborative electronic assignment with another person.

- 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, to write test cases that do not check the logic of the source code being tested, 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 and protect 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 & plagiarism** to fail to list names of all study group members on written workshop activity submissions.
- It is **cheating** to use AI tools or services, like ChatGPT, to generate partial or complete solutions to assessments in the class. AI tools and services should not be used as a starting point that the student will then edit. The use of these services are prohibited for assessed work in this 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).
- Help from TAs or instructor (with citations in the comments).
- Using code from the textbook or textbook website (with citations in the comments).

Example Citations

➤JAVA

```java
/* Citing Help from another Person: (In method or class level comments)
 * I received help from Dr. King on DATE during his office hours. We discussed X.
 */

/* Citing Help from other Assignments
 * The code for this method is based on Exercise Y that I completed with Z on DATE.
 */

/* Citing Help from the Textbooks
 * The code for this method is based on the mergesort algorithm on page X in the course te
 * "Data Structures and Algorithms" by Goodrich, Tamassia, Goldwasser.
 */
```

Additional Concerns

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 (workshop questions, some workshop code, project requirements, project design, etc.) are the intellectual property of the instructors and the university. Much (but not all) of the provided workshop code is copyrighted by the textbook authors under the terms of the GNU General Public License as published by the Free Software Foundation.

You may not post any assignment artifacts (including assignment descriptions) or solutions to a publicly accessible website, public code repository, or assignment repository (e.g., Chegg, CourseHero, etc.) during **or after** the semester. Teaching staff will request removal of unauthorized materials, and you may be referred to the Office of Student Conduct for academic misconduct (aiding & abetting).

Some companies like to review student code artifacts as part of a hiring process. You may use CSC 316 materials for this code portfolio using the following guidelines: 1) the code must be posted in a private repository or online resource and only shared with the hiring manager or reviewer and 2) you must add a README or additional documentation clarifying the parts of the code you implemented and differentiating what was provided for you (GUI, design, tests, etc.). After the review is complete (about 2 weeks), remove permissions from the reviewer.

Syllabus Modification Statement

The CSC316 course syllabus represents a flexible agreement. It outlines the topics we will cover and the order we will cover them in. Dates for assignments represent the earliest possible time they would be due. Minor changes in the syllabus can occur if we need to slow down or speed up the pace of instruction, or due to weather, illness, etc. Course policies may change, but any changes will be communicated with students electronically via email or message board post.

Tentative Course Schedule

NOTE that the course schecule may change throughout the semester due to weather, illness, etc.

⌄ Click to Expand CSC316-001 (On-Campus) Tentative Course Schedule

⌄ Click to Expand CSC316-002 (On-Campus) Tentative Course Schedule

⌄ Click to Expand CSC316-601 (Distance Education) Tentative Course Schedule

(/engr-csc316-staff/DE/privacy)