**NOTES:**

* Part 1: use Python’s IDLE interactive tool. Write your answer beside each command in this sheet in **bold**.
* Part 2: upload a separate .py file that contains your script/code.

**PART 1:**

1. **Dictionary in Python:** Define the following *dicts*:

*#dictionary literals*

d1={"name": "Lisa", "age": 22, (3, 'm'):['a1', 'b2', 'c3'], 5: "Seasons", 20: 556, 12:64}

*#dictionary using sequences*

d2 = dict([("name","Chris"), ('age', 36), ((1,2), ['uu', 'vv', 'ww']), (0, 'bike'), (86, 50)])

*#dictionary using keywords*

d3 = dict(id=123, name='James', siblings=['Adam', 'Bob', 'Carly'])

*#dictionary using zip( ) function*

d4 = dict(zip(("id", "name", "quantity"), (2020, "John Smith", 5)))

***Work with dict methods***: Type the following commands at the Python prompt in IDLE interactive mode and indicate the result of each command:

1. d1.keys()

dict\_keys(['name', 'age', (3, 'm'), 5, 20, 12])

1. d2.values()

dict\_values(['Chris', 36, ['uu', 'vv', 'ww'], 'bike', 50])

1. d3.get('id')

123

1. d2.get('age')

36

1. d3.get('age')

empty response without error

1. d3.get('name', 'Tim')

‘James’

1. d2.items()

dict\_items([('name', 'Chris'), ('age', 36), ((1, 2), ['uu', 'vv', 'ww']), (0, 'bike'), (86, 50)])

1. d3['siblings']

['Adam', 'Bob', 'Carly']

1. d2['siblings']

KeyError: 'siblings'

1. d2.update(d3)

dict\_items([('name', 'James'), ('age', 36), ((1, 2), ['uu', 'vv', 'ww']), (0, 'bike'), (86, 50), ('id', 123), ('siblings', ['Adam', 'Bob', 'Carly'])])

1. d2[0]

'bike'

1. d1.get((1,2))

nothing returned

1. d2['siblings']**\***

['Adam', 'Bob', 'Carly']

1. d2['name']**\***

'James'

1. d1 == d2

False

1. len(d2)

7

1. for key in d1.keys():

print(key)

name

age

(3, 'm')

5

20

12

1. for key in d2.keys():

print(d2[key])

James

36

['uu', 'vv', 'ww']

bike

50

123

['Adam', 'Bob', 'Carly']

**PART 2: Write a complete Python script, with comments, to do the following**:

1. Open a text file called “*catalog.txt*”, attached with this lab, for reading. The file contains the items available in a fitness studio, the items categories/classes, and their quantities.
2. Define a list of strings called ***my\_items*** as follows: ***my\_items*** **= ['gym mats', 'rigs', 'boxing gloves', 'ropes', 'treadmill', 'elliptical', 'dumbbell', 'yoga ball']**
3. Create an empty *dict* **d1 = {}**
4. In a loop, do the following:
   1. Check each **item** in ***my\_items*** and see if that **item** matches any of the products in the file.

**Hint**: use the function readline() to read a new line from the file and compare that line with the elements in the list of strings.

* 1. If there is a match, save the category and the quantity corresponding to that item in some variables. Create a *key-value* pair ***item:(category, quantity)***  for dict ***d1.*** Here ***item* (**i.e*.* the *key***)** is the item (string) found in *catalog.txt* and the tuple (***category, quantity)***  (i.e. the *value*) corresponds to the *category* and *quantity* of that **item**. Add the *key-valu*e pair to ***d1*** as {***item:(category, quantity)***}.
  2. Otherwise (i.e. the **item** is not found in *catalog.txt*) print a suitable message indicating that the specified <**item**> is not available in the *catalog*.

1. Next the program should ask the user to enter a string ***s***, (representing a fitness item) as an input.
2. If ***s*** corresponds to a valid *key* in ***d1***, then the program should retrieve the *category* and *quantity* of ***s*** from ***d1*** and display it to the user.

* After displaying the category and quantity corresponding to item ***s***, the program asks the user enter a number ***n*** indicating how many items they want to order.
* If ***n*** is not an integer the program should catch the exception, display an appropriate error message and exit.
* If ***n*** is a negative integer, the program should print ‘Negative values not accepted!’ then prompt the user to input another value for ***n***.
* If ***n*** is a positive integer, the program should print ‘Your order is successful!’ then exit.

1. If the item’s name entered by the user does not correspond to a valid key, the program should catch an exception. When the exception occurs, display an appropriate error message then prompt the user to input another item’s name.