

Objectives

- Assignments and Arithmetic
- Software development practices
 - Testing
 - Debugging
 - Iteration

Handouts on Canvas

Jan 19, 2022

Sprenkle - CSCI111

1

1

Computer Science Student Assistants

Also in Canvas announcements

Day of the Week	Students
Tuesday	Alex Billy
Wednesday	Alyssa* Sam
Thursday	Alyssa* Katie

<https://wlu.box.com/v/CSAssistantZoomLinks>

Alyssa was an in-lab student assistant for my version of CSCI in the past

Jan 19, 2022

Sprenkle - CSCI111

2

2

Review

- What are the two modes for running Python programs?
- How do we tell our program to display output?
- How can we store information?
 - What is the syntax to do that?
- What are the rules and conventions for variable names?
 - What is another term for “variable names”?
 - Describe characteristics of *good* variable names
- What are the primitive types of information in Python?
- What are the arithmetic operators? Describe their syntax and semantics.
- What is our development process?
 - Generally and for the lab specifically
- What are the expectations for a complete program for this class?

Jan 19, 2022

Sprengle - CSCI111

3

3

Printing Multiple Things

- **print** is a special command or a *function*
- To display multiple things on the same line, separate them with commas
 - `print("Hello,", "class")`
 - `print("x =", 5)`
 - `print(x*y, "is the magic number")`
 - `print(r, s, t)`

Syntax: ,
Semantics: display this too, separated by a space in the display

Jan 19, 2022

Sprengle

4

Formalizing Process of Developing Computational Solutions

1. Create a sketch of how to solve the problem (the algorithm)
 2. Fill in the details in Python
 3. Execute the program
 4. If output doesn't match your expectation, debug the program
 - Where is the problem? How do I fix it?
-

Jan 19, 2022

Our development process will evolve over time

5

5

Development Process for Lab

- Develop in IDLE
1. Create a new file
 2. Develop the program (following previous slide)
 3. Close the shell
 4. Run the program again
 5. Save output from program

Jan 19, 2022

Sprengle - CSCI111

6

6

Review: Lab Expectations

- Comments in programs
 - High-level comments, author
 - Notes for your algorithms, implementation
- Nice, readable, clearly labeled understandable output
 - User running your program needs to **understand** what the program is saying
- Honor System

Jan 19, 2022

Sprenkle - CSCI111

7

7

Other Lab Notes

- Trying to set you up for success now
 - Develop good development habits
 - You know the expectations and how you should develop as programs get larger, more complex
- I won't check your labs before every submission
- Learning how to solve problems
 - Every week: new problems, new techniques to solve problems
- I am explicit in directions/reminders early
 - Then stop reminding because you should know the process later
- Labs are due on Friday; review before the next lab


Jan 19, 2022

Sprenkle - CSCI111

8

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

Jan 19, 2022

Sprenkle - CSCI111

9

9

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 - MD% AS

negation

Jan 19, 2022

Sprenkle - CSCI111

10

10

More on Arithmetic Operations

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

Precedence rules: P E - MD% AS

negation

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

Jan 19, 2022

Sprenkle - CSC111

11

Two Division Operators

/ Float Division

- Result is a **float**
- Examples:
 - $6/3 \rightarrow 2.0$
 - $10/3 \rightarrow 3.3333333333333335$
 - $3.0/6.0 \rightarrow 0.5$
 - $19/10 \rightarrow 1.9$

// Integer Division

- Result is an **int**
- Examples:
 - $6//3 \rightarrow 2$
 - $10//3 \rightarrow 3$
 - $3.0//6.0 \rightarrow 0.0$
 - $19//10 \rightarrow 1$

Integer division is the *default* division used in many programming languages

Jan 19, 2022

Sprenkle - CSC111

12

12

Python Division Practice

1. `6.0//12 * 5.0`

2. `12 // 4 * 5.2`

3. `a = 12//5`

4. `b = 6/12`

5. `z = a / b`

Showing a mix of expressions
(just expression and within assignment statements;
integers and floats)

Jan 19, 2022

Sprenkle - CSCI111

13

13

Python Math Practice

1. `5 + 3 * 2`

2. `2 * 3 ** 2`

3. `-3 ** 2`

4. `2 ** 3 ** 3`

Jan 19, 2022

Sprenkle - CSCI111

14

14

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
- Typical use: only with positive integers
- Precedence rules: P E - MD% AS

Jan 19, 2022

Sprenkle - CSCI111

15

15

Modulo Practice

1. $7 \% 2$

2. $3 \% 6$

3. $6 \% 2$

4. $7 \% 14$

5. $14 \% 7$

6. $6 \% 0$

Jan 19, 2022

Sprenkle - CSCI111

16

16

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?

Jan 19, 2022

Sprenkle - CSCI111

17

17

Trick: Type Conversion

- You can convert a variable's type
 - Use the type's **constructor**

Conversion Function/Constructor	Example	Value Returned
<code>int(<number or string>)</code>	<code>int(3.77)</code> <code>int("33")</code>	3 33
<code>float(<number or string>)</code>	<code>float(22)</code>	22.0
<code>str(<any value>)</code>	<code>str(99)</code>	"99"

Jan 19, 2022

Sprenkle - CSCI111

18

18