ECE 498 - Python Homework 10 Fall 2020

Goals: Introduction to classes with Python, more programming practice.

For this homework, submit your work in a ".py" file. Note that some points may be deducted if your code could obviously be written in a "more Pythonic" way. Note: see the code accompanying this lab.

1) Write a class with the following methods:

```
class Mydate:
      '''Class to encapuslate our Julian Day Number functions'''
      def __init__(self, month = None, day = None, year = None, jdn = None):
            '''Constructor possibilities:
            Mydate()   # today
            Mydate(month,day)     # year is this year
            Mydate(month,day, year)
            Mydate(JDN)
            '''
      def getjdn(self):
            '''Return the julian day number'''
      def mdy(self):
            '''Return month, day, year as a tuple'''
      def mdystr(self):
            '''Return month, day,year as a string. e.g., May 5, 2020)'''
      def __repr__(self):
            '''Return month, day,year as a string. e.g., May 5, 2020)'''
      def dow(self):
            '''Returns the day of the week as an int (Sunday = 0, etc.)'''
      def dowstr(self):
            '''Returns the day of the week as a string ('Sunday', etc.)'''
      def ddays(self, date2):
            '''Returns the number of days between given date and this date
            (Positive for date2 before this date)'''
      def dateplusdays(self, ddays):
            '''Date of ddays beyond this date (as Mydate object)'''
      def thisnext(self, dow):
            '''Date of next (or this) day with given dow (as Mydate object)'''
      def isEaster(self):
            '''Returns true if this date is Easter'''
```

Notes:
a) You will be better off if the only "data member" is the Julian day number.
b) We'll assume the Gregorian calendar for everything.
c) You can use my getJDN, JDN2date and easter functions as global functions. Put them in this file as "global functions, so we won't need multiple files. Call them, making the methods above very short.

d) Your constructor should accept the following forms:

```
day1 = Mydate()             # Today
day2 = Mydate(2459156)      # Day of given jdn
day3 = Mydate(7,4)          # Defaults to this year
day4 = Mydate(7,4,1776)     # Month, day year
```

e) Test your class using the test code given in the accompanying file

2) Add at least five "dunders" to your class so it can handle the test code given in the accompanying file. These should be stand-alone methods that don't use "`ddays`" or "`dateplusdays`" - You can comment those out.

`3) Write global functions with the following prototypes`

```
def genrandomdate(date1, date2):
      '''Generator function with generates random dates between the
      two given dates, inclusive'''

def earliest(dates):
      '''return the earliest among a list of dates'''

def latest(dates):
      '''return the latest among a list of dates'''
```

4) Write test code using the above that:

a) Creates a list of 20 random dates in the current year.
b) Prints the list with a simple print
c) Prints the earliest date in the list
d) Prints the latest date in the list
e) Creates and prints a second list from the first list, keeping only the dates that fall on Tuesdays