

Assignment #0 (*welcome assignment*)

Problem Solving and Programming in C++
Department of Computer Science
Old Dominion University

Objectives:

The main objective of this assignment is to help you review some of the concepts related to functions in C++ as well as working with files.

General Instructions:

1. Read the Problem Descriptions below.
2. In this assignment, you will actually start with a working program.
3. Files provided for this assignment can be found in the directory “**Supporting files for Assg0**”.

Problem 1. Description:

The program in `dayOfYear.cpp` works, but it is **ugly** and harder to understand than it should be. That’s because it was written without functions other than `main`. As a consequence, a lot of code gets repeated, and the overall logic of the program is hidden by the mass of details.

Rewrite the program by grouping the calculations into functions. In particular, your program should include the following 4 functions:

1. A function named `numberOfDaysInMonth` that returns an `int` containing the number of days in a month, given the month and year, (passed, in that order, as `int` parameters).
2. A function named `dateIsValid` that tests a date to see if it is a valid date in the Gregorian calendar, returning `true` if it is valid and `false` if it is not. The inputs are the year, month, and day (passed, in that order, as `int` parameters).
3. A function `isALeapYear` that takes a year (`int`) as a parameter and determines whether a given year is a leap year or not, returning `true` or `false` as appropriate.
4. A function named `dayOfTheYear` to compute the day number within the year. This function will have no return value. Its parameters, all integers, will be, in order, the year, month, and day (inputs) and the day number (output). For example, 01/14/2009 is the 14th day of the year and 02/14/2009 is the 45th day. Clue: pass day number by reference.

As you introduce each function, replace the code in `main()` by calls to your new functions as appropriate. In particular, note that some of these new functions may be called from within the bodies of some of the other functions.

Remember that this program was already working. You should not alter the input or output of the program, as it would be observed by someone running the program, in any way.

Hopefully, once you are done it will be obvious to you that your revised code is simpler and easier to understand than the original. In real life, code that is easier to understand, is easier to get working in the first place, so you should try to develop the code in small, function-based chunks from the very beginning.

Sample output:

```
Enter a date in the form YYYY-MM-DD: 2015-08-23
08/23/2015 is day #235 of that year.
```

Problem 2. Description:

Repeat the functionality for problem 1, but include this **exception**:

Instead of collecting date input from the console (black screen), collect input from a file called `date-File.txt` which contains the date in format: YYYY-MM-DD. This functionality should be implemented in a function called `readDateFromFile()`. Make sure you call `readDateFromFile()` from the main function after the solution for problem 1.

Sample output:

```
Date read: 2015-8-23
08/23/2015 is day #235 of that year.
```

Important notes:

- Avoid using global variables in your solution for problem #1 and #2.
- Create a new project for your solution of each problem.

Submission notes:

- Submit **all files** from your project(s), especially the **.cpp**, **.h**, and **.cbp** file(s)
- Zip all the files and name it as `"Asg0_cslogin"`, where the **cslogin** is your login ID for the computers at the Department of Computer Science at ODU.
- Submit the zipped file in the respective Blackboard link.