AP CS-A Text Adventure Project Specification

End-of-Unit Project — Conditionals, Loops, and Variables

Due: October 31st, Friday

Learning Goals

- Write a branching story using **nested `if/else`** statements.
- Control program flow with `while` loops (replay + input validation).
- Use at least one `for` loop for a mini-challenge or visual effect.
- Track **state** with variables (e.g., inventory, health, score).

Strict Requirements (Must Haves)

Story Scope

- At least **8 distinct scenes** (unique prompts/locations), not counting title/replay.
- At least **3 distinct endings** (e.g., win, lose, secret/alt).
- At least **2 major decision points** that lead to meaningfully different mid-game paths (not just cosmetic flavor text).

Control Flow

- **Conditionals:** At least **10 total** `if/else if/else` statements.
- **`while` loops:**
  - A replay loop so the player can play again.
  - At least one input-validation `while` loop (keep asking until valid input).
- **`for` loop:** Use at least one meaningful `for` loop for a mini-challenge (e.g., 3 tries to pick a lock), a countdown/progress effect, or rendering health (`<3`).

State & Variables

- At least one **boolean** (e.g., `hasKey`) or one **counter** (e.g., `coins`).
- A changing **health or score** (`int`) that affects outcomes.
- At least one decision that **checks previous state** (e.g., a gate only opens if you collected a torch).

User Experience

- **Input validation:** The program never crashes on bad input; it re-prompts.
- **Clear text formatting:** Use blank lines and separators for readability.

Code Quality (No Methods Required)

- All code can remain in `main()`, but should include clear comment sections (e.g., `// Scene 1:  Cave Entrance`).
- Use descriptive variable names and inline comments for tricky logic.
- **Do not use** `System.exit(...)` to skip around the story.

Grading Rubric (20 Points Total)

Story Depth & Design (6 pts)

- 8+ scenes (2)
- 3+ endings (2)
- Meaningfully different branches (2)

Control Flow (6 pts)

- Nested conditionals (2)
- Replay + input-validation `while` loops (2)
- Meaningful `for` loop (2)

State Management (4 pts)

- Examples like Inventory + health/score affect outcomes (3)
- Clean variable use (1)

UX & Clarity (4 pts)

- Input validation & formatting (2)
- Readable structure via comments/sections (2)

Bonus Opportunities (+3 pts)

- ASCII art or creative title screen.

- Multiple mini-games or difficulty levels.

- A “choice log” or summary of key decisions at the end.

Deliverables

1. **TextAdventure.java** (single file) Include header comment (name, date, period, brief description, how to play).

2. **README.txt** (5–8 sentences) Describe your story map.