2026 - 2027 COMPUTER SCIENCE I SYLLABUS

Sean “Mr. Mugg” Muggivan

10th and 11th Grade • Fall Semester

Overview

Computer Science I is about directing machines to build real things. Rather than copying code out of a book, students write durable specifications precise enough for an AI system to build from, then read, test, and judge what comes back. What we build is real: web apps, classroom tools, data visualizations, and sites deployed live at an address anyone can visit.

We also open the machine up: what a CPU, GPU, and memory actually do, how an AI model is trained and served, and what happens to a request between the Chromebook and the hardware answering it, some of which sits in this room. No prior programming experience is assumed.

Resources and Materials

No textbook, and everything we use is free or provided. Two places matter, and they do different jobs: the course site is where content comes from, and Google Classroom is where work goes in.

- School-issued Chromebook, charged daily. The primary tool of this class.
- **Course site, muggsofcompsci.net/cs1:** where I organize and release content each day, and where the course calendar shows the activities we will be working on in class.
- **Google Classroom:** where you submit work to me for grading. I post nothing there beyond rubrics and activity outlines, and those are on the course site as well. If you are looking for content, look at the site.
- **Google Forms:** in-class instruments and surveys.
- Classroom AI tools served from classroom hardware, introduced as released.
- A notebook or folder kept all semester. Paper days happen, planned and unplanned.
- Your own GitHub and Cloudflare accounts, created in class when we reach deployment. They belong to you, not to the school year, and the work in them outlasts this course.

Students are responsible for their device and charger. Lost, damaged, or stolen equipment follows school policy; report it to the front office and to me the same day. Every activity, including every major assignment, has a paper fallback, so a device problem is never a reason to sit out.

Course Description

Computer Science I is a one-semester introductory course in which students specify, direct, verify, and ship computing projects built with AI systems. Its premise: the abilities that once required writing code are the ones that matter most now. Breaking a problem down, saying exactly what should happen, judging whether the result is correct, fixing what isn't. They can be built without authoring code by hand.

Students begin with data and the machine, move into directing and verifying AI output, and finish by deploying their own project to a public address. Expectations are plain: charged device, do the thinking work, submit evidence of it, and be honest about what the machine did and what you did.

Objectives

By the end of the semester, students will be able to:

- Write a specification precise enough that a person or an AI can build from it without follow-up questions.
- Judge AI output for correctness, completeness, and honesty instead of accepting it as finished.
- Explain in plain language how a computer stores and processes information, and how an AI model is trained, served, and queried.
- Collect, code, and interpret real data, and name what the instrument could not ask.
- Use version control and deploy a working project to a public address.
- Describe their own use of AI accurately: what was specified, what was generated, and what was changed.

ACT preparation is built in rather than bolted on: students help build and then use a classroom math item bank as part of our work on data and verification. This course carries no state end-of-course exam; the final is a course assessment.

Course Outline: Main Topics and Subtopics

Fall 2026 semester: three units, August 10 through December 15.

Main Topic & Subtopics	Learning Outcomes & Objectives	Assessment
Unit 1. Foundations: Reading the Data and the Machine (Aug 10 – Sep 11) ● Course launch, routines, non-negotiables ● Baseline instruments: a knowledge benchmark, and a survey of what you believe and what you do ● Survey design: data dictionary, theming open responses, validating results ● How a model is made: pretraining, post-training, tokenization, and inference ● Inside the machine: CPU, RAM, storage, GPU and VRAM ● Local vs. cloud inference: tracing one request end to end ● Writing the class AI policy from our data **REFERENCES** Students should have on hand: ● Course site: muggsofcompsci.net/cs1 ● Class-built Google Forms instruments ● Live classroom hardware demonstration	Students write a data dictionary against their own survey, code the open responses by hand, and author a class AI policy from the result. They explain how a computer computes and how a request reaches a model, distinguishing local from cloud inference. CSTA 2026 CORE claims recorded in the semester plan on file.	Formative: do-now and exit tickets, data dictionary, theming worksheet, chart critique. Summative: baseline cognitive instrument (repeated in December), class AI policy artifact.
Unit 2. Directing and Verifying (Sep 14 – Oct 8) ● Pretraining, post-training, and inference as distinct	Students predict where an AI system is likely to fail, design a check that	Formative: failure-taxonomy entries, quality-control runs,

phases ● Where AI misleads: building a failure taxonomy ● Prompting vs. specifying: what makes an instruction durable ● Building a verification pipeline, on paper first, then for real ● Evaluating sources and claims ● Judging generated output against a quality-control instrument **REFERENCES** ● Course site and classroom verification tools ● Articles and documentation posted to the course site	would catch it, and apply it to real output. They distinguish a prompt from a specification and revise a weak instruction into a durable one. Q1 benchmark falls in this window. CORE claims recorded in the semester plan on file.	verification logs, quizzes. Summative: verification pipeline artifact, specification revision as tracked changes, midterm.
Unit 3. Building and Shipping (Oct 14 – Dec 15) ● Specify → generate → judge, the craft of the course ● The specification portfolio: durable specs, revised ● Version control and a real repository ● Deployment: hosting, DNS, a live address ● Cohort build, then individual capstone ● December post-assessment **REFERENCES** ● Course site and project repositories ● Platform docs for hosting and version control	Students produce a specification portfolio and ship at least one working project to a public URL, and account for what they specified, what the machine generated, and what they changed. Fall LEAP and the Q2 benchmark fall in this window. CORE claims recorded in the semester plan on file.	Formative: build logs, spec drafts, peer spec swaps, code-reading predictions, exit tickets. Summative: specification portfolio reviewed as a GitHub repository, deployed capstone, post-test instrument, final exam.

Grading Scale

Student grades will be based on guidelines in accordance with Inspire Nola's Pupil Progression Plan.

Grade	Score
A	90-100
B	80-89
C	70-79
D	60-69
F	Below 60

Grading Breakdown

- 40% Major — eight tests, one every two weeks: Aug 28, Sep 11, Sep 25, Oct 8, Oct 23, Nov 6, Nov 20, and Dec 11. A test day is half review, half test. Unit projects and deployed builds also land here.
- 40% Minor — two grades every week, one for daily work and one for classwork. See below.
- 20% Exam — the midterm in the first quarter; in the second quarter, the average of the final exam and your specification portfolio. This course carries no state EOC.

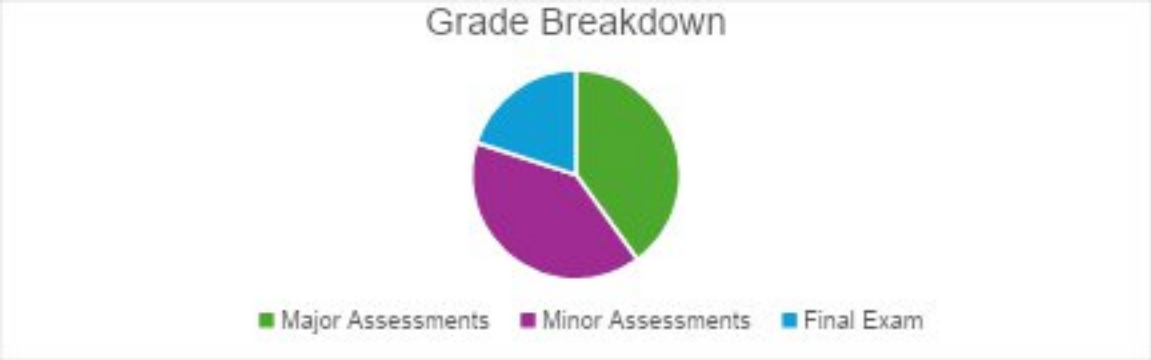
Grades come from evidence of your thinking, not the polish of the output. A good-looking project with no specification, no build log, and no account of what you changed is not finished.

Minor grades. Two of them go in every week. The first is your daily work — the Do Now that opens class and the Exit Ticket that closes it. Each is scored out of 10, and the grade that lands in PowerSchool is your average across that week. The second is classwork: build logs, spec drafts, verification runs, short writing. Those are scored out of 15, and again, what gets recorded is the week's average. A weekly classwork grade is worth 1.5 times a weekly daily-work grade, because classwork is the heavier artifact — what you build and write counts for more than how you open and close class.

How many pieces go into each average depends on the week. A week with a test day has fewer, because test days carry no Do Now or Exit Ticket. That is normal and it costs you nothing — an average of three sits on the same scale as an average of five. There is always at least one piece in each category, so there is always a grade to earn.

Grades post to PowerSchool on a fixed weekly rhythm, each covering the week before: daily work on Tuesday, classwork on Thursday.

- **Late work: the posting day is the deadline.** Anything submitted before the grade posts counts in full, with no email needed — Tuesday for daily work, Thursday for classwork. Classwork goes through the standing form on the course site.
- **After the grade posts,** email me to request the work and I will consider it. After Thanksgiving, that request means a conference with your household.
- **Do Nows and Exit Tickets cannot be made up.** There is nothing to hand back three days later. An excused absence drops the missed piece out of that week's average; an unexcused absence or a suspension averages it in as zero.
- **If missing work is a pattern rather than a single incident,** we talk with your household while there is still semester left to do something about it.
- Specifications and projects may be revised through the end of the unit. The revised grade stands.
- Using AI is the course. Misrepresenting what you did and what the machine did is the academic dishonesty line, handled under the school's academic integrity policy.
- Nothing is lost to a bad tech day; paper versions exist for every major assignment.



Progress Reports:

Progress reports are issued each quarter. Grades are posted and updated regularly, and students are expected to check their own and share them at home. Bringing important information home is the student's responsibility.

Classroom-Household Communication

Students should not wonder whether their computer science teacher is in regular contact with their household. Expect it. Communication with home is a planned part of this class, not a sign something has gone wrong.

I check in with every student's household at three points in the semester:

- At the beginning, to introduce myself and the course and hear what I should know about your student.
- At the midway point, an honest account of where your student stands, with time left to act.
- At the end, to close out the semester and talk about what comes next.

These happen regardless of grades or behavior. If your student is doing well, you will hear it from me directly, not only when there is a problem.

Beyond those three, I reach out whenever something warrants it, good or otherwise. Contact me any time; I would far rather hear from you in August than December. Texting and email are the most reliable ways to reach me:

- **Text:** (504) 407-5523
- **Email:** sean.muggivan@inspire35.org

Parents and guardians are welcome to set up a phone call or an in-person conference, and I am glad to do either. The most reliable way to arrange one is a text or an email.

Behavior Expectations

- Be here and be working. Class opens with a do-now, closes with an exit ticket. Both are graded.
- The Chromebook is the tool of this class. Screens stay on task; off-task browsing or gaming ends device privileges for the period and work moves to paper.
- Talk to each other the way you would want to be talked to about work you are still learning. Everyone here is building in public.
- Be honest about your work. Say what you specified, what the machine produced, and what you changed.
- Take care of the equipment. It is shared, expensive, and some of it is mine.

Soft Skills

The school messages a lot of soft-skills development in preparation for post-secondary life — colleges and universities, trades and tech, employment, and enlistment. The following are especially emphasized in this course:

- **NO SLEEPING IN CLASS.** I will continually try to wake you up. If I cannot keep you awake, I will contact support from outside of the classroom and notify your household.
- **DO NOT DISRUPT THE CLASS.** This includes both causing a disruption and responding to one. When I taught younger students, I gave them a phrase for the second part: ignore the foolishness.

- **WORK.** If you are not interested in computer science at all — congratulations. This is the first job you have where you have to do something you don't want to do in order to get to something you do.
- **NO YELLING.** This applies to students and to the teacher. If I start talking, students stop talking. I am the adult in the room, responsible for the computer science education of about thirty students; you are the student in the room, responsible for your own education. If I start talking while you are talking, it is because I am responsible for keeping an eye on a bigger picture than you are.

Cell Phone Policy

All cell phones are to be placed into the bin by the door upon entering the room. They remain in the bin until exiting the room.

Student failure to place a cell phone inside of the bin will result in notification to the culture team and a communication to the student's household.

This is not personal and not negotiable. The work here asks for sustained attention, and a phone in a pocket takes that away before you notice it is gone.

Restroom Policy

To prevent constant departure from class, students may not leave for the restroom during the first or last 20 minutes of the period. Restroom use is permitted mid-class, one student at a time, with the pass. Go directly to the closest restroom and come straight back. You have five minutes. After five minutes, the culture team is notified that a student is out of class.

Students are responsible for anything they miss while out: directions, demonstrations, deadlines, work. Instruction does not stop and will not be repeated on return.

Students with a documented medical need or a 504 or IHP accommodation are exempt from these restrictions. See me privately in the first week.

Syllabus Acknowledgment

2026 - 2027 Computer Science I • Sean “Mr. Mugg” Muggivan • 10th and 11th Grade

By signing below, the student and the parent or guardian confirm that this syllabus has been read and reviewed together, and agree to abide by it, including the grading, attendance, behavior, cell phone, and restroom policies.

Full syllabus online: https://muggsofcompsci.net/cs1/syllabus.pdf

Adjustments may be made to this syllabus over the course of the semester. Any change will be communicated directly to households.

The adult who signs should be a parent or guardian listed in SchoolRunner. The contact information below is used for the check-ins described in Classroom-Household Communication.

Return this page to Mr. Mugg by the end of the first week.

Student

Student name (printed)

Student signature

Date

Parent or Guardian

Parent or guardian name (printed, as listed in SchoolRunner)

Relationship to student

Parent or guardian signature

Date

Best phone number for calls and texts

Best email address