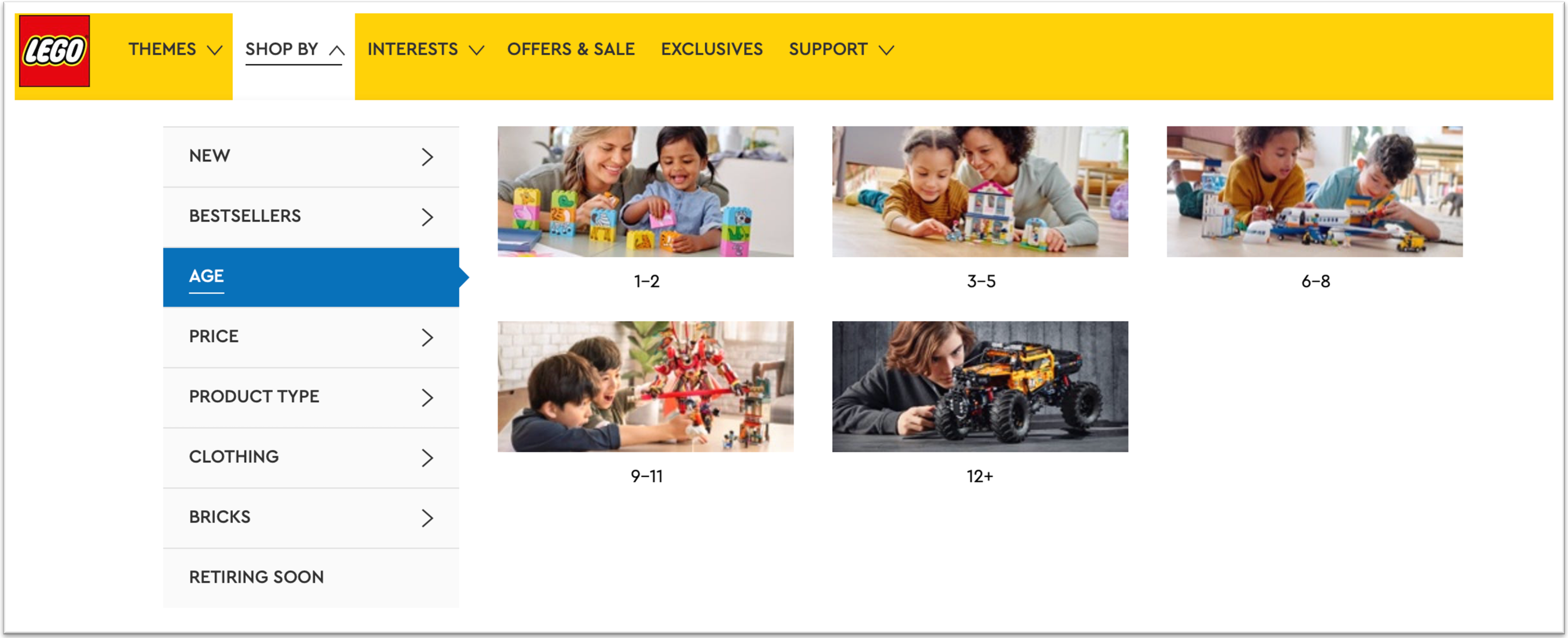
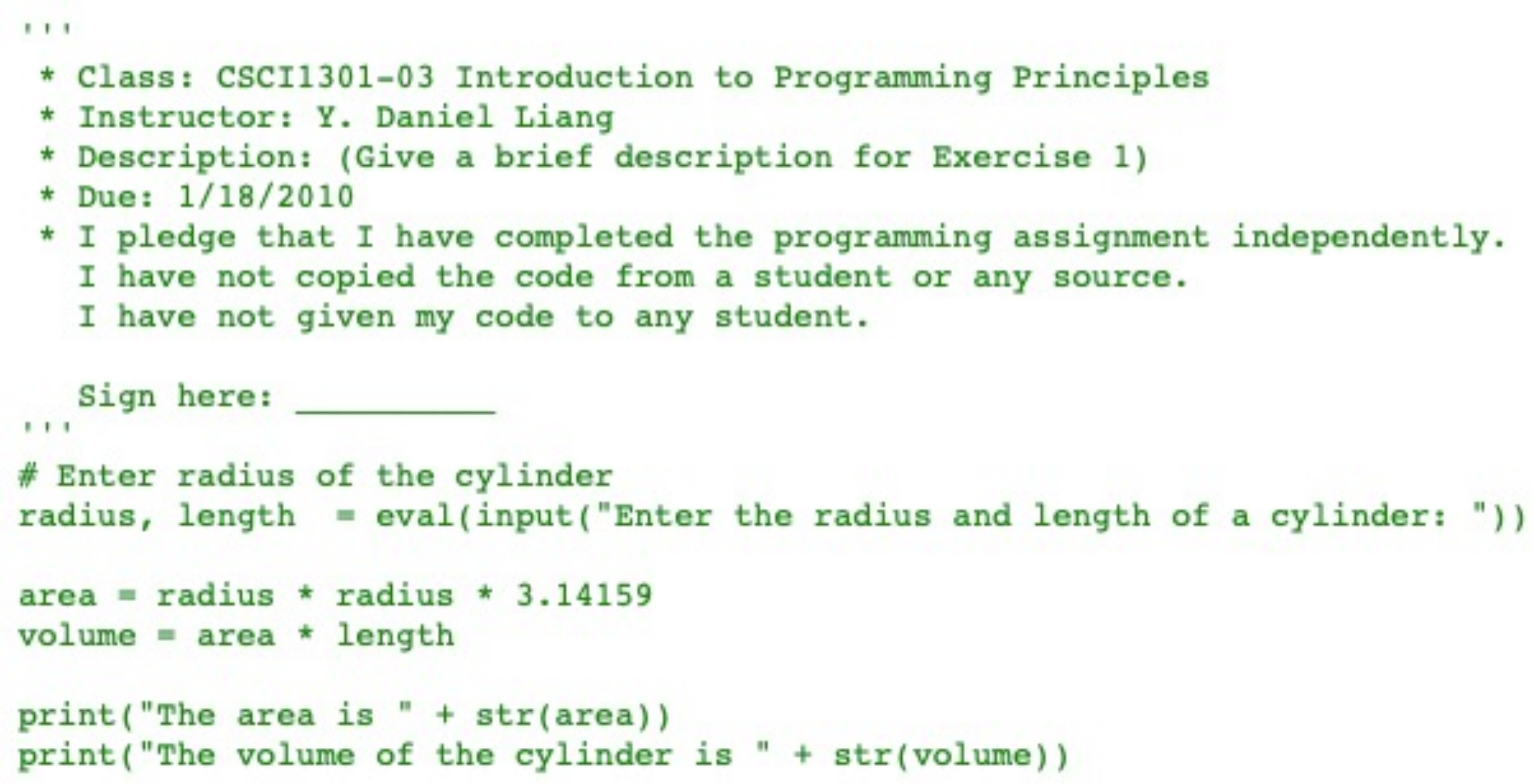
Chapter 2 Elementary Programming
Part - 1



LEGO
THEMES
SHOP BY
INTERESTS
OFFERS & SALE
EXCLUSIVES
SUPPORT
NEW
BESTSELLERS
AGE
PRICE
PRODUCT TYPE
CLOTHING
BRICKS
RETIRING SOON
1–2
3–5
6–8
9–11
12+

What to be covered next

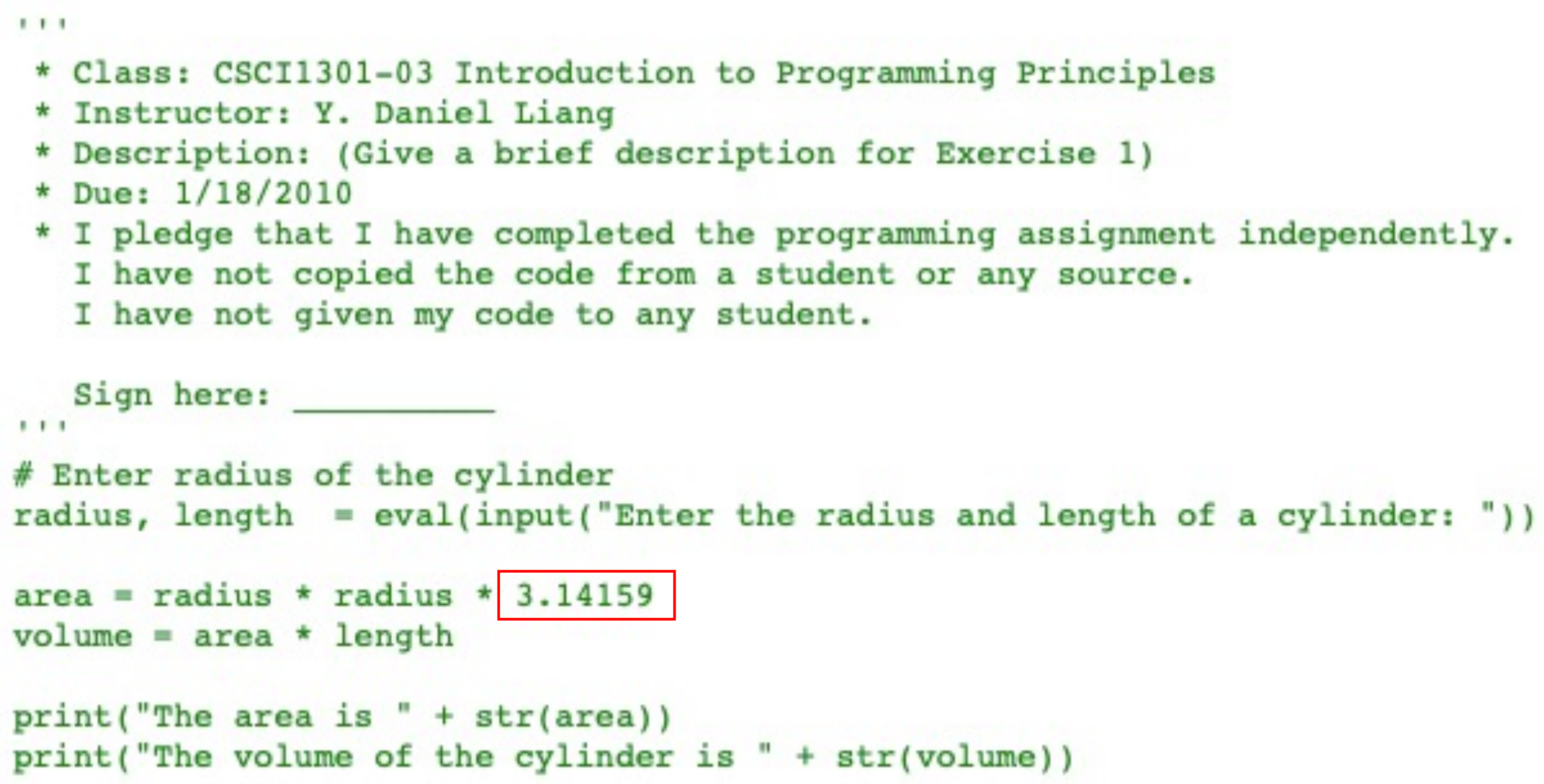
- Constants
- Reserved Words
- Variables
  - Name Rules
- Assignment Statement
- Expressions
  - Numeric Expressions
  - Order of Evaluation
  - Operator Precedence Rule
- Input Statement



```
'''
 * Class: CSCI1301-03 Introduction to Programming Principles
 * Instructor: Y. Daniel Liang
 * Description: (Give a brief description for Exercise 1)
 * Due: 1/18/2010
 * I pledge that I have completed the programming assignment independently.
   I have not copied the code from a student or any source.
   I have not given my code to any student.

   Sign here: __________
'''
# Enter radius of the cylinder
radius, length  = eval(input("Enter the radius and length of a cylinder: "))

area = radius * radius * 3.14159
volume = area * length

print("The area is " + str(area))
print("The volume of the cylinder is " + str(volume))
```

What to be covered next

- **Constants**
- Reserved Words
- Variables
  - Name Rules
- Assignment Statement
- Expressions
  - Numeric Expressions
  - Order of Evaluation
  - Operator Precedence Rule
- Input Statement



```
'''
 * Class: CSCI1301-03 Introduction to Programming Principles
 * Instructor: Y. Daniel Liang
 * Description: (Give a brief description for Exercise 1)
 * Due: 1/18/2010
 * I pledge that I have completed the programming assignment independently.
   I have not copied the code from a student or any source.
   I have not given my code to any student.

   Sign here: __________
'''
# Enter radius of the cylinder
radius, length  = eval(input("Enter the radius and length of a cylinder: "))

area = radius * radius * 3.14159
volume = area * length

print("The area is " + str(area))
print("The volume of the cylinder is " + str(volume))
```

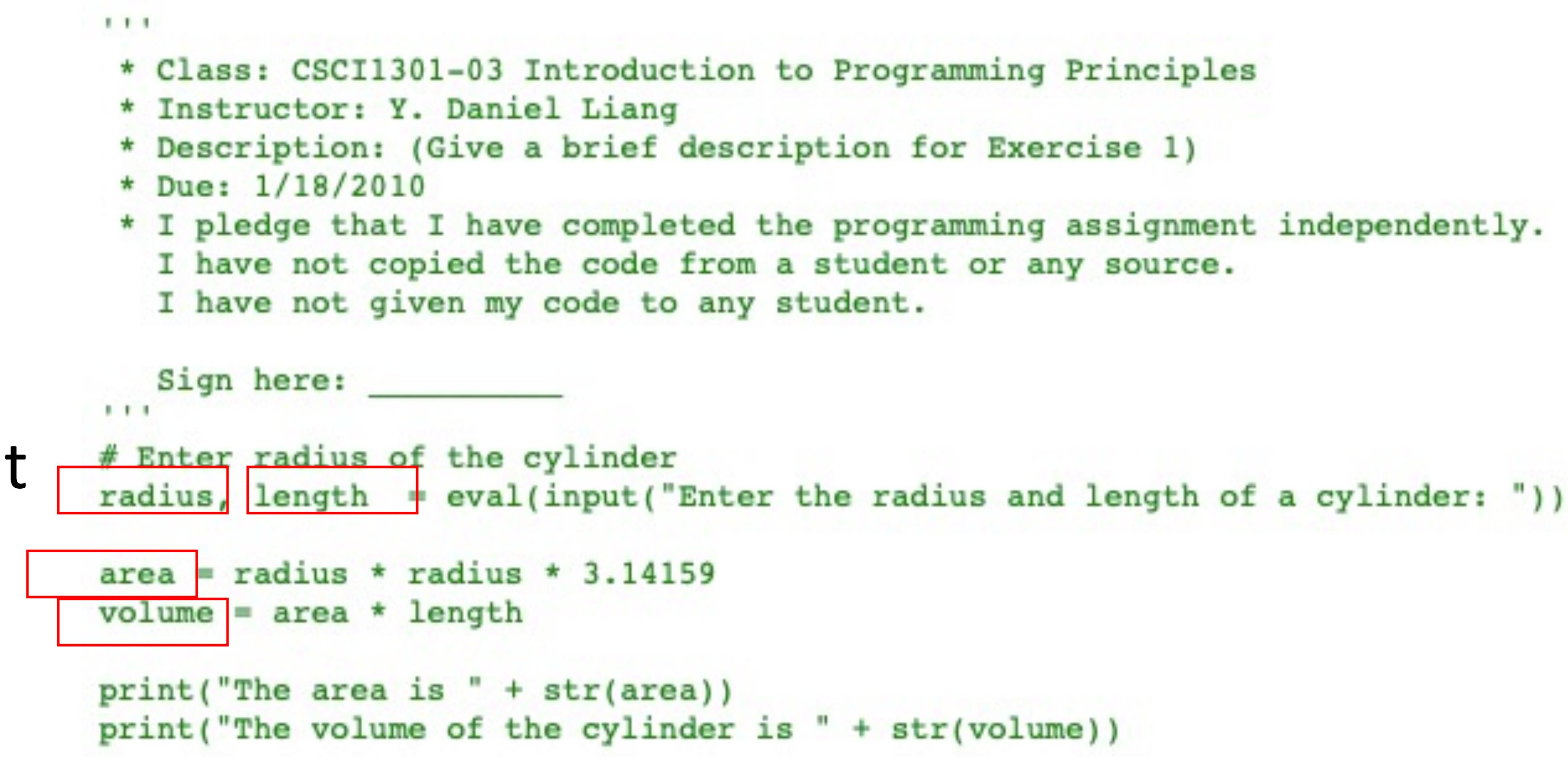
Constants

- Fixed values such as numbers, letters, and strings, are called “constants” because their value does not change
- Numeric constants are as you expect
- String constants use single quotes (') or double quotes (")

```
>>> print(123)
123
>>> print(98.6)
98.6
>>> print('Hello world')
Hello world
```

What to be covered next

- Constants
- **Variables**
    - **Name Rules**
- **Reserved Words**
- Assignment Statement
- Expressions
    - Numeric Expressions
    - Order of Evaluation
    - Operator Precedence Rule
- Input Statement

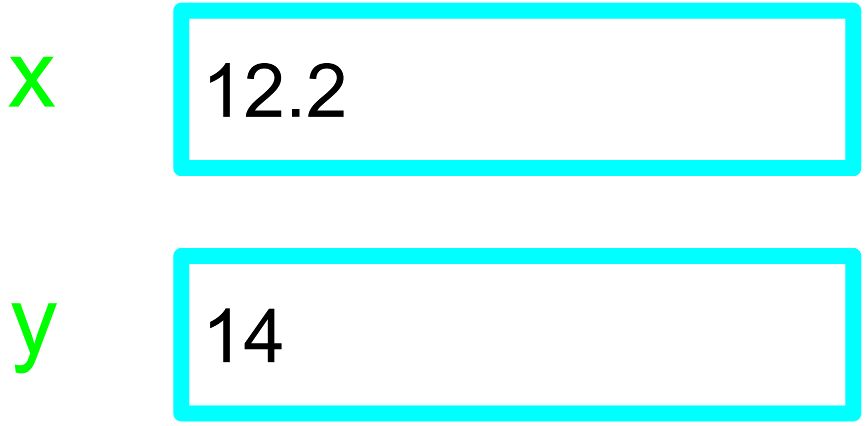


```
'''
 * Class: CSCI1301-03 Introduction to Programming Principles
 * Instructor: Y. Daniel Liang
 * Description: (Give a brief description for Exercise 1)
 * Due: 1/18/2010
 * I pledge that I have completed the programming assignment independently.
   I have not copied the code from a student or any source.
   I have not given my code to any student.

   Sign here: __________
'''
# Enter radius of the cylinder
radius, length = eval(input("Enter the radius and length of a cylinder: "))

area = radius * radius * 3.14159
volume = area * length

print("The area is " + str(area))
print("The volume of the cylinder is " + str(volume))
```

Variables

- A variable is a named place in the memory where a programmer can store data and later retrieve the data using the variable "name"
- Programmers get to choose the names of the variables
- You can change the contents of a variable in a later statement

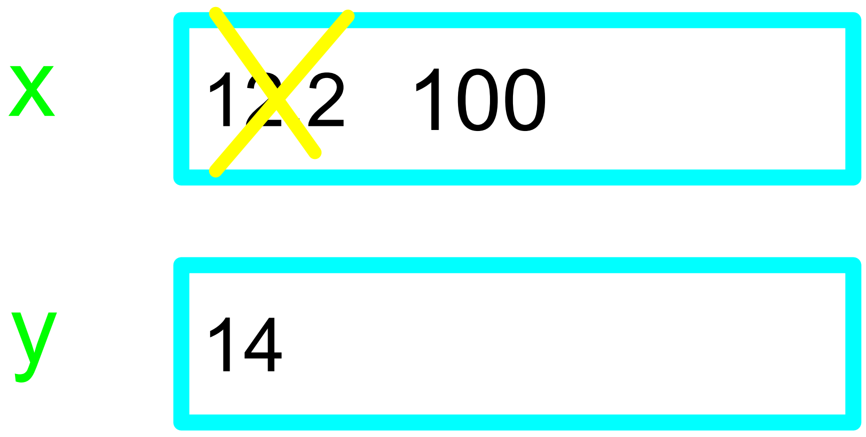
```
x = 12.2
y = 14
```



Variables

- A variable is a named place in the memory where a programmer can store data and later retrieve the data using the variable "name"
- Programmers get to choose the names of the variables
- You can change the contents of a variable in a later statement

```
x = 12.2
y = 14
x = 100
```



Python Variable Name Rules

- Must start with a letter or underscore _
- Must consist of letters, numbers, and underscores
- Case Sensitive
- Cannot be reserved words

```
Good:      spam    eggs   spam23    _speed
Bad:       23spam     #sign  var.12
Different:     spam   Spam   SPAM
```

Reserved Words

- You cannot use reserved words as variable names / identifiers

```
False   class   return  is      finally
None    if      for     lambda  continue
True    def     from    while   nonlocal
and     del     global  not     with
as      elif    try     or      yield
assert  else    import  pass
break   except  in      raise
```

Statements (Sentences or Lines)

```
x = 2
x = x + 2
print(x)
```

Assignment statement

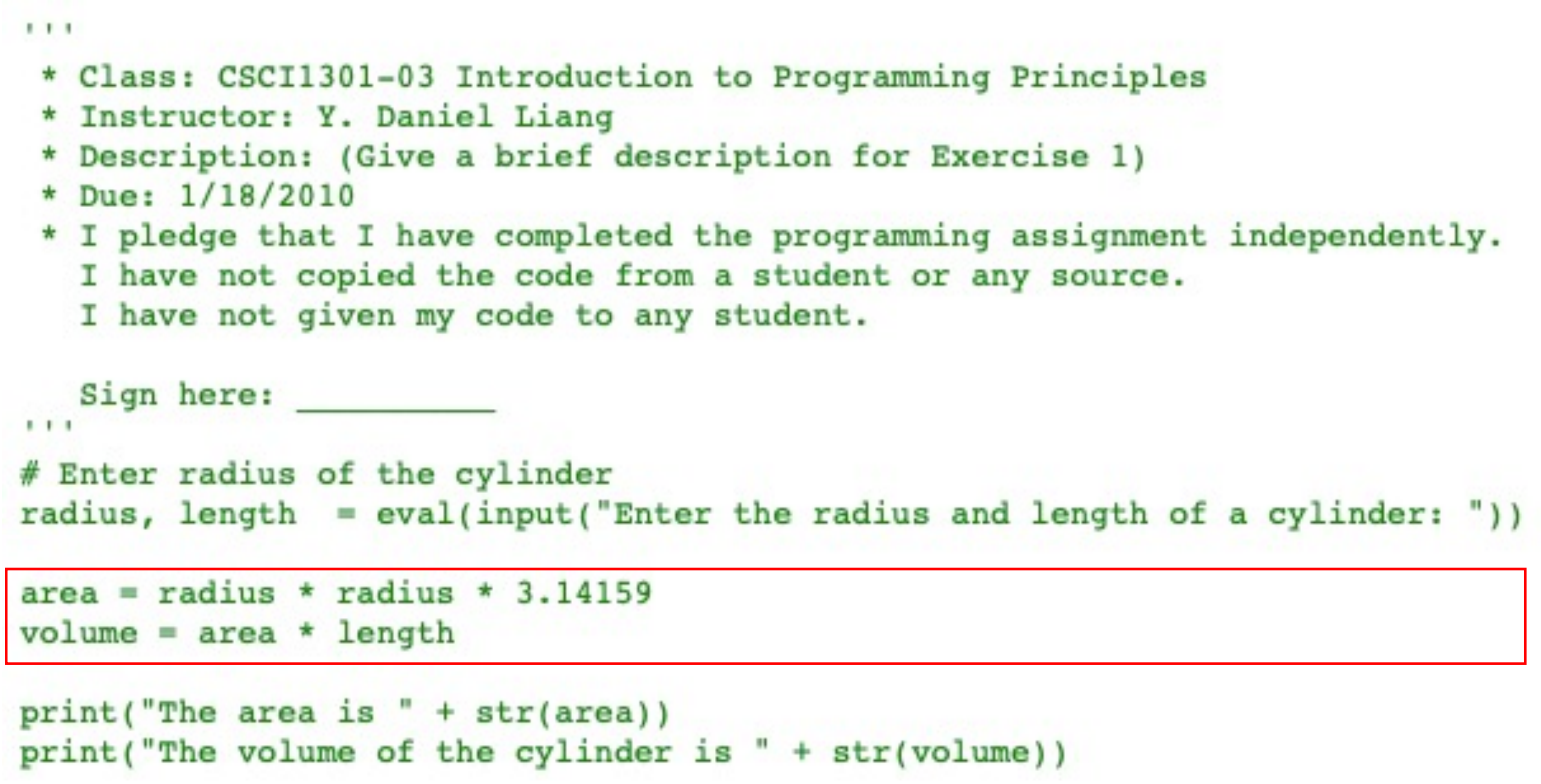
Assignment with expression

Print statement

Variable Operator Constant Function

What to be covered next

- Constants
- Reserved Words
- Variables
  - Name Rules
- **Assignment Statement**
- **Expressions**
  - Numeric Expressions
  - Order of Evaluation
  - Operator Precedence Rule
- Input Statement



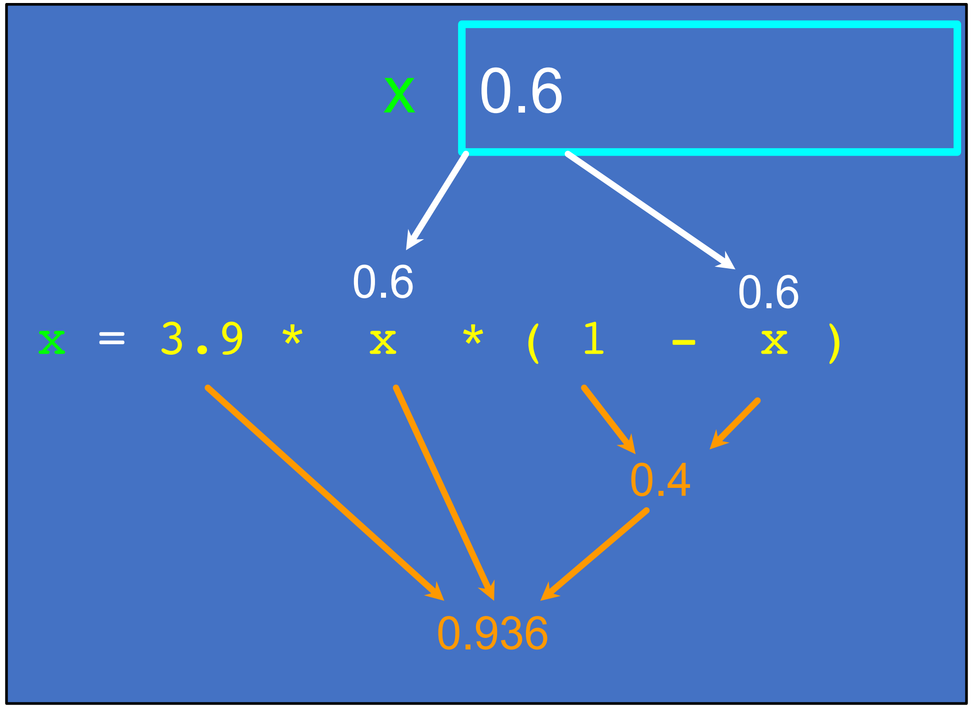
```
'''
 * Class: CSCI1301-03 Introduction to Programming Principles
 * Instructor: Y. Daniel Liang
 * Description: (Give a brief description for Exercise 1)
 * Due: 1/18/2010
 * I pledge that I have completed the programming assignment independently.
   I have not copied the code from a student or any source.
   I have not given my code to any student.

   Sign here: __________
'''
# Enter radius of the cylinder
radius, length  = eval(input("Enter the radius and length of a cylinder: "))

area = radius * radius * 3.14159
volume = area * length

print("The area is " + str(area))
print("The volume of the cylinder is " + str(volume))
```

Assignment Statements

- We assign a value to a variable using the assignment statement (=)
- An assignment statement consists of an expression on the right-hand side and a variable to store the result

```
x = 3.9 * x * ( 1 - x )
```

A variable is a memory location used to store a value (0.6)



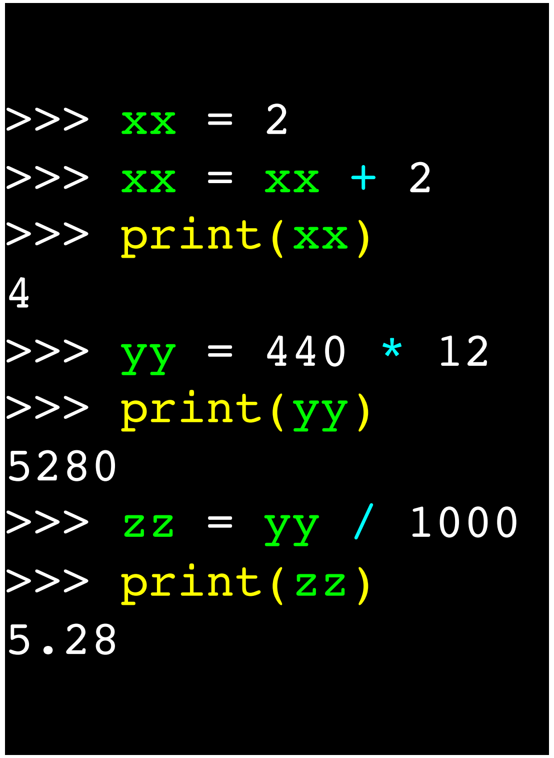
```
x = 3.9 * x * ( 1 - x )
```

The right side is an expression.
Once the expression is evaluated,
the result is placed in (assigned to) x.

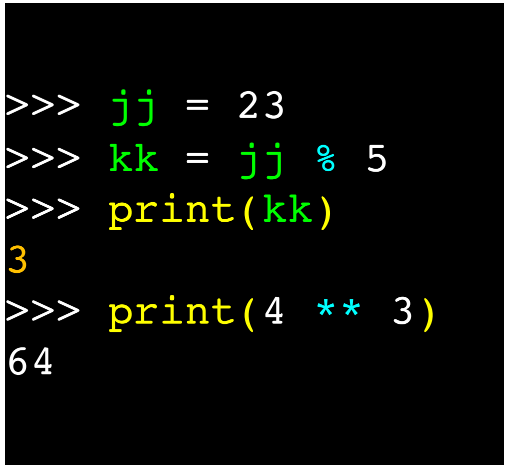
Expressions

- An expression is a combination of values, variables, and operators.
- A value all by itself is considered an expression, and so is a variable
- If you type an expression in interactive mode, the interpreter evaluates it and displays the result.
- But in a program, an expression all by itself doesn't do anything! This is a common source of confusion for beginners.

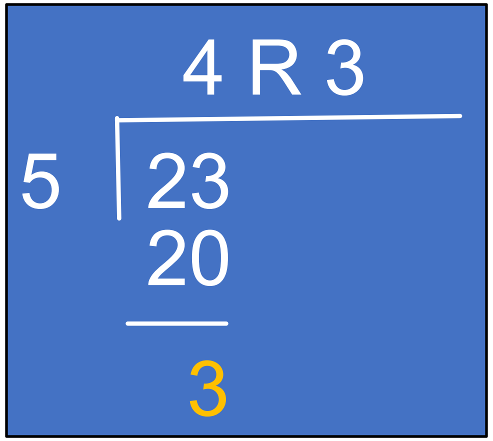
Numeric Expressions



```
>>> xx = 2
>>> xx = xx + 2
>>> print(xx)
4
>>> yy = 440 * 12
>>> print(yy)
5280
>>> zz = yy / 1000
>>> print(zz)
5.28
```



```
>>> jj = 23
>>> kk = jj % 5
>>> print(kk)
3
>>> print(4 ** 3)
64
```



Operator	Operation
+	Addition
-	Subtraction
*	Multiplication
/	Division
**	Power
%	Remainder

Input Statement

- We can instruct Python to pause and read data from the user using the input() function
- The input() function returns a string

```
nam = input('Who are you? ')
print('Welcome', nam)
```

```
Who are you? Yuan
Welcome Yuan
```

Converting User Input

- If we want to read a number from the user, we must convert it from a string to a number using a type conversion function.
- Later we will deal with bad input data.

```
inp = input('Europe floor?’)
usf = int(inp) + 1
print(‘US floor’, usf)
```

```
Europe floor? 0
US floor 1
```

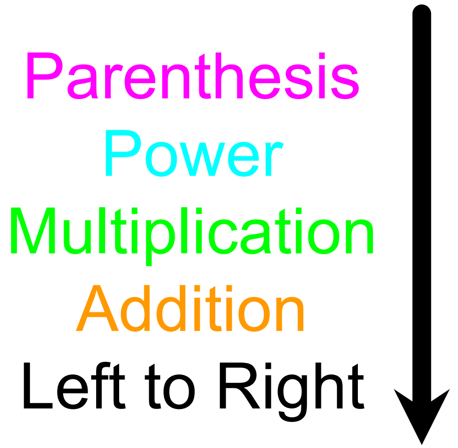
Order of Evaluation

- When we string operators together - Python must know which one to do first
- This is called “operator precedence”
- Which operator “takes precedence” over the others?

```
x = 1 + 2 * 3 - 4 / 5 ** 6
```

Operator Precedence Rules

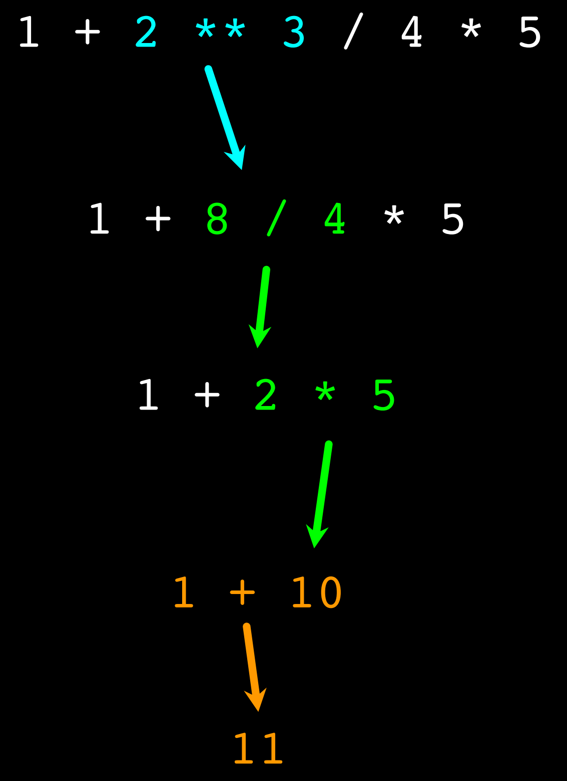
- Highest precedence rule to lowest precedence rule:
    - Parentheses are always respected
    - Exponentiation (raise to a power)
    - Multiplication, Division, and Remainder
    - Addition and Subtraction
    - Left to right



Parenthesis
Power
Multiplication
Addition
Left to Right

```
>>> x = 1 + 2 ** 3 / 4 * 5
>>> print(x)
11.0
>>>
```

Parenthesis
Power
Multiplication
Addition
Left to Right



1 + 2 ** 3 / 4 * 5

1 + 8 / 4 * 5

1 + 2 * 5

1 + 10

11

Summary

- Constants
- Reserved Words
- Variables
    - Name Rules
- Assignment Statement
- Expressions
    - Numeric Expressions
    - Order of Evaluation
    - Operator Precedence Rule
- Input Statement