# Files

Hints exists in this document they are in white text to reveal them copy the text after the Hint!

Try it on the hint under this line.

Hint! Example hint

External files used in this exercise:

* Test.txt
* Words.txt

## Creating a file

Create a file called new\_file.txt and write the text “Hello World” to it. Open the file and check that the text was written to it. Notice where the file was created this is you working directory

## Reading from file

Download the file test.txt and place it in your working directory. Open the file and print its content to the console.

Example output:

Hello

World

Foo

Bar

Lorem

ipsum

dolor

sit

amet

## Printing content on one line

Use the file test.txt. Print the content of the file on a single line.

Example output:

Hello World Foo Bar Lorem ipsum dolor sit amet

## Read n lines from file

Use the file test.txt. Open the file and read n lines from it print these lines. How many lines to read should be given by the user

Example data:

3

Example output:

Hello

World

Foo

## File length in lines

Write a program that reads a filename from the user and determines how many lines long that file is.

Example data:

Test.txt

Example output:

9

## Logging actions

Write a program that reads input from the user and write that input to a log file using the format “Log: User entered {text}” this log file should not be overwritten and new log entries should be added to the bottom of the file i.e appended to the file.

Optional

* Add a time stamp when logging i.e 2020-09-20 15:30:15 Log: User entered Test

Example data:

Test

Example output:

Log: User entered Test

## Word frequency in file

Write a program that reads a filename from the user and counts how many times each word was used in that file.

Optional

* Sort the result so that the most used words are printed first

Hint! Because we count unique words, we can use the word as a key in a dictionary where value is the count of that word.

Example data:

Words.txt

Example output:

('the', 2), ('fool', 2), ('wise', 2), ('doth', 1), ('think', 1), ('he', 1), ('is', 1), ('but', 1), ('man', 1), ('knows', 1), ('himself', 1), ('to', 1), ('be', 1), ('a', 1)

## Parsing a custom file format

Use the file encoded.bff. Write a program that can parse the file encoded.bff. The content of the file should be placed into a dictionary and printed. Encoded.bff is using a custom file format that consists of key-value pair where the key and value is separated by a space and each entry is on a new line. The keys must be unique if a key is not unique an error should be given stating which key was repated. Parse this file and place its content into a dictionary and print it.

Note! The values can also contain space

Example output:

{ “username”: “Foo”, “age”: “28”, “email”: “[foo.bar@nonsense.gov](mailto:foo.bar@nonsense.gov)", “blabla”: “Test Bar”}