CS 210, Fundamentals of Computer Science I

Homework #3
Instructor: Tatiana Harrison

This is the third of 6 homework assignments, and the entire assignment is worth 100 points. This homework assignment covers and assumes understanding of the concepts that are covered in lectures 1 – 8, you'll use conditional control statements (if, if-else, if-else-if, etc.), as well as the switch statement.

The **"Homework 3 Q&A" Canvas quiz** is worth the point values indicated below (a total of 26 points). The one programming exercise is worth 74 points. If you need help, ask me, email me, come to my office hours, etc. Instructions on what to hand in are given in each of the below sections. A rubric for the homework assignment has been posted to the course website.

What to hand in:

Complete "Homework 3 Q&A" quiz on Canvas – 26 points

For the programming task, upload the **.java** file to your Canvas account. The rubric has been posted to the course website.

Programming Question – How many days left in a year? - 74 points

This part of the homework requires you to submit a single .java file, called *NumberDaysInAYearSoFar.java*. Make a folder called **homework3** inside of the cs110Submissions folder in your computer, and you can save and edit your code as you work on this programming assignment, but it is only the file that you upload to Canvas that will be scored. Always upload only *.java files.

For this programming task, you'll use conditional control statements (if, if-else, if-else-if, etc.), as well as the switch statement, that you learned about in Lecture8. Your program will prompt the user to input the day (a number), month (a string), and year (a number). You can use the Scanner class, or the JOptionPane class, to receive input from the user. Using that information, your program will calculate the number of **complete** days since the beginning of the year till the date of the year, all three specified by the user.

For full credit, your program will need to take into account whether the year supplied is a leap year, which is governed by the rules in Figure 1. Samples of code were also given in class.

1. If the year is evenly divisible by 4, go to step 2. Otherwise, go to step 5.
2. If the year is evenly divisible by 100, go to step 3. Otherwise, go to step 4.
3. If the year is evenly divisible by 400, go to step 4. Otherwise, go to step 5.
4. The year is a leap year (it has 366 days).
5. The year is not a leap year (it has 365 days).

Figure 1

Use a **switch** statement to process the month (of type `String`) that is input by the user, and any combination of if, if-else, if-else-if, if-else-then, etc. conditional statements, to calculate the number of days since the beginning the year. An example of a pseudo-code for the entire program is given to you in Figure 2.

```
// declare variables dayOfMonth and year of type int
// declare variables monthName of type String, dayOfMonth and year of type int.

// set up scanner and ask user to provide month, day of month, and year
// using nextLine() for monthName, and nextInt() methods for dayOfMonth and year

// declare variable daysSoFarInYear of type int, and assign it the value 0
// declare variable isLeapYear of type boolean, and assign it the value false

// Determine whether the year is a leap year, using the following rules:
// 1. If the year is evenly divisible by 4, go to step 2. Otherwise, go to step 5.
// 2. If the year is evenly divisible by 100, go to step 3. Otherwise, go to step 4.
// 3. If the year is evenly divisible by 400, go to step 4. Otherwise, go to step 5.
// 4. The year is a leap year (it has 366 days).
// 5. The year is not a leap year (it has 365 days).

// Use a switch statement to determine the number of days since beginning of the year.
// For example, December 1 would be day 335 in a leap year and day 334 in a non-leap year.
// days, so daysSoFarInYear = 335 + dayOfMonth - 1.
// For November, there are 305 days from the beginning of the year, so
// daysSoFarInYear = 305 + dayOfMonth - 1 in a leap year, 304 + dayOfMonth - 1 – in a non-leap year.
// For January, it doesn’t matter if a year is a leap year, so
// daysSoFarInYear = dayOfMonth - 1..

// output the correct text (don’t forget plural vs singular), to indicate how many complete days are so far in the year.
```

Figure 2

The partial code for the switch statement is shown in Figure 3.

```
switch(monthName){
        case "January": // on January 1st daysSoFarInYear = 0
          daysSoFarInYear = dayOfMonth - 1;
          break;
        case "February": // on February 1st daysSoFarInYear = 31
          daysSoFarInYear = 31;
          daysSoFarInYear += dayOfMonth - 1;
          break;
        case "March": // on March 1st, 31 + 29 days in a leap year, otherwise + 28
         ...
}
```

Figure 3

To test your code, be sure to use a variety of month/day/year combinations. Using the leap year algorithm in Figure 1, the year 2000 is a leap year, but the 1900 was not. Also, the year 2007 was not a leap year, but 2008 was. Use at least these four years, to test your program for days in months that were before the leap day, and after the leap day. Sample invocations are shown in Figure 4.

Sample invocation 1 (2007 is NOT a leap year):

```
What month is it? December
What day of the month is it? 30
What year is it? 2007
There are 364 complete days since the beginning of the non-leap year 2007.
```

Sample invocation 2 (2008 is a leap year, and date input is in January):

```
What month is it? January
What day of the month is it? 2
What year is it? 2008
There is 1 complete day since the beginning of the leap year 2008.
```

Sample invocation 3 (2009 is NOT a leap year):

```
What month is it? January
What day of the month is it? 15
What year is it? 2009
There are 14 complete days since the beginning of the non-leap year 2009.
```

Figure 4

As you have done for homework #2, you should also submit as comments in your java file:

- The pseudocode of your program
- A description of two ways that your program crashes when it is run.