Lab Manual: Conditional Statements in Python

Lab Objective:

By the end of this lab, students will:

1. Understand the concept of **conditional statements** in Python.
2. Learn how to use `if`, `else`, and `elif` statements to make decisions based on conditions.
3. Explore the use of logical operators (`and`, `or`, `not`) in conditional expressions.
4. Practice nesting conditionals and using them in combination with loops.

Part 1: Introduction to Conditional Statements

Objective:

- Learn how to use `if`, `elif`, and `else` statements to execute blocks of code based on specific conditions.

Tasks:

1. **Basic `if` Statement:**
   - Use the `if` statement to execute a block of code when a condition is `True`.

   Code Example:

   ```
   num = 10

   if num > 5:
       print(f"{num} is greater than 5")
   ```

2. **`if-else` Statement:**
   - Use the `else` block to handle the case when the condition in the `if` statement is `False`.

   Code Example:

   ```
   num = 3

   if num > 5:
       print(f"{num} is greater than 5")
   else:
       print(f"{num} is less than or equal to 5")
   ```

3. **`if-elif-else` Statement:**
   - Use `elif` to check multiple conditions, and `else` to handle cases that don’t meet any of the specified conditions.

Code Example:

```
num = 7

if num > 10:
    print(f"{num} is greater than 10")
elif num == 7:
    print(f"{num} is equal to 7")
else:
    print(f"{num} is less than or equal to 10 but not 7")
```

Exercise 1:

- Write a Python program that checks whether a number is **positive**, **negative**, or **zero** using an `if-elif-else` structure. Print an appropriate message for each case.

Part 2: Logical Operators with Conditionals

Objective:

- Learn how to use logical operators (`and`, `or`, `not`) to combine multiple conditions in an `if` statement.

Tasks:

1. **Using `and` Operator:**
   - Combine two conditions using the `and` operator, which returns `True` only if both conditions are `True`.

   Code Example:

   ```
   python
   Copy code
   age = 25
   income = 50000

   if age > 18 and income > 30000:
       print("You are eligible for a loan.")
   ```

2. **Using `or` Operator:**
   - Combine two conditions using the `or` operator, which returns `True` if either condition is `True`.

   Code Example:

   ```
   age = 16
   has_parent_permission = True

   if age >= 18 or has_parent_permission:
   ```

```
    print("You can attend the event.")
else:
    print("You cannot attend the event.")
```

3. **Using `not` Operator:**
   - Use the `not` operator to invert the result of a condition.

   Code Example:

   ```
   is_member = False

   if not is_member:
       print("You need to become a member to access this area.")
   ```

Exercise 2:

- Write a Python program that checks if a person is eligible to vote. A person is eligible if they are **18 years or older** and are a **citizen** of the country.

Part 3: Nested Conditionals

Objective:

- Learn how to nest `if` statements to check multiple conditions in a more complex decision-making process.

Tasks:

1. **Using Nested `if` Statements:**
   - Write nested `if` statements to evaluate multiple conditions within a parent condition.

   Code Example:

   ```
   num = 15

   if num > 10:
       print(f"{num} is greater than 10")
       if num % 2 == 0:
           print(f"{num} is also even")
       else:
           print(f"{num} is odd")
   ```

Exercise 3:

- Write a Python program that asks the user for their **age** and **membership status**. If the person is 18 or older and a member, print "Access granted." If they are under 18, print "Access denied." If they are 18 or older but not a member, print "Membership required."

Part 4: Combining Conditionals with Loops

Objective:

- Combine conditionals with loops to control the flow of execution in iterative processes.

Tasks:

1. **Using Conditionals in a Loop:**
    - Use a `for` loop to iterate over a list of numbers and print whether each number is **even** or **odd**.

    Code Example:

    ```
    numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9]

    for num in numbers:
        if num % 2 == 0:
            print(f"{num} is even")
        else:
            print(f"{num} is odd")
    ```

2. **Using `break` and `continue` in Conditionals:**
    - Use `break` to exit a loop when a condition is met and `continue` to skip the current iteration of the loop.

    Code Example:

    ```
    for num in numbers:
        if num == 5:
            print("Number 5 found, stopping the loop.")
            break
        elif num % 2 == 0:
            print(f"{num} is even, continuing the loop.")
            continue
        print(f"{num} is odd")
    ```

Exercise 4:

- Create a Python program that checks for **prime numbers** between 1 and 20 using a loop. Print each prime number along with a message that says "Prime number found: X". Use conditionals within the loop to check if a number is prime.

Part 5: Ternary Conditional Operator

Objective:

- Learn how to use the ternary conditional operator for concise conditional expressions.

Tasks:

1. **Using the Ternary Operator:**
    - Use a ternary operator for a quick, one-line `if-else` expression.

    Code Example:

    ```
    age = 20
    message = "Eligible to vote" if age >= 18 else "Not eligible to vote"
    print(message)
    ```

Exercise 5:

- Write a Python script that checks if a number is **even** or **odd** using a **ternary operator** and prints the result.

Assignment:

1. **Problem 1:**
    - Write a Python program that asks the user to input their **exam score** (between 0 and 100). Use an `if-elif-else` structure to print the grade based on the score:
        - `90-100`: Grade A
        - `80-89`: Grade B
        - `70-79`: Grade C
        - `60-69`: Grade D
        - Below `60`: Grade F
2. **Problem 2:**
    - Create a Python program that asks the user to input three numbers. Use conditional statements to find and print the **largest** of the three numbers.
3. **Problem 3:**

   - Write a Python program that checks whether a **year** is a **leap year**. A year is a leap year if:
      - It is divisible by 4.
      - If it is divisible by 100, it must also be divisible by 400.
4. **Problem 4:**
   - Write a Python program that simulates a **basic calculator**. The program should ask the user for two numbers and an operator (`+`, `-`, `*`, `/`). Use `if-elif-else` statements to perform the appropriate calculation and print the result. Handle cases where the user tries to divide by zero.
5. **Problem 5:**
   - Create a Python program that asks the user for their **age** and checks if they are eligible for a **driver's license**. If the user is **under 18**, print "Not eligible". If the user is **18 or older** but **less than 21**, print "Eligible for a provisional license". If the user is **21 or older**, print "Eligible for a full license".