File Manipulation

- Opening and Closing File
- File Modes
- Handling closing of files
- Reading files with the for loop

Files

- File is a way of data persistence.
- File is simply a named location on non-volatile/permanent storage that holds some information.
- File Processing:
  1. Open File
  2. Process File Data (Fetch/Store)
  3. Close the File

trainer.cpp@gmail.com

Opening and Closing File

- Syntax:

  fileObject = **open**(<name of file>, <modes>)

  fileObject.**close**()

- Open method opens the file specified as a string and returns a **File** Object, which can be used to access the file
- The name of file can contain relative or absolute path.

trainer.cpp@gmail.com

File modes

Mode	Operation	File Pointer
r	Read in text mode	Beginning
rb	Read in binary mode	Beginning
r+, rb+	Read and write text mode	Beginning
w	Write, truncate if exist	Beginning
w+, wb+	Write and read, truncate	Beginning
a	Append	End
ab	Append binary	End
a+, ab+	Append and reading	End

trainer.cpp@gmail.com

Reading files with the for loop

- for <lineObject> in <fileObject>:

  …

- Reads line by line till reaches end
- Reduces the complexity given by while loops (checking empty return value)
- Optimized in comparison to using readlines(), which reads all lines in a list.

trainer.cpp@gmail.com

Question

- WAP to dump everything in a file to the screen.
- Time to update our vowel counting skills.
  Writing a method to count vowels from a file.

trainer.cpp@gmail.com

Reading, Writing and Moving in files

- Read from file
  - **read**([no of bytes])
  - **readline**()
  - **readlines**() # return empty string when reaches end
- Write to file
  - **write()**
  - **flush()**
- Roam around in file
  - **seek(offset, pos)** # pos = 0:beginning; 1: current; 2:end
  - **tell()**

Automatic closing of files

- Syntax:

```
with open(<name of file>, <modes>) as <fileObject>:
    ...
```

Handles automatic closing of file object.

Printing to File Directly

- Syntax:

  fileObject = open(<name of file>, <modes>)

  fileObject.close()

- Methods:

  read([no of bytes]), readline()

  write()

  flush()

  seek(offset, pos) # pos = 0:beginning; 1: current; 2:end

  tell()

Some os Operations

- **os** module contains the following functions:
- *getcwd()* : gives current working directory
  chdir(<path>) : changes current working directory
- *mkdir(<name of directory>)* : create folder in current directory or absolute path
  makedirs(< >) : creates multiple folders appearing in the path if they don't already exist
- *rmdir(<path>)* : the directory to be deleted must be empty
  rename(<source>, <dest>) : source and destination should be on same drive