CS135 - Computer Science I Syllabus
Spring 2026

```
REVISION HISTORY:
2026-01-16: Original
```

Table of Contents

Course Information

Section(s)	**1003, 1005** (Web-Online, asynchronous)
Semester	Spring 2026
Duration	16 Weeks, *Jan 20, 2026 - May 17, 2026*
Weekly Contact Hours	3 Hour Lecture Every Week
	1 Hour Lab Every Week

All students are asked to attend the live lecture sessions (Saturdays 9:30 AM to 12:20 PM Pacific Time) via BigBlueButton, schedule permitting. These sessions are recorded for those who cannot attend live.

Instructor Information

Instructor	Kevin Mess
Office Location	Henderson B233
Office Hours	Virtually, using Discord and Microsoft Teams Mondays and Wednesdays 9:30 AM to 12:00 PM Pacific Time and by appointment
Voicemail	702-651-3049
CSN Email	kevin dot mess at csn dot edu

Discord is the primary channel for course questions and announcements. Email is for private matters only (grades, accommodations, and personal issues).

Important Note:

If you have any concerns about this course and/or me, please contact me first. If I cannot resolve your issue, please contact the CIT Department Office at 702-651-5976. You will be directed to the appropriate Program Director or the Department Chair. You will remain anonymous, if possible, and all communications will be strictly confidential. Please DO NOT wait until the last minute to make your concerns known to me and/or to the CIT Department.

Though I typically respond within minutes or hours, I guarantee I will reply to emails sent to my CSN email address within 72 hours, including nights and weekends. If you do not receive a reply within 72 hours, please re-send your email from another account or leave a voicemail message.

> You must use your CSN email address for official correspondence, such as requesting information about a grade.

Course Description

This course is intended for students in computer science or engineering majors. It covers: a) Program development in a complex operating environment; b) Problem-solving methods and algorithm development in a high-level programming language; c) Program design, coding, debugging, and documentation using techniques of a good programming style.

Prerequisite(s): A grade of C or better in either MATH 127 or MATH 128; or MATH 181 or SAT score of 630 or higher or ACT Math score of 28 or higher
Corequisite: None.
Graded: Letter Grade

Learning Outcomes

1. Develop algorithmic solutions to problems and translate their algorithms into C++ programs that meet a set of specifications.
2. Compile and execute their programs in the Linux operating environment and use appropriate testing and debugging strategies.
3. Use appropriate control structures (sequence, selection, and iteration) in their programs.
4. Develop modularized programs using functions and passing parameters.
5. Use strings and file streams.
6. Use one-dimensional arrays and records.
7. Use good programming style and adequately document programs.

Course Materials

Required Texts

- Malik, D.S. *C++ Programming: From Problem Analysis to Program Design, 8th Edition*. Cengage, 2017. ISBN-13: 978-1-33710-208-7. This book has a green cover. Purchase new or used, physical or electronic. We do not use the publisher's online resources (if any). For possible savings, check out https://www.cengage.com/unlimited. (Note: this is the same textbook used in CS 202.)

- Shotts, William E. Jr. *The Linux Command Line: A Complete Introduction, 7th Internet Edition*. 2026. Download the current edition (free): http://linuxcommand.org/tlcl.php.

Software Requirements

You will receive login credentials to access the CIT Department's Linux server that has all the tools needed for this course. You will submit assignments through this server. Look for an email at your CSN email address with login information. The server will use your CSN email address for system messages pertinent to your account.

CSN Student Email

All students enrolled at CSN have a CSN Student Email account. All information from the college will be sent to your CSN-issued student email address (enrollment information, financial aid information, cashier information, college events, etc.). It is extremely important that you check your student email daily. You can access your student email through GoCSN (https://go.csn.edu).

> Once you validate your student email address, you will have access to Microsoft Office 365 for up to five devices and 1TB of OneDrive storage. Visit the CSN Information website for additional information.

Internet Access

Internet access is your responsibility. This class can be accessed from any computer with Internet access anywhere in the world. Therefore, excuses such as "My computer is not working" or "My provider was down" are not acceptable. If you find yourself in a real jam, you may drive to any CSN site and use one of the computers in the CSN open computer labs. Of course, if there is a computer problem originating from CSN, then I will take corrective action. But in all other instances, it is your responsibility to ensure your own Internet access.

We support using **X2Go Client** to access the remote server. This program is free to download and use for Windows, macOS, and Linux.

Host operating system	Required software
Windows	http://code.x2go.org/releases/X2GoClient_latest_mswin32-setup.exe
macOS	http://code.x2go.org/releases/X2GoClient_latest_macosx_10_11.dmg
Linux	Available from your package manager.

CSN also provides computer labs for you to complete your work. The CSN computer labs each have specific pods of computers that have the X2Go Client software installed. Ask a lab monitor to direct you to the correct pod. You may also use any of the Apple iMac computers.

> While you may use any operating system, compiler, or integrated development environment of your choosing, *your programs must compile, build, and execute correctly on the department's Linux server*.

This class makes use of Canvas, CSN's Learning Management System. You can access Canvas using most modern Web browsers by going to https://csn.instructure.com. Some Canvas assessments may require the Respondus LockDown Browser. This browser is unique to each institution. The link to download Respondus for CSN is https://download.respondus.com/lockdown/download.php?ID=367436054.

NOTE: The midterm and final exams are proctored. You must take the exams:

- online using Respondus Monitor (requires webcam) at your convenience, *or*
- in person at a CSN Testing Center with an appointment, *or*
- coordinate with me for alternatives at least three weeks in advance of the scheduled exam.

Meeting Schedule

- All students **may join Saturday morning lectures virtually** if your schedule permits. You'll find BigBlueButton meeting invitations in Canvas each week.
- These sessions are recorded for those who cannot attend live.
- **Attendance is recorded using available logs when you attend live or watch the full recording.**

Canvas Course Site

- The Canvas site contains all lecture links, recordings, and course materials.

Evaluation Methods

Your performance in this course will be assessed through a combination of programming assignments, quizzes, and exams. Each of these components plays a vital role in determining your understanding and application of the course material.

- **Programming Assignments:** These assignments are graded by automated test harnesses on the department Linux server. Submissions that do not compile or do not run in the automated environment receive a score of 0 for that assignment. All other points are determined by the test harness and the rubric shown later in this syllabus.
    - Weight: 25% of your final grade.
- **Lab Assignments:** Labs are lightweight assignments emphasizing relevant topics. They are graded Satisfactory/Unsatisfactory.
    - Weight: 10% of your final grade.
- **Quizzes:** Regular chapter quizzes will assess your ongoing understanding of the course material.
    - Weight: 20% of your final grade.
- **Midterm Exam:** This exam evaluates your knowledge of the first half of the course.
    - Weight: 20% of your final grade.
- **Final Exam:** The final exam encompasses the entire course content, with an emphasis on post-midterm material.
    - Weight: 25% of your final grade.

How programming assignments are graded

Programming assignments are graded automatically on the department Linux server.

- **Correctness (scored):** A test harness ("judge") runs your program against a suite of tests. Your score is based on how many tests you pass and any required-output checks.
- **Readability and documentation (reviewed with tools):** Static analysis tools (and related checks) may be run to detect readability, style, and documentation issues. These reports are used to flag problems and may be used for deductions, consistent with the course style guidelines and rubric.
- **Hard failures:** Submissions that do not compile or do not run in the automated environment receive a score of **0**.

  Submissions may also be temporarily assigned a sentinel score pending review. See *Sentinel Score Review Policy (Code Review Required)*.

Midterm Mastery Policy

If the final exam score is higher than the midterm exam score, the course grade calculation will use the average of the two scores for the midterm's contribution to the formula. In other words, the midterm score used in the weighted average may increase, but only by averaging it with the final score.

Otherwise, the midterm exam score is used as originally earned.

This rule affects only the course grade formula. Official exam records remain unchanged. Both exams must be completed for the rule to apply.

This policy is designed to reward demonstrated improvement.

How Grades are Determined

Your final grade will be calculated based on your cumulative score from each evaluated component. The grading scale is as follows:

Letter Grade	Overall Score (%)
A	[94,100]
A-	[90,94)
B+	[87,90)
B	[84,87)
B-	[80,84)
C+	[77,80)
C	[74,77)
C-	[70,74)
D+	[67,70)
D	[64,67)
D-	[61,64)
F	[0,61)

Note: To transfer this course or use it as a prerequisite for subsequent courses, a minimum grade of C or higher is required.

Note: To pass the course with a grade of C or better, **you must sit for both exams**. Attendance and participation, while not quantified in your grade, are crucial for fully grasping the course content and can positively influence your final grade in borderline cases.

Note: Canvas letter grades are approximations. I may adjust final letter grades **upward** in borderline cases based on documented participation (attendance records and Discord activity) and sustained effort.

Withdrawal

You may withdraw from the course without academic penalty if you do so by the date published in the academic calendar. Enrollment in the course will appear on your college record, and you will receive a “W” for the class. You should not stop attending the class without officially withdrawing in person or online. Failure to properly withdraw from the class may result in the assignment of an “F” grade to your permanent record.

Withdrawal Policy

Instructors do not have the option of withdrawing students, except as mandated during the first week of classes for non-participation. For official withdrawal dates and other important college dates, look at the online schedule at the CSN Calendar and locate the calendar for the current semester.

You are strongly encouraged to discuss your decisions with an academic counselor, academic adviser, or success coach AND Student Financial Services, because these decisions may affect your financial aid and Satisfactory Academic Progress. Students receiving financial aid may find their awards reduced.

If you wish to receive a W in lieu of a grade, you MUST withdraw yourself officially from the class. Once you have withdrawn (dropped), you must discontinue attending class. Alternatively, you may wish to change from Credit to Audit and continue to attend the class.

Change to Audit

Locate the Auditing Classes Form at https://www.csn.edu/sites/default/files/documents/registrar-resources/auditing_classes_form_0.pdf. Print the form and fill it out, check the boxes, sign and date. Submit the completed form in person at the Registrar's Office. Or, to avoid visiting the registrar's office in person, you may scan or photograph the signed form and send it, using your CSN student email, along with a clear color copy of your state ID to MyCSN.Updates@csn.edu.

Course Agenda

Unless stated otherwise, all work is due by 9:30 am on the date specified. All due dates fall on Saturdays.

Date	Lecture Topic	Due / Event
Jan 24, 2026	Introduction & SyllabusLinux Survival (pa00)	quiz00 due
Jan 31, 2026	Malik Ch. 1 - Overview	pa00 due
Feb 7, 2026	Malik Ch. 2 - Basic Elements of C++	pa01 due quiz01 due
Feb 14, 2026	Malik Ch. 3 - Input/Output	pa02 due quiz02 due
Feb 21, 2026	Malik Ch. 4 - Selection Structures	pa03 due quiz03 due
Feb 28, 2026	Malik Ch. 5 - Repetition Structures	pa04 due quiz04 due
Mar 7, 2026	Midterm Review	pa05 due quiz05 due
Mar 14, 2026	—	**Midterm Exam**
Mar 21, 2026	**Spring Break**	No class
Mar 28, 2026	Malik Ch. 6 - User-Defined Functions	—
Apr 4, 2026	Malik Ch. 7 - User-defined simple data types, namespaces, and the std::string type	pa06 due quiz06 due
Apr 11, 2026	Malik Ch. 8 - Arrays and Strings	pa07 due quiz07 due
Apr 18, 2026	Malik Ch. 8 (continued)	pa08a due
Apr 25, 2026	Malik Ch. 9 - Records	pa08b due quiz08 due
May 2, 2026	Final Review	pa09 due quiz09 due
May 9, 2026	—	**Final Exam**
May 16, 2026	To Be Announced	—

Important Dates

DATES	Spring 2026 SEMESTER CALENDAR
Jan 19, 2026	Martin Luther King, Jr. Holiday - College Closed
Jan 20, 2026	First day of semester
Jan 26, 2026	Last Day to Register by 11:59 P.M. \ Last Day for 100% Refund Wait list will be canceled
Feb 2, 2026	Last Day for 50% Refund Last Day to Drop a class WITHOUT a grade of W
Feb 16, 2026	President's Day Holiday - College Closed
Mar 1, 2026	Last Day to Apply for Spring 2026 Graduation
Mar 8, 2026	Note: Daylight Saving Time Begins
Mar 16-22, 2026	Spring Break - College Closed
Apr 3, 2026	Last Day to Drop a class WITH a grade of W Last Day to Change from Credit to Audit
Apr 3, 2026	Return to Title IV Milestone (100% of Financial Aid earned for full term classes)
May 17, 2026	Last day of semester
May 18, 2026	Commencement
May 20, 2026	Grades Due from Instructors

Policies

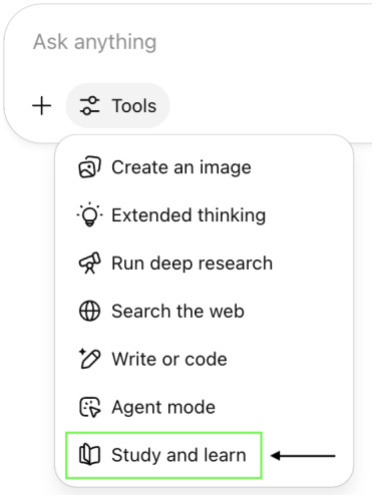
Academic Integrity

You are expected to complete your own work in this class. Cheating on exams or lab exercises is not fair to students who are studying honestly. Cheating is also subject to penalties. For a first offense, you will receive a zero for the assignment and a full letter grade reduction in your overall course grade. A second offense will result in an automatic F for the course. These penalties align with the CSN Student Academic Integrity Policy, which outlines the full list of possible consequences. Please familiarize yourself with this policy: https://www.csn.edu/_csnmedia/documents/policies-and-procedures/2017_academic-integrity-policy_2_0.pdf.

> Some submissions are selected or flagged for code review and may receive a sentinel score pending that review. See *Sentinel Score Review Policy (Code Review Required)*.

Artificial Intelligence

This course permits **limited AI use for learning only**. **ChatGPT in "Study and Learn" mode is the only approved tool** unless the instructor grants specific written permission.



How to select "Study and Learn" mode from the Tools menu

Allowed Assistance (Study and Learn mode only)

- Explain course concepts using questions and stepwise reasoning.
- Help you interpret compiler errors and runtime errors.
- Help you analyze logic you wrote (trace, invariants, edge cases).
- Suggest tests you should write (in English), including what each test should validate.

Disallowed Assistance

- Generating C++ source code for submission (functions, classes, entire files).
- Editing or repairing your code by providing replacement code.
- Producing assignment solutions, even if rewritten.

Rule of thumb: AI may help you think and diagnose. It may not write the code you submit.

Required Disclosure

If you used ChatGPT for any assignment-related work (concept review, debugging, tracing, or test planning), **each affected C++ source file must contain the following lines within the required document header comment block**:

```
/**
 *  ... required document header
 *  @note: ChatGPT Study and Learn session(s)
 *  https://chatgpt.com/share/your-unique-url
 */
```

Session Prompt Requirement

Every documented ChatGPT session must begin with this prompt as the **first message**, after customizing only the bracketed text:

```
I am using ChatGPT in Study and Learn mode
to help me understand [topic]. I am a student
in CS2, working on [brief goal] using C++14.
Please guide me step-by-step using questions
and explanations without providing any direct answers.
```

Students must replace:

- **[topic]** with the specific course concept being studied (e.g., *pointer updates in pop_front*, *linked list traversal invariants*, or *binary search correctness*).
- **[brief goal]** with a short, factual description of their attempt (e.g., *trace algorithm behavior*, *analyze complexity tradeoffs*, or *validate loop invariants*).

All other wording in the prompt must remain **unchanged**.

Valid Session Requirements

Submitted session links must also:

1. Contain **no requests for C++ code or direct solutions**. (Note: You are not responsible for the AI offering code when not requested. The rule is to not request it.)
2. Demonstrate **active learning, reasoning, attempts, and reflection**

Undocumented or non-compliant AI use is treated as academic dishonesty and may result in penalties as described in the Academic Integrity policy.

> Some submissions are selected or flagged for code review and may receive a sentinel score pending that review. See *Sentinel Score Review Policy (Code Review Required)*.

Instructor Consultation Rule

Questions about AI approval must be asked **before using AI**, not after completing the work.

> This policy is enforced in all CS135 sections. Thank you for supporting academic integrity in CS135.

Reference Links

- **Valid single-session link Example:** https://chatgpt.com/share/695b6e78-3a38-8012-9d98-3949ae879fa9

  Check: Shared links must begin with "https://chatgpt.com/share/..." to be valid.
- **ChatGPT Study and Learn FAQ:** https://help.openai.com/en/articles/11780217-chatgpt-study-mode-faq

Attendance Policy

Attendance and participation are expected.

- **Online/hybrid environment:** You must log into Canvas and actively participate during the first week to avoid being administratively dropped, as stated in the Important Notification below. Continued weekly participation is also required.
- Discord is the required communication channel for this course (see Software Lab).

> **Important Notification:**
>
> At the end of the first week of classes, all teaching faculty **MUST** verify **ACTIVE** student participation. **Any student who does not participate during this time will be administratively dropped from this course.**

College assumes maturity, seriousness of purpose, and self-discipline for meeting the responsibilities associated with each course. If you will need the instructor to sign documents testifying about your attendance, YOU must come to the instructor after each class you attend to let them know you were there. Class participation is a strong aspect of this course, and your participation is always encouraged.

While not a specific component of your overall score, I use attendance and participation statistics when computing final letter grades. You are responsible for all topics presented in class or posted/discussed in Canvas.

College Library Services

CSN Libraries provides support for students completing assignments that require research and the use of information. Librarians are available to students for one-on-one assistance locating and citing quality information either online (https://library.csn.edu/ask) or at one of our campus libraries. Find more information on our website (https://library.csn.edu).

Counseling & Retention Services

Counselors assist students who are on academic warning/probation/suspension and financial aid warning/suspension. Counselors prepare academic suspension and financial aid appeals. Counselors help students who are having academic challenges and also those who have been referred by their instructors through the MyCoyotePLAN Early Alert program. Counselors connect students to college and community resources and help them evaluate their options to make informed decisions. For more information about Counseling & Retention Services, please visit: https://www.csn.edu/counselingdepartment.

Counseling and Psychological Services (CAPS)

The Counseling and Psychological Services (CAPS) offers short-term, problem-focused counseling to CSN students who may feel overwhelmed by the responsibilities of college, work, family, and relationships. Clinicians are available to help students cope with stresses and personal issues that may interfere with their ability to perform in school. The service is provided confidentially and free to currently enrolled students. To schedule an appointment, please call CAPS at West Charleston 702-651-5518, or at North Las Vegas 702-651-4099, or at Henderson 702-651-3099.

CSN Americans with Disabilities Act (ADA) Statement and current Disability Resource Center (DRC) Contact Information

CSN is committed to equal opportunity and access in education for all students, including those with disabilities. The CSN Disability Resource Center provides support to students with documented medical, mental health, or learning disabilities and students who need support due to pregnancy. To register with the DRC, students should complete the new student application in the AIM DRC tile in GoCSN. Students are encouraged to register with the DRC as early as possible in or prior to the start of the academic term. The granting of any accommodation will not be retroactive and cannot jeopardize the academic standards or integrity of the course. Please Note - Instructors are not allowed to provide classroom accommodations to a student until appropriate verification from the DRC has been provided. If you have questions, please contact the DRC directly (702-651-5644) or visit https://www.csn.edu/disability-resource-center for additional information.

Early Alert Referral Program

The Early Alert Referral Program (MyCoyotePLAN) is a referral program to connect students with college resources when assistance is needed to achieve success. Referrals may be initiated by faculty and staff as well by students through MyCoyotePLAN. After a referral is submitted, students will receive an email notification and will be contacted by the department to which they were referred to offer assistance.

Food and Housing Insecurity Support

Any student who has difficulty affording groceries or accessing sufficient food to eat every day, or who lacks a safe and stable place to live, and believes this may affect their performance in the course, is urged to contact Counseling & Psychological Services (CAPS), for a list of resources and support.

Immigrant/ UndocuAlly Support

CSN is committed to the inclusion and representation of a diverse campus population. As such, CSN acknowledges that our campus population consists of individuals whose status is undocumented, Deferred Action for Childhood Arrivals (DACA) recipients, as well as immigrant students. If your immigration status presents obstacles to engaging in specific activities or fulfilling specific course criteria, please reach out to the Immigrant and UndocuAlly Committee at: https://news.csn.edu/csn-applauds-daca-decision/

Late Assignments / Makeup Work

I do not accept late assignments, nor do I provide makeup assignments. You are responsible for your own Internet connection when working remotely. Work ahead if you know you are going to have conflicts or time constraints. Please schedule yourself accordingly. Students administratively withdrawn for lack of participation will not be eligible for makeup work unless officially re-enrolled by the college.

Late Instructor Policy

When meeting synchronously (classroom or remotely), students must wait for 15 minutes if the instructor is late. The class is excused if the instructor is delayed beyond 15 minutes. Check Canvas for updates.

Programming Guidelines

General Programming Assignment Requirements

1. For each program, you must work individually unless instructed otherwise. You may discuss the problem with classmates, but at no time should you share/discuss code in any form. *You may not show another student your code, share your file with another student, look at another student's code, or tell another student what to type.* Evidence of academic dishonesty will result in a score of zero (see Academic

Integrity section of the syllabus). This applies to *all* students involved. If you're unsure about something, please ask in advance.

2. Your program must adhere to the problem statement requirements and coding standards below. Violations will lead to deductions.

    1. A header comment *must* be included at the top of *each* submitted file. Submissions without this header comment will receive a grade of zero. The header comment must consist of the following information, including documentation annotations (annotations begin with the character '@'):

```
/**
 *  @file FILE_NAME
 *  @author YOUR_NAME
 *  @date THE_DATE
 *  @note I pledge my word of honor that I have complied with
 *  the CSN Academic Integrity Policy while completing this
 *  assignment.
 *  @brief A brief description of the program (no more
 *  than one or two paragraphs)
 *  @note Time taken to develop, write, test, and debug the
 *  solution.
 *  @note: ChatGPT Study and Learn sessions
 *  https://chatgpt.com/share/your-URL1
 *  https://chatgpt.com/share/your-URL2
 */
```

    Failure to disclose assistance, regardless of source, may be interpreted as academic dishonesty.

    2. Basic blocks:

        1. Blocks will *always* use braces using methods demonstrated in class. Brace placement is at your discretion, but must be consistent.
        2. Statements in the block should be indented consistent with logical nesting. Use 4 *spaces* per indent level. Do not use *tabs*.

    3. Variables:

        1. Use descriptive names for variables using naming standards discussed in class.
        2. Reduce the scope of variables so that they are only visible in the scope where they're used. Global *variables* are *never* permitted; global *constants* are permitted, when appropriate.
        3. Use one declaration for each variable (i.e., do not use the comma operator to declare multiple variables at one time).

4. Variable declarations must appear at the beginning of the block of code in which they're used (i.e., do not intermix declarations with code).
5. Document the purpose of *every* identifier you create (e.g., constants, variables, functions, etc.).

4. Statements:

    1. No more than one statement may be written on a single line.
    2. The following may not be used: 'exit', 'continue', 'goto', and 'break' *not* in a switch structure.
    3. Handle errors graciously wherever possible.
    4. Lines of code should be no more than 80 characters long.
    5. Diagnostic/debug print statements should be disabled/deleted in the final submission (i.e., always submit a "shipping" version of your program). In advanced courses, use conditional compilation to enable/disable debugging statements.

5. Functions:

    1. Use descriptive names for functions using naming standards discussed in class.

    2. All functions must be documented with the following information:

        1. Purpose - A statement or a set of statements that describes the purpose of the function.

        2. Parameter(s) - The purpose of each parameter should be described.

        3. Return value - For value-returning functions only, describe what the function returns.

        4. When using a single source code file, function documentation should be placed in comments directly above the function definition, not the prototype.

        5. When implementing a large project with multiple source code files, function documentation should be placed in comments directly above the function prototypes *in the interface file*, not in the implementation file.

    - Example function documentation header:

```
/**
 *  @brief The function foo.
 *
```

```
 *  Description of what the function does. This part may
 *  refer to the parameters of the function, like
 *  @p param1 or @p param2.
 *
 *  @param param1 Description of the first
 *  parameter of the function.
 *  @param [in,out] param2 The second parameter
 *  which follows param1.
 *  @return Describe what the function returns.
 *
 *  @see http://website/
 *  @note Something to note.
 *  @warning Warning.
 */
 int foo(int param1, int& param2);
```

3. Function bodies should not be of extended length when easily separated into multiple functions (i.e., functions should do one thing and nothing more).

4. Non-recursive functions should contain *exactly* one return statement. Functions should have one entry point and one exit point (i.e., non-recursive functions should have no more than one `return` statement).

5. Refer to the Programming Grading Rubric for detailed expectations on meeting program specifications and correctness. (See: Programming Grading Rubric section below)

3. Programs must be submitted on time. Late programs will not be accepted.

4. All programs must conform to the C++14 language standard. You may only use features or libraries from C++14 that have already been introduced in the textbook, lectures, or prior sample solutions. Do not use deprecated features from older standards or features from newer versions of the language.

5. Submit the program file(s) electronically using the procedure shown on the problem statement requirements. You may submit your program file(s) as many times as you want before the deadline. *Each submission will replace any earlier submission*, so I can only see and grade your most recent submission. Be sure to submit *all* required files with *each* submission. You cannot submit after the deadline (i.e., the drop box is closed).

Programming Grading Rubric

This document lays out common criteria used to grade programming assignments. Each criterion has several different levels of achievement, with a description of how a submission will attain that level and the number of points assigned for reaching it. Please

email or ask me if you have any questions about this rubric. Please refer to the grading rubric below.

Every criterion will make up an approximate percentage of the grade given to a single programming problem as indicated in the "Approx. % of Grade" column. Points will be assigned for a criterion roughly along the lines of the guidelines of the "Excellent," "Above Average," "Average," "Below Average," and "Not Met" evaluations.

For example, a problem that was "Above Average" in the Program Specifications/Correctness criterion, "Average" in readability, and "Excellent" in all other areas would receive:

```
(0.4 * 0.8) + (0.2 * 0.6) + (0.2 * 1.0) + (0.1 * 1.0) + (0.1 * 1.0) = 84% = B
```

Program Specifications / Correctness

This is the most important criterion. A program must meet its specifications (whether from a textbook problem or as written in the assignment) and function correctly. This means that it behaves as desired, producing the correct output, for a variety of inputs. This criterion includes the need to meet specifications by writing a program in a specified way or using a required language feature, if such a thing is specified in the problem.

If a specification is ambiguous or unclear, you have two choices: You can either make a reasonable assumption about what is required, based on what makes the most sense to you, or you can ask the instructor. If you make an assumption about an ambiguous specification, you should mention that somewhere in a comment so that the reader/grader knows what you were thinking. Points may be taken off for poor assumptions, however.

As special cases, if a submission

- lacks the name of the author, or,
- does not meet the specifications at all (entirely incorrect), or,
- is written in a language other than used in the course, or
- does not compile,

no credit will be received for the other criteria either, i.e., these conditions will result in a score of zero.

		Excellent	Above Average	Average	Below Average	Not Met
Criterion	**Weight**	**100%**	**80%**	**60%**	**40%**	**0%**
Program Specifications / Correctness	40%	No errors. No warnings. Program always works correctly and meets the specification(s).	No warnings. Minor details of the program specification(s) are violated. Program functions incorrectly for some inputs.	One or more warnings. Program functions incorrectly for some inputs.	Significant details of the specification are violated. Program often exhibits incorrect behavior.	Program only functions correctly in very limited cases or not at all. Program does not compile and/or link.
Readability	20%	Code is clean, understandable, and well-organized.	Minor issues with consistent indentation, use of whitespace, variable naming, or general organization.	Multiple minor issues with consistent indentation, whitespace, variable naming, or general organization.	At least one major issue with indentation, whitespace, variable names, or organization.	Major problems with three or more of the readability subcategories.
Documentation	20%	No errors. All variables documented. All functions correctly documented. Code is well-commented. No spelling errors.	One or two places that could benefit from comments are missing them, or the code is overly commented. One or two variables not documented properly. One or two spelling errors.	Multiple places that could benefit from comments are missing them. More than two variables not documented properly. More than two spelling errors.	File header missing, complicated lines or sections of code not documented or lacking meaningful comments. More than four variables not documented. More than four spelling errors.	No file header present. No comments present. More than eight spelling errors.
Code Efficiency	10%	No errors. Code uses the best approach in every case.	Code uses poorly-chosen approach in one place.	Code uses poorly-chosen approaches in two places.	Code uses poorly-chosen approaches in three places.	Many things in the code could have been accomplished in an easier, faster, or otherwise better fashion.
Miscellaneous	10%	No errors.	One minor detail of the assignment specification is violated, such as incorrect filename.	More than one minor detail of the assignment specification is violated, such as incorrect filename.	Input/output varies significantly from that specified.	Significant details of the specification are violated, such as extra instructions ignored or entirely misunderstood.

Programming Grading Rubric

Readability

Code needs to be readable to both you and a knowledgeable third party. This involves:

- Using indentation consistently (e.g., every function's body is indented to the same level).
- Adding whitespace (blank lines, spaces) where appropriate to help separate distinct parts of the code (e.g., space after commas in lists, blank lines between functions or between blocks of related lines within functions, etc.).

- Giving variables meaningful names. Variables named ‘a’, ‘b’, and ‘c’ or ‘foo’, ‘bar’, and ‘baz’ give the reader no information whatsoever about their purpose or what information they may hold. Names like ‘principal’, ‘maximum’, and ‘counter’ are much more useful. Loop variables are a common exception to this idea, and loop variables named ‘i’, ‘j’, etc. are okay.
- The code should be well-organized. Code should be organized into functions, and functions should have meaningful names.

Documentation

All interface and implementation files must begin with the required header comment shown above. Submissions without names or your academic pledge will receive a score of zero.

All code should also be well-commented. This requires striking a balance between commenting everything, which adds a great deal of unneeded noise to the code, and commenting nothing, in which case the reader of the code (or you, when you come back to it later) has no assistance in understanding the more complex or less obvious sections of code. In general, document the purpose of every programmer-defined identifier, and aim to put a comment on any line of code that you might not understand yourself if you came back to it in a month without having thought about it in the interim. Like code organization, appropriate commenting is also something we will be learning as we write code throughout the semester, so while corrections may be made, points will only be taken off for things that have been emphasized in class already.

Code Efficiency

There are often many ways to write a program that meets a specification, and several of them are often poor choices. They may be poor choices because they take many more lines of code (and thus your effort and time) than needed, or they may take much more of the computer’s time to execute than needed. For example, a certain section of code can be executed ten times by copying and pasting it ten times in a row or by putting it in a simple for loop. The latter is far superior and greatly preferred, not only because it makes it faster to both write the code and read it later, but because it makes it easier for you to change and maintain.

Miscellaneous

Assignments will usually contain specifications and/or requirements outside of the programming problems themselves. For example, the way you name your files to submit them to the course website will be specified in the assignment. Other instructions may be included as well, so please read the assignments carefully.

Pregnant Students

CSN prohibits discrimination based on sex in education programs and activities. This prohibition on discrimination extends to pregnancy and related conditions–including

childbirth, lactation, false pregnancy, termination of pregnancy, and recovery therefrom– as well as to parental and family status. If you are pregnant or have a pregnancy-related condition, and you are in need of accommodation because of your pregnancy or pregnancy-related condition, you must contact Dr. Armen Asherian, Title IX Coordinator, at titleixcoordinator@csn.edu or 702-651-7481, or the Disability Resource Center at 702-651-5644 for West Charleston, 702-651-3795 for Henderson, and 702-651-4045 for North Las Vegas to explore reasonable accommodation.

Public Health Directives

Students must follow all active CSN public health directives while enrolled in this class, such as properly worn face coverings when required in classrooms as well as inside campus buildings. CSN public health directives are found at https://www.csn.edu/wellness. Students who do not comply with these directives will be asked to leave the classroom. Refusal to follow the guidelines may result in further disciplinary action according to the CSN Student Conduct Code https://www.csn.edu/sites/default/files/documents/student_conduct_code_policy_1.pdf, including being dropped from the course.

Sex-Based Harassment and Discrimination

CSN is committed to creating a safe and open learning environment for all students. In accordance with Title IX of the Education Amendments of 1972, CSN prohibits unlawful sex-based harassment against any participant in its education programs or activities. Sexual-based harassment includes quid pro quo (this for that) harassment, a hostile environment, and criminal sexual violence (including sexual assault, dating/domestic violence, and stalking.) This prohibition applies to CSN students, employees, and visitors. Incidents of sex-based harassment or discrimination should be reported to CSN's Title IX Coordinator, Dr. Armen Asherian, at titleixcoordinator@csn.edu, or 702-651-7481 or University Police Department at 702-895-3669 to report a crime.

Sentinel Score Review Policy (Code Review Required)

To protect academic integrity and ensure accurate grading, some programming submissions may be **flagged for review**. When a submission is flagged, it will be recorded in the gradebook with a **sentinel score of -100**. A submission may be flagged for reasons including, but not limited to: similarity detection results (e.g., MOSS), indicators of prohibited AI assistance, unusual submission patterns, or random audit selection.

A sentinel score is **not a final grade**. It indicates that the submission is **on hold** pending a brief, one-on-one code review.

If a submission receives a sentinel score, the student must **contact the instructor promptly** and **schedule a code review meeting**. During the meeting, the student must be prepared to explain the program's design, key implementation decisions, and how the solution was developed. If the code review is completed satisfactorily, the sentinel score

will be removed and the submission will be assigned a **regular score** based on the course grading procedures.

If the student does not schedule and complete the code review **within seven calendar days of the sentinel score being posted (or by the assignment's last day of availability, whichever comes first)**, the **-100** sentinel score will remain and will be included in the course grade calculation. The instructor may grant extensions in writing for documented circumstances.

This process is part of the course's normal grading and academic integrity enforcement.

Software Lab

Use of Discord is required for all CS135 students. You must join the official course Discord server to receive announcements, access the software lab, participate in course discussions, and request live support during office hours. You are expected to check Discord at least once per week, and again before each quiz, exam, or assignment deadline, for announcements and updates.

The software lab will be virtual and will be available on Jan 20, 2026 via Discord. Office hours are posted weekly in Discord.

Invitation URL (also available in Canvas): https://discord.gg/Pn2nhdTf8K

TutorChat

TutorChat is a free, live, online chat service accessible by CSN students at https://csncentersacademicsuccess.libanswers.com/ or in the navigation menu of your Canvas courses. TutorChat is designed to answer your quick questions, help you form successful study habits, and show you how to use Canvas and other resources for student success, such as online tutoring and MyCoyoteSuccess workshops, accessed in GoCSN. Quick questions are defined as assistance with topic formulation, proper MLA or APA citation formatting, and development of outlines including thesis statement, introduction clarity, and appropriate conclusion. TutorChat staff can start audio/video sessions and give you information on how to connect with a Centers for Academic Success tutor, which is an in-person service FREE to you. For more information, you can chat with us or submit a question. TutorChat is open to all enrolled CSN students.

> If you need assistance outside their normal business hours (11-4, Monday through Sunday), please submit a question at TutorChat. They usually respond within 24 hours.

Course Contract

This syllabus is a course contract for the duration of the semester. By continuing with this course, students agree to the terms set forth in this syllabus and will adhere to all rules and policies outlined.

Information contained in this syllabus, other than the grading, late assignments, makeup work, and attendance policies, may be subject to change with advance notice, as deemed appropriate by the instructor.

Changes will be announced in class and posted in Canvas. It is your responsibility to stay informed.

EOF