L5 SOD A&B

PYTHON HOMEWORK

To be submitted on Monday

INSTRUCTIONS:

1. ***Write codes(python) on answer sheet.***
2. ***Choose 3 questions of your choice.***
3. ***Add comments where is needed.***
4. ***Ensure you use proper indentation as python require indentation***

Write a Python program to process student results with the following requirements:

- Store the student's name (string), age (integer), and class (string) in variables.
- Store marks for 5 subjects in a list, then convert the list to a set to remove duplicates.
- Use a dictionary to store all student information (name, age, class, marks).
- Calculate the student's average marks using arithmetic operators.
- Use an if-elif-else statement to assign a grade based on the average:
    - 90+ → "A"
    - 80-89 → "B"
    - 70-79 → "C"
    - Below 70 → "Fail"
- Use a for loop to print each subject's marks.
- Use **continue** to skip printing marks that are zero.

- Use break if any mark is above 100 (invalid mark).
- Include comments to explain each section of the program.

---

2. Simple Banking System
Write a Python program to manage customer bank transactions with the following features:

- Prompt the user to enter their name and initial balance, storing them in variables.
- Use a dictionary to store account information (name, balance, active status).
- Use assignment operators (+=, -=) to update the balance after deposits or withdrawals.
- Use comparison operators to check if the user has sufficient funds for withdrawals.
- Use logical operators to verify:
    - The account is active.
    - The balance is non-negative.
- Use a while loop to repeatedly ask the user to choose an action: deposit, withdraw, or exit.
- Use break when the user selects "exit".
- Use pass for invalid menu choices.
- Display the account details and transaction results clearly.

---

3. Online Shopping Cart System
Write a Python program for an online shopping cart with the following functionality:

- Store available products in a list and their prices in a dictionary.
- Use a set to store unique items selected by the user.
- Use a while loop to repeatedly prompt the user to enter product names until they type "done".

- Use the in operator to check if the entered product exists in the available products list.
- Use continue to skip unavailable products.
- Use break to exit the loop when the user types "done".
- Calculate the total cost using arithmetic operators.
- Apply a 10% discount if the total is $100 or more; otherwise, no discount.
- Include comments to explain each part of the code.

---

4. Employee Attendance & Salary System
Write a Python program to manage employee attendance and salary with these features:

- Store the employee's name (string), hourly rate (float), and hours worked (integer).
- Use a tuple to store fixed working days of the week.
- Use a for loop to display each working day.
- Use logical operators to check if the employee completed full attendance (e.g., worked all days).
- Use if-elif-else to categorize performance based on hours worked:
    - 40+ hours → "Excellent"
    - 30-39 hours → "Good"
    - Below 30 hours → "Needs Improvement"
- Calculate salary using arithmetic operators, including exponentiation, modulus, or floor division at least once.
- Use pass as a placeholder for a future "bonus calculation" function.
- Include comments for each important step.

---

5. Hospital Patient Management System
Write a Python program to manage patient details and symptoms with these requirements:

- Store the patient’s name (string), age (integer), weight (float), and temperature (float).
- Use a dictionary to store patient details.
- Use a set to store unique symptoms entered by the user.
- Use a for loop to display all symptoms.
- Use membership operators (in, not in) to check for specific symptoms like "fever".
- Use if-elif-else to classify the patient’s condition based on temperature:
    - ≥39°C → "Critical"
    - 37-38.9°C → "Needs Attention"
    - <37°C → "Normal"
- Use a while loop to accept symptoms until the user types "stop".
- Use continue to skip empty entries.
- Use break to exit the loop when "stop" is entered.
- Include comments to explain each code block.