COMP 1023 Midterm Exam - Spring 2026 - HKUST
Answer Book

Date: April 11, 2026 (Saturday)
Time Allowed: 2 hours, 2:00–4:00 pm
Instructions:

1. This is a closed-book, closed-notes examination.
2. There are **8** questions on **13** pages (including this cover page).
3. Write your answers in the space provided in this answer book using black/blue ink. *NO pencils please, otherwise you are not allowed to appeal for any grading disagreements.*
4. All programming code in your answers must be written in Python version as taught in class.
5. Unless otherwise stated, you are **NOT** allowed to define additional classes, helper functions (including inner functions) and use global variables, nor any library functions not mentioned in the questions.
6. **Type hinting does not need to be included in your code.**
7. No electronic devices are allowed, except for an HKEAA approved calculator.

Student Name			
Student ID			
Venue		Seat Number	

Total (For T.A. Use Only)

The HKUST Academic Honor Code

Honesty and integrity are central to the academic work of HKUST. Students of the University must observe and uphold the highest standards of academic integrity and honesty in all the work they do throughout their program of study.

As members of the University community, students have the responsibility to help maintain the academic reputation of HKUST in its academic endeavors. Sanctions will be imposed if students are found to have violated the regulations governing academic integrity and honesty.

Declaration of Academic Integrity

I confirm that I have answered the questions using only materials specifically approved for use in this examination, that all the answers are my own work, and that I have not received any assistance during the examination.

Signature: _____

Consent for Use of Exam Scripts in AI Training

We are seeking your consent to use your exam scripts for the purpose of training an artificial intelligence system that will enhance our online and onsite computer-based examination platform. This AI will help improve the quality and efficiency of our assessments. Please note that your identity, including your name and student ID, **will not** be used in the training process.

Your contributions are invaluable, and we want to ensure that you are comfortable with this process. If you agree to allow us to use your exam scripts for this purpose, please sign to indicate your consent. Thank you for your support in helping us create a better examination experience for everyone.

Declaration of Consent

I confirm that I agree to allow my exam scripts to be used for training the AI.

Signature: _____

You are not required to sign if you do not agree to allow your exam scripts to be used.

For T.A. Use Only

Problem	Topic	Score		Total	Full Mark
1	True/False Questions				10
2	Python Operations				7
3	Branching	(a)			4
		(b)			
4	Looping	(a)			9
		(b)			
5	Lists	(a)			16
		(b)			
		(c)			
6	Functions	(a)			16
		(b)			
		(c)			
		(d)			
7	Fare Helper	(a)			22
		(b)(i)			
		(b)(ii)			
		(c)			
		(d)(i)			
		(d)(ii)			
		(d)(iii)			
8	Strings and Lists Utilities	(a)(i)			16
		(a)(ii)			
		(a)(iii)			
		(b)(i)			
		(b)(ii)			

Problem 1 [10 points] True/False Questions

Answer:

Question	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)
Answer										

Problem 2 [7 points] Python Operations

Answer:

After finishing the statement on line	`x`	`y`	`z`	`check`
5				
7				
9				
11				

Problem 3 [4 points] Branching

(a) [1 point] What is the output of the given program?

Answer:

(b) [3 points] What is the output of the given program?

Answer:

Problem 4 [9 points] Looping

Answer:

(a) [8 points] Complete the tracing table below. Stop when the loop terminates, and leave the remaining table row(s) blank.

Answer:

Iteration	`i`	`val`	`total`
1			
2			
3			
4			
5			
6			
7			

(b) [1 point] What is the final return value of `process_signals(data)`?

Answer:

Problem 5 [16 points] Lists

Answer:

(a) [3 points] Implement `contains_all` with the given criteria.

```
def contains_all(lst: list[int]) -> bool:
```

(b) [6 points] Implement `check_rows_and_columns` with the given criteria.

```
def check_rows_and_columns(grid: list[list[int]]) -> bool:
```

(c) [7 points] Implement `check_boxes` with the given criteria.

```
def check_boxes(grid: list[list[int]]) -> bool:
```

Problem 6 [16 points] Functions

(a) [4 points] Implement the function `get_grade` with the given criteria.

(b) [3 points] Implement the function `count_grades` with the given criteria.

(c) [4 points] Implement the function `apply_bonus` with the given criteria.

(d) [5 points] Implement the function `find_rank` with the given criteria.

Problem 7 [22 points] Fare Helper

(a) [2 points] Implement the function `part_a` with the given criteria.

```
def part_a(stations: list[str], classes: list[str], prices: list[list[int]],
           input_station: str, input_class: str) -> int:
```

(b) [4 points] Implement the function `part_b` with the given criteria.

```
def part_b(stations: list[str], prices: list[list[int]],
           new_station: str, new_price: str) -> None:
```

(c) [6 points] Implement the function `part_c` with the given criteria.

```
def part_c(stations: list[str], prices: list[list[int]]) -> list[str]:
```

(d) [10 points] Implement the function `part_d` with the given criteria.

```
def part_d(stations: list[str], prices: list[list[int]]) \
           -> tuple[list[float], list[str], float]:
```

Problem 8 [16 points] Strings and Lists Utilities

(a) (i) [3 points] Implement the function `join` with the given criteria.

```
from typing import Any
def join(target: list[Any], sep: str) -> str:
```

(ii) [3 points] Implement the function `get_diagonal` with the given criteria.

```
from typing import Any
def get_diagonal(target: list[list[Any]]) -> list[Any]:
```

(iii) [2 points] Implement the function `diag_to_str` with the given criteria.

```
from typing import Any
def diag_to_str(target: list[list[Any]]) -> str:
```

(b) (i) [4 points] Implement the function `mask_diag` with the given criteria.

```
from typing import Any
def mask_diag(target: list[list[Any]], val: Any) -> list[list[Any]]:
```

(ii) [4 points] Implement the function `skip_list` with the given criteria.

```
from typing import Any
def skip_list(target: list[Any], factor: int) -> list[Any]:
```

---------- END OF PAPER ----------