219116/117 Programming II

Lab 3

Answer question 1 in a file named StudentID_Firstname_lab3_ans.pdf, where StudentID is your KU ID and Firstname is your given name

1. **What Would Python Display?**

Analyze pieces of Python code below and figure out what would Python display? Then, verify your answer by running the code in Python interactive mode (python3 -i) If you got it wrong, put a short note explaining what you misunderstood.

```
>>> lambda x: x # A lambda expression with one parameter x
```

```
Predict: Error
```

```
Actual:<function <lambda> at 0x7fa561db68b0>
```

_____

```
>>> a = lambda x: x # Assigning the lambda function to the name a
>>> a(5)
```

```
Predict: 5
Actual: 5
```

_____

```
>>> (lambda: 3)() # Using a lambda expression as an operator in a call exp.
```

```
Predict: 3
Actual: 3
```

_____

```
>>> b = lambda x: lambda: x # Lambdas can return other lambdas!
>>> c = b(88)
>>> c
```

```
Predict: Error
Actual: <function <lambda>.<locals>.<lambda> at 0x7fcad74ac8b0>
```

_____

```
>>> c()
```

Predict: 88

Actual: 88

_____

```
>>> d = lambda f: f(4) # They can have functions as arguments as well.
>>> def square(x):
... return x * x
>>> d(square)
```

Predict: 16

Actual: 16

_____

```
>>> x = None
>>> x
>>> lambda x: x
```

Predict: Error

Actual: <function <lambda> at 0x7fe2794b69d0>

_____

```
>>> z = 3
>>> e = lambda x: lambda y: lambda: x + y + z
>>> e(0)(1)()
```

Predict: 1

Actual: 4

Forgot about z that equal to 3

_____

```
>>> f = lambda z: x + z
>>> f(3)
```

Predict: Error

```
Actual: Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<stdin>", line 1, in <lambda>
TypeError: unsupported operand type(s) for +: 'NoneType' and 'int'
```

```
>>> higher_order_lambda = lambda f: lambda x: f(x)
>>> g = lambda x: x * x
>>> higher_order_lambda(2)(g) # Which argument belongs to which function call?
```

Predict: The first lambda is key and can't take int and float.

```
Actual:Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<stdin>", line 1, in <lambda>
TypeError: 'int' object is not callable
```

```
>>> higher_order_lambda(g)(2)
```

Predict: 4

Actual: 4

_____

```
>>> call_thrice = lambda f: lambda x: f(f(f(x)))
>>> call_thrice(lambda y: y + 1)(0)
```

Predict: 3

Actual: 3

_____

```
>>> print_lambda = lambda z: print(z) # When is the return expression of a  lambda expression executed?
>>> print_lambda
```

Predict: z has no value

Actual: <function <lambda> at 0x7f8644db6ca0>

_____

```
>>> one_thousand = print_lambda(1000)
```

Predict: 1000

Actual: 1000

_____

```
>>> one_thousand
```

Predict: None

Actual: Nothing

_____

```
>>> def even(f):
      def odd(x):
            if x < 0:
                  return f(-x)
            return f(x)
      return odd
steven = lambda x: x
stewart = even(steven)
```

Stewart

Predict: Error

Actual: <function even.<local>.odd at 0x109af70d0>

```
_____
>>> stewart(61)
Predict: 61
Actual:61
_____
>>> stewart(-4)
Predict: -4
Actual: 4


_____
>>> def cake():
... print('beets')
... def pie():
```



```
... print('sweets')
... return 'cake'
... return pie
>>> chocolate = cake()
Predict: beets
Actual: beets
_____
>>> chocolate
Predict: Error
Actual: <function cake.<locals>.pie at 0x7fb832eac940>
_____
>>> chocolate()
Predict: cake pie
Actual: sweet ‘Cake’


_____
>>> more_chocolate, more_cake = chocolate(), cake
Predict: sweets
Actual: sweets
_____
>>> more_chocolate
Predict: cake
```

```
Actual: 'cake'

______

>>> def snake(x, y):
... if cake == more_cake:
... return chocolate
... else:
... return x + y
>>> snake(10, 20)
Predict: Error
```
Actual: <function cake.<locals>.pie at 0x7fb832ea39ACA0>

```
______

>>> snake(10, 20)()
Predict: sweet 'Cake'
Actual: sweet 'Cake'

______

>>> cake = 'cake'
>>> snake(10, 20)
Predict: 30
Actual: 30

______
```

Do not proceed to the next stage until you fully understand the above code and its behavior. Ask the instructor or the TAs if you are confused about any part of the code.

2. lab3.py

Complete the missing code in lab3.py and make sure that it passes all the test cases. **You must use higher-order functions or lambda expressions to get credit for this problem.**

Submission:

- ***Create StudentID_Firstname_lab3 folder, where StudentID is your KU ID and Firstname is your given name***
- ***Put the files to submit, StudentID_Firstname_lab3_ans.pdf and lab3.py, into this folder***
- ***Zip the folder and submit the zip file to the course's Google Classroom before the due date***