

# United International University

## Department of Computer Science and Engineering

DS 1501: Programming for Data Science

Final Examination : Fall 2024

Total Marks: 40      Time: 2 hours

*Answer all questions. Numbers to the right of the questions denote their marks.*

**Any examinee found adopting unfair means will be expelled from the trimester / program as per UIU disciplinary rules.**

1. (a) Write a Python program that can perform the following tasks: create a text file named `students.txt`, and write the following three lines (See Table 1 : Sample Input) into the file. Afterward, close the file, reopen it, and display its contents on the screen as shown in Table 1 (Sample Output). [4]

Sample Input (students.txt)	Sample Output
Alice, 85	"Name": "Alice"
Bob, 78	"Mark": 85
Charlie, 92	"Name": "Bob"
	"Mark": 78
	"Name": "Charlie"
	"Mark": 92

Table 1: Sample Input and Output for Question 1(a)

- (b) In a small town in Bangladesh, a local grocery store wants to introduce a simple billing system. The system should take the name of an item, its price, and the quantity purchased from the user, calculate the total cost, and apply a 10% discount if the total exceeds 1000 Taka. Write a Python program that defines a function `calculate_bill(price, quantity)` to compute the total cost and apply the discount if applicable, then prints the final amount the customer needs to pay.

Refer to Table 2 for a sample input and output.

[6]

Sample Input	Sample Output
Enter item name: Rice	Total cost before discount: 1200 Taka
Enter price per unit: 120	Discount applied: 10%
Enter quantity: 10	Final amount to pay: 1080 Taka

Table 2: Sample Input and Output for Question 1(b)

2. (a) A school in Dhaka is developing a student record system and needs a function that processes a student's full name by removing extra spaces, extracting their initials in uppercase, counting the number of words, and displaying the initials along with the total word count. Write a Python program that defines a function `'process_name(name)'` which removes extra spaces, extracts and returns the initials in uppercase, and counts the number of words. The program should take a full name as input, call the function, and print the initials along with the word count. [6]

**Sample Input:** Enter full name: Ahsan Habib

**Sample Output:** Initials: AH

Total words: 2

- (b) Find the output of the following code:

[4]

```
names_list = ["Leonhard Euler", "Carl Gauss", "Sophie Germain", "David Hilbert", "Emmy
    ↪ Noether"]
index_x = 2
index_y = 4
index_z = 6
selected_name = names_list[index_x]
modified_name = selected_name.replace(selected_name[index_y], selected_name[index_z].
    ↪ upper(), 1)
substring_part = names_list[index_y][-index_z:]
final_result = modified_name + substring_part
print(f"Final result: {final_result}")
```

3. (a) You are given a dictionary where the keys are student names and the values are lists of courses they have taken. Write a Python program to find the names of students who have taken the same set of courses. The order of courses in the list does not matter. A student cannot be compared to themselves, and each student should appear only once in the result list. [6]

**Sample Input:**

```
students_courses = {
    "Arif": ["Data Structures", "Algorithms", "Databases"],
    "Rafiq": ["Operating Systems", "Computer Networks"],
    "Sadia": ["Data Structures", "Algorithms", "Databases"]
}
```

**Sample Output:**

The student names who have the same courses are: Arif,Sadia

- (b) Write the output of the following code: [4]

```
course_list = ["Machine Learning", "Data Wrangling", "Mathematics", "Biology"]
course_dict = {j: len(j) for j in course_list if len(j) % 2 == 0}
print("Original dictionary:", course_dict)
keys_to_delete = []
for name, length in course_dict.items():
    if length > 15:
        keys_to_delete.append(name)
    else:
        print(course_dict.get(name))
for key in keys_to_delete:
    del course_dict[key]
print("Updated dictionary:", course_dict)
```

4. (a) You are given two groups, group\_A and group\_B, representing employees in two different departments. [6]
- Write a function that returns True if all employees of group\_A are in group\_B.
  - Write a function that returns True if all employees in group\_B are in group\_A.
  - Write a function that returns True if group\_A is smaller than group\_B and contains only employees in group\_B, with at least one employee in group\_B not in group\_A.

- (b) Write the output of the following code: [4]

```
import numpy as np

arr = np.array([1, 2, 3, 4, 5])
b = arr + 10
c = arr * 2
arr_2d = np.array([b,c])
arr_resaped = arr_2d.reshape(2,5)
print("Original_Array:", arr)
print("b:",b,"c:",c)
print("arr_resaped:",arr_resaped)
print("Elements:",arr_resaped[1,[1,2]])
```