**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:** The open() function returns a file object which can used to read, write and modify file.
   2. **The read() method:** This method reads a specific number of characters from the file based on the number in the brackets. If the brackets are empty it reads the whole file.
   3. **The readline() method:** Used to read a file line by line.
   4. **The write() method:** Used to write a fixed sequence of numbers to a file.
   5. **The close() method:** Used to close the current window.
2. Research and explain the “Mode” used to open files in a Python program.
   1. **‘r’ mode:** Read Mode. Used when the file is only being read.
   2. **‘w’ mode:** Write Mode. Used to edit or write new information to the file.
   3. **‘a’ mode:** Appending Mode. Used to add new data to the end of the file.
   4. **‘r+’ mode:** Special read and write mode. Used to handle both actions when working with a file.
   5. Explain when and where the mode is used in a Python program

Read is used to read a file. Write is used to edit or write new information to 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.

f=open("HEELLOO/Sample.txt","r") This line opens the folder and file to read

print(f.read()) This prints the written data to the console

f.close() This closes the file

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.

f=open("HEELLOO/Sample.txt","w") This opens the file to write new data

print(f.write("Hi everyone!")) This is where you write new data

f.close() This closes the file

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

**File Object:** An object returned by a call to open

**File Name:** The name of a file, usually passes as an argument to open

**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\n*

*This is a message from your computer\n*

*I hope you are having fun learning to program\n*

*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
2. f=open("HEELLOO/Sample.txt","r")

2. print(f.read())

3. f.close()

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
2. file1 = open("myfile.txt","a")
3. file1.write("Hiiii")
4. file1.close()

I know my program worked because I went to the file and the data I wanted to write was printed in the file. If you do not use the close command the text does not print into the file.

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
2. file = open('myfile.txt', 'w')
3. file.write('hello')
4. file.close()

The difference is that write mode allows you to overwrite date, append writes in a file and creates the file if it does not currently exist.

**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?

This is because the file is in a folder and the code opens the file not the folder.

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

You can add the folder name before the file name and put “/” between the two.

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

1. file1 = open("Resources/myfile.txt","a")
2. file1.write("Hiiii")
3. file1.close()
4. 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 read binary and r mode is only used to read a file.

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

Wb mode is write binary and w mode is only used to write in a file.

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
3. File = open("Resources/Penguin.bmp","rb")
4. print(File.read())
5. File.close()
   1. Explain what you see as output compared to the penguin image itself

xff\xff\xc0\xff\xff\xc0\xff\xff\xc0\xff\xff\xc0\x00\x00\x00'

(This is the last line)