1. Add the current date to the text file today.txt as a string.

Ans.

#import datetime

# datetime.datetime.now() to get

# current date as filename.

filename = datetime.datetime.now()

def create\_file():

# Function creates an empty file

# %d - date, %B - month, %Y - Year

with open(filename.strftime("%d %B %Y")+".txt", "w") as file:

file.write("")

create\_file()

1. Read the text file today.txt into the string today\_string

Ans. #open text file in read mode

text\_file = open("D:/today.txt", "r")

#read whole file to a string

data = text\_file.read()

text\_file.close()

print(data)

1. Parse the date from today\_string.

Ans. #open text file in read mode

text\_file = open("D:/today.txt", "r")

#read whole file to a string

data = text\_file.read()

text\_file.close()

print(data)

1. List the files in your current directory

Ans. import os

path = " PATH"

dir\_list = os.listdir(path)

print("Files and directories in '", path, "' :")

# prints all files

print(dir\_list)

1. Create a list of all of the files in your parent directory (minimum five files should be available).

Ans. SAME AS ABOVE

1. Use multiprocessing to create three separate processes. Make each one wait a random number of seconds between one and five, print the current time, and then exit.

Ans. from multiprocessing import Process, Queue

import time

import sys

def reader\_proc(queue):

"""Read from the queue; this spawns as a separate Process"""

while True:

msg = queue.get() # Read from the queue and do nothing

if msg == "DONE":

break

def writer(count, num\_of\_reader\_procs, queue):

"""Write integers into the queue. A reader\_proc() will read them from the queue"""

for ii in range(0, count):

queue.put(ii) # Put 'count' numbers into queue

### Tell all readers to stop...

for ii in range(0, num\_of\_reader\_procs):

queue.put("DONE")

def start\_reader\_procs(qq, num\_of\_reader\_procs):

"""Start the reader processes and return all in a list to the caller"""

all\_reader\_procs = list()

for ii in range(0, num\_of\_reader\_procs):

### reader\_p() reads from qq as a separate process...

### you can spawn as many reader\_p() as you like

### however, there is usually a point of diminishing returns

reader\_p = Process(target=reader\_proc, args=((qq),))

reader\_p.daemon = True

reader\_p.start() # Launch reader\_p() as another proc

all\_reader\_procs.append(reader\_p)

return all\_reader\_procs

if \_\_name\_\_ == "\_\_main\_\_":

num\_of\_reader\_procs = 2

qq = Queue() # writer() writes to qq from \_this\_ process

for count in [10\*\*4, 10\*\*5, 10\*\*6]:

assert 0 < num\_of\_reader\_procs < 4

all\_reader\_procs = start\_reader\_procs(qq, num\_of\_reader\_procs)

writer(count, len(all\_reader\_procs), qq) # Queue stuff to all reader\_p()

print("All reader processes are pulling numbers from the queue...")

\_start = time.time()

for idx, a\_reader\_proc in enumerate(all\_reader\_procs):

print(" Waiting for reader\_p.join() index %s" % idx)

a\_reader\_proc.join() # Wait for a\_reader\_proc() to finish

print(" reader\_p() idx:%s is done" % idx)

print(

"Sending {0} integers through Queue() took {1} seconds".format(

count, (time.time() - \_start)

)

)

print("")

1. Create a date object of your day of birth.

Ans.

birthday = input("Enter your date of birth: ",)

day = birthday.find("/")

month = birthday.find("/")

year = birthday.rfind("/")

print("Day: ",day)

print("Month: ", month)

print("Year: ", year)

1. What day of the week was your day of birth?

Ans.

birthday = input("Enter your date of birth: ",)

day = birthday.find("/")

month = birthday.find("/")

year = birthday.rfind("/")

print("Day: ",day)

print("Month: ", month)

print("Year: ", year)

1. When will you be (or when were you) 10,000 days old?

Ans.

birthday = input("Enter your date of birth: ",)

day = birthday.find("/")

month = birthday.find("/")

year = birthday.rfind("/")

print("Day: ",day)

print("Month: ", month)

print("Year: ", year)