Practice makes your coding skill perfect. So, whenever you are given exercises, please try and practice them.

Remember that computer coding skills need a systematic approach.

Therefore, every week's learning is built on the current week and previous week learning.

CSE5APG

Week06: Lab06

Function and Functional Programming

OBJECTIVES

- Implement Input, Process and Output steps.
- Learn how to use Function.
- Learn how to write Function.
- Learn how to use Functional Programming.

Function Programming

1. Review

a. What is a function?

b. How many parameters can a function have?

c. How many return statements can a function have?

d. What is a default parameter?

e. If your function has default parameters as well as non-default parameters, what rule should you observe?

f. What is a positional argument?
What is a keyword argument?

g. If your function call has positional arguments as well as keyword arguments, what rule should you observe?

h. What is a local variable?
What is its scope?

i. What is a global variable?
How can you change a global variable inside a function?

----

2. Print Repeatedly

a. Define a function called **printRepeat** that takes a message (a string) and a number ***n*** and prints the message ***n*** times, each on a separate line.

b. Make a function call using positional arguments only.

c. Make a function call using keyword argument only.

d. Make a function call using a mixture of both kinds of arguments.

---

3. Print Repeatedly (Again)

Modify the function in the previous question so that parameter ***n*** has the default value of 1.

---

4. Chat

What kind of variable is myName in function chat? You can delete message and run the program.

```
def chat(msg):
    print(myName, "writes:")    # line 2
    print(msg)
    print()

# call
myName = "CSE5APG"
message = \
    """
    Hello, Everyone!
    Keep in touch.
    """

chat(message)
```

---

5. The Maximum

a. Write a function, called **max2**, that returns the maximum of the two numbers. Do not use the built-in function **max**.

b. Using your previous function, write a function, called **max3**, to return the maximum of three numbers. Do not use the built-in function **max**.

---

6. Removing Duplicates

Write a function that takes a list and returns a list that is the same as the one it receives but with all duplicate elements removed. In other words, if we run the following test program

```
def  removeDuplicates(receivedList):
    # TODO


#  main
list1  =  [2,  4,  1,  3,  5,  2,  3]
print("‘list1:"’,  list1)
# call  the  function
list2  =  removeDuplicates(list1)

# list2  has  no  duplicates print("list2:",  list2)
# list1  remains  the  same print("list1:",  list1)
```

we should get the output

```
list1: [2, 4, 1, 3, 5, 2, 3]
list2: [2, 4, 1, 3, 5]
list1: [2, 4, 1, 3, 5, 2, 3]
```

---

7. Displaying List Vertically

a. Write a function called **show1** that takes a list and displays the list elements vertically, each on one line. Call the function with argument **dir(**__builtins__**)**. In other words, use **dir(**__builtins__**)** as your input list, e.g., l= **dir(**__builtins__**).**

b. Modify the function (call it **show2**) to display only strings that begin with a lowercase letter. Call the function with argument **dir(**__builtins__**)**

---

8. Rounding to Nearest Five Cents

a. Write a function, named **adjust**, that takes a number of cents between 0 and 9, inclusive, rounds it off to the nearest 5 cents by the following rules, and returns the result:

   - 1 and 2 cents are rounded off to 0
   - 3, 4, 6 and 7 cents are rounded off to 5
   - 8 and 9 cents are rounded off to 10 cents

---

b. Write a function named **roundOff** that takes an amount of money in dollars and rounds it off to the nearest 5 cents, and returns the result. The amount is entered as a decimal number with 0, 1 or 2 decimal places. The rounding off is based on the rules given above.

----

9. Account Balance

a. Write a function to calculate the balance of a bank account. The function takes (as parameters)

- The initial balance
- The monthly interest rate (as a decimal, e.g. 5% is represented as 0.05)
- The number of months since the start of the account (an integer).

The function returns the balance, rounded to 2 decimal places.

b. An account starts with 1000 dollars.

Write a program to show the balance of the account

- for each month of the first year (1 to 12 months), and
- for three different monthly interest rates: 0.02, 0.05 and 0.10.

Make the output should look like this (or better):

```
Initial amount:  1000
month    2%          5%          10%
1    1020.00     1050.00     1100.00
2    1040.40     1102.50     1210.00
3    1061.21     1157.63     1331.00
4    1082.43     1215.51     1464.10
5    1104.08     1276.28     1610.51
6    1126.16     1340.10     1771.56
7    1148.69     1407.10     1948.72
8    1171.66     1477.46     2143.59
9    1195.09     1551.33     2357.95
10   1218.99     1628.89     2593.74
11   1243.37     1710.34     2853.12
12   1268.24     1795.86     3138.43
```

-- ---

10. Loan Payment

a. A loan is taken out for an amount of money with a fixed monthly interest rate and monthly payment. The monthly payment required to pay off the loan within a specified number of months is calculated by the formula (*p*):

$$P = \frac{A \times R}{1 - (1 + R)^{-M}}$$

where

- P is the payment amount per month
- A is the amount of loan
- R is the monthly interest rate, enter as a decimal (e.g. 0.05 for 5%)
- M is the number of months

Write a function that takes the amount, interest rate and the number of months and returns the required monthly payment.

b. Bob borrows 5000 dollars with a monthly interest rate of 0.05.

How much does he need to pay per month? And what is the total payment amount? if he pays the loan in:

- 1 year
- 2 years
- 3 years

----

11. Read Integer in a Range

a. Write a function that takes two numbers, **low** and **high**. The function asks the user for an **int** value between **low** and **high**, inclusive. The function keeps asking until a number in the specified range is entered. The function returns the valid int value entered by the user.

b. Write a program that asks for a student's name and a mark between 0 and 100. The program should keep asking until a valid mark is entered. Print out the name and the mark.

Sample run:

```
Enter the name:    Bob
Enter integer between 0 and 100: -10
Enter integer between 0 and 100: 110
Enter integer between 0 and 100: 80
name: Bob mark: 80
```

----

12. Prime Numbers

a. Write a function that takes an integer and returns **True** if a number is a prime number and **False** otherwise. For example, you can assume that the number is >= 2.

b. Write a function that takes an integer n (assumed to be a positive number). The function returns the next prime number that is bigger than n.

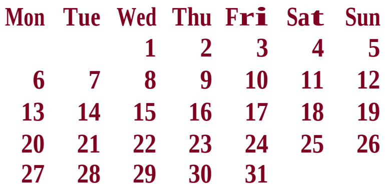
----

13. Printing a monthly Calendar

Write a function that prints the calendar of a month. The function receives as input

- Number of days in the month
- An integer indicating the day-of-week of the first day: 0 for Monday, 1 for Tuesday, etc.

For example, if the month has 31 days and the first day of the month is on Wednesday (entered as 2), the following calendar is displayed



```
Mon  Tue  Wed  Thu  Fri  Sat  Sun
                 1    2    3    4    5
  6    7    8    9   10   11   12
 13   14   15   16   17   18   19
 20   21   22   23   24   25   26
 27   28   29   30   31
```

Functional programming (Lambda, Map and Filter)

Review

1- lambda (): a single line function (aka functions without a name).

- Write a lambda function (fun) returns the sum of two parameters:

```
>>> fun = lambda x, y: x + y

>>> fun (1,4)

>>> 5
```

2- **map ()**: applies a function to all the items in an input list.

```
x = map (function, list)
```

- Write a map function to convert a set of temperature values from Fahrenheit to Celsius.

```
def Celsius(T):

    return (float (5)/9) *(T-32)

F = (80.5, 67, 77.5,89)

C = map (Celsius, F)
```

3- filter (): creates a list of elements for which a function returns true

```
f = filter (function, list)
```

- Given a list of fruit names (fruit = ["Apple", "Pear", "Orange", "Banana", "Apricot"]), write a filter function to create a new list containing only those start with the letter “A”.

```
def starts_with_A(x):
      return x[0] == "A"
fruit = ["Apple", "Pear", "Orange", "Banana", "Apricot"]
new_list = filter (starts_with_A, fruit)
new_list = ["Apple", "Apricot"]
```

A- Write a lambda function to calculate the cube of a given number.

B- Write a lambda function to calculate the power of a given number.

C- Write a map function to calculate the square root for list of elements, e.g. l= [4,6,9,12,15,25].

D- Can we use Lambda with Map to calculate the square root?

E- Write a filter function to remove 'a' letter from the given list. l=['a','b','c','d','a','e','f,'a']