

# Lab/Homework 1

**Deadline:** 23:59 pm, Friday, Nov 3.

## What to submit:

A report with answers to each exercise and corresponding python program(.py file), packaged into a zip file. Named zip as “class\_name\_HW1” (for example:AI1\_刘晔\_HW1), submit to TA (Jiashuo Zheng).

## Requirements on Coding:

1. Adding header to each .py file.  
"""

xxxx.py  
author:  
date:  
description:  
"""

2. Please add a space around the operator and after the comma.

3. add a blank line between code of different functions

## Exercise 1.0 Installing Python

Follow the instructions on the Appendix A of *Python crash course* to install Python and IDLE on your own computer. Install one of IDE chosen from slides. Be sure to install Python 3.x. Ask your TA for help if you run into any trouble. Before continuing, play around with the Python shell a bit and explore how you can use it as a calculator, and show a screenshot in the report.

## Exercise 1.1 Compiler VS Interpreter

1. Summarize the difference between C++ and python.
2. Show difference between running the python program and C++ program.

## Exercise 1.2 Mathematical Operator

Create a new program called `test_operator.py`. Input the following sets of equations, output the type of results. Note the difference between *int* arithmetic and *float* arithmetic, pay attention to the output! And give some explanations to each one.

1.  $3 * 4$ ,  $3.0 * 4$ ,  $3 * 4.0$ ,  $3.0 * 4.0$
2.  $\frac{12}{3}$ ,  $\frac{1}{3}$ ,  $\frac{12}{3.0}$ ,  $\frac{12.0}{3}$
3.  $2^2$ ,  $2^{2.0}$ ,  $2.0^2$  (Both `**` and `math.pow` should be tested)
4.  $\sqrt[2]{7+9}$ ,  $\sqrt[4]{100-19}$

## Exercise 1.3 Calculate average value

Create a new program called `average_calculation.py`. In this program, users can input a number  $N$ , then the program will output the average value from 1 to  $N$ .

## Exercise 1.4 Play with numbers

Create a new program called `convert_three_digit.py`. The user inputs a three-digit natural number, and calculates and outputs the digits in the hundreds, tens and ones digits. For example, when the user input a number 123, the program outputs “The numbers in the hundreds, tens and ones digit are 1, 2, 3 separately.” Draw the diagram or write pseudocode, and finish program.

## Exercise 1.5 Calculate distance

Create a new program called `distance.py`. The user inputs four int or float values  $x_1$ ,  $y_1$ ,  $x_2$ ,  $y_2$  that represent the two points  $(x_1, y_1)$  and  $(x_2, y_2)$ , and outputs the distance between those points as a float. Draw the diagram or write pseudocode, and finish program.

## Exercise 1.6 A simple Chatbot

1. When you start talking with chatbot, it can output a greeting sentence  
“Hello, I am a simple chatbot”
2. Chatbot can ask several questions and output a summary of your personal information  
Q(Chatbot): What’s your name?

A(User): Bob

Q(Chatbot): What is your birthday?

A(User): 2002-02-20

Q(Chatbot): What is your gender?

A(User): male

Q(Chatbot): What is your favorite number?

A(User): 88

3. Chatbot can summarize your personal information.

Chatbot: Bob is a 20 years old young man. His favorite number is 88, which is an even number.

Hints:

1. Chatbot supports at least 2 birthday formats e.g. YYYY-MM-DD or YYYYMMDD, and calculates the correct age.

2. Chatbot can recognize at least 2 gender formats, for example m/boy/man for male, and convert it to decent description of a person, e.g. “young man” in the sentence

3. Chatbot can know your favorite number is odd or even.

4. Consider corner case and error handling as much as you can, to make your chatbot more robust and user friendly. For example if the user's input is an invalid birthday string, the Chatbot should return a hint sentence

like “Please input a valid birthday format (YYYY-MM-DD)” .

5. Feel free to use any knowledge of programming like if...else, string, list etc.