|  |  |  |
| --- | --- | --- |
| **CS 112, Foundations of CS**  **Lab 5: Lambda Functions**  **Submit to Canvas** | | **Computer Science** |
|  |  | |

This lab is worth 100 points. The goal for this lab is:

1. **Writing anonymous functions (lambdas)**

If you do not complete the lab in the time allotted, then please return to the lab in your spare time, and complete it by the **due date, which is specified on Canvas**.

**Preliminaries**

* For this lab, open up the Spyder IDE and create another folder, called **lab5.**

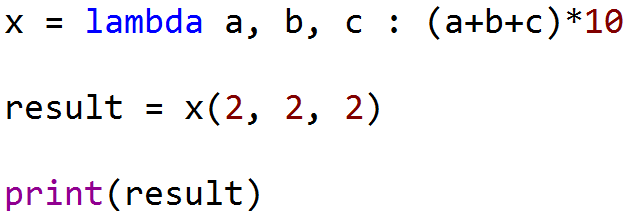
1. **Writing anonymous functions**

For this programing task you will write and test a few lambda functions. Lambda functions, also known as anonymous functions, are very handy in Python when you need a quick function.

Lambda functions have three parts. First part is the “**lambda**” keyword. The second part is the **argument list**, it can have as many or as little parameters as the programmer needs. Arguments are coma separated, if there is more than one. The third part is the **expression**, a lambda function can only have one expression. The argument list and the expression is separated by a colon. A lambda function always returns. Take a look at the example below of a lambda function.



The above lambda function has three arguments (a, b, c), and the expression is (a+b+c)\*10. This function takes in 3 values, adds them together and multiples it by 10. Take a look below to see an example of calling the function and printing out the result.





1. To get started create a new file and save it as ***AnonymousFunctions.py***.
2. In the comment section on top, include your name, todays date and program name (Lab 7, Anonymous Functions.py).
3. Implement the functions from the table below. The second row (highlighted yellow) in the table contains the description for function “**x**”, which is an example on the previous page. Use the information from the table to test your functions once you have written them. Simply call them inside of a print() function. Each print statement should have a message, use the message from the “**Test message**” column from the table below. **You should write a total of 7 lambda functions and call them using 7 print statements. This entire lab can be written on 14 lines of code (excluding comments on top), but do use blank lines to separate your statements.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function name** | **Number of Parameters** | **Expression** | **Test message** | **Test input** | **Output** |
| x | 3 | Adds the 3 parameters together and multiples it by 10 |  | 2, 2, 2 | 60 |
| addTwo | 2 | Adds the 2 parameters together | “Add two : ” | 9, 9 | 18 |
| average | 3 | Adds the 3 parameters together and divides it by 3 | “Average three : ” | 1, 2, 3 | 2.0 |
| powerThree | 1 | Calculates the parameter to the power of 3 | “Power of three : ” | 5 | 125 |
| info | 0 | A string literal “Hello World” | “Info : ” |  | “Hello World” |
| lowerStr | 1 | Strips the white spaces from the parameter and makes it lower case. | “Lower case string : ” | “ CWU ” | “cwu” |
| subUpperStr | 1 | Strips the white spaces from the parameter, makes it upper case and get substring (index 1:4) | “Upper sub string : ” | “ ccWucentral ” | “CWU” |
| halfList | 1 | Gets the first half of the list. | “First half of the list : ” | [1, 2, 3, 4, 5,6] | [1, 2, 3] |

Description of methods/techniques which you will use to complete this assignment :

**x.strip()** – removes the white spaces from a string literal “stored” in x

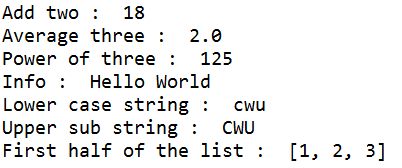
**x.lower()** – makes the string literal lower case

**x.upper()** – makes the string literal upper case

**len(x)** –returns number of elements in the list “x”

**x[0:2]** – returns a sub list, index 0 up to not including 2

1. Sample output of your program can be seen below.



**Rubric**

Upload your source code to Canvas. Here's what we are looking for, when grading your submission

The .py file must be thoroughly commented. If your code breaks (crashes) because you've been unable to fix a syntax error, then the comments will allow you to receive partial credit.

For this lab, make sure that the following file is uploaded to Canvas:

*AnonymousFunctions.py*

|  |  |
| --- | --- |
| **File / task** | **Points** |
| 1. **AnonymousFunctions.py** | 100 |
| Total | 100 |