Lab Week 13 - Classes

Skills Needed to complete this Lab ( You may not use all of these skills )

|  |
| --- |
| * Create and use classes |

## Clock

We’ll use our skills to create a clock class that can keep track of hours, minutes and seconds. The you should be able to create an instance with values for hour, minutes and seconds. You’ll want to make a new class with a \_\_ini\_\_ method. You’ll want to set the attributes for hour, minute, and second in the init.

## Sample Output Creating an instance of Clock test

>>> clock = Clock(2, 10, 15)

>>> clock.hour

2

>>> clock.minute

10

>>> clock.second

15

>>>

## \_\_str\_\_ method

Create a string override method so that the hour, minute, seconds output with a nice 24 hour format style. It should output as 02:10:15 for the example above. In order to output leading zeros you can use .format like below.

## Sample Output \_\_str\_\_ method test

>>> print(clock)

02:10:15

>>> clock.hour = 0

>>> print(clock)

00:10:15

>>> "{:02}".format(1)

'01'

>>>

## Adding to our clock example tick method.

Create a tick() method for the clock. When tick gets called, it should increment the # of seconds. If the number of seconds is 60 or over, then then we increment the number of minutes, and set seconds to 0. If the number of minutes gets set to 60 or over, then we increment the hours.

## Sample tick() method tests

>>> clock = Clock(2, 59, 58)

>>> print(clock)

02:59:58

>>> clock.tick()

>>> print(clock)

02:59:59

>>> clock.tick()

>>> print(clock)

03:00:00

>>> clock.tick()

>>> print(clock)

03:00:01

>>>

## Changing the clock type

We’ll want to add a new default argument to the init method which will be the clock type. The parameter will default to 0, which means a 24 hour clock. The other option 1 would indicate a 12 hour clock. The internal clock will remain unchanged, but our output when printed will change depending on what type of clock is represented. The 12 hour clock will only show hours 1 - 12, with am or pm. 00:00:20 would be 12:00:20 am. While 12:00:20 would be 12:00:20 pm

## Sample clock\_type attribute tests

>>> clock = Clock(0, 59, 58, 0)

>>> print(clock)

00:59:58

>>> clock = Clock(0, 59, 58, 1)

>>> print(clock)

12:59:58 am

>>> clock = Clock(13, 59, 58, 0)

>>> print(clock)

13:59:58

>>> clock = Clock(13, 59, 58, 1)

>>> print(clock)

13:59:58 pm

>>> clock.clock\_type = 0

>>> print(clock)

13:59:58

>>>

# Clock Program

Using the clock class we’ve built create a program to ask a user for hours, minutes and seconds, and create a clock based on that. Then write a loop that calls tick() once a second and then sleeps for a second. Use time module to sleep. Try the example below for using time.

## Using time module

>>> import time

>>> seconds = 0

>>> while True:

print(seconds)

seconds += 1

time.sleep(1)

0

1

2

3

4

## Clock Program example

What is the current hour ==> 11

What is the current minute ==> 58

What is the current second ==> 20

11:58:20 am

11:58:21 am

11:58:22 am

11:58:23 am

11:58:24 am

11:58:25 am

11:58:26 am

11:58:27 am

11:58:28 am

11:58:29 am

11:58:30 am

11:58:31 am

11:58:32 am

11:58:33 am

11:58:34 am

11:58:35 am

11:58:36 am

11:58:37 am

11:58:38 am

11:58:39 am

11:58:40 am

11:58:41 am

11:58:42 am

11:58:43 am

11:58:44 am

11:58:45 am

11:58:46 am

11:58:47 am

11:58:48 am

11:58:49 am

11:58:50 am

11:58:51 am

11:58:52 am

11:58:53 am

11:58:54 am

11:58:55 am

11:58:56 am

11:58:57 am

11:58:58 am

11:58:59 am

11:59:00 am

11:59:01 am

11:59:02 am

11:59:03 am

11:59:04 am

11:59:05 am

11:59:06 am

11:59:07 am

11:59:08 am

11:59:09 am

11:59:10 am

11:59:11 am

11:59:12 am

11:59:13 am

11:59:14 am

11:59:15 am