ICT1002 Programming Fundamentals

Lab 3

Topics:

1. Functional Abstraction
2. Functions and modules
3. Higher Order Functions

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. Define one function to calculate the division of given two numbers, return the result:

2. Define one function to print out all the element for one given list – the input argument is one list and nothing needs to be returned.

3. Evaluate the following code and figure out the output and explain why?

```
def printMax(a, b):
    if a > b:
        print(a, 'is maximum')
    elseif a == b:
        print(a, 'is equal to', b)
    else:
        print(b, 'is maximum')
```

Evaluate the following expressions

printMax(3,4)	printMax(3)
printMax(3,3)	printMax(3,4,5)
printMax(4,4)	printMax('charlie', 'hello')

4. Evaluate what is the output of the following code and understand why

```
def say(message, times = 2):
    print(message * times)

say('Hello')
say('World', 5)
```

5. Evaluate the following code and understand why

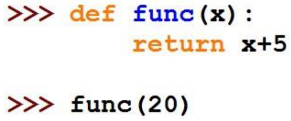
```
def func(a, b=5, c=10):
    print('a is', a, 'and b is', b, 'and c is', c)

func(3, 7)
func(25, c = 24)
func(c = 50, a = 100)
```

6. Evaluate what is the output of the following program and explain why?

```
def func(a, b, names):
     a = a+10
     b = b+20
     names[0] = 12
     names[1] = 18
     return a,b

x,y=10,30
fruits=['apple', 'orange', 'banana']
num1, num2=func(x,y,fruits)
print num1, num2
for fruit in fruits:
     print fruit
```

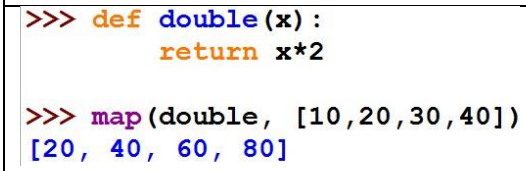
7. According to lab2 Task 1, write one function to decide whether one year is a leap year or not. If it is one leap year, return true. Otherwise, return false.

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



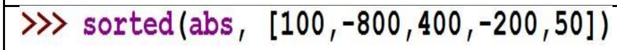
```
>>> def func(x):
        return x+5

>>> func(20)
```

Ans:



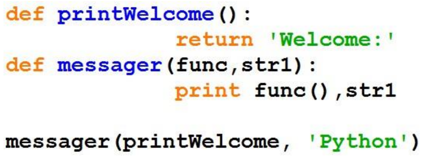
```
>>> def double(x):
        return x*2

>>> map(double, [10,20,30,40])
[20, 40, 60, 80]
```

Ans:



```
>>> sorted(abs, [100,-800,400,-200,50])
```

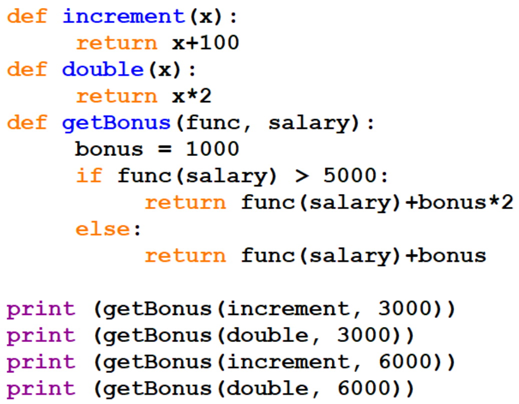
Ans: If there is an error, then fix it.



```
def printWelcome():
            return 'Welcome:'
def messager(func,str1):
            print func(),str1
```

a.)

```
messager(printWelcome, 'Python')
```



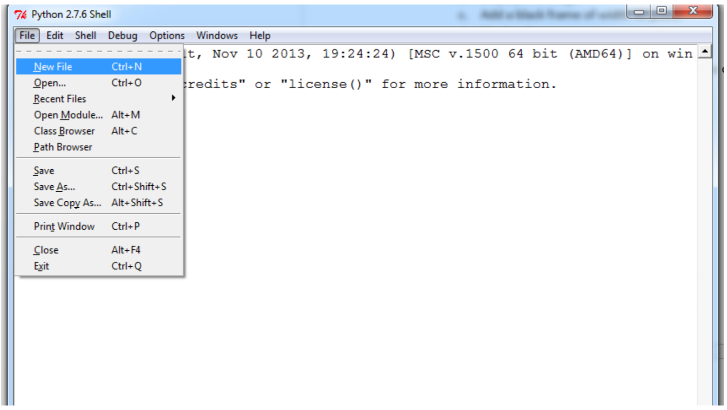
```
def increment(x):
     return x+100
def double(x):
     return x*2
def getBonus(func, salary):
     bonus = 1000
     if func(salary) > 5000:
          return func(salary)+bonus*2
     else:
          return func(salary)+bonus

print (getBonus(increment, 3000))
print (getBonus(double, 3000))
print (getBonus(increment, 6000))
print (getBonus(double, 6000))
```

b.)

Lab Assignment:

In this lab, there will be three tasks that required you to submit. You can download all the two tasks (task 1 and task 2) via LMS Content/Labs/Lab3-submitted tasks. Task 3 is the bonus task, which is a little bit more challenging. If you can do it, you shall be pound of yourself. 😊 In the end, you need to submit your solutions to LMS Assessment/dropbox/Lab3_Px_submission fold where Px is your lab number. Please ensure that you submit your work to the correct folder according to your own lab group. If you can solve task 3, please submit it as well.

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.



```
t, Nov 10 2013, 19:24:24) [MSC v.1500 64 bit (AMD64)] on win

redits" or "license()" for more information.
```