Hour 1

Review: Files -

- Form of persistent data
- Non volatile memory thus can be retrieved after power is turned off
- Persists over time, space and devices
- Can be databases, cloud storage, non volatile memory etc
- Different kinds of files: text, audio, video etc
- 2 Major types: binary and text

Exception handling:

- Try/Except block to find exceptions and handle them
- Exception is when system doesn't know what to do now

```
try:
    x = int(value)
except ValueError:
    print('{} could not be converted to an integer'.format(value))
else:
    print('int({}) is {}'.format(value, x))
```

Using raise:

- Can be used to raise an error

```
def validate_email():
    email_address = input("Enter your email address: ")
    if email_address.count("@") == 1:
        return email_address
    raise ValueError # else

def main():
    try:
        my_email = validate_email()
        print(f"{my_email} is a valid email address")
    except ValueError:
        print("Entered an invalid email")
    print("Thanks for using my validator")
```

Dictionaries:

- Key value pairs
- Mutable thus can be manipulated
- Lists have positions and values whereas dicts have key and values
- Can be created with {} or dict()
- Keys and values can be assigned with dict_name[key] = values
- Can store any datatypes in it
- Can also be nested with other dictionaries

Traversing a Dictionary:

- Dict.items() can be used to get an iterable dictionary
- For key, value in dict.items(): This iterates through the key value tuples in a dict.

Type of files:

- File formats are how a file is made. Ex: text, ppt, music, video, audio etc.
- There are mainly 2 different file categories: Text (sequence of text chars) and Binary (sequence of binary chars)
- Examples of types of text files are: Plain text, Json, CSV etc.
- For each file opened, python creates an object to interact with file.
- File chars are 0-indexed.

Hour 2

- Dictionaries can be converted to lists using list(dictionary)
- Lists should be used when we care about the position of the items
- Dictionaries should be used for a faster lookup and for when we need key value pairs.
- Lists can be used as dictionaries as:

2 lists

'Aquaman'	'Black Panther'	'Greenbook'	'Mary Poppins'
0	1	2	3

(1) movie names
(2) ratings

5	5	4	1
0	1	2	3

One dictionary

value	5	5	4	1
key	'Aquaman'	'Black Panther'	'Greenbook'	'Mary Poppins'

- Building a dictionary from a set of 2 lists:

```
def build_dictionary(words, translation):
    logos = {}
    for index, value in enumerate(words):
        logos[value] = translation[index]
    return logos
```

2 same output functions written with list and dictionary:

Dict:

```
english = ["hello", "goodbye", "thank you"]
french = ["bonjour", "au revoir", "merci"]

translate = build_dictionary(english, french)

word = input("Enter a word: ")
word = word.lower()

if word in translate:
    print(f"Translated word is {translate[word]}")
else:
    print(f"Cannot translate {word}")
```

Lists:

```
english = ["hello", "goodbye", "thank you"]
french = ["bonjour", "au revoir", "merci"]

translator = [english, french]

word = input("Enter a word: ")
word = word.lower()

if word in translator[0]:
    index = translator[0].index(word)
    print(f"Translated word is {translator[1][index]}")
else:
    print(f"Cannot translate {word}")
```

Hour 3

Practice Program Stocks:

```
def main():
    STOCKS = {"IBM" : 140.03,
              "AAPL" : 224.40,
              "NKE" : 92.85}

    asking = True
    while asking:
        stock = input("Enter a stock or :q: to quit: ")
        stock = stock.upper()

        if stock == ":Q:":
            asking = False
            continue

        if stock not in STOCKS:
            print("Stock data not found! Try again")
            continue

        print(f"Current price for {stock} is : {STOCKS[stock]}")

if __name__ == "__main__":
    main()
```

- We first define our stocks and prices dictionary
- Then we in a loop, ask the user for the stock they want to check
- If we don't find the stock, we make them try again
- If we do find it, we just print out the stock name and price

Sets:

- Ordered collection of unique elements
- Not sequences
- Can only contain immutable elements
- Cannot be sliced

Hour 3. 5

Exam Details:

- Tuesday, Online
- Topics:
  - Algorithms
  - Flowcharts
  - Operators
  - Variables
  - Functions
  - Conditionals
  - Sequences
  - Traversing lists
  - Slices
  - Files and Exceptions

No class or lab in the week
4 Hour exam, can be taken anytime on Tuesday before 11:59pm