QUEENSBOROUGH COMMUNITY COLLEGE
The City University of New York
Department of Engineering Technology

Programming Exercises - Functions II

1. Print List
a) Define a function ***printList()*** with **one** parameter *p*.
    1) This function **prints** all values in the list *p*.
b) Define a *main*() function to do the following:
    1) Create a **list** of strings called *lst*. For example, lst = [“apple”, “banana”, “cherry”].
    2) Call the function *printList()* with *lst* as an argument.
c) Call *main*() function to initiate the tasks to be performed.

```
Example Output
apple banana cherry
```

<u>Data Management Functions</u>

2. Name format
a) Define a function ***nameFormat()*** with parameters ***first***, ***middle***, and ***last***.
    1) This function **prints** the first name, the middle initial and the last name using proper **title** format.
b) Define a *main*() function to do the following:
    1) Call the function *nameFormat* with these **positional** arguments: john stu smith
    2) Call the function *nameFormat* with these **keyword** arguments:
        last = ‘kennedy’, first = ‘john’, middle = ‘fitzgerald’
c) Call *main*() function to initiate the tasks to be performed.

```
Example Output
John S. Smith
John F. Kennedy
```

3. Name format
a) Define a function ***nameFormat()*** with parameter *first*, last and middle where middle is an **optional parameter**.
    If all three names are provided **return**: *Last, First, M.*
    If only first and last are provided **return**: *Last, First*
b) Define a *main*() function to do the following:
    1) Call the function with **keyword** arguments for the name: *james bond*
    2) Call the function with **keyword** arguments for the name: *henry indiana jones*
    3) Print the results of the function calls.
c) Call *main*() function to initiate the tasks to be performed.

```
Example Output
Bond, James
Jones, Henry, I.
```

4. Print **Arbitrary** Values
a) Define a function ***printNames()*** with a parameter *names*.
The *names* parameter builds a **tuple** of any number of argument values.
This function **prints** all contents of the *names* **tuple**.
b) Call the function *printNames()* with any number of name arguments (see output below).

```
Example Output
Ann Bianca Coco Dora Emily
```

5. Dictionary
a) Define a function *createUser()* with an **arbitrary** dictionary parameter.
This function **returns** a **dictionary** based upon input arguments.
b) Define a function *printUser()* with a parameter *user* which is a dictionary.
This function prints the contents of the dictionary *user*.
c) Create and print the user: *John, age 43, job Programmer, Hobby Biking*
d) Create and print the user: *Sara, age 20, school QCC, major CSIS*

```
Example Output
name: John
age: 43
job: Programmer
hobby: Biking

name: Sara
age: 24
school: QCC
major: CSIS
```

Computational Functions

6. Averaging
a) Define a function ***average()*** with a parameter *grades* that can accept an **arbitrary** number of integer values.
This function **returns** the average of all values.
b) Define a *main*() function to do the following:
1) Call the *average*() function with the following arguments: 95,87,83,74
2) Create three random integers, *x*, y, and *z*. *x* is from range -100 to -1, y is from range 0 to 1, and *z* is from range 1 to 100 inclusively.
3) Call the *average*() function with *x*, y, and *z*.
4) Print al the results with the average computed to **two** decimal places.
c) Call *main*() function to initiate the tasks to be performed.

```
Example Output
Average of 95,87,83,74: 84.75
Average of any three random numbers, -7, 0, 66: 19.67
```