Basic Python Programming
Basic Math Operations

CSCI-101
Python

• In this side deck, we will look at the following:
  – Basic math operators
  – Precedence rules

• The operator is the math operator (addition, multiplication, etc)
• The operands are the values operated on by the operator
Basic Syntax

• The syntax of all math operations is
  \[ \text{result} = \text{expression} \]

• Here, the result represents the variable that will be assigned the result of the expression on the right; expressions may consist of properly initialized variables or numbers
  \[ \text{sum} = a + b + c \]
  \[ \text{answer} = (a + b) \times 2 \]
Addition

• Addition is evaluated from left to right (if more than 3 operands)
  
  \[ \text{sum} = a + b + c \]

• Here, \(a\) is added to \(b\) first, and then \(c\) is added
Subtraction

• Subtraction is evaluated from left to right (if more than 3 operands)
  
  \[ \text{sum} = a - b - c \]

• Here, b is subtracted from a first, and then c is subtracted
Multiplication

• Multiplication is evaluated from left to right (if more than 3 operands)
• Use the asterisk symbol for multiplication
  \[ \text{prod} = a \times b \times c \]
• Here, a and b are multiplied, then the result is multiplied by c
Integer Division

- There are 2 division operators: `/` indicates floating point division and `//` indicates integer (whole number) division.
- Division is evaluated from left to right (if more than 3 operands):
  \[ \text{sum} = a // b // c \]
- Here, \( a \) is divided by \( b \), and that result is divided by \( c \).
- The values are treated as integers.
Integer Division

• If you want the integer (whole number) remainder, use the % operator
  \[
  \text{sum} = a \% b
  \]

• Here, \( a \) is divided by \( b \), and the remainder is assigned to \( \text{sum} \)

• The values are treated as integers
Floating Point Division

• Floating point division is evaluated from left to right (if more than 3 operands)
  \[ \text{sum} = a / b / c \]

• Here, \(a\) is divided by \(b\), and that result is divided by \(c\)

• The values are treated as floating point (real) numbers
**Integer vs Floating Point Division**

- You previously use the interactive interpreter – shown below are two (nearly) identical expressions and the result.
- Below, the answer is 1 because it’s only dealing with whole numbers.

```python
>>> 3.5 // 2
1.0
```

Below, this answer included the fractional portion because the operands are treated as floating point values.

```python
>>> 3.5 / 2
1.75
```
Exponentiation

• You can raise a number to a power using double asterisks

  >>> 4**2
  16
  >>> 16**.5
  4.0
  >>> 3**4
  81
Operator Precedence

- Operators are evaluated using this hierarchy:
- Unary operators
- Exponentiation
- Multiplication and division
- Addition and subtraction
- All can be overwritten by using parenthesis
Examples

>>> 3 + 4 ** 2
19
>>> 1 + 2 * 4
9
>>> -1 * 16
-16
Examples

>>> (3 + 4) ** 2
49

>>> (1 + 2) * 4
12
Examples – Integer Division and Remainder

```python
>>> 1 % 2
1
>>> 2 % 2
0
>>> 17 // 2
8
>>> 17 % 2
1
```
Examples – Floating Point Division

>>> 17 // 2
8

>>> 17.0 / 2.0
8.5
Evaluate the following

1 + 2 * 3 % 4

2 * 4 ** 2 - 6

1234 // 10 % 10