IHF: CODE - PYTHON WORKSHEET 5

File Handling
File handling is an important part of any web application. Python has several functions for creating, reading, updating, and deleting files.

The key function for working with files in Python is the open() function.
The open() function takes two parameters; filename, and mode.
Example: f = open("demofile.txt", "r")

There are four different modes for opening a file:
"r" - Read - Default value. Opens a file for reading, error if the file does not exist
"a" - Append - Opens a file for appending, creates the file if it does not exist
"w" - Write - Opens a file for writing, creates the file if it does not exist
"x" - Create - Creates the specified file, returns an error if the file exist

File Handling - Read
The open() function returns a file object, which has a read() method for reading the content of the file: f = open("test.txt", "r")
print(f.read()) - this will read the entire contents of test.txt as the document defined in the f variable.

You can return one line by using the readline() method:
f = open("test.txt", "r")
print(f.readline())

By looping through the lines of the file, you can read the whole file, line by line:
f = open("test.txt", "r")
for x in f:
print(x)

Or find out the total number of lines:
total = 0
for x in open("test.txt"):
    total += 1
print(total)

File Handling - Write
To write to an existing file, you must add a parameter to the open() function:
"a" - Append - will append to the end of the file
"w" - Write - will overwrite any existing content

Append
f = open("test2.txt", "a")
f.write("Now the file has more content!")
f.close()
#open and read the file after the appending:
f = open("test2.txt", "r")
print(f.read())

Overwrite
f = open("test3.txt", "w")
f.write("Woops! I have deleted the content!")
f.close()
#open and read the file after the appending:
f = open("test3.txt", "r")
print(f.read())

File Handling - Close
It is a good practice to always close the file when you are done with it.
f = open("test.txt", "r")
print(f.readline())
f.close()

File Handling - Create a New File
To create a new file in Python, use the open() method, with one of the following parameters:
"x" - Create - will create a file, returns an error if the file exist
"a" - Append - will create a file if the specified file does not exist
"w" - Write - will create a file if the specified file does not exist

f = open("myfile.txt", "w")

KPMG

CODING EXAMPLES

```
1. f = open("<file>.txt", "r")

2. f = open("<file>.txt", "r")
       print(f.read())

3. f = open("<file>.txt", "r")
   for x in f:
       print(x)

4. f = open("example.txt", "w")
   f.write("Hello World")
   f.close()

5. f = open("example.txt", "w")
   f.write("Hello World")
   f.close()

6. f = open("example.txt", "a")
   f.write("It's nice to be here")
   f.close()

7 . f = open("names.txt", "a")
   name = True
   while name:
      name = input("Enter a name: ")
      f.write(name + "\n")
   f.close()
```

QUESTIONS

SECTION A

1. Read the file 'jabberwocky.txt' and print its content to the screen.
2. Read the file 'austen.txt' and print the number of lines in the file.
3. Each line of the file 'numbers.txt' contains a number, write a script to add up all the values in the file.

SECTION B

1. Ask the user to enter their name and append this to a file called 'register.txt'
2. Create a new file called 'even.txt' that contains only the even numbers from the file 'numbers.txt'
3. 'secret.txt' contains a secret message. Each number represents the letter of the alphabet where 1 = A, 2 = B ... Z = 26. Work out what the secret message says
4. Benford's Law states that the leading digits in a collection of data are probably going to be small. For example, most numbers in a set (about 30%) will have a leading digit of 1, when the expected probability is 11.1% (one out of nine digits). Fake data is usually evenly distributed, where as real data is not.
   The files 'accounts_1.txt', 'accounts_2.txt' and 'accounts_3.txt' contain financial transaction data.
   Work out which of the files contains fake data.

KPMG