Diagnostic Assessment Test – Unit 02

Collections and Functions

(MCQ + Subjective; designed to identify learning gaps)

Instructor: Tofik Ali **Time Limit:** None
Submission Deadline: March 31, 2026
Student Name: ____________________ **Roll No.:** _________

Instructions

- **This is a diagnostic assessment, not a quiz/test.** The goal is to identify learning gaps so you know what to revise next.
- **Total marks: 100.** Easy: 10 × 2 = 20, Medium: 10 × 3 = 30, Hard: 10 × 5 = 50.
- **No time limit.** Submit on or before **March 31, 2026**.
- **Important (70% rule): To have your assignment marks counted, you must score at least 70% (70/100 or more) in this assessment.** If you score below 70%, you may reattempt and resubmit. Your latest submission will be considered.
- **Marking policy (honest attempt):** Marks are deducted only for (i) not attempting a question and (ii) high similarity with other submissions (copying). Correctness is **not** the main focus; your reasoning and effort matter more.
- **Show your real effort:** if you tried multiple approaches for any question, write them all (e.g., Attempt 1, Attempt 2). Partial or incorrect attempts are acceptable.
- **Many questions are open-ended by design:** multiple solutions/variants are acceptable. Use your own example data and explain your choices.
- Answer **all** questions.
- For MCQs, follow the instruction in the question (some are **select all that apply**).
- For subjective questions, write assumptions and show steps. We care more about your thinking process than a perfect final answer.
- Do not use external libraries unless a question explicitly allows it.

Easy (10 Questions)

E1. [2 marks] [MCQ – Select all that apply] Which statements are **true** about lists? (Write a 1-line justification for any **two** options you select.)

(A) `append(x)` adds **one** element to the end of a list.

(B) `extend(iterable)` adds each element of the iterable to the list.

(C) `sorted(a)` sorts the list `a` in-place and returns `None`.

(D) `a.sort()` returns a new sorted list without changing `a`.

(E) Slicing like `a[1:4]` produces a new list.

(F) `remove(x)` removes by index position.

E2. **[2 marks] [MCQ – Select all that apply]** Which operations will **modify** the original list `a`?

(A) `a.append(5)`

(B) `a = a + [5]`

(C) `a.extend([5])`

(D) `a.sort()`

(E) `b = a + [5]`

(F) `a.insert(0, 5)`

E3. **[2 marks] [Subjective]** Create your own list `nums` of at least 8 integers (include duplicates and at least one negative). Write code to produce:

(a) a set of unique values

(b) count of unique values

(c) a sorted list of unique values

(Many correct solutions are possible.)

E4. **[2 marks] [Subjective]** Write a **single** list comprehension (use **if–else**) that converts a list of integers into labels `"even"` and `"odd"`. Use any input list of your choice and show the output.

E5. **[2 marks] [Subjective]** Show **two different** ways to swap variables `a` and `b` in Python. (Use any numbers, ideally based on your roll number digits.)

E6. **[2 marks] [MCQ – Select all that apply]** Which statements are **true** about tuples?

(A) Tuples are immutable (you cannot change elements after creation).

(B) A tuple can contain mutable objects (like lists) inside it.

(C) `(5)` creates a one-element tuple.

(D) `(5,)` creates a one-element tuple.

(E) Tuples have `append` and `remove` methods.

(F) Tuple unpacking can assign multiple variables in one line.

E7. **[2 marks] [Subjective]** Write a function `clamp(x, low=0, high=100)` that returns:

- `low` if `x < low`
- `high` if `x > high`
- otherwise returns `x`

Show at least 3 sample calls.

E8. **[2 marks] [Subjective]** Create a dictionary `profile` about yourself with at least 5 keys (e.g., name, city, branch, year). Write code to: add a key, update a key, delete a key, and print all key-value pairs.

E9. **[2 marks] [Subjective]** Create two sets `A` and `B` (size ≥ 5) from any real-life categories (e.g., favorite apps, hobbies). Print: union, intersection, and symmetric difference. Explain each in 1 line.

E10. **[2 marks] [Subjective]** Explain (with a small code example) when `d.get(key, default)` is safer than `d[key]`. Give at least **two** examples of missing keys.

Medium (10 Questions)

M1. [3 marks] [MCQ – Select all that apply] Which of the following create a **shallow copy** of a list `a`?

(A) `b = a`

(B) `b = a[:]`

(C) `b = list(a)`

(D) `b = a.copy()`

(E) `b = [*a]`

(F) `b = a + []`

M2. [3 marks] [Subjective] Write **two** solutions (two different styles) for this task: "From a list of integers, keep only numbers divisible by 3 and square them."

- Solution 1: list comprehension
- Solution 2: `filter` + `map`

Use any input list of your choice.

M3. [3 marks] [Subjective] Write a function `word_freq(sentence)` that returns a dictionary of word frequencies **case-insensitive**. Ignore punctuation `.,!?` by replacing them with space. Test using a sentence you choose.

M4. [3 marks] [Subjective] Create a list of at least 5 tuples `(name, marks)` (you choose the data). Sort by marks (descending) and then by name (ascending). Print the sorted list.

M5. [3 marks] [Subjective] Implement `stats(*nums)` that returns `(min, max, avg)`. If no numbers are passed, return `None`. Show at least 3 sample calls (including empty call).

M6. [3 marks] [MCQ – Select all that apply] Which statements about dictionaries are true?

(A) `d.keys()` returns a view of keys (not a list).

(B) `d.items()` returns key-value pairs.

(C) `d.get(k)` raises `KeyError` if `k` is missing.

(D) `d.update(x)` can merge another dictionary into `d`.

(E) `d.pop(k)` removes and returns the value for key `k`.

(F) Dictionaries can only have integer keys.

M7. [3 marks] [Subjective] Invert a mapping:

- Given: `student_to_branch` (you create it)
- Build: `branch_to_students` where each branch maps to a list of student names

M8. [3 marks] [Subjective] Create a 2D list (matrix) using your roll number digits (any shape at least 3x3). Write code to compute: transpose, row sums, and column sums.

M9. [3 marks] [Subjective] Write a small code snippet that demonstrates the difference between: **aliasing** (`b = a`) and **copying** (`b = a[:]`) for lists. Explain what happens in 3–5 lines.

M10. [3 marks] [Subjective] Write a recursive function `gcd(a, b)` (Euclid's algorithm) **or** `sum_to_n(n)`. Show at least 2 test cases.

Hard (10 Questions)

H1. [5 marks] [MCQ – Select all that apply] Which functions have a **mutable default argument** problem?

(A) `def f(x, items=[]): ...`

(B) `def g(x, items=None): ...`

(C) `def h(x, d={}): ...`

(D) `def k(x, t=(1, 2)): ...`

H2. [5 marks] [Subjective] Rewrite **one** problematic function from H1 to avoid the mutable-default pitfall. Explain in 4–6 lines why the original was risky.

H3. [5 marks] [Subjective] Write a function `is_balanced(s)` that checks whether the brackets in a string are balanced. Consider `()`, `{}`, and `[]` and use a stack (list). Provide at least 5 test cases.

H4. [5 marks] [Subjective] Write a recursive function `flatten(lst)` that converts a nested list into a flat list. Example: `[1, [2, 3], [4, [5]]]` → `[1, 2, 3, 4, 5]`. Provide at least 3 test cases.

H5. [5 marks] [Subjective] Implement `top_k_frequent(words, k)` using a dictionary + sorting. Use a paragraph you choose as input text and show the output for `k=3`. (Many correct tie-breaking strategies are acceptable; mention yours.)

H6. [5 marks] [Subjective] Design and implement a mini "attendance tracker" (your choice of format).

- Propose a data structure (nested list/dict/tuple/set).
- Write function signatures for: mark present, mark absent, get attendance percentage, and list defaulters.
- Implement at least **two** of the functions and show a small demo run.

H7. [5 marks] [Subjective] Write a generic `group_by(items, key_func)` that returns a dictionary: key → list of items. Demonstrate with at least two examples (e.g., group words by length, group numbers by parity).

H8. [5 marks] [Subjective] Use a list comprehension with **two loops** and a condition: build coordinate pairs `(i, j)` for `i=1..m` and `j=1..n`, excluding pairs where `i == j`. Choose `m` and `n` based on your roll number digits and show output. Also write an alternative nested-loop solution.

H9. [5 marks] [Subjective] Implement one of these (your choice) and show a demo:

(a) `apply_all(funcs, x)`: given a list of functions, return a list of results.

(b) `compose(*funcs)`: compose multiple single-argument functions into one.

(Any valid approach is accepted; show at least 2 small functions/lambdas.)

H10. [5 marks] [Subjective] Short reasoning: compare membership checking in a **list** vs a **set**. Write 6–10 lines (no timing required) and include a small code snippet illustrating the idea.

End of Assessment