

# Unit 2 — Using Objects

**Big Idea:** MOD (Modularity) | **Code.org unit:** Unit 2 | **Teacher:** \_\_\_\_\_ **Period:** \_\_\_\_\_

Your one-page map of everything graded in Unit 2: point values, weights, and the AP objective each item builds toward. Use it alongside the answer keys (folder 02) and the feedback bank (folder 05).

## 1 · Graded items and point values

Graded item	Points	Category	AP objective
Constructors & method calls	15	Practice	MOD-1.C/1.E
String methods practice	20	Practice	MOD-1.F
Math methods & random	10	Practice	MOD-1.G
Unit 2 quiz (25 Q)	25	Assessment	all MOD-1
Handwritten String-method FRQ <b>AI-PROOF</b>	20	Performance	MOD-1.F
<b>Unit total</b>	<b>90</b>		

**AI-proofing:** items marked **AI-PROOF** are the handwritten, in-class replacements (folder 03) you assign when a student's digital work trips the AI red-flags. Same AP objective, no shortcut.

## 2 · Suggested category weights (gradebook)

Category	Weight
Practice (formative)	35%
Assessment	40%
Performance task	25%

These weights are pre-loaded in the gradebook CSV (folder 04). Change one cell and every grade recalculates.

### 3 · AP learning-objective coverage (put this on the wall)

---

Track per class. If the class average on any objective is below 70%, re-teach it *before* the AP exam.

Objective	Students must be able to...
MOD-1.C	Create objects by calling constructors (new).
MOD-1.E	Call methods on objects, passing arguments.
MOD-1.F	Use String methods (length, substring, indexOf, equals, compareTo).
MOD-1.G	Use Math methods (abs, pow, sqrt, random).
VAR-1.H	Understand references, aliasing, and null.

### 4 · Grading order that saves time

---

1. **Quiz/assessment first** — objective, tells you instantly where the class stands.
2. **Skills work by station** — grade one item across the whole stack, then the next. 2–3× faster.
3. **Performance task last**, with the interactive grader (folder 06) open — click criteria, paste feedback.