ICT1002 Programming Fundamentals - Lab 1

Topics:

1. Variables/types/converter
2. Operators
3. Conditional/selection
4. Input/output from the keyboard & data formatting

Warmup exercises:

The following lab assignment requires the use of all topics discussed so far in the module. You may wish to practice some of the concepts with simple exercises before attempting the lab assignment. You are not required to include these exercises in your submission, though you may wish to do so, to help you in the test.

1. Use the Python shell to evaluate the following:

```
a. print ('Hello')          ________
b. print ("Hello")          ________
c. print ('Let's Go')             ________
d. print ("Let's go!")      ________
e. print ‘hello’            ________
f. print ('Hello World\n')            ________
g. print ('\nHello World')            ________
h. print ('Hello\nWorld')       ________
i. print (1, '\n', 2, '\n', 3)        ________
j. print ('\n', 1, '\n', 2, '\n', 3)  ________
```

2. Compute the following:

```
a. 1976 / 10   ______          e. 1976.0/10.0 ______
b. 1976 // 10  ______          f. 1976.0//10  ______
c. 1976 % 10   ______          g. 1976.0%10   ______
d. 1976.0 / 10 ______          h. 4**3 ______
```

3. Perform the following type conversions:

```
a. int(5.7)     ______          d. float(5)     ______
b. int(‘5’)     ______          e. float(‘5’)  ______
c. int(‘5.7’)   ______          f. float(‘5.7’)______
```

4. Perform the following logic operations:

a. True or False _____
b. True or 0 _____
c. True or 1 _____
d. true and False _____
e. True and True _____
f. True and 0 _____
g. True and 0 _____
h. not True _____
i. not False _____
j. not 1 _____
k. not 0 _____
l. False and 1 _____
m. False or 1 _____
n. False or a=1 _____
o. `False or (a=0)`

5. Check the ASCII codes of the following characters (Refer to the reading material in the lecture notes):

a. '1'	___	g. 'A'	___
b. '2'	___	h. 'B'	___
c. '3'	___	i. 'C'	___
d. 'a'	___	j. '!'	___
e. 'b'	___	k. '@'	___
f. 'c'	___	l. '#'	___

6. Write an input statement that requests the user to enter a name. Save the name into a variable 'firstName' and print the name (write the lines of code below): HINT- input() is used to input something from keyboard.

7. Write an input statement that requests the user to enter an age. Save the age as an integer in a variable 'age'. Increment the age by one and print the result (write the lines of code below):

8. Write an input statement that requests the user to enter his salary. Save the salary as an integer in a variable 'salary'. If the salary is smaller than 1000, increment the salary by 500 and print the new salary. If the salary is bigger than 5000, decrease the salary by 600 and print the new salary. If the salary is 6000, directly print his salary. Otherwise, increment the salary by 20 and print the new salary. (write the lines of code below and IDLE):

Lab Assignment:

To help you better practice, you need to perform a set of tasks in one auto-grading system, CodeDr. CodeDr will provide you immediate feedback of your program, such that you will know the issue of your program. Below is the link for you to access the system:

http://172. 27. 54. 87/codeDr/public

PS. The codeDR system is only accessible via the ICT network. You can access it outside via the ICT VPN. The ICT VPN account confidential should have been sent to your sit email. If you have not or have any issue using the ICT VPN, please contact ICT program professional officer: Remy Remy.Mohamed@singaporetech.edu.sg.

For your CodeDr account, please get it from your lab instructor in the lab. Please remember your user name and password for the future usage of the system. And, you are not allowed to change the password.

In this lab, you need to finish three tasks in CodeDr system, including the 1-1: Average Calculator, 1-2: BMI Calculator and 1-3:Weekly Payment Calculator.

P.S. CodeDr system will provide you immediate feedback about the correctness of your program. You can view the feedback of your system by clicking the view detail beside your grade. CodeDr adopts a test cases-based approach to check your program. Your grade is depending on the number of test cases that you program can pass. Note that to train you to have a good programming practice, you have to write your program strictly according to the requirement of the tasks, including your input and output format. If there is any difference (even one more or less space), your program will fail on the test cases. SO TRAIN YOURSELF TO BE AN EXACT THINKER!

To help you practice, you are allowed to do multiple attempts for each task. Enjoy your learning!

You can use either the Python packaged IDLE (Python GUI) or PyCharm to create the Python program. To create one new program, you can create a new file by clicking the NEW FILE under the FILE menu. See below figure.