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

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

3. Parse the date from today\_string.

4. List the files in your current directory

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

6. 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.

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

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

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

Ans:

1. To add the current date to the text file "today.txt" as a string, you can use the following Python code:

```python

import datetime

current\_date = datetime.date.today().strftime("%Y-%m-%d")

with open("today.txt", "w") as file:

file.write(current\_date)

```

This code retrieves the current date using the `datetime` module, formats it as a string in the "YYYY-MM-DD" format, and writes it to the "today.txt" file.

2. To read the contents of the "today.txt" file into a string, you can use the following code:

```python

with open("today.txt", "r") as file:

today\_string = file.read()

```

This code opens the "today.txt" file in read mode and reads its contents into the `today\_string` variable.

3. If you want to parse the date from the `today\_string` variable, you can use the `datetime` module to convert it to a date object. Here's an example:

```python

import datetime

parsed\_date = datetime.datetime.strptime(today\_string, "%Y-%m-%d").date()

```

This code uses the `strptime()` method to parse the `today\_string` according to the specified format ("%Y-%m-%d") and converts it into a date object stored in the `parsed\_date` variable.

4. To list the files in your current directory, you can use the `os` module. Here's an example:

```python

import os

current\_directory = os.getcwd()

file\_list = os.listdir(current\_directory)

for file\_name in file\_list:

print(file\_name)

```

This code retrieves the current directory using `os.getcwd()`, lists the files in that directory using `os.listdir()`, and then iterates over the `file\_list` to print each file name.

5. To create a list of all the files in your parent directory, you can use the `os` module. Here's an example:

```python

import os

parent\_directory = os.path.dirname(os.getcwd())

file\_list = os.listdir(parent\_directory)

for file\_name in file\_list:

print(file\_name)

```

This code uses `os.path.dirname()` to get the parent directory of the current directory, then lists the files in that directory using `os.listdir()`. Finally, it iterates over the `file\_list` and prints each file name.

6. To use multiprocessing to create three separate processes that wait a random number of seconds between one and five, print the current time, and then exit, you can use the `multiprocessing` module. Here's an example:

```python

import multiprocessing

import random

import time

import datetime

def process\_function():

wait\_time = random.randint(1, 5)

time.sleep(wait\_time)

current\_time = datetime.datetime.now().strftime("%H:%M:%S")

print(f"Process ID: {multiprocessing.current\_process().pid}, Current Time: {current\_time}")

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

processes = []

for \_ in range(3):

process = multiprocessing.Process(target=process\_function)

process.start()

processes.append(process)

for process in processes:

process.join()

```

This code defines a `process\_function()` that generates a random wait time, sleeps for that duration, retrieves the current time, and then prints the process ID and current time. In the main block, it creates three separate processes, starts them, and waits for each process to finish using `process.join()`.

7. To create a date object of your day of birth, you need to provide the year, month, and day values. Here's an example:

```python

import datetime

birth\_date = datetime.date(1990, 1, 1)

```

In this code, the `datetime.date()` function is used to create a date object representing the birth date. Adjust the year, month, and day values accordingly.

8. To determine the day of the week for your day of birth, you can use the `strftime()` method to format the date object and extract the day of the week. Here's an example:

```python

import datetime

birth\_date = datetime.date(1990, 1, 1)

day\_of\_week = birth\_date.strftime("%A")

print(day\_of\_week)

```

This code formats the `birth\_date` using `%A`, which represents the full weekday name. The `strftime()` method returns the day of the week as a string, and it is then printed.

9. To calculate when you will be or when you were 10,000 days old, you can add 10,000 days to your birth date and obtain the resulting date. Here's an example:

```python

import datetime

birth\_date = datetime.date(1990, 1, 1)

future\_date = birth\_date + datetime.timedelta(days=10000)

print(future\_date)

```

In this code, the `timedelta()` function is used to create a time delta representing 10,000 days. It is then added to the `birth\_date`, resulting in the `future\_date`. Finally, the `future\_date` is printed.