Teacher Exercise

Computer Science Assessment 3 Set S of 2026

Staff only | unmarked standardisation exercise

The materials accompany the EcoQuest Learning sample coursework brief and staff mark scheme, revised on 14 October 2026. All three tasks are addressed: Python game 70 marks, database 15 marks and spreadsheet 15 marks.

The examples are synthetic materials for teaching and standardisation. Test records describe execution of the supplied generated examples; they do not represent observed learner activity. No learner signature or supervision record is supplied.

The revised staff mark scheme and native supporting files should be made available. The Teacher Reference and Proposed Marking and Commentary should be withheld until the independent exercise has been completed.

1. Repeated incorrect guesses

Relevant criteria: I2; T1–T2

Source: synthetic response SYN02

```
if letter in self.guessed:
    if letter not in self.term:
        self.attempts -= 1
        return {
            "status":"repeated_incorrect",
            "attempts":self.attempts,
            "word":self.revealed(),
        }
    return {
        "status":"repeated_correct",
        "attempts":self.attempts,
        "word":self.revealed(),
    }
```

Recorded check	Expected	Actual
G05 | Repeated incorrect guess does not use another attempt	status: repeated_incorrect; attempts: 5	status: repeated_incorrect; attempts: 4 [FAIL]

Exercise

The extract from the guess method and the G05 record are to be reviewed. The change to the attempt counter after the same incorrect letter is entered twice should be traced. The affected implementation point should be identified, with an explanation of the separate treatment of testing and correction evidence.

Proposed judgment and supporting evidence

2. Revealing letters and selecting another term

Relevant criteria: I2–I3

Source: synthetic response SYN03

```
    def revealed(self) -> str:
        shown = []
        remaining = set(self.guessed)
        for letter in self.term:
            if letter in remaining:
                shown.append(letter)
                remaining.remove(letter)
            else:
                shown.append("_")
        return " ".join(shown)

    def start_round(self):
        if not self.unused:
            return None
        pair = self.rng.choice(self.terms)
        return GameRound(*pair)
```

Recorded check	Expected	Actual
G02 | Case and surrounding spaces are normalized; all matching letters reveal	status: correct; attempts: 6; word: _ a _ _ _ a _	status: correct; attempts: 6; word: _ a _ _ _ _ _ [FAIL]
G09 | All 12 supplied pairs are selected once without term reuse	count: 12; unique_terms: 12; matching_pairs: true	count: 12; unique_terms: 1; matching_pairs: true [FAIL]

Exercise

The result of guessing a in habitat should be explained. The selection of later terms should then be reviewed. The distinction between revealing all matching positions and preventing reuse across rounds should be recorded, with a suitable check for each issue.

Proposed judgment and supporting evidence

3. Database validation and relationships

Relevant criteria: B4–B5

Source: synthetic response SYN02

```
PRAGMA foreign_keys=ON;
CREATE TABLE Clues (ClueID VARCHAR(3) PRIMARY KEY NOT NULL
    CONSTRAINT ClueID_must_not_be_blank
        CHECK(length(trim(ClueID))>0),
    Word VARCHAR(10) NOT NULL
    CONSTRAINT Word_must_not_be_blank
        CHECK(length(trim(Word))>0),
    Clue VARCHAR(200) NOT NULL
    CONSTRAINT Clue_must_not_be_blank
        CHECK(length(trim(Clue))>0),
    Difficulty INTEGER NOT NULL
    CONSTRAINT Difficulty_must_be_an_integer_from_1_to_5
        CHECK(typeof(Difficulty)='integer' AND Difficulty BETWEEN 1 AND 5),
    Subject VARCHAR(20) NOT NULL
    CONSTRAINT Subject_must_not_be_blank
        CHECK(length(trim(Subject))>0));
CREATE TABLE Games (GameID VARCHAR(10) PRIMARY KEY NOT NULL
    CONSTRAINT GameID_must_not_be_blank
        CHECK(length(trim(GameID))>0),
    ClueID VARCHAR(3) NOT NULL,

    CONSTRAINT ClueID_must_reference_Clues
        FOREIGN KEY(ClueID) REFERENCES Clues(ClueID));
```

Recorded check	Expected	Actual
D07 | Unknown Subject is rejected	REJECTED	ACCEPTED [FAIL]

Exercise

The schema and unknown-Subject test should be compared with the required Subject values: Ecology, Resources and Energy. Credit for the enforced constraints should be separated from the omitted constraint. The role of GameID and the link through ClueID should also be explained.

Proposed judgment and supporting evidence

4. Spreadsheet calculations

Relevant criteria: S1; S4–S6

Response	Word percentages	Mean calculation	Displayed
SYN01	=F6/E6 (filled down)	=AVERAGE(G6:G17)	61.6%
SYN02	=F6/E6 (filled down)	=SUM(F6:F17)/SUM(E6:E17)	62.8%
SYN03	Typed values	Typed value	62.8%

Totals in the supplied data: 435 games played and 273 games won. The completed workbooks contain the twelve per-word records and are the evidence for sorting, recommendations and limitations.

Exercise

The summary formulas should be compared. The required mean should be calculated from the twelve word percentages and compared with the ratio of total wins to total games. The formula evidence, displayed precision, sort order and workbook recommendations should be considered separately when marks are proposed.

Proposed judgment and supporting evidence

5. Maintenance proposals

Relevant criteria: E3

Source: synthetic response SYN01

Two maintenance changes would be appropriate if the prototype were extended. First, the console could accept a documented command to repeat the clue without changing attempts; implementation would add a separate command branch before letter validation, and regression checks would confirm that the clue is redisplayed while the guessed set and counter remain unchanged. Second, a configurable session length could allow supervised practice without exhausting all 12 terms; implementation would select a random sample from the unused pairs, and tests would verify that no pair repeats, that the chosen number of rounds is completed and that the existing exit path still works. These changes are proposals only and have not been implemented or tested.

Exercise

The maintenance proposals should be assessed against E3. For each change, the explanation of implementation and the explanation of testing should be identified. Credit should be supported by the extract; the proposal itself is not evidence that the change has already been implemented.

Proposed judgment and supporting evidence