Homework #3

CS 110, Programming Fundamentals I
Central Washington University
Instructor: Tatiana Harrison

Computer Science

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 one programming exercise is worth 100 points. Check the course web page for TA office hours. If you need help, ask them, 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 canvas page.

What to hand in:

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

Programming Question – Number of days left in this year? - 100 points

This part of the homework requires you to submit a single .java file, called *NumberDaysLeftThisYear.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 Lecture 8. 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 that are remaining in the year specified by the user.

For full credit, your program will need to consider 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 left in a 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 variable monthName of type String

// 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 daysLeftInYear 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 full days left in the year.
// For example, December has 31 days, so daysLeftInYear = 31 - dayOfMonth.
// For November, there are 30 days + 31 days in December, so if user inputs November, daysLeftInYear = 61 - dayOfMonth.
// You will need to look up the number of days in each month.
// For February, IF it is a leap year, then daysLeftInYear = 335 - dayOfMonth, and if it is NOT a
// leap year, then daysLeftInYear = 334 - dayOfMonth.
// For January, if it is a leap year, then daysLeftInYear = 366 - dayOfMonth.
// If it is not a leap year, then if user indicates January, then daysLeftInYear = 365 - dayOfMonth.

// output the correct text, (don't forget plural vs singular), to indicate how many days there are left in the year.
```

Figure 2

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

```
switch(monthName){
        case "December": // always has 31 days
          daysLeftInYear = 31;
          daysLeftInYear -= dayOfMonth;
          break;
        case "November": // always has 30 days
          daysLeftInYear = 61;
          daysLeftInYear -= dayOfMonth;
          break;
…
}
```

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 is 1 complete day left in 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? 31
```

```
What year is it? 2008
There are 335 complete days left in 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 350 complete days left in 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.

RUBRIC

Item	Points Available
Code compiles and correct logic	
Program compiles w/o errors	20
Switch statement is correct	10
If statements for leap year logic calculation is correct calculation is correct	10
Use of `boolean` variables is correct	10
Program outputs proper messages (leap / non-leap year plural vs singular number of days so far)	10
Output is correct • Year 2007: non-leap year • Year 2008: leap year • Year 2009: non-leap year • Leap/non-leap year day counts are correct	20
Code commented / indented • Name included • Date included • Short description included	5
Good variable names	5
Inputs that break code (A description of two ways that your program crashes when it is run)	5
Pseudocode	5
Total	**100**