Modularization Design

LAB 03 : CALENDAR PROGRAM

Due Saturday at 5:00 PM MST

This week, we will implement the calendar design in Python. Deliverables include the following:

1. Your source code
2. A demonstration video

Program Description

Implement in Python the calendar program we designed last week. This is the second lab of a two-part sequence:

1. Lab 02 : Calendar Design
2. Lab 03 : Calendar Program

Assignment

The following is needed to turn this in:

1. Your source code
2. A demonstration video with eight test cases

Source Code

Submit your source code as a file attachment in I-Learn. It must be possible to open this file in Python and execute it without any additional work. You might want to double-check this before pressing [Submit]. In other words, open a fresh Python window and paste the code in to make sure it works as you expect.

Please start with the following template: Lab03.py. In this file, you will find the program header and one function written for you. This function is `display_table()`, the same function whose pseudocode was provided last week.

At the top of your program, include a comment answering these five questions:

```
# 1. Name:
#      -your name-
# 2. Assignment Name:
#      Lab 03: Calendar
# 3. Assignment Description:
#      -describe what this program is meant to do-
# 4. What was the hardest part? Be as specific as possible.
#      -a paragraph or two about how the assignment went for you-
# 5. How long did it take for you to complete the assignment?
#      -total time in hours including reading the assignment and submitting the program-
```

Sample output of the program is the following (where user input is underlined):

```
Enter the month number: 1
Enter year: 1753
  Su  Mo  Tu  We  Th  Fr  Sa
      1   2   3   4   5   6
   7   8   9  10  11  12  13
  14  15  16  17  18  19  20
  21  22  23  24  25  26  27
  28  29  30  31
```

Note: Your program must fulfill the requirements from last week. This means it must count the number of days between January 1, 1753 and the 1st of the month that you will be displaying. It cannot use Python's `datetime` library including `datetime.date()` and `isoweekday()`.

Demonstration Video

Record a short video demonstrating the execution of your program. The video must be very short. No video longer than one minute will be accepted. This means you might need to practice once or twice before recording the video to make sure that you demonstrated everything that is necessary.

Your demonstration video must cover the following test cases:

1. January 1753
2. February 1753
3. January 1754
4. February 1756
5. February 1800
6. February 2000
7. Month: "error", 0, 13, 11
   Year: "error", -1, 1752, 2019

After the video is recorded, provide a voice-over mentioning what test case you are covering. Post your video on some streaming service (like YouTube) and provide a link in your assignment submission.

Assessment

Your grade for this activity will be according to the following rubric:

	Exceptional 100%	Good 90%	Acceptable 70%	Developing 50%	Missing 0%
Efficiency 20%	There is no way possible to make this more efficient	A very efficient solution was found	One or two minor efficiency issues were found	Algorithmic efficiency is one level worse than it should be	Code is extremely inefficient
Maintainability 30%	Everything is obvious and data-driven	Everything is at least straightforward and adjustable	One aspect is not adjustable or straightforward	One aspect is misleading or refactorable	One aspect is puzzling or prohibitive
Modularization 10%	The best possible use of functions	Functions used effectively	A few functions besides display_table() are used	There is only one function: display_table()	There are no defined functions in the code
Functionality	All the test cases execute perfectly and	Everything works but there	One test case fails to execute	At least one test case works as	Code does not run, is missing, no link to demonstration video, or

40%	are demonstrated in the video	are minor cosmetic defects	as expected	expected	does not resemble a working solution