Instructor: Anne Lavergne

This examination has 8 pages.

Let us verify that we have a complete paper.

**Instructions:**

- No books, calculators, computers, cell phones, or other materials may be used.
- Read each question carefully before answering it.
- We must use **Python 3**.
- Let us always **comment our code** and use the **Good Programming Style** (GPS) discussed in class and posted on our course web site.
- We can assume that all Python code fragments given in this exam are syntactically correct, unless otherwise stated.
- The marks for each question are given in [ ]. Let us use this to manage our time:
  - 1 mark correspond to 1 minute of work.

Good luck!
Part 1 - Theory and Understanding [10 marks in total - 1 mark each - No part marks]

Please, answer on the Scantron sheet - in pencil.

1. Consider the Python fragment below:

   ```python
   gameOver = False
   userResponse = 26
   ______________
   print("userResponse is {}).format(userResponse))
   ```

Which of the following Python statement would create an error if it were used to fill in the blank line above?

A. `if gameOver :

B. `if userResponse.isdigit( ) :

C. `if not (gameOver) :

D. All of the above.

E. None of the above.

2. What can we say about the following 2 Python code fragments:

   **Python Code Fragment A**
   ```python
   age = int(input("Age: "))
   if age < 15 :
       print("drink milk")
   if age < 18 :
       print("drink coffee")
   else:
       print("drink beer")
   ```

   **Python Code Fragment B**
   ```python
   age = int(input("Age: "))
   if age < 15 :
       print("drink milk")
   elif age < 18 :
       print("drink coffee")
   else:
       print("drink beer")
   ```

A. They both produce the same result for any input the user enters.

B. They do not always produce the same result.

C. Python code fragment A produces an error.

D. Python code fragment B produces an error.

E. None of the above.
3. What would the variable `vegetable` contain at the end of the 2nd iteration of the following Python loop:

```python
vegetable = ["radish", "carrot", "turnip", "celery"]
for oneVeggie in vegetable:
    print(oneVeggie)
```

A. "turnip".
B. "carrot"
C. ["radish", "carrot", "turnip", "celery"]
D. Python loop produces an error.
E. None of the above.

**Explanation**

Considering the above Python code fragment:

- the 1st time the for loop executes (1st iteration), the variable `oneVeggie` contains "radish", i.e., the element of the list `vegetable` located at index 0
- the 2nd time the for loop executes (2nd iteration), the variable `oneVeggie` contains "carrot", i.e., the element of the list `vegetable` located at index 1
- the 3rd time the for loop executes (3rd iteration), the variable `oneVeggie` contains "turnip", i.e., the element of the list `vegetable` located at index 2
- the 4th time the for loop executes (4th iteration), the variable `oneVeggie` contains "celery", i.e., the element of the list `vegetable` located at index 3

So, in terms of numbering the iterations of a Python loop, we start counting at 1 (as we do in natural language).

In terms of the index or position numbers of characters in a string or elements of a list, we start counting at 0 (as we do as computing scientists).

4. What does the following Python code fragment produce:

```python
length = len("apple")
aList = [length, 0, len("apple")]
print(aList[3])
```

A. "[length, 0, len("apple")]
B. 'len("apple")'
C. 'length'
D. There is an error in the code.
E. None of the above.
5. What does the following Python code fragment produce:

```python
import random
ideas = random.randint(1, 10)
if ideas == 1:
    print("You only have 1 idea!")
else:
    print("You are full of ideas!")
```

A. You are full of ideas!
B. There are no errors in the code, but we cannot tell what the result will be.
C. You only have 1 idea!
D. There is an error in the code.
E. None of the above.

6. To what value should we initialize a variable (i.e., assign an initial value to the variable before using it) if we want to use this variable as a running count variable in our program?

A. Set the variable to 1.
B. Set the variable to an empty string.
C. Set the variable to any number.
D. If the variable is to be used as a running count variable, it should not be initialized at all.
E. None of the above.

**Explanation**

Normally, we initialize a running count variable to the value 0. See an example here: [http://interactivepython.org/runestone/static/thinkcspy/MoreAboutIteration/SentinelValuesAndValidation.html](http://interactivepython.org/runestone/static/thinkcspy/MoreAboutIteration/SentinelValuesAndValidation.html)

However, sometimes, we initialize a running count variable to the value 1. See an example here: [http://interactivepython.org/runestone/static/thinkcspy/MoreAboutIteration/ThewhileStatement.html](http://interactivepython.org/runestone/static/thinkcspy/MoreAboutIteration/ThewhileStatement.html)
7. What does the following Python code fragment produce:

```python
string = "WPLM"
ending = "ow"
for letter in string:
    print(letter + ending)
```

A. Wow
   Pow
   Low
   Mow

B. Wow Pow Low Mow
C. WPLMow
D. There is an error in the code.
E. None of the above.

8. What would the variable `aSection` contain once the following Python code fragment has executed?

```python
prov = "Alberta"
aSection = prov.upper()[2:5:1]
```

A. "BERT"
B. "BER"
C. "" (empty string)
D. There is an error in the code.
E. None of the above.

9. Which of the following is NOT a string method/function?

A. `strip()`
B. `isdigit()`
C. `upper()`
D. `input()`
E. All are string methods/functions.

10. What does `print((16 - 6 // 2 * 3) * 3)` prints on the screen?

A. -11
B. 21
C. 45
D. There is an error in the code.
Part 2 – Coding

1. [15 marks] On the last page of this exam, you will find an incomplete encryption program. Your task is to complete this encryption program by adding the necessary Python code fragment to encrypt the message the user enters.

In order for you to know which algorithm you need to implement in completing this encryption program, have a look at the following table. It contains examples of plain text messages and their corresponding cipher texts. From these examples, can you guess the algorithm you must implement in completing this encryption program?

<table>
<thead>
<tr>
<th>Plain Text Messages</th>
<th>Corresponding Cipher Texts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can you read my message?</td>
<td>C!n y?&amp; r*!d my m<em>ss!g</em>?</td>
</tr>
<tr>
<td>HOW ARE YOU?</td>
<td>H?W !R* Y?&amp;?</td>
</tr>
<tr>
<td>I am well, not need to SHOUT!</td>
<td>@ !m w*ll, n?t n**d t? SH?&amp;T!</td>
</tr>
<tr>
<td>I know SOMETHING you do not know!</td>
<td>@ kn?w S?M*TH@NG y?&amp; d? n?t kn?w!</td>
</tr>
<tr>
<td>Then you must tell me now!</td>
<td>Th<em>n y?&amp; m&amp;st t</em>ll m* n?w!</td>
</tr>
</tbody>
</table>

You can assume there is already a header at the top of this program.
2. [20 marks] Problem Statement: On the last page of this exam, write a LunchOrderBot program that takes your lunch order and outputs the total cost of the lunch. Using your LunchOrderBot the user can buy a sandwich for $5, a salad for $4, a juice for $2 and an apple for $1.

Requirements:

- Your program must accept "Y" or "y" as a yes and "N" or "n" as a no. If the user enters anything else, your program must print "Hum... I did not get this so I’ll assume you said No." and continue as if the user has said no.

- You program must make use of a for loop that uses the range function.

Here are a few sample input/sample output:

<table>
<thead>
<tr>
<th>Lunch Menu:</th>
<th>Lunch Menu:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sandwich -&gt; $5</td>
<td>Sandwich -&gt; $5</td>
</tr>
<tr>
<td>Salad -&gt; $4</td>
<td>Salad -&gt; $4</td>
</tr>
<tr>
<td>Juice -&gt; $2</td>
<td>Juice -&gt; $2</td>
</tr>
<tr>
<td>Apple -&gt; $1</td>
<td>Apple -&gt; $1</td>
</tr>
<tr>
<td>For lunch, would you like ...</td>
<td>For lunch, would you like ...</td>
</tr>
<tr>
<td>a Sandwich (y/n)? y</td>
<td>a Sandwich (y/n)? N</td>
</tr>
<tr>
<td>a Salad (y/n)? n</td>
<td>a Salad (y/n)? Y</td>
</tr>
<tr>
<td>a Juice (y/n)? n</td>
<td>a Juice (y/n)? y</td>
</tr>
<tr>
<td>a Apple (y/n)? Y</td>
<td>a Apple (y/n)? n</td>
</tr>
<tr>
<td>Total for your lunch is $6.</td>
<td>Total for your lunch is $6.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lunch Menu:</th>
<th>Lunch Menu:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sandwich -&gt; $5</td>
<td>Sandwich -&gt; $5</td>
</tr>
<tr>
<td>Salad -&gt; $4</td>
<td>Salad -&gt; $4</td>
</tr>
<tr>
<td>Juice -&gt; $2</td>
<td>Juice -&gt; $2</td>
</tr>
<tr>
<td>Apple -&gt; $1</td>
<td>Apple -&gt; $1</td>
</tr>
<tr>
<td>For lunch, would you like ...</td>
<td>For lunch, would you like ...</td>
</tr>
<tr>
<td>a Sandwich (y/n)? n</td>
<td>a Sandwich (y/n)? y</td>
</tr>
<tr>
<td>a Salad (y/n)? N</td>
<td>a Salad (y/n)? y</td>
</tr>
<tr>
<td>a Juice (y/n)? n</td>
<td>a Juice (y/n)? y</td>
</tr>
<tr>
<td>a Apple (y/n)? n</td>
<td>a Apple (y/n)? y</td>
</tr>
<tr>
<td>Total for your lunch is $0.</td>
<td>Total for your lunch is $12.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lunch Menu:</th>
<th>Lunch Menu:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sandwich -&gt; $5</td>
<td>Sandwich -&gt; $5</td>
</tr>
<tr>
<td>Salad -&gt; $4</td>
<td>Salad -&gt; $4</td>
</tr>
<tr>
<td>Juice -&gt; $2</td>
<td>Juice -&gt; $2</td>
</tr>
<tr>
<td>Apple -&gt; $1</td>
<td>Apple -&gt; $1</td>
</tr>
<tr>
<td>For lunch, would you like ...</td>
<td>For lunch, would you like ...</td>
</tr>
<tr>
<td>a Sandwich (y/n)? y</td>
<td>a Sandwich (y/n)? y</td>
</tr>
<tr>
<td>a Salad (y/n)? n</td>
<td>a Salad (y/n)? y</td>
</tr>
<tr>
<td>a Juice (y/n)? n</td>
<td>a Juice (y/n)? y</td>
</tr>
<tr>
<td>a Apple (y/n)? N</td>
<td>a Apple (y/n)? y</td>
</tr>
<tr>
<td>Hum... I did not get this so I’ll assume you said No.</td>
<td>Total for your lunch is $10.</td>
</tr>
<tr>
<td>a Apple (y/n)? n</td>
<td>With tax, the total for your lunch is $11.5.</td>
</tr>
<tr>
<td>Total for your lunch is $5.</td>
<td></td>
</tr>
</tbody>
</table>

BONUS (1 mark): Add 15% tax to the total:
# Midterm 1 - Part 2 - Question 1
# Description: Encryption program where
# a -> !, e -> *, i -> @, o -> ? and u -> &
# Author: AL
# Date: June 2019

# Ask the user for a message
plainText = input("Please, enter a message to encrypt: ")

# Put your code here
https://repl.it/repls/PreciousBurdensomeParentheses

Possible solution: (there are many)

# Initialize the cipherText
cipherText = ""

# for char in the message :
for char in plainText :
    if char == 'a' or char == 'A':
        cipherText += '!
    elif char == 'e' or char == 'E':
        cipherText += '*'
    elif char == 'i' or char == 'I':
        cipherText += '@'
    elif char == 'o' or char == 'O':
        cipherText += '?'
    elif char == 'u' or char == 'U':
        cipherText += '&'
    else :
        cipherText += char

# Print the original message and the encrypted message
print("'{}' becomes '{}'.".format(plainText, cipherText))
Part 2 - Question 2 – LunchOrderBot Program [https://repl.it/repls/FixedBadFeeds](https://repl.it/repls/FixedBadFeeds)

Possible solution: (there are many)

```
# LunchOrderBot
# Description: LunchOrderBot program that takes your lunch order
and outputs the total cost of the lunch. Using your LunchOrderBot
the user can buy a sandwich for $5, a salad for $4, a juice for $2
and an apple for $1.
# Author: AL
# Date: June 2019

# Print the lunch menu
print("\nLunch Menu:")
print("Sandwich -> $5\nSalad -> $4\nJuice -> $2\nApple -> $1")

# Set up the variables: lunch items, their associated cost, and a
running total variable
menu = ["Sandwich", "Salad", "Juice", "Apple"]
cost = [5, 4, 2, 1]
total = 0

# Ask the user ...
print("For lunch, would you like ...")
for item in range(len(menu)) :
    response = input("\tn {} (y/n)? ".format(menu[item]))
    # If the user says yes ...
    if response in ["Y", "y"] :
        # Add the cost of the item to the running total
        total += cost[item]
    # If the user has neither said yes nor no ...
    elif response not in ["N", "n"] :
        # ... state so ...
        print("Hum... I did not get this so I’ll assume you said No.")

# Print the total cost of lunch
print("Total for your lunch is ${}.".format(total))

# BONUS PART:
# Compute the tax
total *= 1.15

# Print the total cost of lunch with tax
print("With tax, the total for your lunch is ${}.".format(total))
```