

CSC 4800 Python Applications Programming

Lab #2 – Calendar for a Year

Due: Wednesday, January 18, 2017

The purpose of this assignment is another warm-up exercise in Python 3 programming. Create a Python 3 program that outputs a yearly calendar to a text data file, formatted similar to the example below. Prompt the user for a 4-digit year value (e.g., 2014) and create the output file 'calendar2014.txt'. Your implementation must use at least 2 programmer-defined functions in the solution (be sure to document them properly).

Example output on file 'calendar2014.txt': (one longer file, split up for printing in this handout):

January 2014						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
			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	
February 2014						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
						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	
March 2014						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
						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					
April 2014						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
			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		
May 2014						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
					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						
June 2014						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
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					
July 2014						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
			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	
August 2014						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
						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					
September 2014						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
	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				
October 2014						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
				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
November 2014						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
						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						
December 2014						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
	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			

Turn in printed assignment in class.

→ Be sure to use "Courier New" fixed-pitch font for source and output listings.

- 1) Print out of source code.
- 2) Three printouts of sample execution runs, for year 1996, year 2000 and year 2001.

Hints about calendars:

January 1 in year x begins on day

$$\left(x + \left\lfloor \frac{x-1}{4} \right\rfloor - \left\lfloor \frac{x-1}{100} \right\rfloor + \left\lfloor \frac{x-1}{400} \right\rfloor \right) \bmod 7$$

where $\lfloor x \rfloor$ denotes the greatest integer less than or equal to x , and $m \bmod n$ denotes the remainder when m is divided by n . Sunday corresponds to 0, Monday to 1, and so on. For example, if $x = 1998$,

$$\begin{aligned} & \left(1998 + \left\lfloor \frac{1998-1}{4} \right\rfloor - \left\lfloor \frac{1998-1}{100} \right\rfloor + \left\lfloor \frac{1998-1}{400} \right\rfloor \right) \bmod 7 \\ &= (1998 + 499 - 19 + 4) \bmod 7 = 2482 \bmod 7 = 4 \end{aligned}$$

thus, January 1, 1998 begins on Thursday.

Year x is a leap year if

x is divisible by 4 and not by 100

or

x is divisible by 400

For example, 1998 is divisible by neither 4 nor 400, so 1998 is not a leap year. 1996 is a leap year since 1996 is divisible by 4 and not by 100. 2000 is a leap year since 2000 is divisible by 400. 1900 is not a leap year since 1900 is divisible by 4 and by 100, and 1900 is not divisible by 400.

Note:

Your program and code may not import and use the “calendar” module. The purpose of the lab is to develop and improve your Python programming skills. Your program must analyze the year value and determine the day of the week for January 1st, and whether it is a leap year or not. Then your program must format and output the table for each of the months in that year.

Be sure to implement a clean, well-designed, and commented Python program solution. Every module/program that you write must have a docstring `""" ... """` heading at the beginning that identifies you and gives a brief synopsis of the purpose of the program. Each function must have a docstring comment on the line following the `"def"` states the purpose of the function, a brief description of the input parameters required by the function, and a clear description of what values the function returns, if any.