CSC 111: Intro to Computer Science through Programming

Spring 2017
Prof. Sara Mathieson
After viewing the video lecture that accompanies these slides, take the quiz on Moodle by Tuesday night.

I will be at TA hours on Tuesday 2/7 from 7:30-9:30pm.
Outline: 2/6

+ Recap last time
+ Data types and casting
+ Python math module
+ Triangle loop loop exercise
Recap
Recap last time

String review

The plus (+) operator is used to **concatenate**
two strings to form a new string
i.e. `word1 + word2`

In the print function, use commas to
separate **arguments** (including strings)
i.e. `print(word1, word2)` or `print(word1, 5)`
Recap last time

String review
The plus (+) operator is used to concatenate two strings to form a new string
i.e. word1 + word2

In the print function, use commas to separate arguments (including strings)
i.e. print(word1,word2) or print(word1,5)

Naming conventions
• Underscores to separate words
• All lowercase

spam_and_eggs
Recap last time

String review

The plus (+) operator is used to **concatenate** two strings to form a new string  
i.e. \texttt{word1 + word2}

In the print function, use commas to separate **arguments** (including strings)  
i.e. \texttt{print(word1,word2)} or \texttt{print(word1,5)}

Naming conventions

- Underscores to separate words
- All lowercase

\texttt{spam\_and\_eggs}
Recap last time

String review
The plus (+) operator is used to concatenate two strings to form a new string i.e. \texttt{word1 + word2}

In the print function, use commas to separate \texttt{arguments} (including strings) i.e. \texttt{print(word1,word2)} or \texttt{print(word1,5)}

Naming conventions
• Underscores to separate words
• All lowercase

spam\_and\_eggs

Swap function

```python
>>> x = 7
>>> y = 10
>>> x = y
>>> y = x
```

Main takeaways for assignment:
• The equals sign (=) is the assignment operator
• Assignment is not symmetric
• The LHS is the variable being changed
• The value of the expression on the RHS is used
Recap last time

Getting ready for a chatbot with the echo program

```python
# CSC 111, Day 4
# Author: Sara Mathieson and CSC 111 class
# Program to echo user input and add a question mark

def main():

    # execute the for loop a fixed number of times
    for i in range(5):
        print()  # blank line
        print("Exchange", i)
        response = input("Say something: ")
        echo = response + "?"
        print(echo)

    # outside the for loop, print ending statement
    print("Conversation over")

    # invoke main
    main()
```
Data types and casting
Types we will use for now

- String (abbreviation `str`)
  - i.e. “hello”, “coffee”, “-------”
Types we will use for now

- **String** (abbreviation `str`)
  - i.e. “hello”, “coffee”, “--------”

- **Integer** (abbreviation `int`)
  - i.e. -10, -2, 0, 3, 5, 89
Types we will use for now

- **String** (abbreviation `str`)
  - i.e. “hello”, “coffee”, “-------”

- **Integer** (abbreviation `int`)
  - i.e. -10, -2, 0, 3, 5, 89

- **Floating point**; think decimal (abbreviation `float`)
  - i.e. 5.5, 0.336721, 3.141592...
Types we will use for now

- String (abbreviation \texttt{str})
  - i.e. “hello”, “coffee”, “--------”

- Integer (abbreviation \texttt{int})
  - i.e. -10, -2, 0, 3, 5, 89

- Floating point; think decimal (abbreviation \texttt{float})
  - i.e. 5.5, 0.336721, 3.141592...

Called data types since they hold data (numbers, names, etc).

We will use other types later on in this course.
Types we will use for now

- **String** (abbreviation `str`)
  - i.e. "hello", "coffee", "-------"

- **Integer** (abbreviation `int`)
  - i.e. -10, -2, 0, 3, 5, 89

- **Floating point**; think decimal (abbreviation `float`)
  - i.e. 5.5, 0.336721, 3.141592...

Called data types since they hold data (numbers, names, etc).

We will use other types later on in this course.
Python math module
Python math module examples

```python
>>> import math
>>> math.sqrt(2)
1.4142135623730951
>>> math.pi
3.141592653589793
>>> math.sin( math.pi / 4 )
0.7071067811865475
```

More information: https://docs.python.org/3/library/math.html

Credit: Joe O’Rourke
Triangle loop example
Create a file called `triangle.py`

```python
>>> main()
Enter n = 10
#
# #
# # #
# # # #
# # # # #
# # # # # #
# # # # # # #
# # # # # # # #
# # # # # # # # #
# # # # # # # # # #
# # # # # # # # # # #
# # # # # # # # # # # #
# # # # # # # # # # # # #
# # # # # # # # # # # # # #
# # # # # # # # # # # # # # #
# # # # # # # # # # # # # # # #
```

Try to produce this output!