**Level 1: File Handling Definitions**

Use the following resources to answer the questions about file handling in Python.

* <https://www.pythonforbeginners.com/files/reading-and-writing-files-in-python>
* <https://www.pythonforbeginners.com/cheatsheet/python-file-handling>

1. Explain the function of each of the following file handling commands
   1. The open() function: opens a file
   2. The read() method: return one big string
   3. The readline() method: return one line at a time
   4. The write() method: This method writes a sequence of strings to the file.
   5. The close() method: use close() to close a file and free up any system resources taken up by the open file
2. Research and explain the “Mode” used to open files in a Python program.
   1. ‘r’ mode
   2. ‘w’ mode
   3. ‘a’ mode
   4. ‘r+’ mode
   5. Explain when and where the mode is used in a Python program

r’ – Read mode which is used when the file is only being read

‘w’ – Write mode which is used to edit and write new information to the file (any existing files with the same name will be erased when this mode is activated)

‘a’ – Appending mode, which is used to add new data to the end of the file; that is new information is automatically amended to the end

‘r+’ – Special read and write mode, which is used to handle both actions when working with a file

1. Provide example code which opens a text file for reading and prints the contents of the file to the console display.
   1. Explain what each line of the program does.

file = open("myfile.txt","r")

text= file.read()

print(text)

1. Provide example code which opens a text file for writing and writes some data to the file.
   1. Explain what each line of the program does.

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

file.write("Hello World")

file.close()

1. Research and explain the difference between a “File Name” (type Python string) and   
   a File Object (type Python object).

When you use the open function, it returns something called a file object. File objects contain methods and attributes that can be used to collect information about the file you opened. They can also be used to manipulate said file. For example, the mode attribute of a file object tells you which mode a file was opened in. And the name attribute tells you the name of the file that the file object has opened. You must understand that a file and file object are two wholly separate – yet related – things.

**Level 2: Reading & Writing Files**

1. Add a text file to your project as follows:
   * Click on “Add File” icon in the files pane/window.
   * Type “myfile.txt” and return.
   * “myfile.txt” is now open in the editor pane/window.
   * Type some text into “myfile.txt”
   * Make sure to add several lines of text. A sample file contents could look like:

*Hello kind student*

*This is a message from your computer*

*I hope you are having fun learning to program*

*Remember to ask Mr. Nestor questions when you don’t understand*

1. Write a program that opens “myfile.txt” for reading and prints the contents to the file to the console display.
   1. The program should also print out the number of lines in the file
   2. Provide a listing of your program below

file = open("myfile.txt","r")

text= file.read()

print(text)

1. Write a program that opens “myfile.txt” for appending new contents to the file.
   1. You can “hard code” some commands to write new text to the file
   2. Make sure to use the close() method when your are finished.   
      (What happens if you don’t?)
   3. How can you tell that your program worked? (That the contents changed?)
   4. Provide a listing of your program below

file = open("myfile.txt","a")

file.write("Hello World")

file.close()

1. Write a program that opens “myfile.txt” for writing new contents to the file.
   1. You can “hard code” some commands to write new text to the file
   2. Explain the difference between appending and writing to a file.
   3. Provide a listing of your program below

Writing erases existing content in the file. Appending adds to what already exists in the file.

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

file.write("Hello World")

file.close()

**Level 3: Folders & Binary Files**

1. Add a folder called “resources” to your project as follows:

· Click on “Add Folder” icon in the files pane/window.

· Type “resources” and return.

2. Drag and drop your “myfile.txt” file into the “resources” folder.

3. Run you program from Level 2 to see what happens.

a. Why does it give an error?

b. How can you modify the file name string used by the open() function so that it also includes the “resources” folder?

c. Fix the open() function so that the program runs correctly and provide your program listing below.

file = open("resources/myfile.txt","w")

4. Research and explain the “Binary Mode” used to open files in a Python program.

a. What is the ‘rb’ mode and how is it different from the ‘r’ mode

b. What is the ‘wb’ mode and how is it different from the ‘w’ mode

According to oreilly.com, “When you open a file in binary mode, you are instructing Python to use the data in the file without any decoding; binary mode file reflects the raw data in the file. A file opened in text mode, on the other hand, treats its contents as if it contains text strings of the str type”.

5. Add the “Penguin.bmp” binary image file to your repl project as follows:

a. Download the “Penguin.bmp” file from the GitHub repository to your desktop

b. Drag and drop the “Penguin.bmp” from your desktop to the “resources” folder in your repl project

c. Click on the “Penguin.bmp” to make sure everything is ok.

6. Modify your Level 2 program to open the “Penguin.bmp” and print its contents to the screen.

a. Provide a listing of your modified code below

file = open("resources/Penguin.bmp","rb")

pic= file.read()

print(pic)

file.close()

b. Explain what you see as output compared to the penguin image itself

The output is a series of three digit numbers and letters combined separated by the / symbol, not the penguin image itself.