

ICT1002 Programming Fundamentals

Lab 5

Topics:

1. Lambda Expression

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 lab test.

1. The following examples of writing higher order functions and lambda expression may help your better understand the basics covered by the lecture. Please evaluate the following code, and explain and understand why the output is that:

```
>>> def func(x):  
        return x+5  
  
>>> func(20)
```

Ans:

```
>>> fun=lambda x:x+5  
>>> fun(20)
```

Ans:

```
>>> map(lambda x:x*2, [10,20,30,40])
```

Ans:

```
>>> map(lambda x,x+30, [3,4,6,7])
```

Ans:

```
>>> filter(lambda x: x<=20, [10,20,30,40])
```

Ans:

```
>>> reduce(lambda x,y: x+y, [10,20,30,40])
```

Ans:

```
def increment(x):  
    return x+100  
def double(x):  
    return x*2  
def getBonus(func, salary):  
    bonus = 1000  
    if func(salary) > 5000:  
        return func(salary)+ bonus*2  
    else:  
        return func(salary)+bonus  
  
print (getBonus(increment, 3000))  
print (getBonus(double, 3000))  
print (getBonus(increment, 6000))  
a.) print (getBonus(double, 6000))
```

```
def getBonus(salary):  
    bonus = 1000  
    if (lambda x: x+100)(salary) > 5000:  
        return (lambda x: x+100)(salary) + bonus*2  
    else:  
        return (lambda x: x+100)(salary) + bonus  
  
print (getBonus(3000))  
b.) print (getBonus(6000))
```

c.) Similarly, rewrite the `getBonus(salary)` function in (c) to replace the `lambda x:x+100(salary)` into another lambda expression to double the number of the salary like the `double` function in (b). And evaluate `print (getBonus(3000))` and `print (getBonus(6000))` then see whether you can get the same result as the `print (getBonus(double,3000))` and `print (getBonus(double,6000))` in (b).

d.) If you still cannot understand these concepts well to this end, please refer to your lecture notes to try all the examples there.

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 from 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.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 two tasks in CodeDr system, including the [Lab5_Task1](#), and [Lab5_Task2](#).

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 your 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!