

CS 122 – Lab #6 – Practice with static methods in Java

Today we will practice the techniques you have seen in sections 2.1 and 2.2 in the book. You will implement some static methods (functions), and then call these functions from `main()`. The specific application deals with the calendar-related functions.

On your CS server account, create a new directory called `lab06` and do all your work in there. From the class Web site (or from `~chealy/www/cs122`), copy the 3 Java files from the `lab06` folder. You should see `Astrology.java`, `Calendar.java` and `Driver.java`. You will make changes to the last two files.

Part 1 – Calendar function implementations

Inside `Calendar.java`, we need implementations of these functions:

- `isLeapYear` – given a year number, is it a leap year or not
- `monthLength` – determine the number of days in a month, given the month and year
- `julian` – calculate the Julian day of a certain day/month/year
- `zodiac` – determine the Zodiacal sign when given the month and day

Currently, the implementations are just skeletons just to make the compiler happy. You need to complete the implementations of these 4 functions. At the bottom of `Calendar.java`, I have included a special `main()` method that will help you test your implementations. Please do not change `main()`. For the `zodiac()` method, use the following definitions of the signs:

Sign	Date range
Aries	March 21 – April 19
Taurus	April 20 – May 20
Gemini	May 21 – June 20
Cancer	June 21 – July 22
Leo	July 23 – August 22
Virgo	August 23 – September 22
Libra	September 23 – October 22
Scorpio	October 23 – November 21
Sagittarius	November 22 – December 21
Capricorn	December 22 – January 19
Aquarius	January 20 – February 18
Pisces	February 19 – March 20

Here is a hint for implementing `zodiac()` – you can convert the month/day into a single value equal to $100 * \text{month} + \text{day}$. Then this single number can be compared to a range of values. For example, Libra would correspond to the numbers 923 to 1022, inclusive. You need to be careful with the definition of Capricorn since it wraps around to the next year.

Run `java Calendar` to verify that your implementations are correct.

Part 2 – Interactive I/O in Driver.java

Now edit the file `Driver.java`. This contains the `main()` method for the interactive version of the program. It will ask the user to enter a date, and then call the various `Calendar` methods, plus one method in `Astrology` to print the horoscope. I have implemented the horoscope for you already.

To finish `Driver.java`, you need to complete the 5 assignment statements in the middle of the program. In these assignment statements, you need to write function call expressions. You need to call the correct function, specify the proper source file where its implementation can be found, and also supply the required actual parameters.

When you are finished, please demonstrate your completed program to the instructor or lab aide.

Have fun!