كلية التقنيات الطبية والصحية
قسـم الانـظـمـة الـطبية الـذكـيـة

Lecture: (4)

Example "If conditions & Loops" in python

Subject: Object oriented programming II
Class: Second
Lecturer: Dr. Maytham N. Meqdad

Q\ Design a program to check a patient's condition

Ask the user to input the patient's temperature

```python
temperature = float(input("Enter the patient's temperature: "))

if temperature > 37.5:
    print("The patient has a fever.")
    print("They may need medical advice.")
else:
    print("The patient does not have a fever.")
    print("They seem to be in good condition.")
```

Q\ Program to check eligibility for voting

```python
# Ask the user to input their age
age = int(input("Enter your age: "))

if age >= 18:
    print("You are eligible to vote.")
else:
    print("You are not eligible to vote yet.")
```

Q\Program to check blood pressure

```python
# Ask the user to input their systolic and diastolic blood pressure
systolic_bp = int(input("Enter the systolic blood pressure: "))
diastolic_bp = int(input("Enter the diastolic blood pressure: "))

if systolic_bp <= 120 and diastolic_bp <= 80:
    print("Your blood pressure is within the normal range.")
    print("You are in good health.")
elif systolic_bp <= 130 and diastolic_bp <= 85:
    print("Your blood pressure is slightly elevated.")
    print("Consider making some lifestyle changes.")
else:
    print("Your blood pressure is high.")
    print("You should consult a healthcare professional.")
```

```
Q\Program to calculate and assess BMI

# Ask the user to input their weight (in kilograms) and height (in meters)
weight = float(input("Enter your weight (in kilograms): "))
height = float(input("Enter your height (in meters): "))

# Calculate BMI
bmi = weight / (height ** 2)

print("Your BMI is:", bmi)

if bmi < 18.5:
    print("You are underweight.")
    print("Consider gaining some weight.")
elif 18.5 <= bmi < 25:
    print("Your weight is normal.")
    print("You are in good health.")
elif 25 <= bmi < 30:
    print("You are overweight.")
    print("Consider losing some weight for better health.")
else:
    print("You are obese.")
    print("Consult a healthcare professional for guidance.")
```

- In this program, the user is asked to input their weight and height. The program calculates the BMI and then provides an assessment based on the BMI value. This can help in evaluating a patient's weight status.

```
Q\ Program to track weight loss progress

# Ask the user to input their initial weight (in kilograms) and target weight
initial_weight = float(input("Enter your initial weight (in kilograms): "))
target_weight = float(input("Enter your target weight (in kilograms): "))

# Set up a loop to simulate a weight loss journey
current_weight = initial_weight
days = 0

while current_weight > target_weight:
    current_weight -= 0.1  # Assuming a daily weight loss of 0.1 kg
    days += 1

print("It took", days, "days to reach your target weight.")
print("Congratulations on your weight loss journey!")
```
- In this program, the user provides their initial weight and a target weight. The program then simulates a weight loss journey using a `while` loop, assuming a daily weight loss of 0.1 kg. It calculates how many days it takes to reach the target weight and congratulates the user on their weight loss journey.

```
Q \ Program to monitor heart rate

# Set up a loop to monitor the patient's heart rate
heart_rate = 80  # Initial heart rate
minutes = 0

while heart_rate < 100:
    minutes += 1
    heart_rate += 10  # Simulating an increase in heart rate

print("After", minutes, "minutes, the patient's heart rate reached a critical
level.")
print("Medical attention is required.")
```

- In this program, the patient's initial heart rate is set to 80, and the program simulates monitoring the heart rate with a `while` loop. It increases the heart rate by 10 beats per minute until it reaches a critical level (100 beats per minute). The program then alerts that medical attention is required.

```
Q\ Program to track medication schedule

# List of medications and their dosages
medications = [
    ("Aspirin", 1),
    ("Antibiotic", 2),
    ("Painkiller", 1),
    ("Vitamin C", 1),
]

# Simulate a 7-day medication schedule
days = 7

for day in range(1, days + 1):
    print("Day", day)
    for med, dosage in medications:
        print(f"Take {dosage} {med}(s)")
    print("")

print("Medication schedule completed. You are on the road to recovery!")
```

- In this program, we have a list of medications and their dosages. The program uses a `for` loop to simulate a 7-day medication schedule, printing the medications and dosages for each day. At the end of the schedule, it indicates that the medication schedule is completed.

```
Q\ Initialize an empty list to store medications
medications = []

# Ask the user to input medication details
num_medications = int(input("How many medications are you taking? "))
for i in range(num_medications):
    med_name = input(f"Enter the name of medication {i + 1}: ")
    med_dosage = input(f"Enter the dosage for {med_name}: ")
    medications.append((med_name, med_dosage))

# Simulate a 7-day medication schedule
days = 7

for day in range(1, days + 1):
    print("Day", day)
    for med, dosage in medications:
        print(f"Take {dosage} {med}(s)")
    print("")

print("Medication schedule completed. Wishing you a quick recovery!")
```

In this program, the user is asked to input the details of their medications, including the medication name and dosage. Then, the program uses a `for` loop to simulate a 7-day medication schedule, printing the medications and dosages for each day. At the end of the schedule, it indicates that the medication schedule is completed.

```
Q\ Program to analyze medical lab results with user input

# Initialize empty lists to store test results
test_names = []
test_results = []

# Ask the user to input test names and results
num_tests = int(input("How many medical tests were conducted? "))
for i in range(num_tests):
    test_name = input(f"Enter the name of test {i + 1}: ")
    test_result = input(f"Enter the test result for {test_name}: ")
    test_names.append(test_name)
    test_results.append(test_result)

# Analyze and report on the test results
print("\nMedical Test Results:")
for i in range(num_tests):
    print(f"Test: {test_names[i]}, Result: {test_results[i]}")

# Check for abnormal results
abnormal_tests = []
for i in range(num_tests):
    if test_results[i].lower() == "abnormal":
        abnormal_tests.append(test_names[i])
```

```
if abnormal_tests:
    print("\nAbnormal Test Results:")
    for test in abnormal_tests:
        print(f"{test} - Abnormal result. Further evaluation may be
required.")
else:
    print("\n please return to doctor .")
```

- In this program, the user is asked to input the names and results of medical tests. The program stores this data in lists and then analyzes and reports on the test results. It also checks for abnormal results and suggests further evaluation when needed.

-