

# STUDENT-FACING RUBRIC (Syllabus / Assignment Page)

## AP CSA GitHub Track — How Your Code Is Graded

Each GitHub assignment is graded on **Java correctness, reasoning, and clarity**, aligned with **AP Computer Science A expectations**. GitHub is used only for submission — even though it's an invaluable skill to have and will give you an edge at college and in any job market, **you are not graded on Git usage**.

## Important GitHub Expectations

- You may make **as many commits as you want** before the assignment deadline.
- Only the **state of your repository at the deadline** is graded.
- There is **no penalty** for frequent commits.

Use commits to:

- Fix bugs
  - Improve logic
  - Clean up code
  - Improve tests or explanations
- 

## Grading Breakdown (100 points per assignment)

### 1. Completion & Submission (10 pts)

- All required files and classes are present
  - Repository is accessible and complete
- 

### 2. Compilation & Runtime Stability (15 pts)

- Code compiles successfully
- Program runs without crashing

 **Important:**

If your code does **not compile**, your score for the assignment **cannot exceed 70%**, even if parts of your solution are correct!

---

 **3. Functional Correctness (30 pts)**

- Meets all required specifications
- Produces correct output
- Handles edge cases (e.g., empty input, zero values, boundary conditions)

This is the **largest category** — correctness matters most!

---

 **4. Use of AP CSA Concepts (20 pts)**

You are graded on *how* you solve the problem.

Full credit requires:

- Appropriate use of loops, conditionals, methods, arrays, and/or objects
- Logic that reflects concepts taught in class

Correct output using incorrect or avoided concepts may lose credit.

---

 **5. Code Organization & Readability (15 pts)**

Your code should be:

- Clearly structured into methods
- Properly indented
- Written with meaningful variable and method names

This mirrors expectations for the **AP Free Response section**.

---

 **6. JUnit Testing (when assigned—10 pts)**

- Tests compile and run
- Tests meaningfully verify behavior
- Includes at least one edge or boundary test

Tests should **fail if the implementation is incorrect**.

---

## 7. Explanation / Reflection (10 pts)

In your README or comments:

- Explain what your program does
- Explain *why* your logic works
- Mention challenges, assumptions, or decisions you made

Clear explanation is part of AP-level reasoning.

---



## How Your Final GitHub Grade Is Determined

You will complete **5 GitHub assignments**.

Your overall GitHub track grade is based on **mastery across all five assignments**, not a straight average:

Average Across 5 Assignments	Final GitHub Grade
88–100	A (100%)
80–87	B (90%)
72–79	B- / C+ (85%)
65–71	C (75%)
Below 65	D (65%)

---

# Resubmission & Improvement Policy

Learning from your mistakes is always encouraged!

If you want to improve an assignment **after it has been graded**, you may complete a **resubmission reflection**.

## How to Resubmit

1. Fix or improve your code based on feedback.
2. Record a **screen recording with audio voice-over** (maximum **10 minutes**) explaining:
  - What went wrong in your original submission
  - What you changed
  - Why the revised code is better or more correct
3. In your GitHub repository:
  - Create a new folder named **Resubmission**
  - Upload the video file into that folder
  - Commit the change before the resubmission deadline (if provided)

## Notes

- The original submission is still graded as-is.
  - Resubmissions are evaluated based on **understanding and improvement**, not just correctness.
  - Further instructions on how resubmissions affect scores will be provided per assignment.
- 

## What This Means for You

- ✓ You can commit as often as needed before the deadline
- ✓ Only your final submitted code is graded
- ✓ Clean, correct Java matters more than perfection
- ✓ Testing and explanation are expected
- ✓ You have a structured way to show learning after feedback