STOP 0 – Lists Laboratory Exercise

```
1 from csc1017fvuladb import csc1017f_vula_list
2 var_vula = csc1017f_vula_list()
3 type (var_vula) # <class 'list'>
4
5 first_record =  # DO THIS
6 total_entries = # DO THIS
7
8 var_vula[10] # 'Xx, Yy;XY;XY@myuct.ac.za;Observer'
9
10 var_record = var_vula[10]
11 type(var_record) # <class 'str'>
12
```


STOP 0 – Lists Laboratory Exercise

```
1 var_record = var_vula[10]
2
3 var_record = var_vula.split(“;”) # string method
4 #['Xx, Yy', 'xy', 'XY@myuct.ac.za', 'Observer']
5
6 var_record[0].replace(“,”, “”)
7 #['Xx Yy', 'xy', 'XY@myuct.ac.za', 'Observer']
8
9 # WHAT FANCY TECHNIQUES DID YOU GOOD PEOPLE USE
10
11
12
```


UCT Department of Computer Science
Computer Science 1017F

Dictionaries

Lighton Phiri <lphiri@cs.uct.ac.za>
April 2015

Real-world Examples

- Online account details
  - Username; password; email
- User records/databases
  - Lists laboratory exercise
- Addresses
  - Name, street, city, zipcode, country
- Unusual examples
  - Windows registry keys; environment variables

- What examples can you good people think of?
  - ???

Introduction

- Dictionaries are used to store lookup information with sets of key-value pairs
- Dictionaries are defined using curly brackets {}
- Key separated from value by colon (:)
- Items (key/value pairs) separated by commas
- Keys are unique; values may not
- Values can be of any data type; keys MUST be immutable
- Keys are used to access values
- Dictionaries are mutable

Creating Dictionaries

```
1 var_dict = {'name':'Xx, Yy', 'user_id':'xy',
'email':'XY@myuct.ac.za', 'role':'Student'}
2 type (var_dict) # <class 'dict'>
3
4 var_another_dict = dict() # empty dictionary
5 type (var_another_dict) # <class 'dict'>
6
```

- Notice the use of the dict() function in Line 4 to optionally create an empty dictionary

Dictionaries in Pespective

```
1 var_dict = {'name':'Xx, Yy', 'user_id':'xy',
'email':'XY@myuct.ac.za', 'role':'Student'}
```

- Dictionaries are defined using curly brackets {}
- Key separated from value by colon (:)
  - Keys—???
  - Values—???
- Items (key/value pairs) separated by commas
  - How many items does var_dict have?

Dictionaries in Pespective

- Keys are unique; values may not

```
1 var_dict = {'name':'Xx, Yy', 'name':'xy',
'email':'XY@myuct.ac.za', 'role':'Student'}
# later duplicate value is used for multiple keys
```

- Values can be of any data type; keys MUST be immutable

```
1 var_dict = {'name':'Xx, Yy', 'user_id':'xy',
'email':'XY@myuct.ac.za', 'role':'Student'}
# This is an error!!!
```

Dictionaries in Pespective

- Keys are are used to access values

```
1 var_dict = {'name':'Xx, Yy', 'user_id':'xy',
'email':'XY@myuct.ac.za', 'role':'Student'}
2 var_dict['name'] # 'Xx, Yy'
3 type(var_dict['name']) # <class 'str'>
```

- Values can be of any data type; keys MUST be immutable

```
1 var_dict = {['name']:'Xx, Yy', 'user_id':'xy',
'email':'XY@myuct.ac.za', 'role':'Student'}
# Error!
type(['name']) # <class 'list'>
```

Dictionaries in Pespective

- Dictionaries are mutable

```
1 var_dict = {'name':'Xx, Yy', 'user_id':'xy',
'email':'XY@myuct.ac.za', 'role':'Student'}
2
3 var_dict['name']= 'Doe, Jane' # updating value
4 var_dict['gender'] = 'Female' # Adding new entry
5 del var_dict['name']; # Remove value with key
'name'
6 var_dict.clear(); # Purge entire dict
```


STOP 1 – Questions

STOP 2 – Methods vs Functions

```
1 import math
2
3 def add (a, b):
4    return a+b
5
6 math.pow(10, 2) # function invocation from module
7 add (1, 1) # standalone function invocation
8 course_name = “Singh, Shekhar”
9 course_name.upper () # method call—class function invocation
```

- Dict **methods** are invocked on instances of dict using “.” operator

Dictionary Methods

- <dict>.get(key, default=None)
  - Return value or default is key not present
  - Parameter(s): key and optional default; Return value: None or default data type

```
1 var_dict = {'name':'Xx, Yy', 'user_id':'xy',
'email':'XY@myuct.ac.za', 'role':'Student'}
2 var_dict.get(“name”) # 'Xx, Yy'
3 var_dict.get(“gender”, default=”Err”) # 'Err'
```

Dictionary Methods

- <dict>.keys()
  - Returns list of dictionary keys
  - Parameter(s): no parameters Return value: List—dict_keys class

```
1 # 1 var_dict = {'name':'Xx, Yy', 'user_id':'xy',
'email':'XY@myuct.ac.za', 'role':'Student'}
2 var_dict.keys() # dict_keys(['User ID', 'Name',
'Email Address', 'Role'])
```

Dictionary Methods

- <dict>.values()
  - Returns list of dictionary values
  - Parameter(s): no parameters Return value: List—dict_values class

```
1  var_dict  =  {'name':'Xx,  Yy',  'user_id':'xy',
'email':'XY@myuct.ac.za', 'role':'Student'}
2 var_dict.values() # dict_values(['XY@myuct.ac.za',
'Student', 'Xx, Yy', 'xy'])
```


Dictionary Methods

- <dict>.items()
  - Returns set-like object of dict items
  - Parameter(s): no parameters Return value: List—dict_keys class

```
1  var_dict  =  {'name':'Xx,  Yy',  'user_id':'xy',
'email':'XY@myuct.ac.za', 'role':'Student'}
2  var_dict.items()  #  dict_items([('User  ID',
'brrale004'),  ('Name',  'Barry  Alexandra'),  ('Email
Address',      'BRRALE004@myuct.ac.za'),      ('Role',
'Student')])
```


Dictionary Methods

- <dict>.pop(key, <default>)
  - Remove and return key
  - Parameter(s): key& default; Return value: value data type

```
1 # Try this
```

- <dict>.popitem()
  - Remove and return arbitrary key/value pair
  - Parameter(s): no parameters; Return value: tuple

```
1 # Try this
```


Dictionary Methods

- <dict>.update(<other>)
  - Update dictionary with key/value from another dictionary—existing keys overwritten
  - Parameter(s): other—optional; Return value: None

- <dict>.copy()
  - Return copy of dictionary
  - Parameter(s): No parameters; Return value: Dict

Dictionary Methods

- <dict>.clear()
  - Purge all items from dictionary
  - Parameter(s): No parameters; Return value: None

```
1  var_dict  =  {'name':'Xx,  Yy',  'user_id':'xy',
'email':'XY@myuct.ac.za', 'role':'Student'}
2 var_dict.clear() # {}
```


STOP 4 – CSC1017 Vula DB 1/2

- How would we convert out Vula list into a dictionary?
  - Suggestions? Ideas?

```
['Marufu,
Anesu;mrfmuf001;MRFMUF001@myuct.ac.za;Support staff',
'Mbogo, Chao;mbgcha002;MBGCHA002@myuct.ac.za;Support
staff', 'Zhou, Yin
Hong;zhxyin002;ZHXYIN002@myuct.ac.za;Student', 'Zide,
Zikhona;zdxzik001;ZDXZIK001@myuct.ac.za;Student',
'Zimuto,
Tanaka;zmttan001;ZMTTAN001@myuct.ac.za;Student']
```


STOP 4 – CSC1017 Vula DB 2/2

```
1 # Follow instruction from laboratory exercise 8 on
how to return list used here...
2
3 var_record = var_vula[35].split(";")
4 # ['Barry, Alexandra', 'brrale004',
'BRRALE004@myuct.ac.za', 'Student']
5
6 var_header = var_vula[0].split(";")
7 # ['Name', 'User ID', 'Email Address', 'Role']
8
9
10
```