Applied Algorithmic Thinking (Fall 2024)

Final Project

Part 1: Skills Workshops-Registration System

A university wants a simple program to manage skills workshop registrations, seat assignments, and student check-ins. You are tasked with developing a program that enables students to view available workshops, register for sessions, make payments, and check in for workshops.

Q1- Workshop Scheduling

Workshop Schedule Data Structure: Create a data structure with at least five workshops. Each workshop should contain details like workshop code, name, instructor, timing, duration, total seats, seats available for registration, and course fee.

Example data structure:

```
workshops = [
    {"workshop_code": "W201", "workshop_name": " Assignment Writing",
"instructor": "Dr. Johnson", "timing": "10:00 AM", "duration": 2, "total_seats":
40, "seats_available": 40, "course_fee": 150},
    {"workshop_code": "W202", "workshop_name": "Using AI for Exam Prep",
"instructor": "Dr. Lee", "timing": "1:00 PM", "duration": 3, "total_seats": 35,
"seats_available": 35, "course_fee": 200},
]
```

Or

```
workshops = [

    ["W201", "Effective Assignment Writing", "Dr. Johnson", "10:00 AM", 2, 40,
40, 150],
    ["W202", "Using AI for Exam Prep", "Dr. Lee", "1:00 PM", 3, 35, 35, 200],
]
```

(Note: The selection of data structure is up to you; the above are just examples.)

View Workshop Schedules: Write a function that displays all available workshops with the workshop name, timing, duration, and course fee so that students can choose one.

Q2- Registration Process

a. Registering for a Workshop: Allow students to reserve a seat by selecting a workshop code and specifying the number of seats. Check the workshop schedule data structure to see if that number of seats is available, and then book the seats. If the workshop is fully booked, display an appropriate message and offer to book another workshop. If seats are available, update the available seat count for the workshop and proceed with registration.

b. Create Registration: For each registration, collect student information, including name, age, student ID, and payment details, and store it in a registration data structure with the selected workshop code. This registration data structure should store each registration as a new entry against a registration number, formatted as a serial number starting from REG1. Each new registration number should increase sequentially (e.g., REG2, REG3, etc.) based on the highest existing registration number.s

c. Payment Details: Each registration requires a payment. The payment function should calculate the total payment as `(course fee * number of seats) + 5% VAT`.

d. Registration Confirmation: Display a registration confirmation showing the registration number, workshop details, student information, and payment details.

e. Continue Registering: The program should allow students to continue registering for workshops until they choose to exit.

Part 3: Check-In and Attendance Pass

a. Check-In Function: Allow students to check in by entering their registration number. Retrieve the registration details based on the registration number. The workshop code should be retrieved from the registration details, and the relevant workshop details should be extracted from the workshop schedule. If the provided registration number is invalid, display an appropriate message and prompt the student for another registration number.

b. Generate an Attendance Pass: After check-in is complete, generate an attendance pass for each checked-in student, including workshop details, seat assignment, and student name.

Important Guidelines

1. Your program should use appropriate user-defined functions for all the processes, such as registration, check-in, payment, and confirmation.
2. Choose appropriate two-dimensional data structures, such as lists, tuples, sets, and dictionaries, to store data for workshops, registrations, and payments.

Sample Output:

Welcome to the University Skills Workshop Registration System

---

1. Viewing Available Workshops
Available Workshops:
Workshop Code | Workshop Name | Instructor | Timing | Duration | Seats Available | Course Fee
W201 | Effective Assignment Writing | Dr. Johnson | 10:00 AM | 2 hours | 40 | 150 AED
W202 | Using AI for Exam Prep | Dr. Lee | 1:00 PM | 3 hours | 35 | 200 AED

2. Registering for a Workshop
Please enter the Workshop Code to register for a workshop or type 'exit' to quit: **W202**
How many seats would you like to register for? (Max seats per student: 5): **2**
Please enter your details:
Enter Student Name: **John Doe**
Enter Age: **20**
Enter Student ID: **U12345**

Payment Details
Course Fee per Seat: 200 AED
Number of Seats: 2
Subtotal: 400 AED
VAT (5%): 20 AED
Total Payment: 420 AED
Registration completed successfully!

Registration Confirmation:
Registration Number: REG001
Workshop: Using AI for Exam Prep
Instructor: Dr. Lee
Timing: 1:00 PM
Duration: 3 hours
Student Name: John Doe
Student ID: U12345
Total Payment: 420 AED
Would you like to register for another workshop? (yes/no): **no**

---

3. Check-In

To check in, please enter your registration number: **REG001**

Registration Found:

Registration Number: REG001
Workshop: Using AI for Exam Prep
Instructor: Dr. Lee
Timing: 1:00 PM
Duration: 3 hours
Student Name: John Doe
Check-in successful!

Attendance Pass:

Workshop: Using AI for Exam Prep
Instructor: Dr. Lee
Seat Assignment: A12, A13
Student Name: John Doe
Thank you! Please proceed to the workshop at the scheduled time.

Part 2: Data Analysis and Visualization

Your task is to analyze and visualize data using Python and relevant libraries. Choose a dataset from the provided link, which includes additional details about each option. Write and document a Python program to answer the questions below:

1. Import a dataset into a pandas DataFrame
2. Display the dataset's first 15 and last 30 rows, and show row and column count. Discuss any patterns or unusual values you observe.
3. Summarize the dataset's context, columns, and any patterns in the data.
4. Select a column in your dataset that contains text data (such as names, categories, or locations). Using Python, perform the following steps:
   a) Convert all values in this column to lowercase to ensure consistency.
   b) Sort the values alphabetically and display the first and last entries in the sorted list.
   c) Describe any patterns or notable observations, such as frequently occurring entries, unique values, or inconsistent data.

1. Select a categorical column with repeated values in your dataset. Using Python:
   a) Identify the unique values in this column and display how many times each unique value appears.
   b) Calculate the percentage distribution of each unique value.
   c) Create a well-formatted **pie chart** to visualize this distribution.
   d) Briefly describe any patterns or insights from the chart.
6. Select one unique value from the list of unique values identified in the previous question.
   a) Create a new data frame containing only the rows where the column has this selected value.
   b) Create a well-formatted **scatter plot** to examine if there is a relationship between two numerical columns of your choice in this new data frame.
   c) Explain why you chose these two columns and describe any patterns or relationships you observe in the scatter plot.
7. Using the dataset you have chosen and the analysis you have previously conducted, formulate a specific question related to data filtering and analysis. Then, provide the code to answer your question, along with a brief explanation of the results.

Assignment Guidelines and Instructions:

Assignment Guidelines: As you solve each part of the problem, follow the steps of algorithmic thinking as outlined below.

Section	Description
Step 1: Algorithm and Flowchart (required only for Part 1)	Use an algorithm and a flowchart to develop and express your algorithm that accomplishes the given task. Remember, you must be very explicit and clear to ensure one can achieve the required solution following the steps of your algorithm. Describe the input(s), output(s), and algorithm processes. Follow the flowchart shape conventions reading, which is available here.
Step 2: Python Code	Implement the algorithm in Python using the basic concepts and structures we covered in class (IMPORTANT: ONLY USE CONCEPTS COVERED IN CLASS). 1 - Download this Python file (.ipynb) to your local machine and save it with name student_name_AATPrimary_Final 2 – Run Jupyter Notebook or open Colab on your browser 3 - Open the Python file you downloaded in the first step. 4 - You can start working on your assignment by answering the questions in the corresponding cells. 5 - If you have any questions, please contact your instructors or the CIS tutors.
Step 3: Program Testing	**Part 1**: a. Create and conduct a Test Plan with at least 2 test cases that demonstrate your code works as intended. b. Clearly explain your choice of test cases and how they help ensure your code functions properly. **Part 2**: Run the code to generate the required data, DataFrames, and graphs.
Step 4: Program Documentation	a. Be sure to use Python comments to clearly show you understand what every line of the code is intended to accomplish. b. Ensure your code is well structured and easy to read and understand.
Step 5: Contribution	A paragraph mentioning the individual contribution of each group member.
Step 6: References	List any references used in APA format, if any.

Instructions for Students: You will work on this assignment in groups of two students. Carefully and thoroughly read each of the instructions below thoroughly:

Instruction	Description
1- Code restrictions:	• Code with advanced tools or programming structures not covered in the course will get a maximum score of 2 in #Computationaltools and a maximum score of 4 in #Compprogramdesign LOs. • You must understand every single line of code submitted. • Your instructor might call you for a technical interview on your assignment explaining your approach to coding.
2- Plagiarism:	• Make sure that you submit your original work. • AI-generated work is not considered original and will be treated as a case of plagiarism. • Suspected plagiarism cases will be treated as possible academic misconduct and will be reported to the College Academic Integrity Committee for formal investigation. • As part of this procedure, your instructor may require you to meet with them for an oral exam on the assignment.
3- Submission:	3.1 Two files need to be submitted: 1. **Primary Resource:** A single PDF file of your Python (.ipynb) workbook. A neat, clearly presented, and easy-to-read file should be submitted. The file should be submitted using the filename "student_name_AATPrimary_Final.pdf". 2. **Secondary Resource:** A Zip file. The file should be submitted using the filename "student_name_AATSecondary_Final.zip": 1. Python Code File (.ipynb file): This file should include: o The algorithm flowchart, code, and test cases for Part 1. o The code and a test for Part 2. 2. Flowchart Image: An image file of your flowchart for Part 1. 3. CSV File: The dataset used for Part 2.
4 Important Notes:	1. The PDF file of your Python code file (ipynb) should be submitted as a primary resource. 2. No PDF submission will result in a 0 grade in #Compprogramdesign and #ProfessionalWorkProduct Los and a maximum of 4 in other Los. 3. If the Python (ipynb) file is not submitted, the final grade will be 0 in #Compprogramdesign and a maximum of 4 in #ProfessionalWorkProduct and #Computationaltools LOs.

Assignment LOs Grading:

Component	Learning Outcomes (LO)
Algorithm and Flowchart	#Algorithmicstrategies
Program effectiveness, efficiency, and testing	#Compprogramdesign, #Computationaltools
Documentation	#QuantCommunication
Creating and submitting a professional/coherent PDF File	#ProfessionalWorkProduct

Rubric:

AAT Assignment Rubrics						
LOs	5	4	3	2	1	0
Computational Tools	The submission fully meets the criteria: The code and design provided use standard terminology, notations, and tools. The inputs and output variables are well-defined, with explicit requests for inputs and well-formatted output. Structured programming concepts of sequence, selection, and repetition are applied.	The submission mostly meets the criteria: The code and design provided use standard terminology, notations, and tools. The inputs and output variables are well-defined, with explicit requests for inputs and well-formatted output. Structured programming concepts of sequence, selection, and repetition are applied.	The submission somewhat meets the criteria: The code and design provided use standard terminology, notations, and tools. The inputs and output variables are well-defined, with explicit requests for inputs and well-formatted output. Structured programming concepts of sequence, selection, and repetition are applied.	The submission barely meets the criteria: The code and design provided use standard terminology, notations, and tools. The inputs and output variables are well-defined, with explicit requests for inputs and well-formatted output. Structured programming concepts of sequence, selection, and repetition are applied.	The submission almost fails to meet the criteria: The code and design provided use standard terminology, notations, and tools. The inputs and output variables are well-defined, with explicit requests for inputs and well-formatted output. Structured programming concepts of sequence, selection, and repetition are applied.	No submission OR The submission totally fails to meet the criteria: The code and design provided use standard terminology, notations, and tools. The inputs and output variables are well-defined, with explicit requests for inputs and well-formatted output. Structured programming concepts of sequence, selection, and repetition are applied.
Algorithmic Strategies	The submission fully meets the criteria: Algorithms and flowcharts provide a sequence of steps that indicate the flow of control and an ordered set of concrete steps that a machine could implement. There is no ambiguity in the steps; all steps are numbered and explained clearly. All notations and expressions used follow the appropriate standard.	The submission mostly meets the criteria: Algorithms and flowcharts provide a sequence of steps that indicate the flow of control and an ordered set of concrete steps that a machine could implement. There is no ambiguity in the steps; all steps are numbered and explained clearly. All notations and expressions used follow the appropriate standard.	The submission somewhat meets the criteria: Algorithms and flowcharts provide a sequence of steps that indicate the flow of control and an ordered set of concrete steps that a machine could implement. There is no ambiguity in the steps; all steps are numbered and explained clearly. All notations and expressions used follow the appropriate standard.	The submission barely meets the criteria: Algorithms and flowcharts provide a sequence of steps that indicate the flow of control and an ordered set of concrete steps that a machine could implement. There is no ambiguity in the steps; all steps are numbered and explained clearly. All notations and expressions used follow the appropriate standard.	The submission almost fails to meet the criteria: Algorithms and flowcharts provide a sequence of steps that indicate the flow of control and an ordered set of concrete steps that a machine could implement. There is no ambiguity in the steps; all steps are numbered and explained clearly. All notations and expressions used follow the appropriate standard.	No submission OR The submission totally fails to meet the criteria: Algorithms and flowcharts provide a sequence of steps that indicate the flow of control and an ordered set of concrete steps that a machine could implement. There is no ambiguity in the steps; all steps are numbered and explained clearly. All notations and expressions used follow the appropriate standard.
Comp Program Design	The submission fully meets the criteria: The code submitted is error-free and provides the required output. The code provided covers all exceptional cases and provides a well defined and consistence prompts to the users. The program is user friendly and provides clear use guidelines to the user. The submitted test cases validate all possible input values and offer appropriate messages to guide the user to use the program as intended.	The submission mostly meets the criteria: The code submitted is error-free and provides the required output. The code provided covers all exceptional cases and provides a well defined and consistence prompts to the users. The program is user friendly and provides clear use guidelines to the user. The submitted test cases validate all possible input values and offer appropriate messages to guide the user to use the program as intended.	The submission somewhat meets the criteria: The code submitted is error-free and provides the required output. The code provided covers all exceptional cases and provides a well defined and consistence prompts to the users. The program is user friendly and provides clear use guidelines to the user. The submitted test cases validate all possible input values and offer appropriate messages to guide the user to use the program as intended.	The submission barely meets the criteria: The code submitted is error-free and provides the required output. The code provided covers all exceptional cases and provides a well defined and consistence prompts to the users. The program is user friendly and provides clear use guidelines to the user. The submitted test cases validate all possible input values and offer appropriate messages to guide the user to use the program as intended.	The submission almost fails to meet the criteria: The code submitted is error-free and provides the required output. The code provided covers all exceptional cases and provides a well defined and consistence prompts to the users. The program is user friendly and provides clear use guidelines to the user. The submitted test cases validate all possible input values and offer appropriate messages to guide the user to use the program as intended.	No submission OR The submission totally fails to meet the criteria: The code submitted is error-free and provides the required output. The code provided covers all exceptional cases and provides a well defined and consistence prompts to the users. The program is user friendly and provides clear use guidelines to the user. The submitted test cases validate all possible input values and offer appropriate messages to guide the

						user to use the program as intended.
Quant Communication	The submission fully meets the criteria: The code submitted is well-formatted and documented. The code is readable and includes comments that describe all significant functionalities. The functions (if applicable) and variables use intuitive names that are easy to read and understand.	The submission mostly meets the criteria: The code submitted is well-formatted and documented. The code is readable and includes comments that describe all significant functionalities. The functions (if applicable) and variables use intuitive names that are easy to read and understand.	The submission somewhat meets the criteria: The code submitted is well-formatted and documented. The code is readable and includes comments that describe all significant functionalities. The functions (if applicable) and variables use intuitive names that are easy to read and understand.	The submission barely meets the criteria: The code submitted is well-formatted and documented. The code is readable and includes comments that describe all significant functionalities. The functions (if applicable) and variables use intuitive names that are easy to read and understand.	The submission almost fails to meet the criteria: The code submitted is well-formatted and documented. The code is readable and includes comments that describe all significant functionalities. The functions (if applicable) and variables use intuitive names that are easy to read and understand.	No submission OR The submission totally fails to meet the criteria: The code submitted is well-formatted and documented. The code is readable and includes comments that describe all significant functionalities. The functions (if applicable) and variables use intuitive names that are easy to read and understand.
Professional Work Product	The submission fully meets the criteria: Work is presented coherently, with appropriate consistent formatting and impeccable spelling and grammar. Formal academic language has been used, and appropriate subject specific terminology is used effectively. The assessment is original work produced by the student/s.	The submission mostly meets the criteria: Work is presented coherently, with appropriate consistent formatting and impeccable spelling and grammar. Formal academic language has been used, and appropriate subject specific terminology is used effectively. The assessment is original work produced by the student/s.	The submission somewhat meets the criteria: Work is presented coherently, with appropriate consistent formatting and impeccable spelling and grammar. Formal academic language has been used, and appropriate subject specific terminology is used effectively. The assessment is original work produced by the student/s.	The submission barely meets the criteria: Work is presented coherently, with appropriate consistent formatting and impeccable spelling and grammar. Formal academic language has been used, and appropriate subject specific terminology is used effectively. The assessment is original work produced by the student/s.	The submission almost fails to meet the criteria: Work is presented coherently, with appropriate consistent formatting and impeccable spelling and grammar. Formal academic language has been used, and appropriate subject specific terminology is used effectively. The assessment is original work produced by the student/s.	No submission OR The submission totally fails to meet the criteria: Work is presented coherently, with appropriate consistent formatting and impeccable spelling and grammar. Formal academic language has been used, and appropriate subject specific terminology is used effectively. The assessment is original work produced by the student/s.