

**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
777 <function <lambda> at 0x....>
```

*Handwritten notes:  $\lambda$  expression, parameter x, return x*

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

*Handwritten notes: parameter x, return x*

```
>>> (lambda: 3)() # Using a lambda expression as an operator in a call exp.
777 3
```

*Handwritten notes:  $\lambda$  expression, parameter x, return x*

```
>>> b = lambda x: lambda: x # Lambdas can return other lambdas!
>>> c = b(88)
>>> c
88 <function <lambda>. <locals>. <lambda> at 0x....>
```

```
>>> c()
88
```

```
>>> d = lambda f: f(4) # They can have functions as arguments as well.
>>> def square(x):
...     return x * x
>>> d(square)
16
```

```
>>> x = None
>>> x
>>> lambda x: x
777 <function <lambda> at 0x....>
```

*Handwritten notes:  $\lambda$  expression, parameter x, return x*

```
>>> z = 3
>>> e = lambda x: lambda y: lambda: x + y + z
>>> e(0)(1)()
4
```

```
>>> f = lambda z: x + z
```

```
>>> f(3)
```

x is not defined so return error

```
>>> 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?
```

Error

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

4

```
>>> call_thrice = lambda f: lambda x: f(f(f(x)))
```

```
>>> call_thrice(lambda y: y + 1)(0)
```

3

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

```
>>> print_lambda
```

<function <lambda> at 0x... >

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

1000

```
>>> one_thousand
```

None

```
>>> 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
```

<function even.<locals>.odd at 0x... >

```
>>> stewart(61)
```

61

```
>>> stewart(-4)
```

4

```
>>> def cake():
```

```
...     print('beets')
```

```
...     def pie():
```

```

...     print('sweets')
...     return 'cake'
...     return pie
>>> chocolate = cake()
sweets
>>> chocolate
<function cake.<locals>.pie at 0x...>
>>> chocolate()
sweets ✓
'cake' (returned)
>>> more_chocolate, more_cake = chocolate(), cake
sweets
>>> more_chocolate
'cake'
>>> def snake(x, y):
...     if cake == more_cake:
...         return chocolate
...     else:
...         return x + y
>>> snake(10, 20)
<function cake.<locals>.pie at 0x...>
>>> snake(10, 20)()
sweets
'cake'
>>> cake = 'cake'
>>> snake(10, 20)
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