

## Homework #2 – Class Date (second implementation)

In this assignment you are asked to implement a Date class representing a month, day, and year. Your implementation will differ from the previous assignment in several ways (see below). The following UML class diagram shows all attributes and behaviors for class Date.

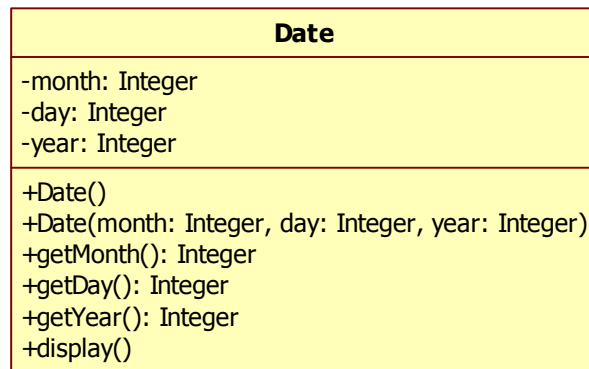


Figure 1. UML class diagram for class Date

1. **(1 point)** Create files named “Date.h” and “Date.cpp” to hold your implementation of class Date.
2. **(1 point)** Define a namespace in which class Date will be defined. The namespace should follow the Pascal naming convention and be named to match your first name followed by your last name (e.g. “RayMitchell”).
3. **(1 point)** Define class Date (in the files and namespace you defined in the previous steps) to have three private data members – a month (type int), a day (type int), and a year (type int).
4. **(1 point)** Define a default constructor that sets the month, day, and year to the current date. Hint: Use the time function to obtain the current system time.
5. **(1 point)** Define a constructor taking three parameters - month, day, and year; this constructor should do the following:
  - a. If the month is not in the range [1, 12] output an error message
  - b. Else if the day is not legal for the given month output an error message
  - c. Else if the year is less than zero output an error message
  - d. Else initialize the month, day, and year data members to the parameter values
6. **(1 point)** Define member functions getMonth, getDay, and getYear as *explicit inline functions*; these functions should return the value of the corresponding data member.
7. **(1 point)** Define a member function display that outputs the month, day, and year separated by forward slashes (/) (e.g. “6/4/2011”).
8. **(1 point)** Make Date’s member functions const where appropriate.
9. **(1 point)** Write a test program that demonstrates class Date’s capabilities. Your test program should demonstrate all constructors and public member functions for class

Date . It should also demonstrate all error conditions being tested. Your test program should be placed in a file named "hw2.cpp".

10. **(1 point)** Make sure your source code is well-commented, consistently formatted, uses no magic numbers/values, follows a consistent style, and is ANSI-compliant.

**Place all source code and a screen capture of the output produced by your program in a single Word or PDF document. Submit this document.**