**Week 6 – Notes from the Lecture**

What is a Boolean Expression and Why is It Relevant for Conditionals?

A Boolean expression is an expression that evaluates to either True or False. In Python, Boolean expressions often involve comparison operators (like `==`, `>`, `<`, `!=`) or logical operators (`and`, `or`, `not`).

Relevance for Conditionals:

- Boolean expressions are essential for conditionals because they determine the flow of control in the program. In an `if`, `elif`, or `else` statement, the condition is a Boolean expression. If the expression evaluates to `True`, the code inside the conditional block is executed; otherwise, it's skipped, or another condition is checked.

Example:

age = 18

if age >= 18: # Boolean expression: True

print("You are an adult.") # This will be executed

In this case, `age >= 18` is a Boolean expression that evaluates to `True`, so the block of code inside the `if` statement runs.

**Difference Between Chained Conditionals and Nested Conditionals**

1. Chained Conditionals:

- Chained conditionals involve multiple `if`, `elif`, and `else` statements used in sequence. Each condition is checked one after the other until one evaluates to `True`, after which its associated block is executed, and the rest of the chain is ignored.

- Example:

if score >= 90:

print("A")

elif score >= 80:

print("B")

else:

print("C")

2. Nested Conditionals:

- Nested conditionals involve placing one conditional inside another. That is, within the block of an `if`, `elif`, or `else` statement, there can be another `if` statement.

- Example:

if score >= 90:

if attendance >= 80:

print("A")

else:

print("A but poor attendance")

else:

print("B or lower")

Difference:

- Chained conditionals check multiple independent conditions in sequence.

- Nested conditionals allow for more complex logic by having an `if` statement inside another, creating a hierarchical flow of decision-making.

**What Does the Term "Branch" Refer to in the Context of Conditionals?**

In the context of conditionals, a branch refers to one of the possible paths that a program can take based on the evaluation of a Boolean expression in a conditional statement (`if`, `elif`, `else`).

* If branch: The path taken when the `if` condition is `True`.
* Else branch: The path taken when none of the previous conditions are `True`.
* Elif branch: An additional branch checked if the previous `if` or `elif` conditions were `False`.

Example:

if condition\_1: # Branch 1

# Code for branch 1

elif condition\_2: # Branch 2

# Code for branch 2

else: # Branch 3

# Code for branch 3

Each `if`, `elif`, or `else` statement represents a different branch of the control flow, depending on which condition is met.