Objectives

- More arithmetic operators
- Software development practices
  - ➢ Testing
  - ➢ Debugging
  - ➢ Iteration
- Broader Issue

Review

- How can we tell our program display output?
- How can we store information?
- What is the syntax to do the last step?
- What are the rules and conventions for variable names?
  - ➢ What is another word for "variable names"
- What are the types of information we can store?

Review: NOT Math Class

- Need to write out all operations explicitly
  - In math class, a (b+1) meant $a * (b+1)$

Write this way in Python

What are the values?

- After executing the following statements, what are the values of each variable?
  - a = 5
  - y = a + -1 * a
  - z = a + y / 2
  - a = a + 3
  - y = (7+x)*z
  - x = z*2

What are the values?

- After executing the following statements, what are the values of each variable?
  - ➢ a = 5
  - ➢ y = a + -1 * a
  - ➢ z = a + y / 2
  - ➢ a = a + 3
  - ➢ y = (7+x)*z
  - ➢ x = z*2

Runtime error:
> x doesn't have a value yet!
- We say "x was not initialized"
- Can't use a variable on RHS until seen on LHS!*
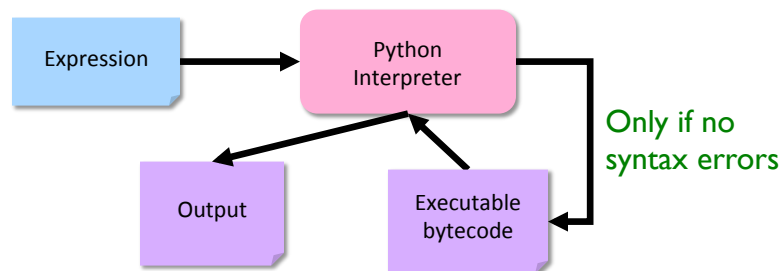
Sept 15, 2017　　　　Sprenkle - CSCI111　　　　5

Python Interpreter

1. Validates Python programming language expression(s)
   - Enforces Python syntax rules
   - Reports syntax errors
2. Executes expression(s)　← Have a lot of these early on!

Expression → Python Interpreter

Only if no syntax errors

Output　　　Executable bytecode

Sept 15, 2017　　　　Sprenkle - CSCI111　　　　6

3

Two Modes to Execute Python Code

- **Interactive/Shell**: using the *interpreter*
  - ➤ Try out Python expressions

- **Batch**: execute *scripts* (i.e., files containing Python code)
  - ➤ What we'll write usually

Sept 15, 2017   Sprenkle - CSCI111   7

Bringing It All Together:
A simple program

```
# Demonstrates arithmetic operations and
# assignment statements
# by Sara Sprenkle

x = 3
y = 5

print("x =", x)
print("y =", y)

print("x * y =", x*y)

# alternatively:
# result = x * y
# print("x*y =", result)
```

Comments: human-readable descriptions. Computer does not execute.

arith_and_assign.py

Sept 15, 2017   Sprenkle - CSCI111   8

Parts of an Algorithm

- Input, Output
- Primitive operations
  - What data you have, what you can do to the data
- Naming
  - Identify things we're using
- Sequence of operations
- Conditionals
  - Handle special cases
- Repetition/Loops
- Subroutines
  - Call, reuse similar techniques

Two Division Operators

/　　Float Division

- Result is a `float`
- Examples:
  - 6/3 → 2.0
  - 10/3 → 3.333333333333335
  - 3.0/6.0 → 0.5
  - 19/10 → 1.9

//　　Integer Division

- Result is an `int`
- Examples:
  - 6//3 → 2
  - 10//3 → 3
  - 3.0//6.0 → 0.0
  - 19//10 → 1

> Integer division is the default division used in most programming languages

Division Practice

- `a = 12//4`
- `4 // 6 * 5.0`
- `b = 6/12`
- `6.0//12`
- `z = a / b`

More on Arithmetic Operations

Symbol	Meaning	Associativity
+	Addition	Left
–	Subtraction	Left
*	Multiplication	Left
/	Division	Left
%	Remainder ("mod")	Left
**	Exponentiation (power)	Right

Precedence rules: P E - DM% AS

↑ negation

Associativity matters when you have the same operation multiple times. It tells you where you should start computing.

Math Practice

```
5 + 3 * 2
2 * 3 ** 2
-3 ** 2
2 ** 3 ** 3
```

How should we verify our answers?

Modulo Operator: %

- Modular Arithmetic: Remainder from division
  - x % y means the remainder of x//y
  - Read as "x mod y"
- Example: 6 % 4
  - Read as "six mod four"
  - 6//4 is 1 with a remainder of 2, so 6%4 evaluates to 2
- Works only with integers
  - Typically just positive numbers
- Precedence rules: P E - DM**%** AS

Modulo Practice

- 7 % 2
- 3 % 6
- 6 % 2
- 7 % 14
- 14 % 7
- 6 % 0

Brainstorm

- What useful thing does % 10 do?
  - 3 % 10 =
  - 51 % 10 =
  - 40 % 10 =
  - 678 % 10 =
  - 12543 % 10 =
- What useful thing does // 10 do (integer division)?
  - 3 // 10 =
  - 51 // 10 =
  - 40 // 10=
  - 678 // 10 =
  - 12543 // 10 =
- What useful thing does % 2 do?

Broader Issue Groups

Introduce yourselves!

Aimee Tristan Turner Utkrist	Annie B. Daniel Isaac Max Prakriti	Amalia Chris Drew Pranam Sam	Abhi Alex Angel Katlin Landon	JD Liam Pengrui Rinn

Sept 15, 2017   Sprenkle - CSCI111   17

Broader CS Issues

- Good summaries!
  - Good English, complete sentences
- Good, thoughtful questions
- Mechanics details
  - Follow instructions on BI Forum about what summary should contain
  - Should be able to edit your own posts
  - Characters from Word
    - Click button "Paste from Word"
  - Don't attach Word documents

Sept 15, 2017   Sprenkle - CSCI111   18

AI Everywhere

- "An algorithm is, essentially, a brainless way of doing clever things… Brainlessness, in other words, is no impediment to intelligence. "
- What are examples of algorithms that you do every day?
- What is AI (which is based on algorithms) useful for?
  - ➤ What aren't algorithms useful for?
- What would be some useful algorithms, specific to W&L students?
  - ➤ What are problems that are difficult—but useful—to solve?

Sept 15, 2017   Sprenkle - CSCI111   19

Looking Ahead

- Pre-lab assignment due before lab on Tuesday

Sept 15, 2017   Sprenkle - CSCI111   20