CSE1322L - Lab 8

Objectives:

In this lab you'll allow the user to enter in 2 times in 24hr (military) format. You'll calculate how many seconds are between those times.

For example, the user may enter 12:00:00 and 14:00:00. In this case your method would return 7200 since there are 60 seconds per minute, 60 minutes per hour, and this is 2 hours of a difference = 60*60*2 = 7200.

This week's lesson is on exception handling, so you'll need to handle all the exceptions that could happen here. Think about what the user could do wrong. Here are a few examples:

25:20:20 - not a valid hour
0:0:89 - not a valid second
14:-2:09 - not a valid minute
12:20 - not valid because they didn't specify seconds
"B" - not a valid time
"\n" - ie, they just hit enter, not valid at all

You'll need to handle all these, but your error should be able to produce at least 4 different errors:
1) Hour not valid
2) Minute not valid
3) Second not valid
4) Time format not valid (Covers "12:20", "B" and "\n")

Tasks

Write a method which takes in a string, and returns an integer. The string should be in the format HH:MM:SS for example 12:20:30

Your method should break the string apart into 3 pieces. In java you might find string.split(":",3) useful, in C# string.Split(":",3)

Convert each number to an int, and validate the sanity of each number. Hours should be 0-23, minutes and seconds should be 0-59. Anything outside those ranges should throw an exception of type InvalidTimeException with an appropriate message indicating the issue.

Assuming the given time is valid, calculate the number of seconds since midnight of that time. The formula is (Hours*60*60) + (Minutes*60) + Seconds. Return that result.

InvalidTimeException should be a class that you define that extends Exception. It should have a constructor which takes in a string, and calls its parent's constructor with that string.

Finally write a driver program which prompts the user to enter 2 time strings, and calls above method with the string, gets a number seconds back, and subtracts the second from the first. The main method should deal with all exceptions thrown. See sample run below

Sample Output:

```
Enter time 1 in 24hr format as follows (HH:MM:SS)
```

```
12:00:00
Enter time 2 in 24hr format as follows (HH:MM:SS)
16:30:01
Difference in seconds: 16201

****Separate Run****

Enter time 1 in 24hr format as follows (HH:MM:SS)
26:00:00
Hour must be below 24

****Separate Run****


Enter time 1 in 24hr format as follows (HH:MM:SS)
12:20
Enter a valid time

****Separate Run****

Enter time 1 in 24hr format as follows (HH:MM:SS)
12:76:01
Minutes must be less than 60

****Separate Run****

Enter time 1 in 24hr format as follows (HH:MM:SS)
2:15:00
Enter time 2 in 24hr format as follows (HH:MM:SS)
4:20:60
Seconds must be less than 60

****Separate Run****

Enter time 1 in 24hr format as follows (HH:MM:SS)

Enter a valid time
//In this last run, I just hit enter when asked for a time.

****Separate Run****

Enter time 1 in 24hr format as follows (HH:MM:SS)
16:00:00
Enter time 2 in 24hr format as follows (HH:MM:SS)
17:00
Enter a valid time
```

Submission Guidelines:

You will turn in one program and a custom exception.

Please follow the posted submission guidelines here:

https://ccse.kennesaw.edu/fye/submissionguidelines.php

Ensure you submit before the deadline listed on the lab schedule for CSE1322L here:
https://ccse.kennesaw.edu/fye/courseschedules.php