COMP 1900 - Fall 2019
Lab 3 Homework

Total Points: 15

As with all other labs, this should be done individually. Contact your instructor and/or visit the Computer Science Learning Center (`https://www.memphis.edu/cs/current_students/cslc.php`) if you need help. You may also work with other students on general solution ideas, but the code you submit must be your own.

Due: By the beginning of your next lab session.

Grader: Your lab TA will grade your work. Questions about grading should be directed to him/her. TA contact info can be found on the course syllabus.

Coding Style: Use consistent indentation in your code; include a reasonable amount of comments throughout your code; and use standard Java conventions for `ClassNames`, `variableNames`, and `CONSTANT_NAMES`. Your TA may deduct points for poor coding style.

Within the `1900` folder on your desktop, create a new folder named `Lab3HWLastnameFirstname`.

(6 pts) Problem 1

As you know, the number of days in each month of our calendar varies:

- February has 29 days in a leap year, or 28 days otherwise.
- April, June, September, and November have 30 days.
- All other months have 31 days.

Usually, years that are divisible by 4 (e.g., 2008, 2012, 2016) are leap years. However, there's an exception: years that are divisible by 100 (e.g., 2100, 2200) are not leap years. But there's also an exception to that exception: years that are divisible by 400 (e.g., 1600, 2000) are leap years.

Within your `Lab3HW` folder, write a program named `DaysInMonth.java` that asks the user to enter a month (1-12) and year (1000-3000). Your program should then show the number of days in that month. If the user enters a month or year beyond the specified ranges, show an appropriate error message.

Here are some examples of what your completed program might look like when you run it. Underlined parts indicate what you type in as the program is running.

Example 1

```
Enter month (1-12): 0
Enter year (1000-3000): 2001
Error - month and/or year out of bounds.
```

Example 2

```
Enter month (1-12): 2
Enter year (1000-3000): 2016
2/2016 contains 29 days.
```

Example 3

```
Enter month (1-12): 2
Enter year (1000-3000): 2900
2/2900 contains 28 days.
```

(9 pts) Problem 2

As indicated at `https://www.flymemphis.com/parking`, Memphis International Airport uses the following pricing schedule for their short-term parking fees:

Time	Pricing
First 30 minutes	Free
31-60 minutes	$2
Each additional 30 minutes	$1
24-hour maximum	$24

Within your `Lab3HW` folder, write a program named `AirportParking.java` that allows the user to enter their number of minutes parked, then computes and prints the appropriate short-term parking charge in dollars. Assume that the airport will always round up to the next half-hour – for example, parking exactly 60 minutes would cost $2, but 61 minutes would make it $3. After all, they want to make as much money as possible!

If the user enters a negative number of minutes, display an appropriate error message. If the user enters a number of minutes that exceeds 24 hours, the 24-hour maximum charge should be applied to each complete day, and any partial day should be charged according to the schedule in the table. For example, parking for 1471 minutes would result in a charge of $26. The first 1440 minutes (24 hours) costs $24, the next 30 minutes is free, and the next 1 minute costs $2.

Here are some more examples of parking charges to help you test your program:

Minutes Parked	Charge
0	$0
30	$0
31	$2
90	$3
91	$4
582	$20
1000	$24
21972	$373
22479	$384

Hint: Start by determining three things – 1) how long it takes to reach the 24-hour maximum charge (it is not 24 hours), 2) how many full days are contained in the minutes parked, and 3) how many 30-minute increments are contained in the partial day that's left over.

Submission

Make sure that your `Lab3HW` folder contains all your `.java` files from this assignment. Zip the `Lab3HW` folder and submit that zip file to the appropriate dropbox folder in your eCourseware lab section.