DW Mid-Term Solutions for Part A (student version)

Q.1 [10 points]

Consider the following code in a Python script file and answer the following questions.

```
 1  import math
 2  def combination(credit):
 3      credit = (credit - 10.0)/20.0
 4      out = 2.0*credit - 1.5
 5
 6      return out
 7
 8  def activation(a):
 9
10      probability = 1.0/(1 + math.exp(-a))
11      print probability
12
13  credit = 16.0
14  output = activation( combination(credit) )
15  print credit
16  print output
17
```

(a) A student calculates that when line 3 is executed, `credit` will have a value of 0.3. However, when line 15 is executed, he is surprised to see `16.0` displayed on the screen. Explain why this is the case. (4 points)

(b) When line 14 is executed, describe the sequence of the function calls and how data is passed among these functions and back to the variable `output`.
Hence, explain why when line 16 is executed, `None` is displayed on the screen. (6 points)

a) The variable “credit” at line 3 is a local variable that exists only within the scope of the function called "combination". The variable “credit” at line 13 is a global variable. This variable at line 13 is not accessed by the function "combination" and is not modified in any way. Hence, line 15 access the value of “credit” at line 13 and prints out 16.

Examiner Remark: *some students seemed to think (wrongly) that the two variables named “credit” are the same variable.*

b) The "combination" function is called and the value of 16 is passed to it and stored in the local variable or formal argument “credit”.

The "combination" function makes some calculations and returns the value of “out” which is -0.9. This value of -0.9 is passed to the "activation" function and stored in the local variable or formal argument “a”.

The value of “probability” is calculated to be 0.289 and printed on the screen. The lack of a return statement in the "activation" function causes None to be returned and stored in the variable “output”, hence line 16 prints out None.

Q2.

a) Syntax errors:

- Line 2 should be "import time"
- Lines 6 and 8 should be == instead of =
- Line 13 should convert "times" to string to concatenate
- Line 16 should be one_round(direction)
- Line 22 'clockwise' should have quotes

b) Logical errors

- Line 7 and 9 should be inverted
- Line 10 sleep should be 5 seconds according to specification
- Must stop of robot after line 10 (wheels(0,0))
- Line 15 had wrong range, should be range(times) or range(1,times +1).

Explanation:

If logical errors are not fixed, the robot will move indefinitely counterclockwise.

Correct code:

```
1. from eBot import eBot
2. import time
3.
4. #Rotate and conquer
5. def one_round(direction):
6.    if direction == 'clockwise':
7.       ebot.wheels(1,0)
8.    elif direction == 'counter clockwise':
9.       ebot.wheels(0,1)
10.   time.sleep(5)
11.   ebot.wheels(0,0)
12.
13. def rotate(times, direction):
```

```
14.    print "Rotating " +str(times)+ " times in direction: "+direction
15.    for i in range(times):
16.        one_round(direction)
17.
18. ebot = eBot.eBot() # create an eBot object
19. ebot.connect() # connect to the eBot via Bluetooth
20.
21. #Test case
22. rotate(3,'clockwise')
23. ebot.disconnect() # disconnect the Bluetooth communication
```