Lesson 3

Input and Output Statements:

More on **Print()**

The characters that can appear within strings include letters of the alphabet (A-Z, a-z), digits (0-9), punctuation (, ., ! etc.), and other printable symbols (#, &, %, etc.). In addition to these “normal” characters, we may embed special characters known as control codes.
The backslash symbol (\) signifies that the character that follows it is a control code.

- \n control code represents the message after \n to be printed on the next line,
- \t for tab

Instructions	**Output**

print('A\nB\nC')
print('D\tE\tF')
	A
	B
	C
	D E F

To embed a quotation in a print statement, you can choose one of the methods as shown in the table.
They each can print the statement
Did you know that ‘word’ is a word?
or
Did you know that “word” is a word?

	Instructions
1	**print("Did you know that 'word' is a word?")**
2	**print('Did you know that "word" is a word?')**
3	**print('Did you know that \'word\' is a word?')**
4	**print("Did you know that \"word\" is a word?")**

We can control how the print function separates the arguments it displays. By default, the print function places a single space in between the items it prints.
Print uses a keyword argument named *sep* (separator) to specify the string to use insert between items.
Printsep.py in the table shows the *sep* customizes print’s behavior.

	printsep.py	Output
1	w, x, y, z = 10, 15, 20, 25	
2	print(w, x, y, z)	10 15 20 25
3	print(w, x, y, z, sep=',')	10,15,20,25
4	print(w, x, y, z, sep='')	10152025
5	print(w, x, y, z, sep=':')	10:15:20:25
6	print(w, x, y, z, sep='-----')	10-----15-----20-----25

Activity 1:
Write a **one-line** print statement so that the computer can produce the following message:

My name is “Chan”, not “Chen”.
I live in Hong Kong

User Input

Input ()

The print function enables a Python program to display textual information to the user. Programs may use the input function to obtain information from the user. The simplest use of the input function assigns a *string* to a variable:

x = input()

Program 1	Output	Program 2

print('Please enter an integer value:')
x = input()
print('Please enter another integer value:')
y = input()
num1 = int(x)
num2 = int(y)
print(num1, '+', num2, '=', num1 + num2)
	Please enter an integer value
	13
	Please enter another integer value
	16
	num1 + num2 = 29
		**x = input**('Please enter an integer value: ')
		**y = input**('Please enter another integer value: ')
		num1 = int(x)
		num2 = int(y)
		print(num1, '+', num2, '=', num1 + num2)

Lesson 3

Activity 2:
Modify the program on rectangle (from last lesson) so that the dimensions of the rectangle are entered by the user.

Expressions

Consider the following script:

average.py	output

print(“Please enter three numbers; each is separated by return key”)
x = int(input())
y = int(input())
z = int(input())
avg = (x + y + z)/3   *#avg can be assigned an expression*
print (“The average is “, avg)
	Please enter three numbers; each separated by return key
	3
	4
	5
	the average is 4.0

Expressions can be commonly built using mathematical operators (+, - , *, /, **, // and %). For example *avg = (x + y + z)/3*

More Arithmetic Operators

Variables are often modified in a regular way as programs execute. A variable may increase by one or decrease by five. The statement

x = x + 1

increments x by one. It should be read as x becomes x + 1. In fact it is a *counter*. Python has a shorter statement that accomplishes the same effect:

x += 1

This is the increment statement. A similar decrement statement is available:

x -= 1   (# Same as x = x – 1);

Python provides a more general way of simplifying a statement that modifies a variable through simple arithmetic. For example, the statement

x = x + 5

can be shortened to

x += 5

Any statement of the form

x op= exp

where x is a variable, op= is the operator (+=, -=, *=, /=, //=, and %=); exp is an expression compatible with the variable x.

Lesson 3

Arithmetic reassignment statements of this form are equivalent to

x = x op exp;

For examples:

	x op= exp	x = x op exp
1	x -=4	x = x - 4
2	x *=6	x = x*6
3	x *= y + z	x = x*(y+z)

Activity 3: Using Math Functions
Create a program to compute the circumference of a circle with the radius given by the user.

To use math values or functions, do

1.

2.

Activity 4: swapping the contents of two variables
Suppose x and y are two variables of a program. Their contents are entered by the user. Write a program to interchange the contents of the two variables. (i.e. If the input is 34, 21, the output will be 21, 34)

Your program / code	The output	Sample program

Program practice (for submission):
Write a program to create, either,

1. a Fahrenheit to Celsius converter using the formula C = $\frac{5}{9}x\ (F-32)$. A sample output is as follows:

   Please enter the temperature in Fahrenheit: *200*
   200 degrees F = 93.33333 degrees C

   where 200 is entered by the user.

2. a second to hour converter. A sample output is as follows

   Please enter the time in seconds: *4000*
   4000 seconds = 1 hr 6 min 40 sec

   where 4000 is entered by the user

Save your program as FC-xxyy.py where xx is your name and yy is your class. Pay attention to user friendliness in your program.