Class 3 — Control Flow, Branching, and Conditions

These notes summarize the key ideas from Class 3. They are intended as a reference for understanding how Python programs control execution using if statements and conditions.

1. Straight-Line Execution

Before control flow, Python executes code from top to bottom, one line at a time. Every line runs exactly once unless a control structure changes that behavior.

```
x = 5
y = x + 3
print(y)
```

2. Introducing Control Flow

Control flow allows a program to decide which lines run and which lines are skipped. This introduces decision-making into program execution.

```
if x > 0:
    print("positive")
```

3. Core Rule of if Statements

Code inside an if block runs only if the condition is truthy. Python evaluates the condition and uses its truthiness to decide whether to execute the block.

4. Truthiness (Primitive Types)

Every value in Python has a truthiness that determines how it behaves in an if statement. In this class, we consider only primitive types.

Numbers: 0 and 0.0 are falsey; all other numbers are truthy.

Strings: "" is falsey; any non-empty string is truthy.

None: represents no value and is always falsey.

5. Comparison Operators

Comparison operators compare two values and produce a boolean result.

```
== equal to
!= not equal to
< less than
> greater than
<= less than or equal to
>= greater than or equal to
```

6. Logical Operators

Logical operators work with boolean expressions and allow conditions to be combined.

not — flips truthiness

and — both conditions must be true

or — at least one condition must be true

7. if / elif / else

An if / elif / else chain allows multiple conditions to be tested in order. Exactly one branch will execute.

```
if x > 0:
    print("positive")
elif x < 0:
    print("negative")
else:
    print("zero")
```

8. Key Takeaways

1. Straight-line code executes top to bottom.
2. Control flow introduces decision-making.
3. if statements depend on truthiness.
4. Truthiness depends on type and value.
5. Comparison operators produce booleans.
6. Logical operators combine conditions.