ENGR 102 PROGRAMMING PRACTICE

WEEK 1

Course Organization

- Syllabus is available on LMS
- Instructor: Ali Cakmak
  - Office Hour: Tuesday, 14:00 – 16:00
- Textbook:
  - Programming Collective Intelligence by Toby Segaran. O'Reilly Press

Course Organization

Teaching Assistants and Office Hours

Mehmet Isgoren <mehmetisgoren@std.sehir.edu.tr>	Wed	11-13
Asem Okby <asemokby@std.sehir.edu.tr>	Fri	11-13
Ali Reza Ibrahimzada <aliibrahimzada@std.sehir.edu.tr>	Fri	9-11
Hakan Yurtluk <hakanyurtluk@std.sehir.edu.tr>	Mon	16-18

Course Organization

- 5 mini projects
  - 12% each
- Midterm Exam
  - 20%
- Final Exam
  - 20%

Course Organization

- Mini Projects:
  - Duration: 2-3 weeks
  - Individual or 2-person small groups
- Evaluation:
  - Plagiarism check
  - Grading criteria announced in the project manual
    - What functionality works?
    - Code organization (comments, naming, etc.)

Course Organization

- Mini Project Evaluation:
  - **Your mini project grade <= 2 * ExamGrade**
  - ExamGrade:
    - Midterm Grade for MP1 and MP2
    - Avg. of Midterm and Final for MP3, MP4, MP5

Course Organization

- **Plagiarism**:
  - Zero tolerance
  - Cases will be referred to the Ethics Committee
  - Both parties (provider and receiver) are responsible
  - Process:
    - Automated computerized checks for pre-filtering
    - Human review for confirmation
    - Referral to the Ethics Committee if true positive

Plagiarism Reports

Submissions_2/ _9137_assignsubmission_file_/ (36%)		Submissions_2/ _9061_assignsubmissi (41%)
112-138		106-134
62-70		60-67
7-14		8-15
93-98		85-90

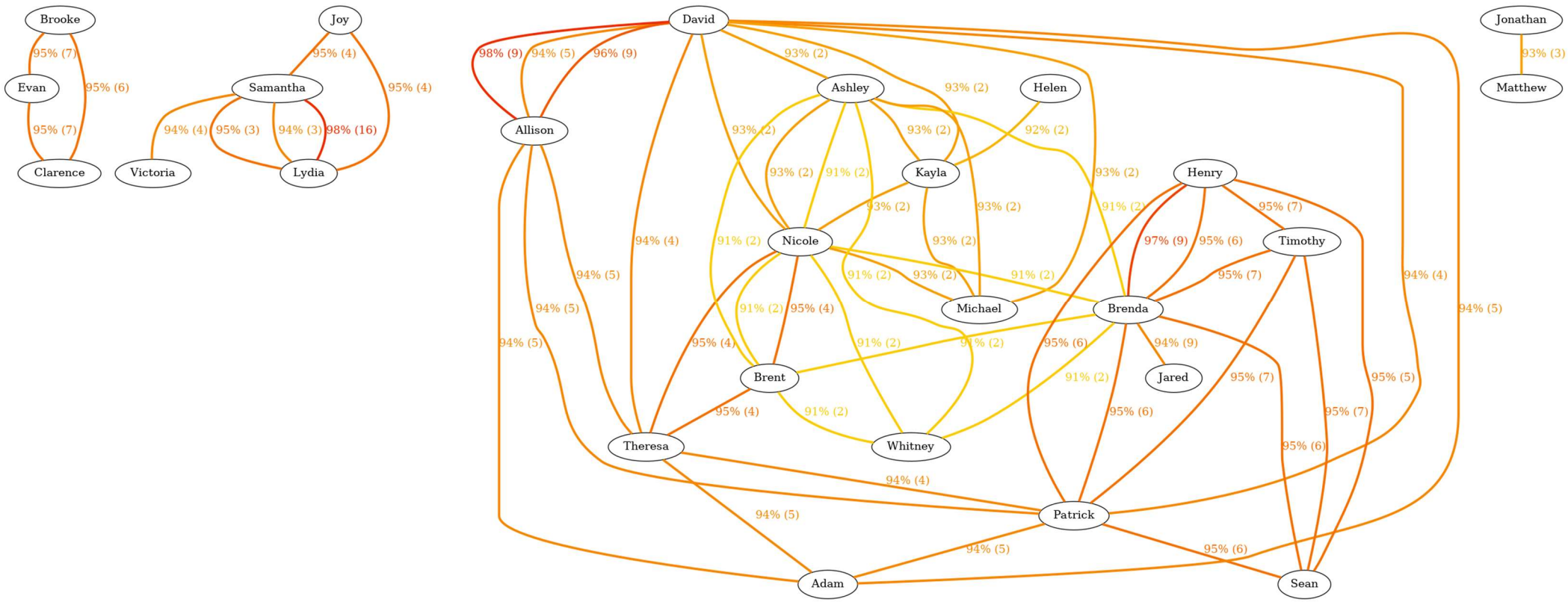
Submissions_2 9137_assignsubmission_file

```
>>>> file:                    .py
import random
def Name_Determiner():
    global x
    global y
    print('-----First Hero-----')


    x=(input('Please type your heros name:'))
    print('-----Second Hero-----')
    y=(input('Please type your heros name:'))
    if x==y:
        while x==y:
            print('Sub-zero is taken,please choose another name!')
            print('----Second Hero-----')
            y=input('Please write Your heros name:')
Name_Determiner()
#Determine the names of hereos

my_list=[]
for i in range(1,101):
    my_list.append(i)
#Every i get %1 possibility - total-->%100

def Coin_toss1():
    global m
    global h1
    global h2
```

Submissions_2/ _9061_assignsubmission_file_/

```
>>>> file:          .py
import random


def start():
    global first_heros
    global second_heros


    first_heros = raw_input('Please type your heros name:')
    print '-----Second Hero-----'
    second_heros = raw_input('Please type your heros name:')
    if first_heros == second_heros:
        while first_heros == second_heros:
            print 'Sub-zero is taken,please choose another name!'
            print '----Second Hero-----'
            second_heros = raw_input('Please write Your heros name:')

my_list = []
for i in range(1, 100):
    my_list.append(i)


start()
def firstattack():
    global first_hero_attack
    global s1
    global s2
```

Plagiarism Reports



Course Organization

- **Other notes**:
  - Attendance policy:
    - Same as the university policy: 75% threshold
    - Attending in a different section than the registered one is **not** allowed
  - Do not work with or talk to others on your projects.
    - You are only allowed to work with your teammate
- Copying from other web sites is plagiarism as well

- "I did not get the solution from anyone! I Google'd by myself, I found a solution by myself, I tested it by myself, … by myself, … by myself, … by myself."

Files

Persistence

- **Transient**:
  - runs for a short time and produce some output,
  - when it ends, its data disappears.
  - if it shuts down and restarts, it starts with a clean state.
- **Persistent**:
  - runs for a long time (or all the time);
  - it keeps at least some of their data in permanent storage (a hard drive, for example);
  - if it shuts down and restarts, it picks up where it left off.

Writing

- To write to a file, you have to open it with mode 'w':

```
fout = open('output.txt', 'w')
```

- If the file already exists, opening it in write mode clears out the old data and starts fresh, so be careful!
- If the file doesn’t exist, a new one is created.

Writing

- The write method puts data into the file.

```
line1 = "Istanbul Sehir\n"

fout.write(line1)
```

- The file object keeps track of where it is, so if you call write again, it adds the new data to the end.

```
line2 = "University.\n"

fout.write(line2)
```

Closing

- When you are done writing, you have to close the file.

```
fout.close()
```

Reading – Alternative ways

```
file = open('newfile.txt', 'r')
for line in file:
    print(line)
file.close()
```

```
for line in open('newfile.txt', 'r'):
    print(line)
```

```
with open('newfile.txt', 'r') as file:
    for line in file:
        print(line)
```

Recommended

Modes

Modes	Description
r	Opens a file for reading only. The file pointer is placed at the beginning of the file. This is the default mode.
r+	Opens a file for both reading and writing. The file pointer placed at the beginning of the file. Does not create the file if it does not exist.
w	Opens a file for writing only. Overwrites the file if the file exists. If the file does not exist, creates a new file for writing.
w+	Opens a file for both writing and reading. Overwrites the existing file if the file exists. If the file does not exist, creates a new file for reading and writing.
a	Opens a file for appending. The file pointer is at the end of the file if the file exists. That is, the file is in the append mode. If the file does not exist, it creates a new file for writing.
a+	Opens a file for both appending and reading. The file pointer is at the end of the file if the file exists. The file opens in the append mode. If the file does not exist, it creates a new file for reading and appending.

Exercise with file functions

Format operator

- The argument of write has to be a string.
- If we want to put other values in a file, we have to convert them to strings. The easiest way is with str:

```
x = 52

f.write(str(x))
```

- Alternative: Use the format operator: %
  - The first operand is the format string, which specify how the second operand is formatted. The result is a string.

Format operator

- For example, the format sequence '%d' means that the second operand should be formatted as an integer (d stands for “decimal”):

```
camels = 42
print('I have spotted %d camels.' % camels)
```

```
'I have spotted 42 camels.'
```

Format operator

- If there is **more than one** format sequence in the string, the second argument has to be a tuple. Each format sequence is matched with an element of the tuple, in order.
- %d: to format an integer,
- %g: to format a floating-point number
- %s: to format a string

```
print('In %d years I have spotted %g %s.' % (3, 0.1, 'camels'))
```

```
In 3 years I have spotted 0.1 camels.
```

Filenames and paths

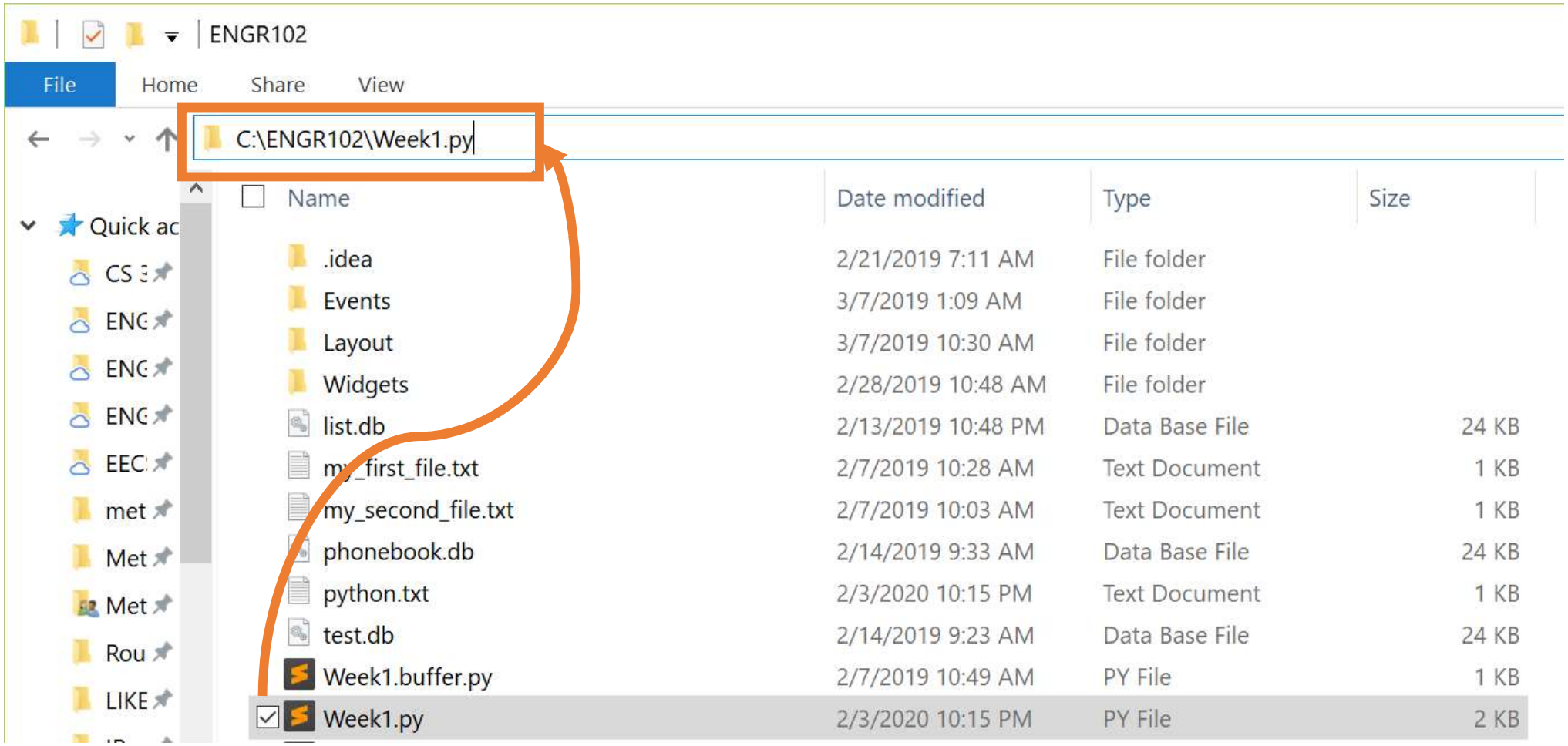
- The **os** module provides functions for working with files and directories (“os” stands for “operating system”).
- os.getcwd returns the name of the current directory:

```
import os
cwd = os.getcwd()
print(cwd)
```

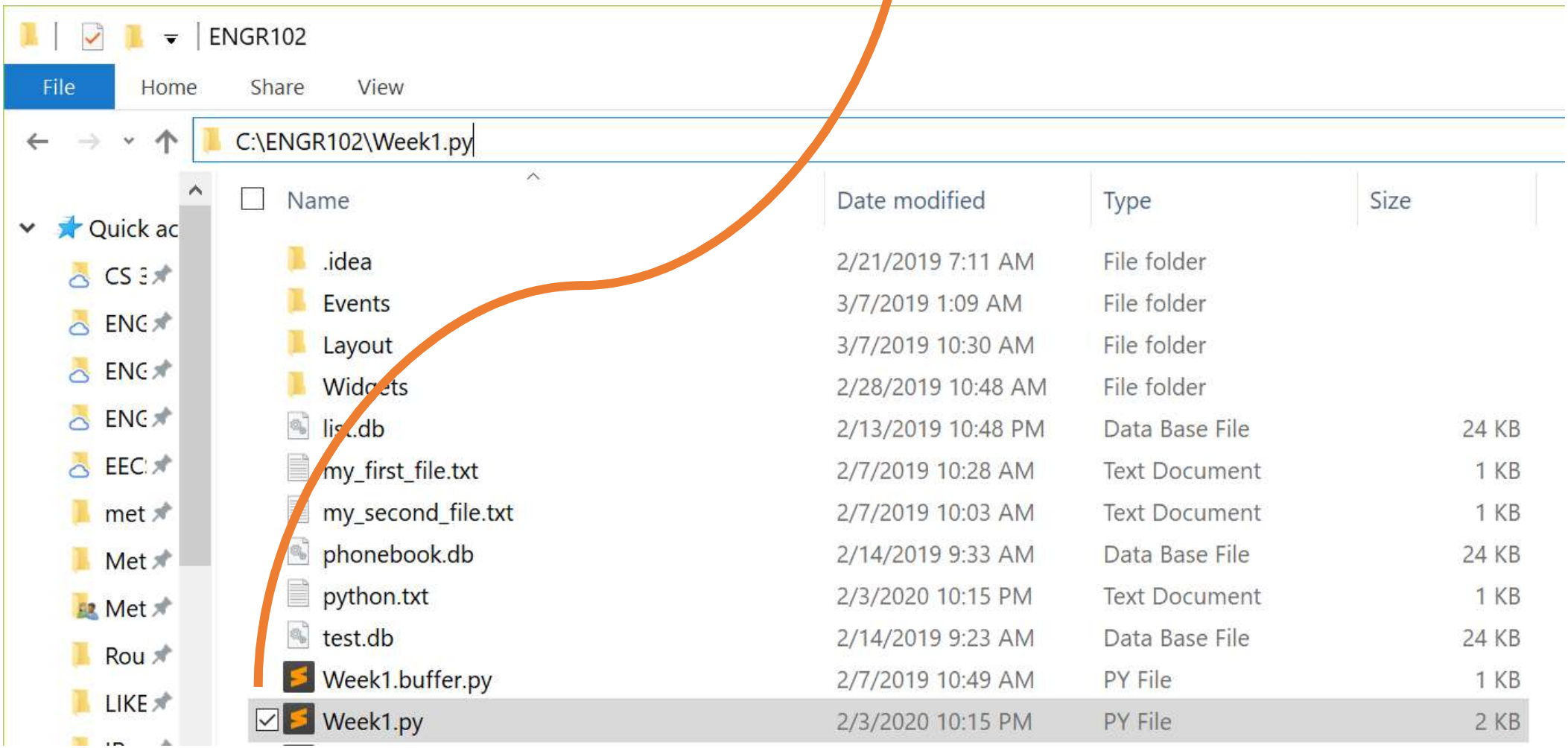
Filenames and paths

- A string that identifies where a file is located is called a **path**.
- A **relative** path starts from the current directory.
- An **absolute** path starts from the topmost directory in the file system.

Absolute Path (C:\ENGR102\Week1.py)



Relative Path (Week1.py)



Filenames and paths

- To find the absolute path to a file, you can use **os.path.abspath**
- **os.path.exists** checks whether a file or directory exists.
- **os.path.isdir** checks whether it's a directory.
- **os.path.isfile** checks whether it's a file.
- **os.listdir** returns a list of the files (and other directories) in the given directory.

Filenames and paths

```
def walk(dir):
    for name in os.listdir(dir):
        path = os.path.join(dir, name)
        if os.path.isfile(path):
            print(path)
        else:
            walk(path)
```