IQM3010

INTO QUALIFICATIONS

LEVEL 3 · CONTROLLED STAFF DOCUMENT

Computer Science

Assessment 3 Applied coursework mark scheme

STAFF ONLY · Mark scheme and marking guidance

Duration 20 supervised hours	**Total marks** 100
Assessment window SAMPLE ASSESSMENT	**Assessment set** Set S of 2026
Audience Staff only	**Release timing** At marking

STAFF ONLY Apply the published task criteria and their exact allocations. Credit valid alternative approaches. No completed answers or model code are supplied to candidates.

General marking principles

Use this scheme with the Set S of 2026 candidate brief and submitted evidence. Mark all three tasks. Award integer marks for the evidence described below; the maximum is 100. Record marks against the criterion references and retain brief comments identifying the supporting evidence.

Python is assessed through analysis, design, implementation, testing and debugging, and evaluation and maintenance. The database and spreadsheet tasks are marked separately, as shown in the allocation table. This assessment contributes one third of the module grade.

Criterion	Marks
Python analysis	10
Python design	15
Python implementation	25
Python testing and debugging	10
Python evaluation and maintenance	10
Relational database	15
Spreadsheet	15
Total	100

- Credit valid alternative algorithms, data structures, field names, relational database applications and spreadsheet layouts where they meet the task. A console game can receive full marks. Do not award extra credit for decorative output, a graphical interface or unrequested links between Python, the database and spreadsheet.
- Award each marking point once within its criterion. The same product may support separate criteria where they assess different skills; for example, implemented input validation and a recorded test of that validation.
- A description of intended functionality is design evidence, not proof of a working implementation. Where a program does not run, award independently evidenced correct analysis, design, code, testing and evaluation. Do not apply an overall cap or award credit for unsupported claims.
- Minor errors affect only the relevant points. For testing, a recorded failure can earn credit when it is accurately reported; fabricated successful results cannot. Code inspection alone does not demonstrate that a learner executed tests.
- Accept evidence in submitted working files, the report, screenshots or a permitted supervised demonstration. Record the basis of a demonstration. Follow centre procedures for academic integrity; do not resolve concerns solely by adjusting marks.
- No completed answer, formula set, database solution or Python model answer is provided to candidates. Keep this staff document and the staff reference database within the controlled marking process.

Task 1 Python environmental science revision game

70 marks. EcoQuest Learning is developing a revision game for GCSE pupils learning environmental science terms. Apply the Set S task requirements, including the supplied 12 term/clue pairs, the session rules and the requirement for at least one purposeful class.

Analysis 10 marks

Ref	Marks	Awarding guidance
A1	2	1 mark identifies EcoQuest Learning and its educational purpose; 1 mark identifies GCSE pupils as users and a relevant need, such as clear clues, readable progress or understandable feedback.
A2	2	1 mark identifies specific functional requirements; 1 mark gives measurable success criteria covering key game behaviour. Criteria must be checkable, not simply “the game works”.
A3	2	1 mark identifies relevant inputs and outputs; 1 mark analyses processing and validation requirements.
A4	2	1 mark identifies a practical constraint, such as the supplied data, console interface or supervised time; 1 mark explains a suitable development approach, such as building and checking the game in small stages.
A5	2	1 mark explains a benefit of digital environmental science revision; 1 mark explains a limitation. At least one point must concern access to or communication of learning information. Credit contextual examples such as immediate feedback or dependence on access to a computer.

Design 15 marks

Ref	Marks	Awarding guidance
D1	2	1 mark decomposes the problem into purposeful components; 1 mark explains how they interact, for example word selection, guess processing, state updates and round completion.
D2	3	1 mark for a coherent round loop; 1 mark for a win condition based on revealing all letters; 1 mark for a loss when no incorrect guesses remain. Credit clear pseudocode or flowcharts.
D3	3	1 mark for normalising and validating input; 1 mark for detecting repeats without a penalty; 1 mark for revealing all matching letters or reducing attempts only for a new incorrect letter.
D4	2	1 mark for random selection from unused supplied pairs; 1 mark for round/exit logic that prevents reuse in the session. Accept removing used pairs or tracking them in a suitable collection.
D5	3	1 mark for purposeful class attributes; 1 mark for constructor and method responsibilities; 1 mark for suitable structures that keep each word and clue associated. Accept one well-designed class or several.
D6	2	1 mark for planned normal tests with expected outcomes; 1 mark for planned boundary and invalid-input tests with expected outcomes.

Task 1 Implementation 25 marks

Mark implemented behaviour and code quality from the submitted program and permitted evidence. Accept valid alternative designs that satisfy the same requirements; do not require a GUI or a particular class design.

Ref	Marks	Awarding guidance
I1 Start and input	5	Uses all 12 supplied word/clue pairs without mismatching them; randomly selects a pair and displays its matching clue and one placeholder per letter; accepts a single letter; ignores case and leading/trailing spaces; rejects blank, non-letter and multiple-character inputs without using an attempt.
I2 Guessing procedure	8	1 mark identifies a correct letter and reveals every occurrence; 1 mark preserves previously revealed letters; 1 mark leaves attempts unchanged after a correct guess; 1 mark starts each round with six incorrect guesses; 1 mark deducts exactly one attempt for a new incorrect guess; 1 mark detects a repeated correct or incorrect guess, reports the repeat and applies no penalty; 1 mark displays guessed letters; 1 mark displays the current revealed word and remaining attempts accurately.
I3 Win, loss and replay	6	1 mark ends a completed word as a win; 1 mark ends at zero attempts as a loss; 1 mark displays the outcome and correct word; 1 mark offers another round or exit; 1 mark resets round state correctly; 1 mark prevents a word from being used again during the session.
I8 Purposeful class	3	Uses a purposeful class containing relevant state; a constructor initialises that state; methods carry out game actions. A class that merely wraps unrelated code does not earn full credit.
I9 Code quality	3	Uses appropriate functions or methods to avoid unnecessary repetition; code is readable and maintainable, with meaningful names, useful whitespace and comments where needed; important decisions are justified.

Task 1 Testing and debugging 10 marks

Ref	Marks	Awarding guidance
T1	8	Tests cover normal correct and incorrect guesses; repeated letters and repeated guesses; blank, non-letter and multiple-character inputs; invalid inputs leaving attempts unchanged; a loss; a win; another round and exit; and words not being reused. Expected and actual results are recorded with supporting evidence.
T2	2	Makes fixes where errors occur and retests those errors. If no errors are found, award both marks where comprehensive recorded tests show that the relevant requirements are met.

Task 1 Evaluation and maintenance 10 marks

Ref	Marks	Awarding guidance
E1	3	Makes judgements against success criteria; supports judgements with test evidence; evaluates the solution rather than merely describing it.
E2	3	Evaluates usability for GCSE pupils; explains a specific limitation; explains its consequence for users or the game. Generic claims such as “easy to use” without support do not earn the relevant point.
E3	4	For each of two realistic maintenance changes: explains the change and how it would be implemented; explains how it would be tested.

Applying the Python criteria

Distinguish analysis and the planned solution from implemented behaviour, actual tests and reasoned evaluation. Working code cannot substitute for an absent plan. Use the full criterion evidence and award an integer mark within each criterion maximum; do not invent submark caps or double-count a point.

Task 2 Relational database 15 marks

The task uses two tables for EcoQuest Learning: Clues and Games. Clues contains the supplied 12 term and clue records. Games contains GameID and ClueID. A relationship through ClueID is required.

Ref	Marks	Awarding guidance
B1	3	1 mark for the five required field names (ClueID, Word, Clue, Difficulty and Subject); 1 mark for suitable data types; 1 mark for text sizes that store all supplied values without truncation.
B2	2	1 mark for importing all 12 supplied records; 1 mark for preserving the word/clue associations and all supplied values.
B3	2	1 mark for choosing a suitable primary key; 1 mark for explaining how its unique value identifies each record. ClueID is the expected choice.
B4	4	1 mark restricts Subject to Ecology, Resources or Energy; 1 mark restricts Difficulty to integers 1–5 inclusive; 1 mark requires values in all five Clues fields and provides suitable validation messages; 1 mark demonstrates that valid values are accepted and invalid or blank values are rejected, including evidence of messages.
B5	4	1 mark creates a Games table with exactly two fields, GameID and ClueID; 1 mark sets GameID as its primary key; 1 mark links Clues and Games through ClueID; 1 mark justifies the relationship.

Database marking notes

- Accept suitable relational database applications and valid alternative physical data types where all required values are preserved and constraints work.
- For B1, award one mark each for the required Clues field names, suitable data types and sufficient text sizes.
- The headerless EcoQuest_Clues.csv contains 12 records in the order ClueID, Word, Clue, Difficulty and Subject. The subject values are Ecology, Resources and Energy; retain all supplied values.
- For B4, the validation evidence must show allowed Subject values, integer Difficulty values from 1 to 5, required values in all five fields and suitable messages. Valid records must be accepted; invalid subject/difficulty and blank values must be rejected and evidenced.
- No queries, forms, macros or connection to Python or Excel are required. The staff reference database contains the 12 supplied Clues records and the required Games structure.

Task 3 Spreadsheet 15 marks

The spreadsheet uses the supplied simulated results for the 12 EcoQuest terms. Mark formula accuracy, totals, the unweighted mean of the 12 per-word win percentages, ascending row-preserving sort, recommendations and a contextual modelling limitation.

Ref	Marks	Awarding guidance
S1	6	1 mark for a correct per-word win percentage formula (wins divided by games played); 1 mark for correct formulas across all 12 words; 1 mark for one-decimal percentage display; 1 mark for total games played; 1 mark for total games won; 1 mark for the unweighted mean of the 12 word percentages. A hard-coded result does not earn formula marks.
S4	3	1 mark for sorting by win percentage; 1 mark for ascending order; 1 mark for keeping each full row together so all values remain associated.
S5	4	1 mark identifies relevant term(s) where observed success and difficulty may be inconsistent; 1 mark supports selections with specific percentages, counts and ratings; 1 mark proposes a plausible clue review or rating adjustment; 1 mark explains the recommendation and treats results as evidence for review, not proof. Recommendations must appear in an identifiable workbook area.
S6	2	1 mark explains a limitation of spreadsheet modelling; 1 mark gives a contextual explanation.

Expected baseline results for staff checking

These are expected calculations from the supplied Set S data. They are reference checks only, not evidence that a learner performed a test. The candidate receives the raw data only; do not supply completed formulas, a sorted solution or recommendations.

Ascending position	ClueID	Word	Difficulty	Played	Won	Win %
1	C12	geothermal	5	35	7	20.0%
2	C01	habitat	1	40	10	25.0%
3	C10	turbine	3	40	12	30.0%
4	C06	compost	4	25	8	32.0%
5	C02	species	3	35	21	60.0%
6	C11	renewable	5	30	20	66.7%
7	C04	prey	2	40	28	70.0%
8	C03	predator	4	30	24	80.0%
9	C08	landfill	2	35	29	82.9%
10	C07	reuse	1	30	27	90.0%
11	C05	recycle	1	45	41	91.1%
12	C09	solar	1	50	46	92.0%

Spreadsheet expected totals and interpretation

Check	Expected result
Total games played	435
Total games won	273
Unweighted mean of the 12 per-word percentages	61.6% (61.6362…% before display rounding)

The mean above is the arithmetic mean of the 12 individual win percentages. It is not the overall total wins divided by total games. If the workbook uses displayed rounded percentages, allow a result consistent with the spreadsheet's documented calculation and rounding method.

A suitable recommendation may review habitat, which has a 25.0% observed win rate despite difficulty 1, or the clue/difficulty balance for geothermal and renewable, both rated 5. Lower success alone does not prove that a rating is wrong. Other evidence-based recommendations can receive full credit.

The results describe the supplied practice data. They do not prove that a clue caused a particular outcome or that all GCSE pupils would perform similarly. Credit a different relevant limitation when explained in context.

Mark recording

Criterion	Maximum	Awarded
Python analysis	10	
Python design	15	
Python implementation of behaviour	25	
Python testing and debugging	10	
Python evaluation and maintenance	10	
Relational database	15	
Spreadsheet	15	
Total	100	

Retain criterion-level evidence and any centre-required records under the applicable academic-integrity and assessment procedures. This sample mark scheme does not itself approve or release candidate materials.