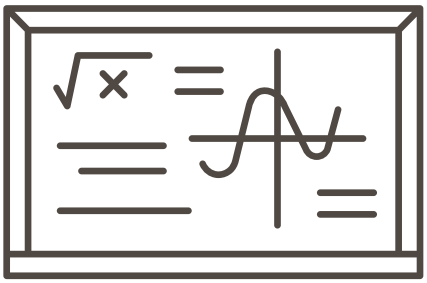
Notes for CSE101 - Programming Fundamentals (Lecture 2)

Mathematical Operator and Comparison Operator



In programming, operators are symbols that perform operations on variables and values. These operations range from basic arithmetic to logical comparisons. Think of operators as the "verbs" in programming—they perform the action! Let's dive into the various types of operators and how they are used.

1. Arithmetic Operators

Arithmetic operators are used to perform mathematical operations such as addition, subtraction, multiplication, and more.

Basic Arithmetic Operators in Python:

1. **+** (Addition) – Adds two values.
    - **Example**: 3 + 2 equals 5
2. **-** (Subtraction) – Subtracts one value from another.
    - **Example**: 5 - 2 equals 3
3. ***** (Multiplication) – Multiplies two values.
    - **Example**: 4 * 2 equals 8
4. **/** (Division) – Divides one value by another.
    - **Example**: 10 / 2 equals 5.0 (Note: The result is always a float in Python.)
5. **%** (Modulus) – Returns the remainder of a division.
    - **Example**: 10 % 3 equals 1 (because 10 divided by 3 leaves a remainder of 1)
6. ****** (Exponentiation) – Raises one number to the power of another.
    - **Example**: 2 ** 3 equals 8 (2 raised to the power of 3)
7. **//** (Floor Division) – Divides and returns the integer part (removes the decimal).
    - **`Example: 10 // 3 equals 3`**

Real-Life Example:
Think of arithmetic operators like basic math you perform in everyday life. If you have 5 apples and you give 2 away, you can use the subtraction operator to find out how many you have left: `5 - 2 = 3 .`

2. **Relational (Comparison) Operators**

Relational operators compare two values or variables and return a Boolean (True or False). They are often used in conditional statements like if statements.

Relational Operators in Python:

1. **==** (Equal to) – Checks if two values are equal.
   - **`Example: 5 == 5 returns True`**
2. **!=** (Not equal to) – Checks if two values are not equal.
   - **`Example: 5 != 4 returns True`**
3. **>** (Greater than) – Checks if the left value is greater than the right.
   - **`Example: 6 > 3 returns True`**
4. **<** (Less than) – Checks if the left value is less than the right.
   - **`Example: 4 < 6 returns True`**
5. **>=** (Greater than or equal to) – Checks if the left value is greater than or equal to the right.
   - **`Example: 5 >= 5 returns True`**
6. **<=** (Less than or equal to) – Checks if the left value is less than or equal to the right.
   - **`Example: 3 <= 4 returns True`**

Real-Life Example:

Imagine you're shopping and comparing the prices of two items. If the price of item A is 200 and item B is 150 , you could use a relational operator to check which one is more expensive: 200 > 150 returns True .

3. Logical Operators

Logical operators are used to combine multiple conditions and return either True or False based on the evaluation of those conditions. They are most often used with if statements to control the flow of a program.

Logical Operators in Python:

1. and – Returns True if both conditions are True .
   - **Example: (5 > 3) and (4 == 4) returns True because both conditions are true.**
2. or – Returns True if at least one condition is True .
   - **Example: (5 < 3) or (4 == 4) returns True because the second condition is true.**
3. not – Reverses the result; returns False if the condition is True , and vice versa.
   - **Example: not (5 > 3) returns False because 5 > 3 is True , and not flips it to False .**

Real-Life Example:
Imagine you're deciding whether to go out for a walk. You might check two conditions:

- Is it sunny outside?
- Do you have free time?

If both conditions are true (sunny and free_time), you go for a walk (just like an and operator).

4. Assignment Operators

Assignment operators are used to assign values to variables. The most basic one is = (equal), but Python has several other assignment operators that combine mathematical operations with assignment.

Assignment Operators in Python:

1. **=** (Assignment) – Assigns a value to a variable.
   - **Example: x = 5 assigns the value 5 to the variable x .**
2. **+=** (Add and Assign) – Adds the right value to the left value and assigns the result.
   - **Example: x += 3 is the same as x = x + 3 .**
3. **-=** (Subtract and Assign) – Subtracts the right value from the left value and assigns the result.
   - **Example: x -= 2 is the same as x = x - 2 .**
4. ***=** (Multiply and Assign) – Multiplies the right value by the left value and assigns the result.
   - **Example: x *= 4 is the same as x = x * 4 .**
5. **/=** (Divide and Assign) – Divides the left value by the right value and assigns the result.
   - **Example: x /= 2 is the same as x = x / 2 .**
6. **%=** (Modulus and Assign) – Takes the modulus of the two values and assigns the result.
   - **Example: x %= 3 is the same as x = x % 3 .**

Real-Life Example:
Imagine your bank account has a balance of 500 dollars. If you deposit 100 dollars, the assignment operator can update the balance:

```
balance = 500
balance += 100  # Now, balance becomes 600
```

Problem Statements and Solutions

Problem 1:

Write a program given two numbers and performs the following operations:

1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Modulus

Solution:

```
num1 = 5;
num2 = 13;


# Performing operations
addition = num1 + num2
subtraction = num1 - num2
multiplication = num1 * num2
division = num1 / num2
modulus = num1 % num2

# Displaying results
print(f"Addition: {addition}")
print(f"Subtraction: {subtraction}")
print(f"Multiplication: {multiplication}")
print(f"Division: {division}")
print(f"Modulus: {modulus}")
```

Problem 2:

Write a Python program to compare two ages and print whether the first person is older, younger, or the same age as the second person.

Solution:

```python
# Taking input
age1 = int(input("Enter the age of the first person: "))
age2 = int(input("Enter the age of the second person: "))

# Comparison using relational operators
if age1 > age2:
    print("The first person is older.")
elif age1 < age2:
    print("The first person is younger.")
else:
    print("Both persons are of the same age.")
```

Fun Fact:

- Did you know? Python was initially created as a hobby project by Guido van Rossum in the late 1980s. It has since grown to become one of the most popular programming languages in the world!

Extra Challenge:

Write a Python program that asks the user to enter three test scores. Use logical operators to check if all three scores are above 50. If true, print "Pass", otherwise print "Fail".