Chapter 12: Operator Precedence

Python operators have a set order of precedence, which determines what operators are evaluated first in a potentially ambiguous expression. For instance, in the expression 3 * 2 + 7, first 3 is multiplied by 2, and then the result is added to 7, yielding 13. The expression is not evaluated the **other way around**, because

* has a higher precedence than +

Below is a list of operators by precedence, and a brief description of what they (usually) do.

Section 12.1: Simple Operator Precedence Examples in python

Python follows **PEMDAS** rule. PEMDAS stands for Parentheses, Exponents, Multiplication and Division, and Addition and Subtraction.

```
print(100/10*5)
```

Note: as per PEMDAS precedence, we have to process 10*5, yielding 50,

The 100 / 50, the net result is 2
But that is NOT correct in this case(Logical error)

If we have multiplicaton and division in same expression (without any parentheses), then it must start the process from LEFT to RIGHT

```
print(100/10*5)
```

now from left to right...100/10 will be evaluated first (ans 10.0), then the 10 is multiplied by 5, yielding 50.0 This is CORRECT

======

```
print(100/(10*5))
```

If we want to 10*5 to be evaluated first, we have to give it inside the parentheses ========

See how the LEFT to RIGHT and parentheses works

```
print(300/300 *200) # 200.0
print(300/(300 *200))#0.005
```

From DS team (Mr.Mansoor)

```
print (7**2)
```

```
print (7*2)
print (7/2)
print (7 % 2)
print (7 % 4)
print (7+2)
print (7-2)
print (7**2+3 % 2)
print (7*2 + 25 % 7*2)
print (3+5**2 % 3 * 2/3)
print (3+5**2-3 % 3 * 2/3)
print (3+(5**2)-3 % 3 * (2/3))
```

Modulus (%) and division has same priority (left to right) Modulus (%) and division and multiplication (excutes in left to right) Parentheses ( ), **, *, /, \, Mod, +, -