2.1. **Rubrics.** Communicate expectations by setting standards for excellence, not effort.
   Criteria: Independent dimensions of judgement.
   Levels: A rating scale for each criterion.
   Descriptors: Model standards of excellence for each criterion and level.

2.2. **Source code.** Communicates to a compiler, other programmers, and end users.
   Syntax: The compiler checks if the language grammar accepts the source code.
   Schematics: Other programmers expect source code to follow conventions.
   Semantics: End users care about what the program does and how it works.

2.3. **Software tests.** Specify behavior and establish confidence in implementation.
   Specification: Tests define how a program should (and should not) behave.
   Coverage: Tests exercise program elements, thereby establishing confidence.

2.4. **Writing.** Convey points, clarify thought, avoiding errors and verbosity.
   Point: Writing must convey points with relevant evidence.
   Style: Writing must avoid errors, verbosity.
   Organization: Sentence, paragraph and section structures must clarify thought.

2.5. **Presentations.** Demonstrate knowledge and engage the audience.
   Knowledge: The presenter must convey knowledge so the audience learns.
   Technique: The presenter must engage the audience.

2.6. **Questions.** Probe productively.

2.7. **Grades.** Assess the quality of work using evidence and criteria.
   Fairness: Assessments of quality reflect reality.
   Feedback: Assessments provide actionable feedback, without giving away solutions.
   Reliability: Independent graders arrive at similar quality assessments.

2.8. **Participation.** Engage during class meetings, do not merely show up.
   Attendance: Show up to class meetings, or justify absence.
   Engagement: Engage during class meetings.

2.9. **Lessons.** Teach with objectives and assessments.
   Learning objectives: Made first, manageable, measurable, and important.
   Assessments: Rank student progress appropriate to learning objective.

TABLE 1. Rubric rubric

Criterion	Unsatisfactory	Needs Improvement	Satisfactory	Excellent
Criteria				
Rubric lists ...	Irrelevant criteria or no criteria	Overlapping, partial, conflicting criteria	Distinct, complete, relevant, consistent criteria	Satisfactory criteria enabling rapid grading
Levels				
Rubric ...	Omits levels	Defines perfection only	Sorts imperfection and perfection into levels	Sorts with objective metrics & evidence
Descriptors				
Rubric ...	Omits descriptors	Describes criteria and levels with adjectives	Describes criteria and levels with sentences	Demonstrates using sentences & example(s)

TABLE 2. Source code rubric

Criterion	Unsatisfactory	Needs Improvement	Satisfactory	Excellent
Syntax				
Code . . .	Does not compile	Compiles with warnings	Compiles, no warnings	Uses safe constructs
Schematics				
Code is . . .	An empty shell	Structured poorly	Structured well	Attentive to detail
Indentation . . .		Is sloppy or absent	Highlights structure	Is spaced consistently
Cohesion is . . .		Low	High	High
Coupling is . . .		Tight	Loose	Loose
Structured as . . .		Duplicated or dead code	All code has a purpose	Delegation to libraries
Methods are . . .		Excessively long	Short (fit one screen)	One-liners
Identifiers . . .		Mislead or are useless	Convey meaning	Obey conventions
Comments . . .		Omitted or restate code	Explain why	Input to doc generator
Semantics				
Specification is. . .	Irrelevant, absent	A problem number	A problem statement	Augmented, exceeded
Program runs . . .	Not at all	Partially, incorrectly	To specification	Beyond specifications
Missing . . .	All cases	Edge or corner cases	No specified cases	No case whatsoever
Input prompts. . .		Confuse, are ambiguous	Reveal expectations	Prevent errors
Performance is. . .		Slow	Fast	A live interaction
Program . . .		Delegates work to users	Performs tasks for user	Minimizes user work
Outputs have . . .		No labels or structure	Labels and structure	Good labels, structure

TABLE 3. Testing rubric

Criterion	Unsatisfactory	Needs Improvement	Satisfactory	Excellent
Specification				
Tests assert ...	Nothing	Program behavior	Behavior & exceptions	Behavior, exceptions Nonfunctional qualities
Tests are ...	Non-existent	Positive, specific	Positive, negative & specific	Positive, negative & generic
Coverage				
Tests cover ...	Nothing	Methods (units)	Statements	Conditions & decisions
Tests exercise ...	Nothing	$\geq 60\%$ of above	$\geq 80\%$ of above	$\geq 90\%$ of above

TABLE 4. Writing rubric

Criterion	Unsatisfactory	Needs Improvement	Satisfactory	Excellent
Point				
Topic sentence...	Is irrelevant	Does not give a topic	Summarizes paragraph	Makes a point
Evidence is ...	Non-existent	Equivocation: *very, quite*	Assertions of topic	Topical citations, proofs
Style				
Words ...	Are bogus: *refudiate*	Are misspelled or unfit: *then/than, its/it's*	Are spelled correctly & proper for context	Replace phrases
Sentences are ...	Partial & passive	Ill-formed, long, passive	Grammatical, active	Concise in active voice
Tense is ...	Unfit to context	Mixed inconsistently	Consistent with context	Consistent throughout
Organization				
Sentences have...	No form (notes)	Duplicate words	Unique words	Varied structure
Sequenced in ...	No logical order	Choppy groups	A logical progression	A flowing narrative