Programming and Scripting
Lab Topic 05-Data Structures

Introduction.

The first activity is a quiz, answers are at the end of this lab sheet.

For the programs in the other activities, I would suggest that you create a folder called labs and a subdirectory **topic05-datastructures**, or (**week05-datastructures,** whichever you wish) in your mywork directory.
You can save the programs you create in this lab in there.
I do not correct the programs you write for the labs, just the weekly tasks.

Quiz:

1. Look at this code and answer the questions below, answers at the end of the lab sheet

```
numberOfQuestions = 5
averageAge = 23.4
debugMode = True
name = "joe"
ages = []
months = ('Jan', 'Feb', 'Mar')
book = {}
stuff = [ 12 , 'Fred', False, {}]
someone = dict(firstname = "joe")
me = {
    "firstName" : "Andrew",
    "teaching"  : [{
        "courseName" : "programming",
        "semester" : 1
    },{
        "courseName" : "Data Representation",
        "semester" : 2
    }
    ]
}
```

Questions

What are the variable types of the following variables in the code above

a. numberOfQuestions
b. averageAge
c. debugMode
d. name
e. ages
f. months
g. months[1]
h. book
i. stuff
j. stuff[2]
k. someone
l. someone["firstname"]
m. me
n. me["teaching"]
o. me["teaching"][0]["semester"]

p is a trick question look at it carefully

p. me["teaching"][0]["coursename"]

2. Create a tuple that stores the months of the year, from that tuple create another tuple with just the summer months (May, June, July), print out the summer months one at a time.

```
May
June
july
```

Answer

```
months =("January",
             "February",
             "March",
             "April",
             "May",
             "June",
             "July",
             "August",
             "September",
             "October",
             "November",
             "December"
)
summer = months[4:7]
for month in summer:
    print(month)
```

3. Create a program that puts 10 random numbers into a queue(list), the program should then output all the values in the queue, then take the numbers from the queue one at a time, print it and the current numbers still in the queue. (the command pop(0) takes the first element out of a list)

```
queue is [17, 73, 31, 89, 42, 19, 83, 86, 49, 62]
current Number is 17 and the queue is [73, 31, 89, 42, 19, 83, 86, 49, 62]
current Number is 73 and the queue is [31, 89, 42, 19, 83, 86, 49, 62]
current Number is 31 and the queue is [89, 42, 19, 83, 86, 49, 62]
current Number is 89 and the queue is [42, 19, 83, 86, 49, 62]
current Number is 42 and the queue is [19, 83, 86, 49, 62]
current Number is 19 and the queue is [83, 86, 49, 62]
current Number is 83 and the queue is [86, 49, 62]
current Number is 86 and the queue is [49, 62]
current Number is 49 and the queue is [62]
current Number is 62 and the queue is []
the queue is now empty
```

Answer

```
import random
queue = []
numberOfNumbers=10
rangeTo=100

for n in range(0,numberOfNumbers):
    queue.append(random.randint(0,rangeTo))

print (f"queue is {queue}")

while len(queue) != 0:

    currentNumber = queue.pop(0)
    print ("current Number is {currentNumber} and the queue is {queue} ")

print ("the queue is now empty")
```

This does not output correctly. What is wrong with this line

4. Write a program that stores a student name and a list of her courses and grades in a dict, the program should then print out her data.

The number of course she has could change.

```
Student: Mary
    Programming   : 45
    History       : 99
```

Answer

```
student = {
    "name":"Mary",
    "modules": [
        {
            "courseName":"Programming",
            "grade": 45
        },
        {
            "courseName":"History",
            "grade":99
        }
    ]
}
print ("Student: {}".format(student["name"]))
for module in student["modules"]:
    print("\t {} \t: {}".format(module["courseName"], module["grade"]))
```

Extra

5. Write a program that will read in the data for the data structure above, ie reads in a student's name, then keeps reading in their modules and grades (until the user enters a blank module name),
   You can break this up into two parts:
   a. Just read in the module names until the user enters blank,
   b. Then read in the grade as well
   This program can just read in one student (and their module details).
6. If you want to go a step further, read in multiple students (until the student_name is blank.
   Next week we will be looking at functions and we will implement something like this.

Answers to Question 1

a. int
b. float
c. boolean
d. str
e. list
f. tuple
g. str
h. dict
i. list
j. boolean (False)
k. dict
l. str
m. dict
n. list (is is nested in the dict)
o. int
p. undefined (the code has a capital N in courseName)