Week 1

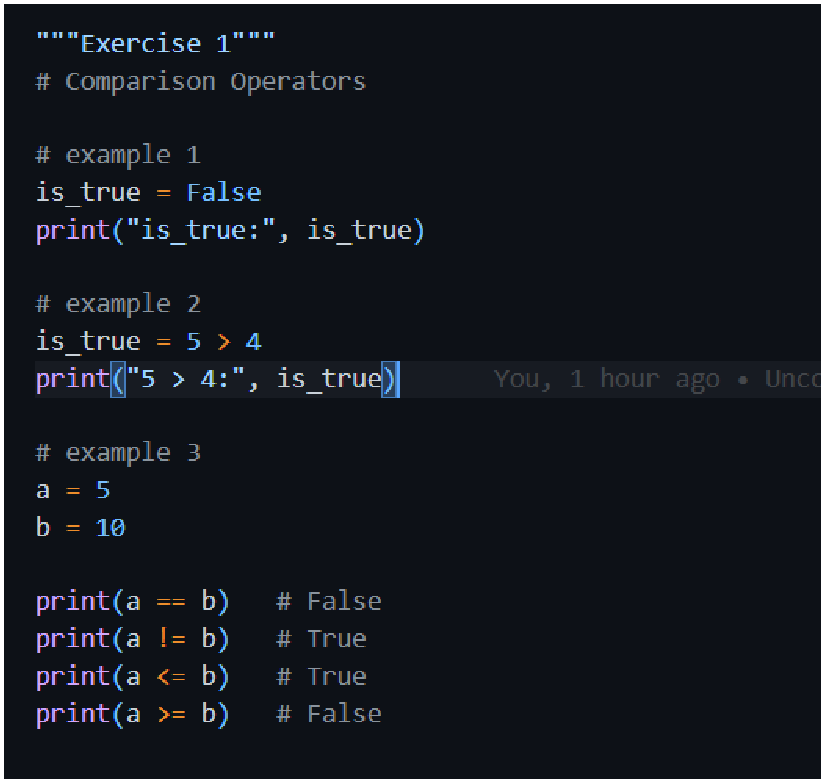
Section 1. Comparisons and Conditionals

Exercise 1: Comparison Operators

Understanding the Task:

This exercise asked us to explore and practice **comparison operators**. We were expected to check how values or variables compare using operators like **==, !=, >, <, >=,** and **<=.** The main goal was to understand how these comparisons return either True or False.

Source Code:



```
"""Exercise 1"""
# Comparison Operators

# example 1
is_true = False
print("is_true:", is_true)

# example 2
is_true = 5 > 4
print("5 > 4:", is_true)        You, 1 hour ago • Unco

# example 3
a = 5
b = 10

print(a == b)   # False
print(a != b)   # True
print(a <= b)   # True
print(a >= b)   # False
```

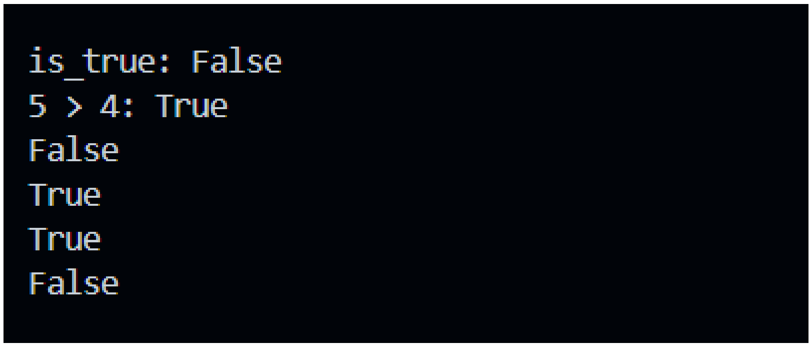
Explanation

- In **Example 1**, I directly assigned a Boolean value ‘False’ to a variable and printed it.
- In **Example 2**, I used a comparison 5 > 4 which is True, and printed the result.
- In **Example 3**, I compared two variables a = 5 and b = 10 using all standard comparison operators to see which conditions are true.

Time and Space Complexity

- **Time Complexity**: O(1) – All operations are constant-time comparisons.
- **Space Complexity**: O(1) – Only a few variables are stored in memory.

Output



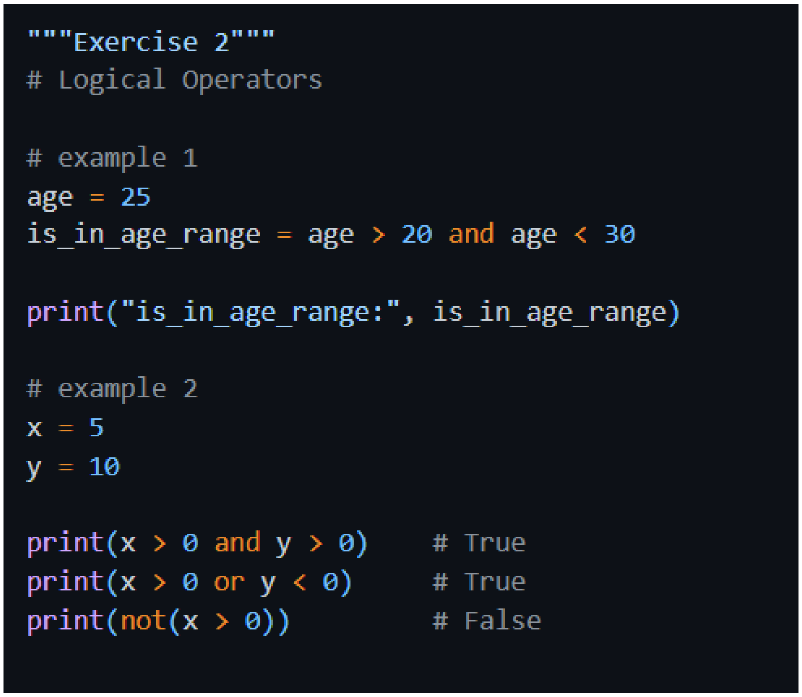
```
is_true: False
5 > 4: True
False
True
True
False
```

Exercise 2: Logical Operators

Understanding the Task

In this exercise, I was supposed to work with **logical operators**. These include **and**, **or**, and **not**. I needed to check how they behave when used with different conditions and see what kind of Boolean result they give, either True or False. The goal was to understand how we can combine multiple conditions using logic.

Source Code

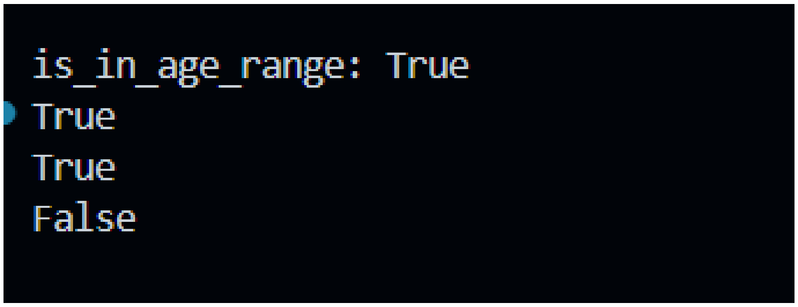


```python
"""Exercise 2"""
# Logical Operators

# example 1
age = 25
is_in_age_range = age > 20 and age < 30

print("is_in_age_range:", is_in_age_range)

# example 2
x = 5
y = 10

print(x > 0 and y > 0)    # True
print(x > 0 or y < 0)     # True
print(not(x > 0))         # False
```

Explanation

- In the first example, I checked if the age falls between 20 and 30. Since 25 is in that range, it returns True.
- In the second part, I used two variables x and y and tested them using different logical operators:
  - x > 0 and y > 0 checks if both values are positive — which they are.
  - x > 0 or y < 0 checks if **at least one** condition is true — which is also true.
  - not (x > 0) flips the result of x > 0. Since x is greater than 0, the original condition is True and not makes it False.

Output



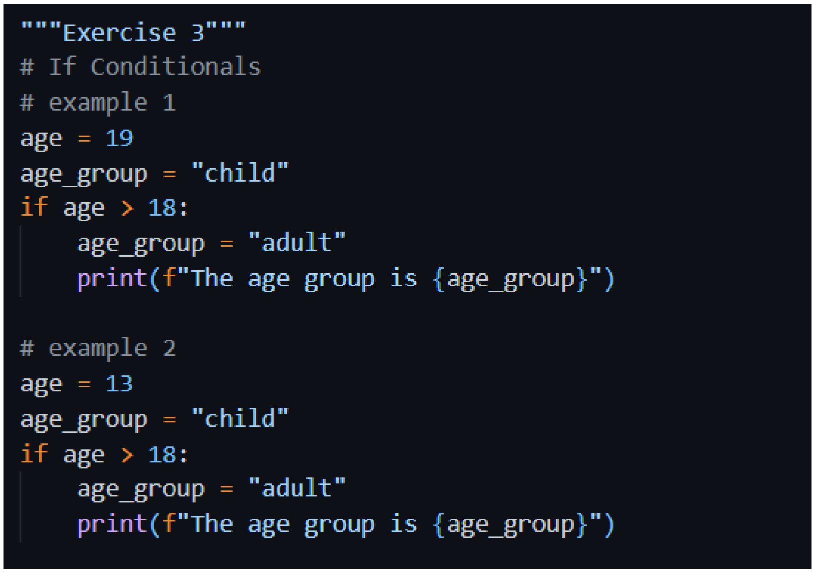
```
is_in_age_range: True
True
True
False
```

Exercise 3: if – Conditionals

Understanding the Task

In this task, I explored the basic use of **if statements** in Python. I had to check if a certain condition is true and then update a variable or display a message accordingly. The main idea was to learn how decisions are made in a program based on conditions.

Source Code



```python
"""Exercise 3"""
# If Conditionals
# example 1
age = 19
age_group = "child"
if age > 18:
    age_group = "adult"
    print(f"The age group is {age_group}")

# example 2
age = 13
age_group = "child"
if age > 18:
    age_group = "adult"
    print(f"The age group is {age_group}")
```

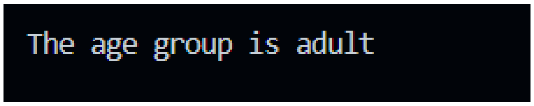
Explanation

- In the first example, the age was 19, which is greater than 18, so it updated age group from "child" to "adult" and printed it.
- In the second example, the age was 13, so the condition age > 18 was false and nothing was printed — the program skipped the if block.

Time Complexity and Space Complexity

- **Time Complexity**: O(1) – A single condition is checked.
- **Space Complexity**: O(1) – Only a few variables are used.

Output



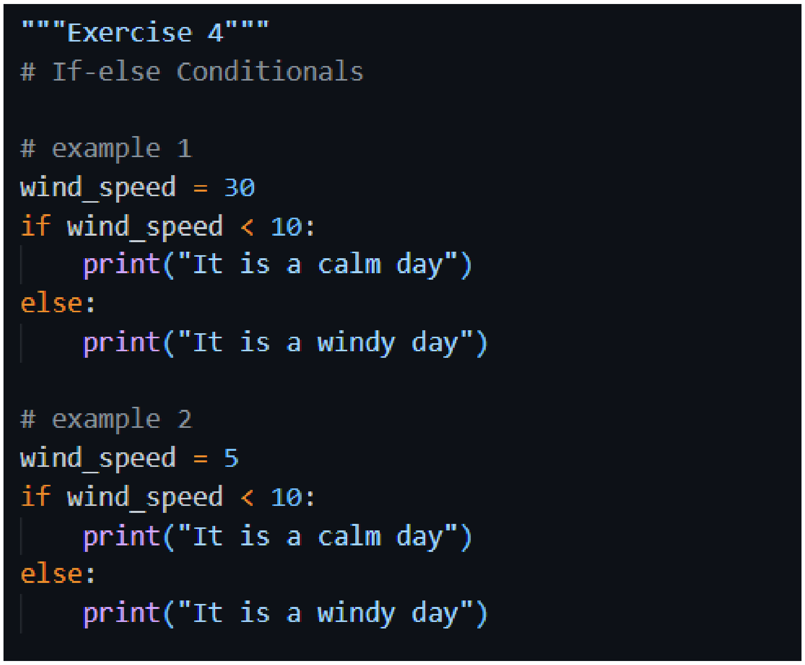
```
The age group is adult
```

Exercise 4: if – else Conditionals

Understanding the Task

This task was about using the **if-else** structure. I had to write code where the program chooses between two possible actions, one if a condition is true, and another if it's false.

Source Code



```
"""Exercise 4"""
# If-else Conditionals

# example 1
wind_speed = 30
if wind_speed < 10:
    print("It is a calm day")
else:
    print("It is a windy day")

# example 2
wind_speed = 5
if wind_speed < 10:
    print("It is a calm day")
else:
    print("It is a windy day")
```

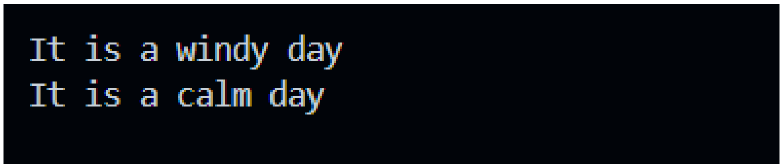
Explanation

- In the first example, wind_speed was 30, which is not less than 10, so it printed "It is a windy day".
- In the second example, windspeed was 5, which is less than 10, so it printed "It is a calm day".

Time Complexity and Space Complexity

- **Time Complexity**: O(1) – It just checks one condition.
- **Space Complexity**: O(1) – Minimum memory usage.

Output



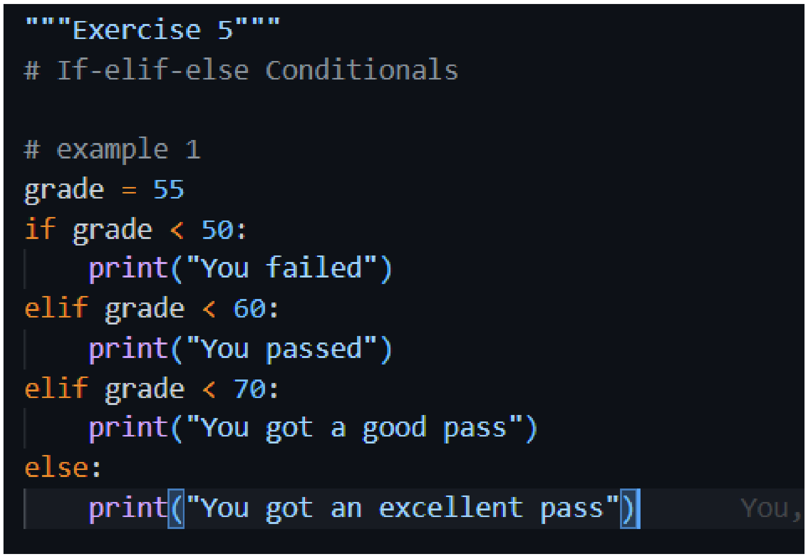
```
It is a windy day
It is a calm day
```

Exercise 5: if – elif - else Conditionals

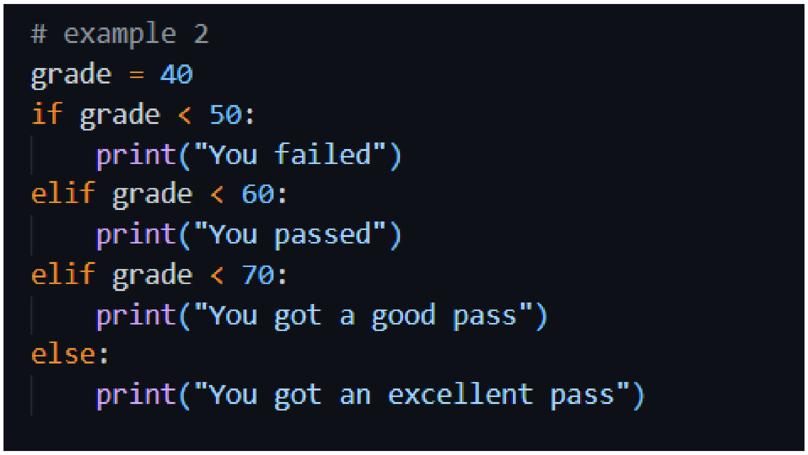
Understanding the Task

In this exercise, I practiced using **if-Elif-else** blocks to handle multiple conditions in a clean way. This helped in writing better decision-based logic where the program chooses the right option from several possibilities.

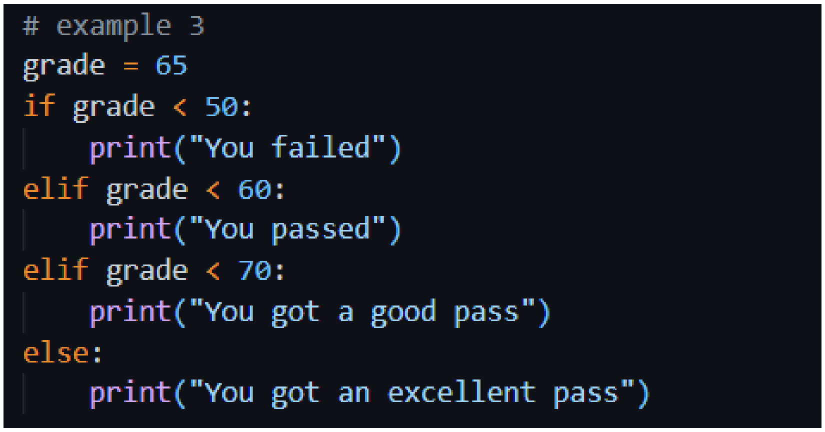
Source Code



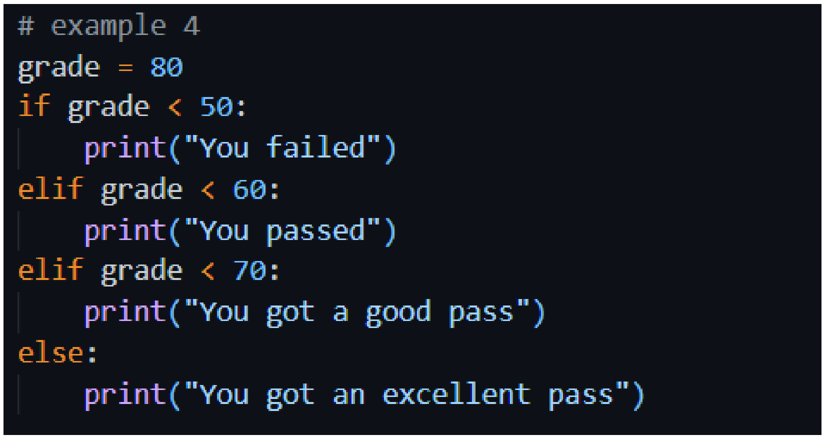
```
"""Exercise 5"""
# If-elif-else Conditionals

# example 1
grade = 55
if grade < 50:
    print("You failed")
elif grade < 60:
    print("You passed")
elif grade < 70:
    print("You got a good pass")
else:
    print("You got an excellent pass")      You,
```



```
# example 2
grade = 40
if grade < 50:
    print("You failed")
elif grade < 60:
    print("You passed")
elif grade < 70:
    print("You got a good pass")
else:
    print("You got an excellent pass")
```



```
# example 3
grade = 65
if grade < 50:
    print("You failed")
elif grade < 60:
    print("You passed")
elif grade < 70:
    print("You got a good pass")
else:
    print("You got an excellent pass")
```



```python
# example 4
grade = 80
if grade < 50:
    print("You failed")
elif grade < 60:
    print("You passed")
elif grade < 70:
    print("You got a good pass")
else:
    print("You got an excellent pass")
```

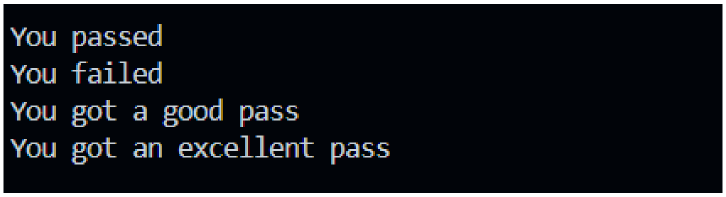
Explanation

- Four different grades were tested.
- The code checked each grade and printed a message:
    - Below 50 → "You failed"
    - Between 50–59 → "You passed"
    - Between 60–69 → "You got a good pass"
    - 70 and above → "You got an excellent pass"
- Each elif allows checking in sequence, and only one block runs depending on the value.

Time Complexity and Space Complexity

- **Time Complexity**: O(1) – Constant time, since only a few checks are made.
- **Space Complexity**: O(1) – Few variables used.

Output



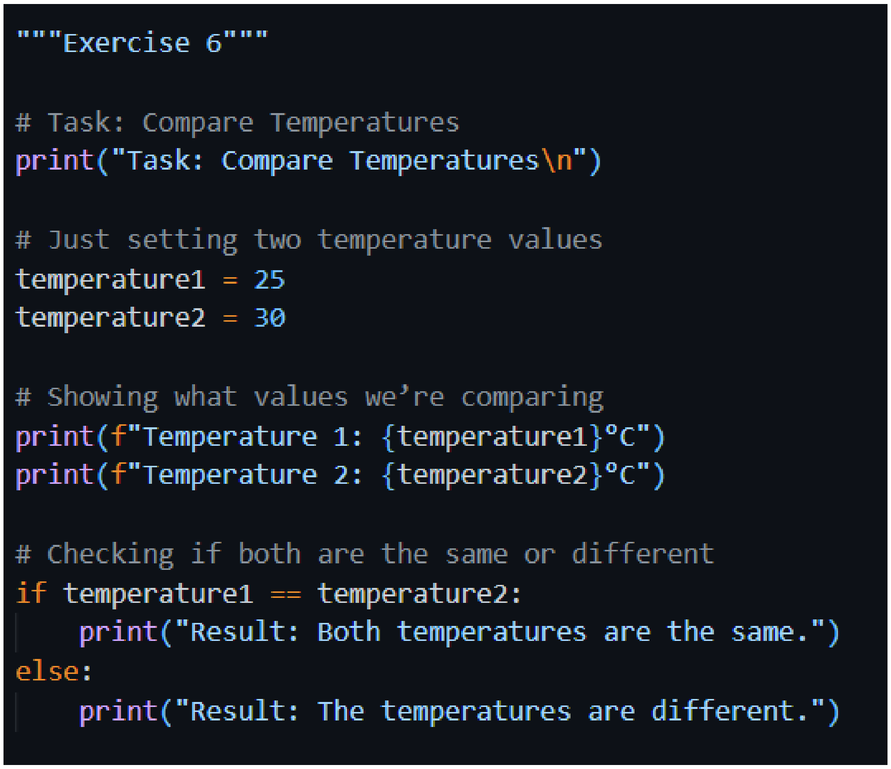
```
You passed
You failed
You got a good pass
You got an excellent pass
```

Exercise 6: Compare Temperatures

Understanding the Task

This task was about comparing two temperature values. I had to check if they were equal or different and show the appropriate message using an if-else structure.

Source Code



```python
"""Exercise 6"""

# Task: Compare Temperatures
print("Task: Compare Temperatures\n")

# Just setting two temperature values
temperature1 = 25
temperature2 = 30

# Showing what values we’re comparing
print(f"Temperature 1: {temperature1}°C")
print(f"Temperature 2: {temperature2}°C")

# Checking if both are the same or different
if temperature1 == temperature2:
    print("Result: Both temperatures are the same.")
else:
    print("Result: The temperatures are different.")
```

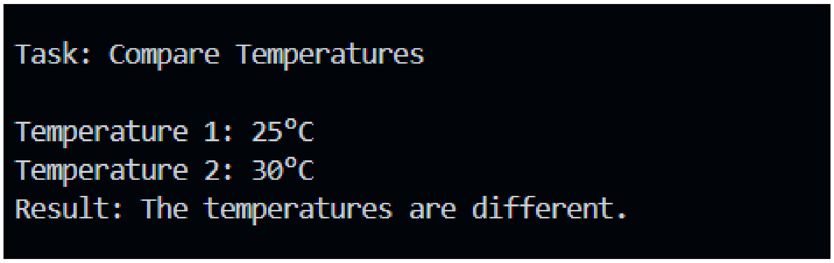
Explanation

- The two temperatures were set to 25 and 30.
- The program displayed both values.
- Then it compared them: since 25 ≠ 30, it printed “The temperatures are different”.

Time Complexity and Space Complexity

- **Time Complexity**: O(1) – One comparison is made.
- **Space Complexity:** O(1) – Two variables only.

Output



```
Task: Compare Temperatures

Temperature 1: 25°C
Temperature 2: 30°C
Result: The temperatures are different.
```