CIDM 4360/5360 Assignment #2: C#-OOP 1

Total Points: 50 (5% of the course's final grade)

Note: CIDM5360 students have extra tasks in this assignment

Due: Wed Sep 18, 11:59pm

Objective : Assess your understanding of the basic concepts of OOP

Skills needed: In this programming assignment, you'll need to Know how to:

1- **use VS Code to create/edit and run C# programs**
2- **declare class**
3- **Create Constructor methods**
4- **Declare static and non-static (instance) attributes**
5- **Declare and call static and non-static methods**
6- **Overload methods to create more than one constructor**
7- **Instantiate class to create objects**
8- **Split strings using / as delimiters**
9- **Parse (convert) string to int**
10- **Given a class description, identify its member attributes and methods, and write C# code to implement, instantiate, and test the class.**

Requirements & Submission:

1. You **must** add **your name** and the course code (CIDM4360 or CIDM5360) as a comment in the first line of the source code
2. You need to submit complete, error free, and runnable C# program source code
3. Submit on WTClass ("Resources >> Assignments >> Assignment#2)
4. You need to use at least 2 source code files, one for the main program, and one for the tDate class. You can compress them with zip (no rar please)
5. DO NOT use any date processing library or DateTime class

Grading:

1. You'll earn full points for correct implementation of each task
2. Non-executable code will get 0 points
3. Partial credit will be given for executable non-complete tasks
4. CIDM5360 has extra tasks, and different scores for each task

Description:

We want to design a Date class to represent a date using three integer numbers for day, month, and year. Our class will be named tDate. The class has two static attributes (***earliestDate***, ***latestDate***) to represent the earliest and latest dates it can hold. These static attributes are strings contain dates formatted as "mm/dd/yyyy" (using US date formatting). The class can be *instantiated* in two ways, the ***first*** is using three integer arguments separated by comma as (mm, dd, yyyy), and the ***second*** is using string contains a date formatted as "mm/dd/yyyy". Each object of this class type can show its date formatted as either (US) mm/dd/yyyy or (Europe) dd/mm/yyyy according to an argument passed to its showDate( ) method.

Tasks [1- 4 for both CIDM4360 and CIDM5360]:

Write C# program that implements the date class described above according to the following:

1. Declare the class named ***tDate*** with the appropriate private ***static*** and ***non-static*** attributes, public **constructors**, and public **methods**. Make this class in a separate **.cs file**

2. The class **constructors** must parse the passed ***arguments*** correctly and do the following verification(s) before initializing the attributes (day,month,year) of the created object:

   -Verify that the ***arguments*** it received **form** a valid date (do just simple verification, like, 0< days<=31, 0 < months <=12, year >0). i.e. using (15,18,2010) should not be accepted as valid date because the month >12. Similarly, "15/18/2010" is not valid too.

   If the verification failed, the constructor should initialize the attributes (day, month, year) of the new object using the "***earliestDate***" date values.
   Hint: The "***earliestDate***" attribute is a string, so you need to parse it to extract its day, month and year components using string split() method.

3. The ***showDate*** method should receive an argument to specify which format the method should use to display the date (US or Europe), then prints the date according to the given format. The argument could be char ('U' for US, 'E' for Europe) or int (0 for US, 1 for Europe) or a string.

4. The main program method **main( )** should do the following :
   - **a)** Set the static attributes (***earliestDate***, ***latestDate***) of the ***tDate*** class as "1/1/1900", and "12/31/2100" respectively.
   - **b)** instantiate the following objects as follows :
     1- **d1** using three integer values (12,15, 1990)
     2- **d2** using the string ("12/15/1990")
     3- **d3** using the string ("15/12/1990"), which is invalid date
     4- **d4** using the string ("11/11/1811"), which is out of range date (CIDM5360 programmers code shouldn't accept this as valid date)

c) Display the dates of the objects above using the ***showDate*** method with both date formats (US, and Europe). That is, call the method ***showDate*** twice on each object.

[Tasks 5 and 6 are for CIDM5360 Only]

5. Write additional static member method **verifyRange( )** in the class ***tDate*** to perform the following verification, and then call it from your 2nd constructor before you initialize the attributes of the new object:

   -Verify that the received date is between the ***earlies*** and ***latest*** dates specified by the class static attributes "***earliestDate***" and "***latestDate***"

   Hint: One way to test if date1 > date2 (i.e. if date1 comes after date2) is to compare if year1 >year2 then date1 comes after date2, and you don't need to check months or days. But if year1==year2 then heck the months, and if they were equal then check the days too. You can find other ways, **but don't use any date library**

6. Make the 2nd constructor, which has a string parameter, process string dates in the format "dd-MMM-yyyy" (like. "13-Sep-2019") in addition to the "mm/dd/yyyy" format.

   Hint: Try to split the string using '-' first, then if you succeed, extract the month name, and use switch statement with a case for each month name. It would be more organized if you made the switch inside an auxiliary (utility) method that you can call and pass the month name as argument, and it returns the month number (index from 1 to 12) corresponding to that month. For example, write a method named **getMonthIndex** that you can call as **int mm = getMonthIndex(mmm);** which should assign 9 to **mm** if **mmm** equals "Sep". Also you can use arrays to store month names, and check their index in the array.

[- 5 points] Penalty:

- For violating any of the submission requirements above (up to -20 points)

Best of success

Grading Rubric:

Task	*Possible Points CIDM 4360*	*Possible Points CIDM 5360*
1-Class declaration		
a-Static member attributes declaration	*3*	*3*
b-Non-Static member attributes declaration	*3*	*3*
c-1st Constructor Declaration	*3*	*3*
d- 1st Constructor Declaration	*3*	*3*
e-show date member method declaration	*3*	*3*
2- Correct logic		
1st constructor		
Correct basic verification	*4*	*2*
Correct assignment of arguments to member attributes	*3*	*2*
2nd constructor		
Correct parsing of "mm/dd/yyyy" date strings	*4*	*3*
Show date method		
Correctly use the passed format selection parameter	*3*	*3*
Correct display of the date in the selected format	*3*	*3*
Main()		
Correct access of the class tDate static members	*2*	*2*
Correct instantiations of the objects	*3*	*3*
Correct show date message passed to the objects	*3*	*3*
3- Execution		
No syntax errors in the code	*5*	*2*
The program gives the expected results	*5*	*5*
CIDM 5360 only		
Correct parsing of the "dd-mmm-yyyy" date strings		*3*
Correct **verifyRange** *method for 2nd verification*		*4*
Total Points	*50*	*50*