**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  
**Topic 12 - Testing Sets of Conditions**  
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Introduction**

In programming, many decisions require evaluating multiple conditions together. Python provides logical operators like and and or to test combined conditions within if statements, allowing us to manage more complex decision-making scenarios efficiently.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**What is Testing Sets of Conditions?**

Testing sets of conditions means evaluating multiple conditions in one if statement to determine the outcome. This approach uses logical operators:

* **and**: Requires all conditions to be true.
* **or**: Requires only one condition to be true.

**Examples of Testing Sets of Conditions:**

*# Example with `and`*

*if weight > 300 and time < 6:*

*status = "try to recruit him"*

*# Example with `or`*

*if SAT > avg or GPA > 2.5 or parent == "alum":*

*message = "Welcome to Leeds College!"*

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Why Use Combined Condition Testing?**

* **Efficiency**: Evaluates multiple conditions in a single statement.
* **Control**: Offers precise control over when code executes by ensuring all or some conditions are met.
* **Readability**: Simplifies complex decision-making logic, making the code easier to understand and maintain.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**How to Test Sets of Conditions in Python**

1. **Using and for Combined Condition Testing**  
   Use and when all specified conditions need to be true for the statement to execute:

*if weight > 300 and time < 6:*

*status = "try to recruit him"*

* + In this example, both weight > 300 and time < 6 must be true to proceed.

1. **Using or for Combined Condition Testing**  
   Use or when any one of the conditions being true will execute the statement:

*if SAT > avg or GPA > 2.5 or parent == "alum":*

*message = "Welcome to Leeds College!"*

* + Here, only one of these conditions needs to be true for message to display.

1. **Combining and and or Conditions with Parentheses**  
   When combining and and or in one if statement, use parentheses to eliminate ambiguity. For example:

*# Example 1: Both conditions in parentheses need to be true for a pass*

*if (age > 65 or age < 21) and res == "U.K.":*

*pass\_status = "Eligible"*

*# Example 2: Either condition outside or inside parentheses needs to be true*

*if age > 65 or (age < 21 and res == "U.K."):*

*pass\_status = "Eligible"*

* + **Explanation**: Parentheses clarify how conditions are grouped, helping Python and readers interpret the logic accurately.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Conclusion**

Testing sets of conditions is a powerful way to manage complex decisions in Python. By understanding how to use and, or, and parentheses, developers can write code that handles multi-condition scenarios accurately and efficiently, keeping logic clean and readable.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**