**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

a.       The open() function

The ***open*** function opens a file. It’s simple.

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.

fileObject = open(“filename.txt”,”mode”)

b.      The read() method

The output of that command will display all the text inside the file, the same text we told the interpreter to add earlier. There’s no need to write it all out again, but if you must know, everything will be shown except for the “$ cat testfile.txt” line.

file = open(“testfile.text”, “r”)

print file.read()

c.       The readline() method

If you want to read a file line by line – as opposed to pulling the content of the entire file at once – then you use the *readline()* function.

Why would you use something like this?

Let’s say you only want to see the first line of the file – or the third. You would execute the *readline()*function as many times as possible to get the data you were looking for.

Each time you run the method, it will return a string of characters that contains a single line of information from the file.

file = open(“testfile.txt”, “r”)

print file.readline():

d.      The write() method

One thing you’ll notice about the file write method is that it only requires a single parameter, which is the string you want to be written.

This method is used to add information or content to an existing file. To start a new line after you write data to the file, you can add an EOL character.

file = open(“testfile.txt”, “w”)

file.write(“This is a test”)

file.write(“To add more lines.”)

file.close()

e.       The close() method

When you’re done working, you can use the ***fh.close()*** command to end things. What this does is close the file completely, terminating resources in use, in turn freeing them up for the system to deploy elsewhere.

file.close()

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

a.       **‘r’ mode -** Read mode which is used when the file is only being read

b.      **‘w’ mode -** 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)

c.       **‘a’ mode -** 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

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

e.       Explain when and where the mode is used in a Python program

f = open(“file.txt”,”w”)

print f

This snippet opens the file named “workfile” in writing mode so that we can make changes to it. The current information stored within the file is also displayed – or printed – for us to view.

3.      Provide example code which opens a text file for reading and prints the contents of the file to the console display.

a.       Explain what each line of the program does.

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

print(file.read())

file.close()

The first line opens the file, and put it into read mode. The second prints the contents of the file when written to be read. The third line close the file.

4.      Provide example code which opens a text file for writing and writes some data to the file.

a.       Explain what each line of the program does.

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

file.write("My ")

file.write("file ")

file.write("has ")

file.write("been ")

file.write("corrupted ")

file.write("for ")

file.write("eternity.")

file.close()

This opens the file “file.txt”, and writes over its contents. Each line below writes a single word with a space, or a period to create a message. The final line will close access to the file and leave it be. The final product is that the contents of the file should have changed to create the message “My files have been corrupted for eternity.”.

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

a File Object (type Python object).

A file object is what allows the alteration to a file as it calls the file. The file name is the name of the file in which you are going to altar.

**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*

2.      Write a program that opens “myfile.txt” for reading and prints the contents to the file to the console display.

a.       The program should also print out the number of lines in the file

b.      Provide a listing of your program below

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

numLines = 0

for line in fH :

print(line)

numLines += 1

print("Number of lines is ", numLines)

3.         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?)

The new text is not written to the file.

* 1. How can you tell that your program worked? (That the contents changed?)

You can check if the program worked by checking the contents of the file.

* 1. Provide a listing of your program below

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

file.write("New text")

file.close()

4.         Write a program that opens “myfile.txt” for writing new contents to the file.

* 1. a.You can “hard code” some commands to write new text to the file
  2. Explain the difference between appending and writing to a file.

Writing to a file replaces any previous information while appending adds new content.

* 1. Provide a listing of your program below

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

file.write("New text")

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.
   1. Why does it give an error?

The file is no longer in the location that it previously was

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

You can modify the string so that it includes the open function by adding it before the file name when opening a file.

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

file = open("resources/test.txt","r")

print(file.read())

file.close()

1. Research and explain the “Binary Mode” used to open files in a Python program.
   1. What is the ‘rb’ mode and how is it different from the ‘r’ mode

‘rb’ mode is used to read binary files which is anything that is not a text file.

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

‘wb’ mode is used to write on binary files.

1. Add the “Penguin.bmp” binary image file to your repl project as follows:
   1. Download the “Penguin.bmp” file from the GitHub repository to your desktop
   2. Drag and drop the “Penguin.bmp” from your desktop to the “resources” folder in your repl project
   3. Click on the “Penguin.bmp” to make sure everything is ok.
2. Modify your Level 2 program to open the “Penguin.bmp” and print its contents to the screen.
   1. Provide a listing of your modified code below

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

print(file.read())

file.close()

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

I see a large amount of binary text. It consists of characters seperated by ‘/’s.

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

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")

print(file.read())

file.close()

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

I see binary text.