CS65: Introduction to Computer Science

File Input

Md Alimoor Reza
Assistant Professor of Computer Science

Review

- Dictionary:
  - What is it and why do we need it?
- Dictionary creation
- Dictionary manipulation
  - Adding or updating
  - Removing
- Iterating over a dictionary

Review: Exercise

1. Set up a dictionary of usernames and passwords

```
passwords = {"urness": "I<3CS", "student19": "Spring2025!"}
passwords["user123"] = "password123"
```

1. Prompt the user for a username and password

```
username = input("What is your username: ")
pw = input("What is your password: ")
```

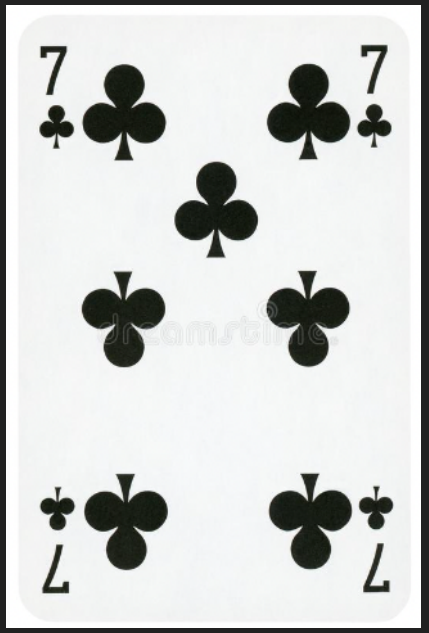
If the username is NOT in the dictionary, response should be **"username not found"**

If the username is in the dictionary, but the password given does not match the password stored in the dictionary, response should be **"password incorrect"**

If the username and password match, response should be **"access granted"**

Dictionaries as Cards

```
my_card = {'value': "7", 'suit': "clubs"}
your_card = {'value': "K", 'suit': "hearts"}
```



Exercise #1

Describe what is happening here in English:

```
deck = []
for suit in ['spades', 'clubs', 'hearts', 'diamonds']:
    for rank in ['A','2','3','4','5','6','7','8','9','T','J','Q','K']:
        deck.append({'value':rank, 'suit':suit})
```

cards.py

What will be the first card in the deck?

What will be output when you execute:

```
print(deck[0])
print(deck[51])
```

Shuffling the deck

The Random module has a simple way of randomizing the elements of a list:

```
import random


deck = []
for suit in ['spades', 'clubs', 'hearts', 'diamonds']:
    for rank in ['A','2','3','4','5','6','7','8','9','T','J','Q','K']:
        deck.append({'value':rank, 'suit':suit})

# The deck is actually a list of 52 dictionaries.

# The shuffle method can be used to shuffle the deck:
random.shuffle(deck)
print("here are 5 cards after shuffling")
for i in range(5):
    print(deck[i])
```

cards2.py

Exercise #2

Develop a **word count** dictionary:

The key should be a word entered, the value should be the number of times is occurred

word_dictionary_ex2.py

```
1  # Alimoor Reza
2  # Exercise # 2
3
4  input_string = input("please input some words: ")
5  word_list = input_string.split()
6
7  word_dictionary = {}
8  for cur_word in word_list:
9      ## yout code goes here
10     # if the word is in the dictionary ...
11     # else if the word is (not yet) in the dictionary ...
12
13
14 print(word_dictionary)
```

```
please input some words: the cat in the hat on the cat
{'the': 3, 'cat': 2, 'in': 1, 'hat': 1, 'on': 1}
```

Agenda

- Files and File I/O operations
  - Read
  - Write
  - Append

General steps for file handling

- Step 1: Open the file using a file variable
  - `file_variable = open(file_name, mode)`
- Step 2: Accomplish the operation using the file variable
  - `file_variable.read(string)`
  - `file_variable.readlines(string)`
  - `file_variable.write(string)`

File Object

- **File object** is a variable associated with a specific file in the disk

```
# reading 'test.txt' file from the current directory/
def read_file_v1():

    file_handler = open('test.txt', 'r')

    my_str = file_handler.read()

    print('Reading the following:')
    print(my_str)
```

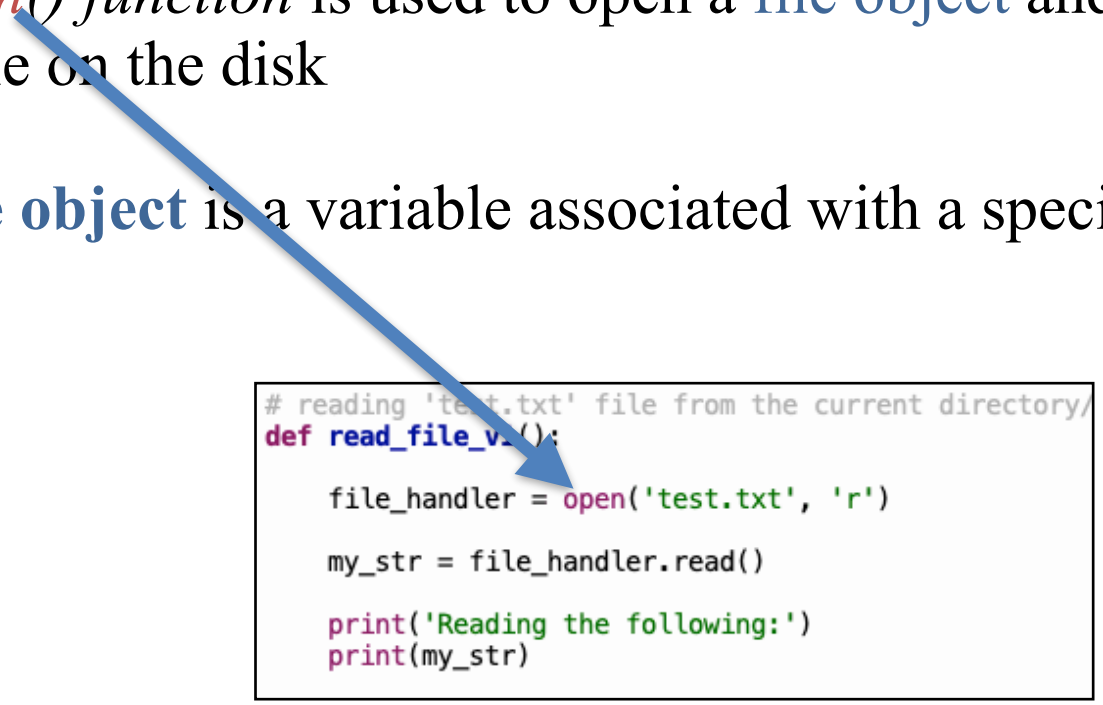
File on the disk (.txt file)

Content after reading from our python program

```
>>> %Run lec15_demo.py

  Reading the following:
  100
  100
  3
  100
  9
  9
```

Open function

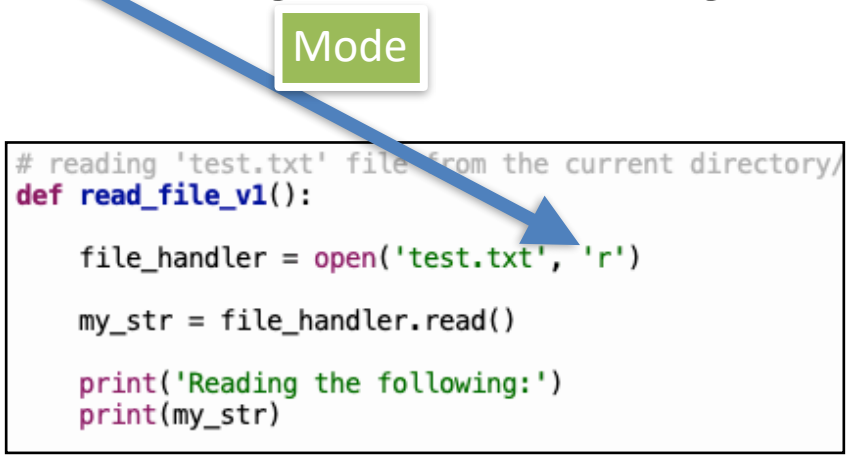
- *open() function* is used to open a file object and associates it with a file on the disk
- **File object** is a variable associated with a specific file in the disk



```
# reading 'test.txt' file from the current directory/
def read_file_v1():

    file_handler = open('test.txt', 'r')

    my_str = file_handler.read()

    print('Reading the following:')
    print(my_str)
```

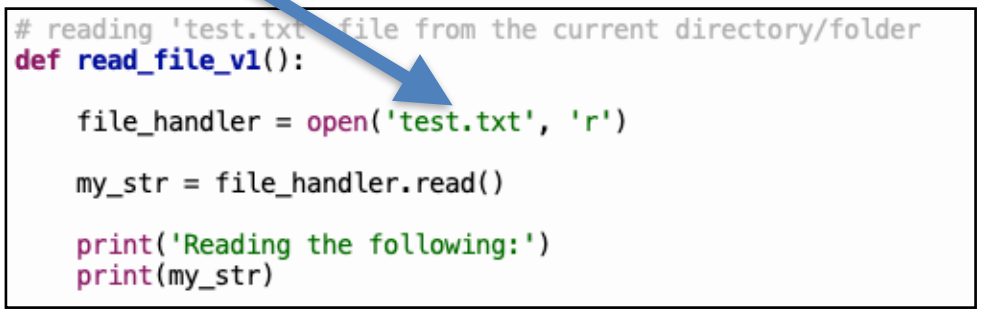
Mode of operation

- **File object** is a variable associated with a specific file in the disk
- *open() function* is used to open a file object and associates it with a file on the disk
- A string argument inside the *open()* specifies how the fill will be opened eg, 'r' —> reading, 'w' —> writing, 'a' —> appending



Name of the file

- **File object** is a variable associated with a specific file in the disk
- *open() function* is used to open a file object and associates it with a file on the disk
- A string argument inside the *open() function* specifies how the fill will be opened eg, 'r' —> reading, 'w' —> writing, 'a' —> appending
- Another string argument inside the *open() function* specifies the name of the file to be accessed



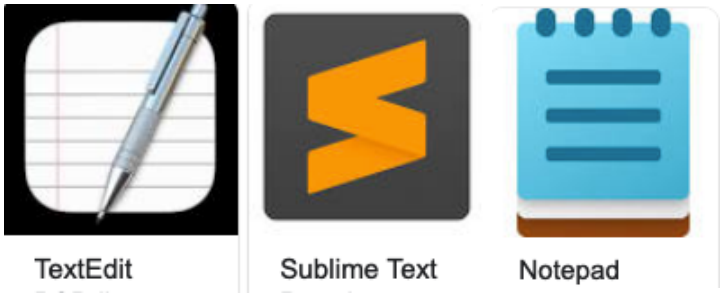
```
# reading 'test.txt' file from the current directory/folder
def read_file_v1():

    file_handler = open('test.txt', 'r')

    my_str = file_handler.read()

    print('Reading the following:')
    print(my_str)
```

Read: different methods

Method with syntax	What it returns
f1.read()	Returns the entire file content as a string
f1.readline()	Returns one line of the file content as a string
f1.readlines()	Returns a list of strings where 1st list item is the content of the 1st line, 2nd list item is the content of the 2nd line, … Last list item is the content of the last line

Reading Data from a File

- Create a text file first using any editor of your choice
  - TextEdit (Mac OSX)
  - Notepad (Windows)



- Make sure you are in the same directory where your python file has been saved (or provide the correct path)

Reading Data from a File

- **Steps:**
  - Open the file with reading mode: `'r'` indicates that. A with statement can be used to open a file, execute a block of statements, and automatically close the file at the end
  - Read the file with method `.read()` method

Demo: reading data from a file using `.read()` method

file_read_demo1.py

```
1  # Alimoor Reza
2  # 11/11/25
3  # reading from a file using .readline() method
4
5  with open('test.txt','r') as f1:
6
7      all_lines = f1.read() # reads the entire file altogether and returns
8      print("all_lines is a single string with the following content")
9      print(all_lines)
10
11     # print the first 3 characters
12     print("the first 3 characters:")
13     print(all_lines[0])
14     print(all_lines[1])
15     print(all_lines[2])
```

Reads everything all at once and places the entire thing in a single string

Reading a File using `.readlines()`

- **Steps:**
  - Open the file with reading mode: `'r'` indicates that
  - Read the file with `.readlines()` method

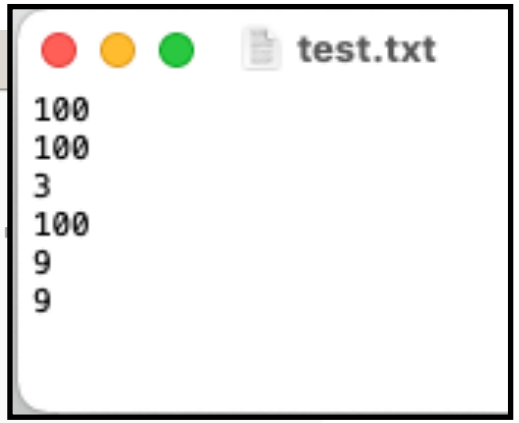
Reads everything all at once and places each line as a separate string in a list

- **Caution:**
  - make sure you are in the same directory or provide the correct path

Demo: Reading a File using `.readlines()`

file_readlines_demo1.py

```
1  # Alimoor Reza
2  # 11/11/25
3  # reading from a file using .readlines() method
4  with open('test.txt', 'r') as f1:      # default mode is 'r'
5
6      file_content_list = f1.readlines() # reads all lines at
7      print(file_content_list)
8      # print the first 3 strings
9      print("the first 3 strings:")
10     print(file_content_list[0])
11     print(file_content_list[1])
12     print(file_content_list[2])
13
14
15     print("showing all the strings using a for loop:")
16     for cur_line in file_content_list:
17         cur_line = cur_line.rstrip('\n') # removes the invisible new line c
18         print(cur_line)
```




```
Reading the following:
['100\n', '100\n', '3\n', '100\n', '9\n', '9\n']
```

Reads everything all at once and places each line as a separate string in a list

Reading a file from an arbitrary location in the disk

Steps:

- Provide the location of the file in your disk
  - as a function parameter of a ***string*** type
- Open the file in reading mode: `'r'` indicates that
- Read using `.readlines()` method
- Iterate through individual *string* with ***for loop***

10.txt

To the People of the State of New York:

THE author of the "Notes on the State of Virginia," quoted in the last paper, has subjoined to that valuable work the draught of a constitution, which had been prepared in order to be laid before a convention, expected to be called in 1783, by the legislature, for the establishment of a constitution for that commonwealth. The plan, like every thing from the same pen, marks a turn of thinking, original, comprehensive, and accurate; and is the more worthy of attention as it equally displays a fervent attachment to republican government and an enlightened view of the dangerous propensities against which it ought to be guarded.

One of the precautions which he proposes, and on which he appears ultimately to rely as a palladium to the weaker departments of power against the invasions of the stronger, is perhaps altogether his own, and as it immediately relates to the subject of our present inquiry, ought not to be overlooked. His proposition is, "that whenever any two of the three branches of government shall concur in opinion, each by the voices of two thirds of their whole number, that a convention is necessary for altering the constitution, or CORRECTING BREACHES OF IT, a convention shall be called for the purpose. "As the people are the only legitimate fountain of power, and it is from them that the constitutional charter, under which the several branches of government hold their power, is derived, it seems strictly consonant to the republican theory, to recur to the same original authority, not only whenever it may be necessary to enlarge,

11.txt

To the People of the State of New York:

IT MAY be contended, perhaps, that instead of OCCASIONAL appeals to the people, which are liable to the objections urged against them, PERIODICAL appeals are the proper and adequate means of PREVENTING AND CORRECTING INFRACTIONS OF THE CONSTITUTION. It will be attended to, that in the examination of these expedients, I confine myself to their aptitude for ENFORCING the Constitution, by keeping the several departments of power within their due bounds, without particularly considering them as provisions for ALTERING the Constitution itself. In the first view, appeals to the people at fixed periods appear to be nearly as ineligible as appeals on particular occasions as they emerge.

If the periods be separated by short intervals, the measures to be reviewed and rectified will have been of recent date, and will be connected with all the circumstances which tend to vitiate and pervert the result of occasional revisions. If the periods be distant from each other, the same remark will be applicable to all recent measures; and in proportion as the remoteness of the others may favor a dispassionate review of them, this advantage is inseparable from inconveniences which seem to counterbalance it. In the first place, a distant prospect of public censure would be a very feeble restraint on power from those excesses to which it might be urged by the force of present motives. Is it to be imagined that a legislative assembly, consisting of a hundred or two hundred members, eagerly bent on some favorite object, and breaking through the

Exercise #3

Given the files 10.txt and 11.txt write a program that will open a file, read in the data, then and create a dictionary that contains the word counts.

In each of the files, what is the word count of:

- whilst
- while
- upon
- the

Exercise #3 hint

word_dictionary_ex3.py

```
1  # Alimoor Reza
2  # Exercise # 3
3
4  with open('11.txt', 'r') as f1:
5      word_list = f1.read().split()
6
7  word_dictionary = {}
8  for cur_word in word_list:
9      if cur_word in word_dictionary:
10         word_dictionary[cur_word] += 1
11     else:
12         word_dictionary[cur_word] = 1
13
14 print('The word count of the word *whilst*: ', word_dictionary['whilst'])
15 print('The word count of the word *the*: ', word_dictionary['the'])
16
```

CSV files

A CSV (comma separated value) file is a special kind of file that stores data

```
Title,Year,Age,IMDb,Rotten Tomatoes,Directors,Genre,Runtime,Country
A Fool There Was,1915,,5.8,80,Frank Powell,Drama,67,United States
The Birth of a Nation,1915,7+,6.4,93,D.W. Griffith,Drama,195,United States
Intolerance: Love's Struggle Throughout the Ages,1916,,7.8,97,D.W. Griffith,Drama,163,United States
Stella Maris,1918,,6.7,86,Marshall Neilan,Drama,84,United States
Tarzan of the Apes,1918,,5.7,83,Scott Sidney,Action,73,United States
Broken Blossoms,1919,13+,7.3,95,D.W. Griffith,Drama,90,United States
Dr. Jekyll and Mr. Hyde,1920,,7,92,Rouben Mamoulian,Horror,98,United States
The Cabinet of Dr. Caligari,1920,7+,8.1,100,Robert Wiene,Fantasy,76,Germany
The Golem: How He Came into the World,1920,,7.2,100,"Carl Boese,Paul Wegener",Fantasy,76,Germany
Way Down East,1920,,7.4,94,D.W. Griffith,Drama,145,United States
Orphans of the Storm,1921,,7.4,92,D.W. Griffith,Drama,150,United States
The Three Musketeers,1921,,7,100,Paul W.S. Anderson,Action,110,United States
```

Title	Year	Age	IMDb	Rotten Tomatoes	Directors	Genre	Runtime	Country
A Fool There Was	1915		5.8	80	Frank Powell	Drama	67	United States
The Birth of a Nation	1915	7	6.4	93	D.W. Griffith	Drama	195	United States
Intolerance: Love's Struggle Throughout the Ages	1916		7.8	97	D.W. Griffith	Drama	163	United States
Stella Maris	1918		6.7	86	Marshall Neilan	Drama	84	United States
Tarzan of the Apes	1918		5.7	83	Scott Sidney	Action	73	United States
Broken Blossoms	1919	13	7.3	95	D.W. Griffith	Drama	90	United States
Dr. Jekyll and Mr. Hyde	1920		7	92	Rouben Mamoulian	Horror	98	United States
The Cabinet of Dr. Caligari	1920	7	8.1	100	Robert Wiene	Fantasy	76	Germany
The Golem: How He Came into the World	1920		7.2	100	Carl Boese,Paul Wegener	Fantasy	76	Germany
Way Down East	1920		7.4	94	D.W. Griffith	Drama	145	United States
Orphans of the Storm	1921		7.4	92	D.W. Griffith	Drama	150	United States
The Three Musketeers	1921		7	100	Paul W.S. Anderson	Action	110	United States
Nosferatu	1922		7.9	97		Drama	60	United States
Robin Hood	1922	7	7.2	100	Ridley Scott	Action	140	United States
The Hunchback of Notre Dame	1923		7.3	91	Gary Trousdale,Kirk Wise	Animation	91	United States

	Title	Year	Age	IMDb	Rotten Tomatoes	Directors	Genre	Runtime	Country
2	A Fool There Was	1915		5.8	80	Frank Powell	Drama	67	United States
3	The Birth of a Nation	1915	7+	6.4	93	D.W. Griffith	Drama	195	United States
4	Intolerance: Love's Struggle Throughout the Ages	1916		7.8	97	D.W. Griffith	Drama	163	United States
5	Stella Maris	1918		6.7	86	Marshall Nei	Drama	84	United States
6	Tarzan of the Apes	1918		5.7	83	Scott Sidney	Action	73	United States
7	Broken Blossoms	1919	13+	7.3	95	D.W. Griffith	Drama	90	United States
8	Dr. Jekyll and Mr. Hyde	1920		7	92	Rouben Mar	Horror	98	United States
9	The Cabinet of Dr. Caligari	1920	7+	8.1	100	Robert Wien	Fantasy	76	Germany
10	The Golem: How He Came into the World	1920		7.2	100	Carl Boese,P	Fantasy	76	Germany
11	Way Down East	1920		7.4	94	D.W. Griffith	Drama	145	United States
12	Orphans of the Storm	1921		7.4	92	D.W. Griffith	Drama	150	United States
13	The Three Musketeers	1921		7	100	Paul W.S. An	Action	110	United States
14	Nosferatu	1922		7.9	97		Drama	60	United States
15	Robin Hood	1922	7+	7.2	100	Ridley Scott	Action	140	United States
16	The Hunchback of Notre Dame	1923		7.3	91	Gary Trousda	Animation	91	United States
17	The Marriage Circle	1924		7.2	100	Ernst Lubitsc	Comedy	85	United States
18	The Joyless Street	1925		7.2	75	Georg Wilhe	Drama	125	Germany
19	The Phantom of the Opera	1925	13+	7.6	91	Joel Schuma	Drama	143	United Kingdom
20	Tumbleweeds	1925	all	6.5	100	Gavin O'Con	Comedy	102	United States
21	The General	1926		8.1	93	Clyde Bruckn	Action	67	United States

Using the csv module

readCSV.py

```
import csv

# open the MoviesData.csv file
with open("MoviesData.csv", mode='r', encoding='utf-8-sig') as infile:
    movies = list(csv.DictReader(infile))
    # movies is a list of dictionaries

# print out the very first movie (a dictionary) in the list
print(movies[0])
```

```
{'Title': 'A Fool There Was', 'Year': '1915', 'Age': '', 'IMDb': '5.8', 'Rotten Tomatoes': '80', 'Directors':
'Frank Powell', 'Genre': 'Drama', 'Runtime': '67', 'Country': 'United States'}
```

What do you notice about the fields here?

```
{'Title': 'A Fool There Was', 'Year': '1915', 'Age': '', 'IMDb': '5.8', 'Rotten Tomatoes': '80', 'Directors':
'Frank Powell', 'Genre': 'Drama', 'Runtime': '67', 'Country': 'United States'}
```

They are all…

What do you notice about the fields here?

```
{'Title': 'A Fool There Was', 'Year': '1915', 'Age': '', 'IMDb': '5.8', 'Rotten Tomatoes': '80', 'Directors':
'Frank Powell', 'Genre': 'Drama', 'Runtime': '67', 'Country': 'United States'}
```

They are all… strings.

If you want to do a numeric calculation, you will need to convert them to a numeric type

What do you notice about the fields here?

```
{'Title': 'A Fool There Was', 'Year': '1915', 'Age': '', 'IMDb': '5.8', 'Rotten Tomatoes': '80', 'Directors':
'Frank Powell', 'Genre': 'Drama', 'Runtime': '67', 'Country': 'United States'}
```

They are all… strings.

If you want to do a numeric calculation, you will need to convert them to a numeric type

```
# note the conversion of the fields to a numeric type
if float(movies[0]['IMDb']) > 5.0 and int(movies[0]['Year']) < 1930:
    print(movies[0]['Title'], " ... is a classic!")
else:
    print(movies[0]['Title'], ".. is a NOT a classic.")
```

readCSV2.py

Exercise #4

Part 1: Write the Python code to determine how many movies are there.

Part 2: Loop through all of the movies, print out all movie **`Title`** that are Action movies (**`Genre`** is **`Action)`** , after 1970 (**`Year`** is greater than **`1970)`** , with an **`IMDb`** rating > 8.3 and a **`Rotten Tomatoes`** rating > 85.

Hint: there should be 8 movies that satisfy the above criteria

```
>>> %Run readCSV_ex3.py

  There are total  5155 movies
  ----------------------------------------
  Star Wars: A New Hope  ...!
  Star Wars: The Empire Strikes Back  ...!
  Raiders of the Lost Ark  ...!
  The Matrix  ...!
  The Dark Knight  ...!
  Inception  ...!
  Dangal  ...!
  Avengers: Endgame  ...!
```

Exercise #4 Hint

```
import csv

# open the MoviesData.csv file
with open("MoviesData.csv", mode='r', encoding='utf-8-sig') as infile:
    movies = list(csv.DictReader(infile))
    # movies is a list of dictionaries

# how many movies?
print("there are", len(movies), "in the file")

# what are the good Action movies after 1970
for i in range(len(movies)):
    if float(movies[i]['IMDb']) > 6.3 and float(movies[i]['Rotten Tomatoes']) > 85 and int(movies[i]['Year']) > 1970 and movies[i]['Genre'] == "Action":
        print(movies[i]['title'])
```

Agenda

- Files and file I/O operations
  - Read
  - **Write**
  - Append